UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

LINEAR AND NONLINEAR ROOM COMPENSATION OF AUDIO RENDERING SYSTEMS

DOCTORAL THESIS

by

Laura Fuster Criado

Supervisors:

Dr. María de Diego Antón

Dr. Miguel Ferrer Contreras

Valencia, Spain

November 2015

Abstract

Common audio systems are designed with the intent of creating real and immersive scenarios that allow the user to experience a particular acoustic sensation that does not depend on the room he is perceiving the sound. However, acoustic devices and multichannel rendering systems working inside a room, can impair the global audio effect and thus the 3D spatial sound.

In order to preserve the spatial sound characteristics of multichannel rendering techniques, adaptive filtering schemes are presented in this dissertation to compensate these electroacoustic effects and to achieve the immersive sensation of the desired acoustic system. Adaptive filtering offers a solution to the room equalization problem that is doubly interesting. First of all, it iteratively solves the room inversion problem, which can become computationally complex to obtain when direct methods are used. Secondly, the use of adaptive filters allows to follow the time-varying room conditions.

In this regard, adaptive equalization (AE) filters try to cancel the echoes due to the room effects. In this work, we consider this problem and propose effective and robust linear schemes to solve this equalization problem by using adaptive filters. To do this, different adaptive filtering schemes are introduced in the AE context. These filtering schemes are based on three strategies previously introduced in the literature: the convex combination of filters, the biasing of the filter weights and the block-based filtering. More specifically, and motivated by the sparse nature of the acoustic impulse response and its corresponding optimal inverse filter, we introduce different adaptive equalization algorithms.

In addition, since audio immersive systems usually require the use of multiple transducers, the multichannel adaptive equalization problem should be also taken into account when new single-channel approaches are presented, in the sense that they can be straightforwardly extended to the multichannel case.

On the other hand, when dealing with audio devices, consideration must be given to the nonlinearities of the system in order to properly equalize the electroacoustic system. For that purpose, we propose a novel nonlinear filtered-x approach to compensate both room reverberation and nonlinear distortion with memory caused by the amplifier and loudspeaker devices.

Finally, it is important to validate the algorithms proposed in a real-time implementation. Thus, some initial research results demonstrate that an adaptive equalizer can be used to compensate room distortions.

Keywords: Room Equalization, Filtered-x Structures, Linear and Non-Linear Distortion, Adaptive Algorithms, Mean-square Error, Convex Combination Filters, Blocked-based Algorithms, Biased Filters, Real-time Implementation.

Resumen

Los sistemas de audio actuales están diseñados con la idea de crear escenarios reales e inmersivos que permitan al usuario experimentar determinadas sensaciones acústicas que no dependan de la sala o situación donde se esté percibiendo el sonido. Sin embargo, los dispositivos acústicos y los sistemas multicanal funcionando dentro de salas, pueden perjudicar el efecto global sonoro y de esta forma, el sonido espacial 3D.

Para poder preservar las características espaciales sonoras de los sistemas de reproducción multicanal, en esta tesis se presentan los esquemas de filtrado adaptativo para compensar dichos efectos electroacústicos y conseguir la sensación inmersiva del sistema sonoro deseado. El filtrado adaptativo ofrece una solución al problema de salas que es interesante por dos motivos. Por un lado, resuelve de forma iterativa el problema de inversión de salas, que puede llegar a ser computacionalmente costoso para los métodos de inversión directos existentes. Por otro lado, el uso de filtros adaptativos permite seguir las variaciones cambiantes de los efectos de la sala de escucha.

A este respecto, los filtros de ecualización adaptativa (AE) intentan cancelar los ecos introducidos por la sala de escucha. En esta tesis se considera este problema y se proponen esquemas lineales efectivos y robustos para resolver el problema de ecualización mediante filtros adaptativos. Para conseguirlo, se introducen diferentes esquemas de filtrado adaptativo para AE. Estos esquemas de filtrado se basan en tres estrategias ya usadas en la literatura: la combinación convexa de filtros, el sesgado de los coeficientes del filtro y el filtrado basado en bloques. Más especificamente y motivado por la naturaleza dispersiva de las respuestas al impulso acústicas y de sus correspondientes filtros inversos óptimos, se presentan diversos algoritmos adaptativos de ecualización específicos.

Además, ya que los sistemas de audio inmersivos requieren usar normalmente múltiples trasductores, se debe considerar también el problema de ecualización multicanal adaptativa cuando se diseñan nuevas estrategias de filtrado adaptativo para sistemas monocanal, ya que éstas deben ser fácilmente extrapolables al caso multicanal.

Por otro lado, cuando se utilizan dispositivos acústicos, se debe considerar la existencia de no linearidades en el sistema electroacústico, para poder ecualizarlo correctamente. Por este motivo, se propone un nuevo modelo no lineal de filtrado-x que compense a la vez la reverberación introducida por la sala y la distorsión no lineal con memoria provocada por el amplificador y el altavoz.

Por último, es importante validar los algoritmos propuestos mediante implementaciones en tiempo real, para asegurarnos que pueden realizarse. Para ello, se presentan algunos resultados experimentales iniciales que muestran la idoneidad de la ecualización adaptativa en problemas de compensación de salas.

Keywords: Ecualización de salas, estructuras de filtrado-x, distorsión lineal y no lineal, algoritmos adaptativos, error cuadrático medio, filtros de combinación convexa, algoritmos basados en filtros, filtros sesgados, implementación en tiempo real.

Resum

Els sistemes d'àudio actuals es dissenyen amb l'objectiu de crear ambients reals i immersius que permeten a l'usuari experimentar una sensació acústica particular que no depèn de la sala on està percebent el so. No obstant això, els dispositius acústics i els sistemes de renderització multicanal treballant dins d'una sala poden arribar a modificar l'efecte global de l'àudio i per tant, l'efecte 3D del so a l'espai.

Amb l'objectiu de conservar les característiques espacials del so obtingut amb tècniques de renderització multicanal, aquesta tesi doctoral presenta esquemes de filtrat adaptatiu per a compensar aquests efectes electroacústics i aconseguir una sensació immersiva del sistema acústic desitjat.

El filtrat adaptatiu presenta una solució al problema d'equalització de sales que es interessant baix dos punts de vista. Per una banda, el filtrat adaptatiu resol de forma iterativa el problema inversió de sales, que pot arribar a ser molt complexe computacionalment quan s'utilitzen mètodes directes. Per altra banda, l'ús de filtres adaptatius permet fer un seguiment de les condicions canviants de la sala amb el temps.

Més concretament, els filtres d'equalització adaptatius (EA) intenten cancel·lar els ecos produïts per la sala. A aquesta tesi, considerem aquest problema i proposem esquemes lineals efectius i robustos per a resoldre aquest problema d'equalització mitjançant filtres adaptatius. Per aconseguir-ho, diferent esquemes de filtrat adaptatiu es presenten dins del context del problema d'EA. Aquests esquemes de filtrat es basen en tres estratègies ja presentades a l'estat de l'art: la combinació convexa de filtres, el sesgat dels pesos del filtre i el filtrat basat en blocs. Més concretament, i motivat per la naturalesa dispersa de la resposta a l'impuls acústica i el corresponent filtre òptim invers, presentem diferents algorismes d'equalització adaptativa.

A més a més, com que els sistemes d'àudio immersiu normalment requereixen l'ús de múltiples transductors, cal considerar també el problema d'equalització adaptativa multicanal quan es presenten noves solucions de canal simple, ja que aquestes s'han de poder estendre fàcilment al cas multicanal.

Un altre aspecte a considerar quan es treballa amb dispositius d'àudio és el de les no linealitats del sistema a l'hora d'equalitzar correctament el sistema electroacústic. Amb aquest objectiu, a aquesta tesi es proposa una nova tècnica basada en filtrat-x no lineal, per a compensar tant la reverberació de la sala com la distorsió no lineal amb memòria introduïda per l'amplificador i els altaveus.

Per últim, és important validar la implementació en temps real dels algorismes proposats. Amb aquest objectiu, alguns resultats inicials demostren la idoneïtat de l'equalització adaptativa en problemes de compensació de sales.

Paraules Clau: Equalització de sales, estructures de filtrat-x, Distorsió lineal i no lineal, algorismes adaptatius, error quadràtic mig, filtres combinats convexes, algorismes basats en blocs, filtres sesgats, implementació en temps real.

Contents

List of Figures

List of symbols

$\mathbf{X}$	Matrix
$\mathbf{x}$	Vector
x	Scalar
$(\cdot)^{\mathrm{T}}$	Transpose
$(\cdot)^{\mathrm{H}}$	Conjugate transpose
$\lvert \cdot \rvert$	Absolute value
$\lceil x \rceil$	To round up the x value towards the nearest integer
$E\{\cdot\}$	mathematical expectation
$diag(\cdot)$	Diagonal matrix with the elements of vector $(\cdot)$
$\lVert \cdot \rVert_p$	ℓ_p norm
$\hat{(\cdot)}$	Estimation
$*$	Element-wise convolution
T_{60}	Reverberation time
$J(n)$	Mean square error
$\xi(\mathbf{x})$	Sparseness measure
k	Source index
j	Loudspeaker index
m	Microphone index of the multichannel system
n	Discrete time sample
l	Coefficient index of the adaptive filter
N	Number of sources
J	Number of loudspeakers
M	Number of microphones of the multichannel system
i	Filter index for the combination scheme, $i = 1, 2$.
L_h	Length of the FIR filter that models $\mathbf{h}$
L_w	Length of the adaptive filter
$x(n)$	Input signal
$y_i(n)$	Output signal of the adaptive filter i
$z(n)$	Microphone signal
$d(n)$	Desired signal
$e(n)$	Error signal
$e_i(n)$	Error signal of the adaptive filter i
$\mathbf{h}$	Acoustic channel

$\hat{\mathbf{h}}$	Estimated impulse response of $\mathbf{h}$ with a L_h-taps FIR filter
$x_{\mathrm{f}}(n)$	Input signal $x(n)$ filtered by $\hat{\mathbf{h}}$
$y_{\mathrm{f}}(n)$	Output signal $y(n)$ filtered by $\hat{\mathbf{h}}$
$w_l(n)$	lth coefficient of the adaptive filter
$\mathbf{w}(n)$	Inverse adaptive filter
$\mathbf{x}(n)$	$[x(n), x(n-1), \ldots, x(n-L_w+1)]^T$
$\mathbf{x}_{\mathrm{f}}(n)$	$[x_{\mathrm{f}}(n), x_{\mathrm{f}}(n-1), \ldots, x_{\mathrm{f}}(n-L_w+1)]^T$
$\mathbf{x}_h(n)$	$[x(n), x(n-1), \ldots, x(n-L_h+1)]^T$
$\mathbf{y}(n)$	$[y(n), y(n-1), \ldots, y(n-L_h+1)]^T$
$\mathbf{y}_i(n)$	$[y_i(n), y_i(n-1), \ldots, y_i(n-L_h+1)]^T$
$\mathbf{w}_{\mathrm{o}}(n)$	Optimal inverse filter, $[w_{o,1}(n), w_{o,2}(n), \ldots, w_{o,L_w}(n)]^T$

For block-based schemes:

m	Block index
q	Coefficient index of each block
M	Number of blocks
Q	$Q = L_w/M$, number of coefficients of each block
$y_{m\mathrm{f}}(n)$	Output signal $y_m(n)$ of the m-block filter, filtered by $\hat{\mathbf{h}}$
$\mathbf{x}_m(n)$	$[x(n-[m-1]Q), \ldots, x(n-mQ+1)]^T$
$\mathbf{x}_{m_{\mathrm{f}}}(n)$	$[x_{\mathrm{f}}(n-[m-1]Q), \ldots x_{\mathrm{f}}(n-mQ+1)]^T$
$\mathbf{w}_m(n)$	$[w_{1+(m-1)Q}(n), \ldots, w_{mQ}(n)]^T$
$\mathbf{w}_{i,m}(n)$	mth inverse filter block of the adaptive filter i
$\mathbf{w}_{\mathrm{o},m}(n)$	mth optimal inverse filter block
Q	$Q = L_w/M$, number of coefficients of each block

For nonlinear systems:

Q	Number of kernels
M_q	Number of coefficients of the Volterra kernel q
i_1	Coefficients index of the first or linear Volterra kernel
i_1, i_2	Coefficients indexes of the second or quadratic Volterra kernel
$\mathcal{H}$	Nonlinear system including linear and nonlinear distortion
$\mathcal{L}$	Linear distortion of the nonlinear system
$\mathcal{N}_{\mathrm{II}}$	Nonlinear q-order distortion of the nonlinear system
$\mathbf{w}_l(i_1; n)$	Linear adaptive filter vector
$\mathbf{w}_q(i_1, i_2; n)$	Quadratic adaptive filter vector

For the frequency domain algorithm:

k	Frequency domain sample
i	Frequency index

Abbreviations and Acronyms

3D	Three dimensions
AE	Adaptive equalization
AEC	Acoustic echo cancellation
ANC	Active noise control
A/D	Analog to digital converter
BBFx	Block-based biased filtered-x
BFx	Biased filtered-x
BMFx	Biased multichannel filtered-x
CBFx	Combined block-based filtered-x
CFBFx	Combined fitted block-based filtered-x
CFx	Combined filtered-x
CMFx	Combined multichannel filtered-x
CPFBFx	Combined partitioned fitted block-based filtered-x
D/A	Digital to analog converter
DFT	Discrete Fourier transform
EMSE	Excess mean square error
FBBFx	Fitted block-based biased filtered-x
FBLMS	Frequency domain block least mean square
FFT	Fast Fourier transform
FIR	Finite impulse response
FLANN	Functional link artificial neural network
FLMS	Frequency domain least mean square
FNLMS	Frequency domain normalized least mean square
Fx	Filtered-x
GPU	Graphic processing unit
IDFT	Inverse discrete Fourier transform
IFFT	Inverse fast Fourier transform
IPNLMS	Improve proportionate least mean square
LEM	Loudspeaker-enclosure-microphone
LMS	Least mean square
LNLR	Linear to nonlinear ratio
MFx	Multichannel filtered-x
MSE	Mean square error

MIMO	Multiple-input multiple-output
MISO	Multiple-input single-output
NFx	Nonlinear filtered-x
NLMS	Normalized least mean square
NPM	Normalized projection misalignment
PFBFx	Partitioned fitted block-based filtered-x
PNLMS	Proportionate least mean square
RIR	Room impulse response
RLS	Recursive least square
SIMO	Single-input multiple-output
SISO	Single-input single-output
SNR	Signal to noise ratio
WFS	Wave Field Synthesis

Introduction and Scope 1

Introduction and Scope 1

1.1 Background

This thesis fits into the field of Information Technology and Communications, especially in the area of Digital Signal Processing. In recent years, multimedia immersive systems that allow real and reliable user experiences, including audio and image processing, are sought. With the focus on signal processing for multichannel audio reproduction systems, the main goal of this thesis is to render a desired audio signal at the listening area.

Audio rendering systems working inside a room can exhibit a behavior different to the desired one due to the acoustical room properties and the elements inside the acoustic space. Thus, the final audio signal at the listening point will contain the contribution of all these echoes, which deteriorates the three dimensional sound effect of the original sound source. In this regard, a comprehensive analysis of room acoustic properties can be found in [1],[2].

Moreover, in sound reproduction systems, the electroacoustic path involves the room enclosure and the loudspeaker and microphone devices. The basic electronic components, such as digital-to-analog (D/A) and analog-

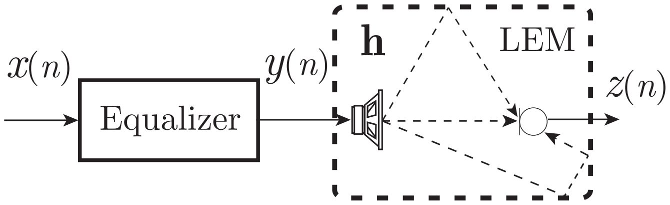


Figure 1.1. Sound equalization system.

to-digital (A/D) converters, amplifiers, loudspeakers and microphones, usually present linear responses, but when they are driven with large amplitude inputs or even when are built with cheap components, nonlinear distortions that severely degrade the audio quality can arise.

In order to remove these effects and to render the desired signal at the listening area, an equalizer is used before driving the output signal through the loudspeakers. That way, the combination of the equalizer filter and the electroacoustic path will reproduce the desired audio signal. Fig.1.1 illustrates a sound equalization system in the simplified case of a single source and a single receiver. Furthermore, this scheme will allow not only to cancel the electroacoustic system, but also to recreate a specific acoustic environment, such as a concert hall.

Those equalization filters can be implemented in two different forms: direct or adaptive. Direct equalization techniques, whether in the time [3] or frequency [4] domain, have been used for years. They usually compute the equalization filter once and at an earlier stage than the rendering one. However, real systems imply time-varying scenarios, when the room conditions change or even when the temperature varies [5], and direct filters do not properly follow those changes. On the other hand, adaptive equalization (AE) strategies allow to iteratively obtain the equalization filter while following the time-varying room conditions. Adaptive algorithms have been used in a wide range of signal processing applications for audio and communications [6].

Regarding the listening area, the equalization can be achieved in a single listening point using a single source and a single receiver (SISO system). Such a setup is the most straightforward to analyze, but real world systems normally imply multiple inputs and multiple outputs (MIMO system), where the inputs are the loudspeaker sources and the outputs the microphone devices. These MIMO systems or global control systems would obtain the desired sound field in a large zone of the space, allowing the algorithm to work independently of the listener position. For MIMO systems, the system inversion can be normally adressed with adaptive algorithms.

These strategies require a high computational burden to implement the adaptive algorithms, particularly for MIMO systems. In the last decade, multi-core processors and many-core hardware accelerators as parallel systems have been designed and used to implement real time applications for audio signal processing, which imply real immersive audio systems with a high number of transducers.

1.2 Motivation and objectives

Audio applications are designed to implement real and immersive systems. However, when rendering with loudspeakers inside an enclosure and due to the weak functionality of the transducers, the global audio effect can be lost and thus the 3D spatial sound. Due to this, equalization filters are required to keep these systems as close as possible to the desired ones. Regarding adaptive equalization, recent hardware advances allow an easy real-time implementation of adaptive equalization filters, although the rendering audio system can imply a high number of loudspeakers and listening positions.

With this in mind, the main objective of this thesis is the following:

To develop and optimize adaptive algorithms to compensate linear and nonlinear distortions of the electroacoustic path when rendering an audio signal inside an enclosure in order to obtain the desired audio signal at the listening area.

In order to achieve this major objective, the following particular scopes should be met:

- To implement adaptive algorithms in order to identify and linearize the nonlinear distortions of a loudspeaker. For this purpose, efficient nonlinear adaptive algorithms have to be developed. Particularly, novel and promising algorithms will be considered such as the nonlinear filtered-x structure using Volterra kernels. These schemes will be developed for channel identification and compensation of the nonlinear response of the loudspeaker. For this last case, a time-varying virtual channel approach will be implemented.

- To design and develop different adaptive algorithms for room equalization applications in order to compensate the electroacoustic path. For this purpose, novel combination and blocked-based schemes will be applied to the linear equalization system using a filtered-x structure.

  The equalization filters of the whole electroacoustic chain, which involves loudspeaker and the acoustic path, will be implemented using two different structures:

  - In the first scheme, referred to as p-order equalization, the process is split into two steps: Firstly, the nonlinearities of the loudspeaker will be cancel. Meanwhile, a second filter, tandemly connected to the first one, will compensate the linearities of the electroacoustic path.
  - The second structure, referred to as whole nonlinear equalization, will allow to equalize at the same time both the linearities and the nonlinearities of the audio chain, with independency of the nonlinearities of the system.

- To implement the adaptive equalization algorithms in real-time systems using current hardware.

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 adaptive signal processing and its use in audio applications, particularly for room equalization, which will be necessary to understand this dissertation. It includes, among other concepts, adaptive filtering, room equalization, and both linear and nonlinear distortion, existent in room rendering scenarios.

- **Chapter 3**. This chapter introduces the adaptive room equalization problem for single-input single-output (SISO) systems, but also for multiple-input multiple-output (MIMO) systems. Different schemes for the room adaptive equalization application are proposed for these systems, which give robustness to the equalization problem for different room conditions. In particular, the filtered-x IPNLMS algorithm is introduced to exploit the different structure capabilities.

- **Chapter 4**. This chapter introduces the block-based adaptive schemes, that can improve the performance of the structures presented in the previous chapter in the context of room equalization applications. The use of filters split into blocks is useful in audio applications due to the sparse nature of the acoustic channels, where each filter block is independently updated depending on the coefficients energy distribution. The drawback of the block-based structures is the huge computational cost required as the number of block increases.

- **Chapter 5**. This chapter presents the nonlinear adaptive equalization scheme. Acoustic devices exhibit usually a nonlinear behavior. For that reason, nonlinear equalization is required when dealing with audio rendering and recording. In this chapter, a particular solution is provided for the loudspeaker-enclosure-microphone path when it presents nonlinearities with memory.

- **Chapter 6**. 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**. In the first appendix, a real time room equalization algorithm is implemented using a multi-core processor. Particularly, the Fx-NLMS algorithm is developed in frequency domain to allow the equalization algorithm be implemented in real-time conditions.

- **Appendix B**. The second appendix shows direct multichannel equalization efficient techniques, that can be employed when rendering with a high number of loudspeaker and microphone signals. In particular, these techniques are employed in a rendering Wave Field Synthesis system.

Problem formulation and Fundamentals 2

Problem formulation and Fundamentals 2

This chapter introduces some concepts that are necessary to better understand this dissertation. An introduction to the room equalization problem is included, where acoustic room properties are described in order to successfully tackle the room inversion problem. Next section presents adaptive filtering, particularly the least-mean square algorithms. Thereby, the implementation of adaptive filtering in the context of room equalization is also introduced. Moreover, the nonlinear characteristics of the electronic devices are also presented, which also affect the acoustic transmission chain and present an additional issue to the inversion problem. Furthermore, the multichannel equalization problem is also discussed. Finally, to study the equalization system behavior, some performance measures are introduced, such as the excess mean square error (EMSE) and the normalized projection misalignment (NPM).

2.1 Room equalization

In sound rendering systems using loudspeakers, the listening room adds echoes not considered by the reproduction system, thus deteriorating the rendered audio signal and loosing the immersive effect of the audio reproduction system. Fig. 2.1 shows the impulse response between two points of an acoustic space, where each line represents the direct sound and echoes present in this acoustic channel, with a different amplitude and time delay. The first line corresponds to the direct path between the sound source and the measurement or microphone position. The first following echoes are due to the first room reflections (walls, ceiling, floor and close objects). They show high amplitude and are sparser than the last echoes, with lower energy and more dispersive behavior, which are called reverberation.

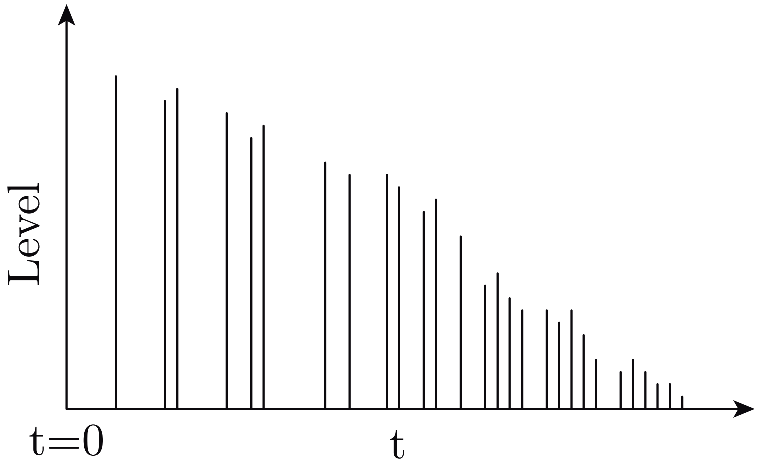


Figure 2.1. Room impulse response between two points inside an enclosure.

To measure this reverberation, the reverberation time (T_{60}) is employed, which was introduced by Sabine [7]. This parameter is defined as the period of time that takes to the level of sound to decay 60 dB from the level of the original sound when it ceases. T_{60} allows to define if a room presents a more reverberant or more 'dry' response, which gives an idea of the level of reflected sound, higher as the T_{60} increases.

The room impulse response of Fig. 2.1 can be transformed into the frequency domain. Fig. 2.2 shows a room frequency response, where the room reflections are represented by peaks and notches at different frequencies. Ideally, when rendering in free field conditions, the frequency response would show a flat response for all frequencies. On the other hand, the ideal room impulse response in free field conditions would correspond to the first echo of Fig. 2.1. In some situations, the aim of the equalizer will be not to get the ideal response, but a smoother or simple response. In fact, to achieve an ideal inverse response usually requires a high computational cost or even it would not be possible to cancel all the peaks and notches of the frequency response. For other applications, such as speech enhancement, the aim is not to cancel all the echoes but to keep the first ones, as they allow to improve the speech intelligibility.

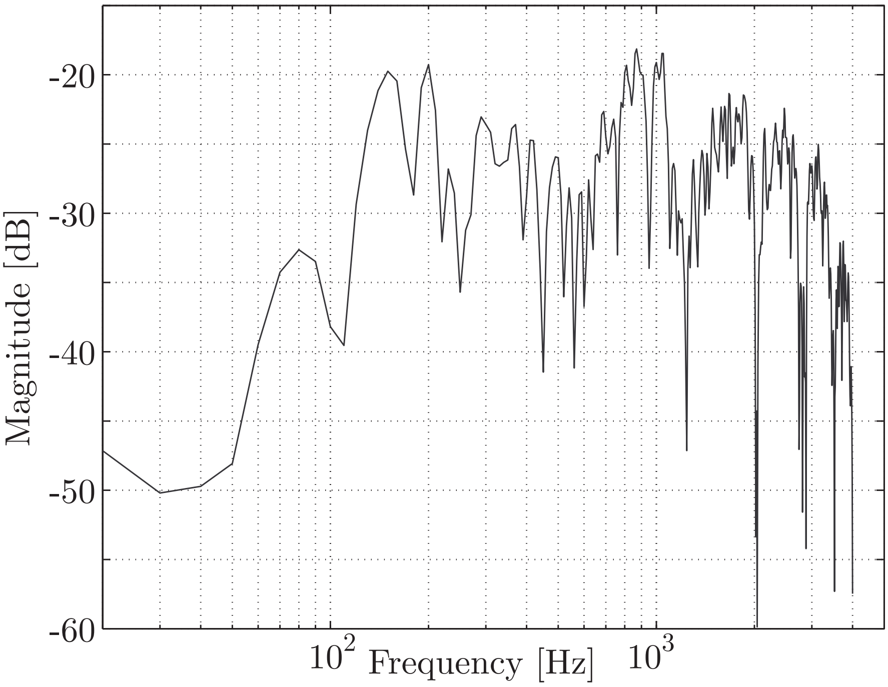


Figure 2.2. Room frequency response between two points inside an enclosure.

A deep study of the acoustic room properties can be found in [1] and [2]. To compensate these echoes, a passive canceller can be used, which consist on using absorbent material that will cancel some of the acoustic reflections. The problem of this method is that it does not work properly for low frequencies and implies to use absorbent material in all possible reflectors. Another possibility is to use active compensation, which is based on using a filter to process the acoustic input signal and rendering the new generated signals. That way, the combined effect of the filter and the acoustic path will give a desired response at the listening area. For an ideal acoustic compensation, this filter will be the inverse of the room impulse response.

2.1.1 Active compensation

For active audio compensation a filter is used in the transmission chain to invert the room impulse response (see Fig. 1.1). This inversion can be achieved using direct or adaptive strategies. Direct strategies normally compute the inverse filter once in a previous stage to the audio rendering. For its inversion, the room impulse response has to be previously measured. Room acoustic responses have normally nonminimum phases, for that reason inverse filters do not provide the exact inverse, [8]. To achieve an exact inverse filter of the acoustic impulse response in a room, a method is proposed in [3] which is based on the multiple-input/output inverse theorem (MINT). This method gets an exact inversion for multiple responses when (2.1) is satisfied

$$L_w = \frac{M\left(L_h - 1\right)}{J - M}, \tag{2.1}$$

where L_w and L_h are the finite impulse response (FIR) filter length of the inverse and the channel filters, respectively. Moreover, J is the number of loudspeakers and M the number of microphones. Thus, from (2.1) the number of loudspeakers must be higher than the number of microphones.

This condition is not always fulfilled and it requires a high computational cost for the inversion. For that reason, there exist another strategies that try to achieve an approximated solution with a more efficient inversion, as the fast deconvolution with regularization in the frequency domain [4].

On the other hand, adaptive strategies do not always converge to the exact inverse response, but tend to an optimal solution. Moreover, their implementation can be efficiently deal in the frequency domain for implementing real-time applications.

In order to obtain this optimal solution, the length of the inverse FIR filter with respect to that of the channel filter has to be chosen to ensure that the convolution between them achieves the desired compensation, but without highly increasing the number of samples. The longer the filter, the higher the computational cost requirements. A good compromise is to choose a length double than that of the acoustic channel. From (2.1) and using 2 loudspeakers an 1 microphone, the exact inversion can be obtained with $L_w = L_h - 1$, whereas a longer filter will provide also a good approximation.

Regarding the inversion of the acoustic channel, which can present nonminimum-phase responses, for both direct and adaptive methods, the desired response should be built using a proper delay to allow a good and stable inversion of the system. For that purpose, a delay close to half the length of the inverse filter is used which provides a inverse filter with a central response, that allows a two-sided stable inverse [9].

2.1.2 Sparsity degree

One of the properties of the acoustic channel is the degree of sparsity. This parameter gives an idea of the impulse response components, telling how many coefficients have a significant magnitude, whereas the rest of them are zero or small. Room channel responses are normally sparse, that means that only a small percentage of the components have high values.

This sparseness characteristic is used in the proportionated algorithms [10], that will be introduced in the next section, to take advantage on the knowledge of the coefficient energy distribution using this sparsity degree.

There are different functions that are used to measure the sparseness of the impulse response $\mathbf{h}$, [11], which are based on different ℓ_p norms for $p = 0, 1, 2$ and ∞. Where $\ell_p = \|\cdot\|_p$ of $\mathbf{h}$ is defined as,

$$\|\mathbf{h}\|_p = \left(\sum_{i=1}^{L_h} |h_i|^p \right)^{1/p} . \tag{2.2}$$

If the sparseness measure gives a value closer to 1, the impulse response is very sparse; on the contrary, the closer the measure to 0, the denser or more dispersive the impulse response. For other responses its value interpolates smoothly between these two extremes. The difference between these measures, that use a different ℓ_p norm, is the variation or transition between the 0 and 1 values. A sparseness measure that is broadly used is that based on the ℓ_1 and ℓ_2 norms [11], which is defined in (2.3). Further in this document, we will refer to these sparseness measure as $\xi(\mathbf{h})$.

$$\xi_{12}(\mathbf{h}) = \frac{L_h}{L_h - \sqrt{L_h}} \left(1 - \frac{\|\mathbf{h}\|_1}{\sqrt{L_h}\|\mathbf{h}\|_2}\right). \tag{2.3}$$

2.2 Adaptive filtering algorithms

Adaptive filters have been applied to signal processing for multiple applications, such as prediction, system identification, equalization and, noise and echo cancellation. A description of the adaptive signal processing and their applications can be found in [6]. Furthermore, a comprehensive description of the main adaptive algorithms can be read in [12].

Adaptive filters update the filter coefficients using an iterative process. Depending on the recursive algorithm they employ, they can be based on: Wiener filter and Kalman filter, which could be computed directly using a statistical knowledge of the signal. Another kind of adaptive filter uses the Least Square method, which employs a deterministic formulation to achieve the adaptive solution.

Among these adaptive methods, the least-mean square (LMS) algorithm, the recursive least-square (RLS) algorithm, the affine projection algorithm and variations and combinations of them can be employed in an equalization context, in both time and frequency domain. In this work, we focus mainly in the LMS-type algorithm.

2.2.1 Least mean-square algorithm

The LMS algorithm has been chosen in this work for its simplicity and also because it exhibits a good performance with a proper configuration. It can be implemented with a transversal filter, where the weights of the filter define its finite impulse response. This algorithm is based on the method of steepest descendent [13], which finds the minimum value of the mean squared error (MSE) $J(n) = E\left\{|e(n)|^2\right\}$, which corresponds to the mean square value of the difference between the desired response and the transversal filter output, see Fig 2.3.

Thus, each recursion follows the direction of the negative of the gradient vector leading to the minimum mean squared error, $J_{\min}$, at which point the weight vector assumes its optimum value or Wiener solution, $\mathbf{w}_{\mathrm{o}}$. μ is

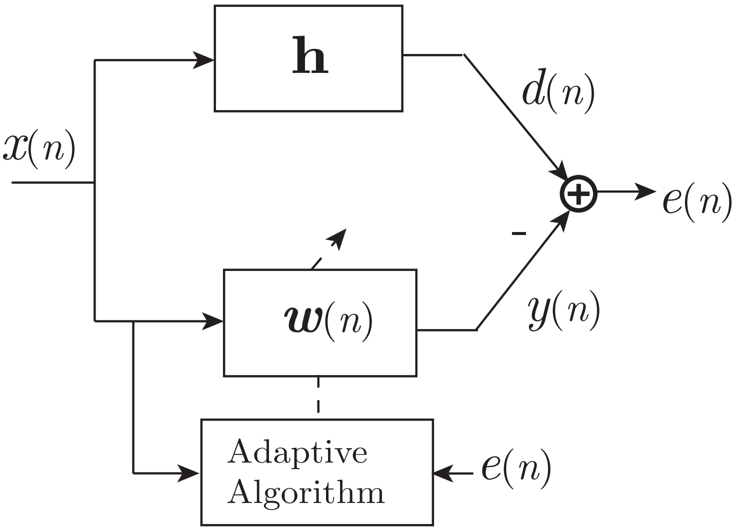


Figure 2.3. Block diagram of an adaptive filter.

the step size of the adaptive algorithm.

$$\mathbf{w}(n) = \mathbf{w}(n-1) - \frac{1}{2}\mu\frac{\partial J(n)}{\partial \mathbf{w}(n)} \tag{2.4}$$

The LMS algorithm uses as the estimator of the mean-square error its instantaneous value, $J(n) = E\left\{|e(n)|^2\right\} = |e(n)|^2$. Thus, for the adaptive algorithm of Fig 2.3, the error signal can be obtained as

$$e(n) = d(n) - y(n), \tag{2.5}$$

being $d(n)$ the output of the system and $y(n)$ the output of the adaptive filter.

The derivation of the MSE in (2.5) with respect to the filter coefficients can be obtained as

$$\frac{\partial J(n)}{\partial \mathbf{w}(n)} = -2e(n)\mathbf{x}(n), \tag{2.6}$$

where $\mathbf{x}(n)$ is a vector containing the last L_w samples of the input signal, being L_w the number of samples of the adaptive filter.

By substituting (2.6) in (2.4), the adaptive filter update can be rewritten as follows:

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \mu e(n)\mathbf{x}(n). \tag{2.7}$$

The choice of the step size μ is critical. A large value accelerates the initial convergence, but it must not be too high to prevent from divergence. On the other hand, a small value has a slow initial convergence, but allows to reduce the final excess mean square error. In order to do the algorithm independent of the input signal, the normalized LMS (NLMS) algorithm is defined, which uses a time-varying adaptation speed $\overline{\mu}(n)$ in (2.7), that has been normalized with the power of the input signal $\mathbf{x}(n)$, as

$$\overline{\mu}(n) = \frac{\mu}{\delta + \|\mathbf{x}(n)\|^2}, \tag{2.8}$$

where $0 \leq \mu \leq 2$ and being δ a small constant to avoid division by zero.

Regarding LMS-type algorithms, one of their main drawbacks is that they distribute the adaptation energy equally among all filter coefficients with μ and thus, they suffer from slow convergence speed. This applies to both the LMS and the NLMS algorithms.

2.2.2 Proportionated LMS algorithms

To overcome this problem, the proportionate adaptive filter (PNLMS) algorithm [14] has been introduced to accelerate filter convergence in scenarios where the optimal solution presents a high degree of sparsity, that means, $\xi(\mathbf{h})$ is close to 1. PNLMS spends more energy on adapting the active coefficients, thus it converges faster than the NLMS. The adaptation filter is defined by

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

$$\text{being } \boldsymbol{\mu}(n) = \frac{\mu\mathbf{G}(n-1)}{\delta + \mathbf{x}^T(n)\mathbf{G}(n-1)\mathbf{x}(n)}, \tag{2.10}$$

where $\mathbf{G}(n-1) = \text{diag}\left\{g_1(n-1), \ldots, g_{L_w}(n-1)\right\}$ is a diagonal matrix of L_w size, that adjusts the individual filter coefficients, using the following expression for each coefficient,

$$g_l(n) = \frac{\max\left\{\rho l'_{\infty}(n), |w_l(n)|\right\}}{\sum\limits_{k=0}^{L_w-1} g_k(n)}, \tag{2.11}$$

$$l'_{\infty}(n) = \max\left\{\epsilon, \max\left\{|w_1(n)|, \ldots, |w_{L_w}(n)\}\right\}, \tag{2.12}$$

where ρ and ϵ are small regulation parameters.

However, it assumes the filter solution is sparse and its performance degrades significantly when the optimal filter is not so sparse. Thus, the improved proportionate NLMS (IPNLMS) [10] tries to alleviate this problem improving filter convergence for different degrees of sparsity using a κ variable. For the IPNLMS algorithm, the elements of the diagonal matrix in (2.10), $\mathbf{G}(n-1) = \text{diag}\left\{g_1(n-1), \ldots, g_{L_w}(n-1)\right\}$, are defined by

$$g_l(n) = \frac{g_l(n)}{\|\mathbf{g}(n)\|_1} = (1-\kappa)\frac{1}{2L_w} + (1+\kappa)\frac{|w_l(n)|}{\varepsilon + 2\sum\limits_{k=0}^{L_w-1} |w_k(n)|}, \tag{2.13}$$

where ε is a small constant to avoid division by zero.

However, its major drawback is that it requires to know the degree of sparsity of the optimal solution, which rarely occurs in practical systems. In [10], a κ value between 0 and -0.5 is recommended to achieve a good behavior of the algorithm. Moreover, the IPNLMS algorithm can be seen as a generalized expression of the LMS-type algorithms, where $\kappa \in [-1, 1]$ arranges from the NLMS algorithm (for a value of $\kappa = -1$) to $\kappa = 1$ for the PNLMS algorithm.

Some successful applications of the proportionated adaptive filters include system identification [15], acoustic echo cancellation [14], ANC [16], as well as AE [17].

2.2.3 Convex combination of adaptive filters

In the last decade, there has been an interest in adaptive combination of two or more filters, where the outputs of several filters are mixed to obtain an improved overall output. These filtering schemes are introduced

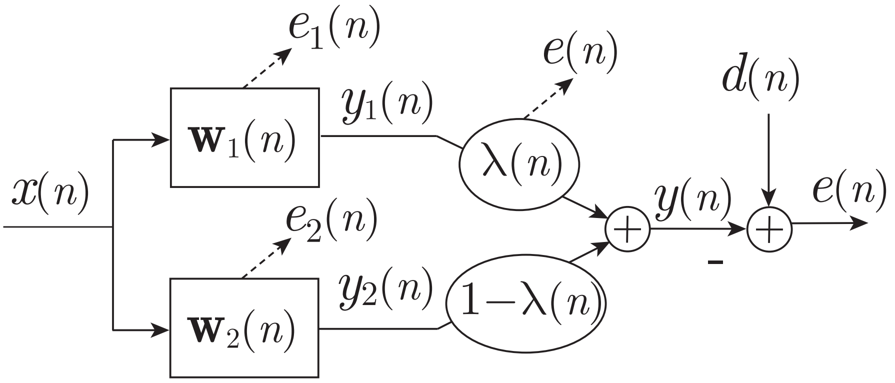


Figure 2.4. Block diagram of a convex combination of adaptive filters.

to improve robustness when several kinds of adverse scenario conditions can impair the filter performance and also looking for the filter that provides the best solution.

In this work, we focus on the convex combination, where two filters are combined with λ and $(1-\lambda)$, where $\lambda \in [0,1]$. In fact, they can alleviate the different tradeoffs to which adaptive filters are subject involving mainly speed of convergence and steady-state misadjustment.

The output of the combination filter $y(n)$ is obtained as the weighted sum of the single outputs $y_1(n)$ and $y_2(n)$, with the combination factor $\lambda(n) \in [0,1]$ for the convex combination scheme.

$$y(n) = \lambda(n)y_1(n) + [1-\lambda(n)]\,y_2(n), \tag{2.14}$$

where $\lambda(n)$ can be adapted using a sigmoid activation function,

$$\mathrm{sgm}[a(n)] = \frac{1}{1+e^{-a(n)}}, \tag{2.15}$$

where $a(n)$ is updated in order to minimize the mean square error of the combination filter, by using for instance a gradient descent method.

The error signals that are used to update each adaptive filters of the combination scheme, are obtained in order to minimize the mean square value of the difference between the desired response and the correspondent filter output as

$$e_i(n) = d(n) - y_i(n), \quad i = 1, 2, \tag{2.16}$$

where the desired response $d(n)$ for each filter is the same as that of the combined filter.

To achieve a good trade off between convergence speed and final mean-square error (MSE), the use of a combination of two LMS filters was first introduced for system identification in [18]. Such adaptive scheme has been successfully applied to other acoustic applications, such as acoustic echo cancellation [18][19] and ANC [20][21][16]. As it is shown in these works, any other algorithm can be used instead of the LMS-type algorithms, as appropriate. As an example of this, two affine projection algorithms are used in [20], and the combination of IPNLMS filters is considered in [16]. Good results are obtained also when different kinds of filters are combined [22]. Several authors have also used the combination strategy to solve the optimal filter-length search problem by using the LMS algorithm either with a sparse filter [23] or in a high noise environment [24].

2.3 Adaptive room equalization

Adaptive filtering algorithms have been broadly used for equalization applications. The adaptive schemes proposed usually consider the LMS algorithm or some variations of it, which represent a stable and simple solution. Several contributions have been recently proposed in the time, frequency or wave domain for adaptive room equalization (AE).

Whereas some authors address the problem of inverse filtering in time domain [25][17], others use the LMS adaptive algorithm in frequency domain for the previous identification of the channel response and a direct frequency or warped domain method with regularization, [26][27][28]. Also the time and the frequency domain version of a decoupled Fx-LMS algorithm are presented in [25] and [29], respectively, for multichannel equalization. Other frequency approaches try to compensate a specific frequency band as in [30]. Finally, different techniques have been developed in wave domain as those in [31][32][33] or with modal decomposition in [34], to reduce the number of adaptive filters.

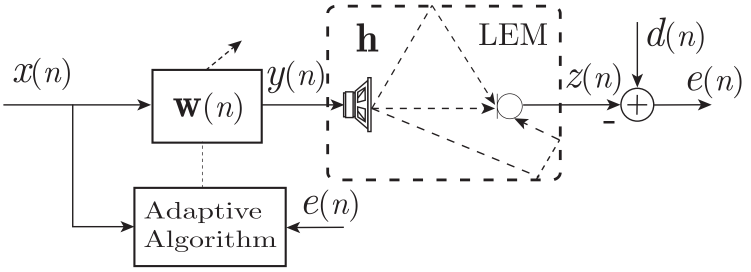


Figure 2.5. Block diagram of an acoustic audio system including equalization.

2.3.1 Filtered-x structure

A remarkable difference introduced by adaptive room equalization (AE) systems (and also by active noise control (ANC) systems) compared to a system identification problem is due to the presence of the LEM channel $\mathbf{h}$ between the adaptive filter output and the error signal. Fig. 2.5 illustrates a scheme where the adaptive algorithm is updated using the input signal $x(n)$ and the error signal $e(n)$, where $e(n)$ is obtained by subtracting the microphone response $z(n)$ from the desired response $d(n)$. Thus, $e(n)$ is obtained after propagating the output of the adaptive filter through the channel response $\mathbf{h}$. This propagation degrades the behavior of the LMS algorithm, which presents an increase in the residual error, a lowered convergence rate and can even become unstable. The usual way to take into account this response $\mathbf{h}$ and avoid negative effects on the algorithm performance consists in using a particular filtering structure named filtered-x scheme [35], that requires a previous estimation of the acoustic channel. Fig. 2.6 shows the same acoustic audio system of Fig. 2.5, but with a filtered-x structure.

This filtering scheme has been frequently used in ANC systems, [36][37][38][39]. For room equalization, a brief description of the filtered-x scheme is given in the next section.

2.3.2 Adaptive equalization

Fig. 2.6 shows the block diagram of a single channel acoustic audio equalization system with a filtered-x embedded structure, where the error signal $e(n)$ is obtained by subtracting from the desired signal $d(n)$, the input signal $x(n)$ filtered through the adaptive filter and then propagated through

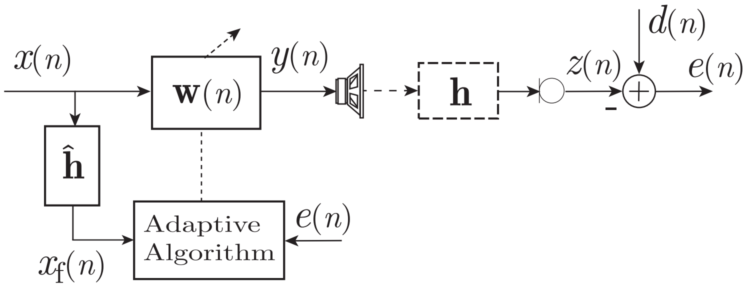


Figure 2.6. Block diagram of an acoustic audio equalization system with a filtered-x structure.

the channel response to the listening position (or microphone). For AE applications, the desired signal is formed with the input signal with a suitable source-listening point delay, $d(n) = x(n-\tau)$. Thereby, the equalizer will correspond to the inverse of the LEM channel. The acoustic signal measured at the microphone can be modeled as

$$z(n) = \mathbf{h} * y(n), \tag{2.17}$$

where $*$ denotes the discrete linear convolution.

For the LMS-type algorithm, the derivation of the MSE, $J(n)$, with respect to the adaptive filter coefficients for AE can be approximated by

$$\frac{\partial J(n)}{\partial \mathbf{w}(n)} = -2e(n)\mathbf{x}_{\mathrm{f}}(n), \tag{2.18}$$

where the f-term of the $\mathbf{x}_{\mathrm{f}}(n)$ vector refers to the filtered-x structure. In order to implement this filtering, a previous estimation of the acoustic channel is needed, thus the input signal is filtered through this estimated acoustic channel $\widehat{\mathbf{h}}$.

The filtered-x version of the NLMS algorithm is called Fx-NLMS filter. The Fx-NLMS weights are updated at each iteration from (2.4) and (2.18) according to

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \overline{\mu}(n)e(n)\mathbf{x}_{\mathrm{f}}(n), \tag{2.19}$$

where $\mathbf{w}(n)$ is the adaptive weight vector of L_w-length. $\mathbf{x}_f(n)$ is a vector containing the last L_w samples of the input signal $x(n)$ filtered through the L_h-length estimated impulse response $\hat{\mathbf{h}}$. Furthermore, the adaptation speed for each filter weight of the Fx-NLMS algorithm has been normalized as in (2.7),

$$\overline{\mu}(n) = \frac{\mu}{\delta + \|x_{\mathrm{f}}(n)\|^2}, \tag{2.20}$$

where δ is a small constant to avoid division by zero.

For the Fx-IPNLMS algorithm, the adaptation of the filter coefficients $w_l(n)$ is obtained as

$$w_l(n) = w_l(n-1) + \mu_l(n)e(n)x_{\mathrm{f}}(n-l),\ l = 0, \ldots, L_w - 1, \tag{2.21}$$

$$\text{being } \mu_l(n) = \frac{\mu g_l(n-1)}{\delta + \sum\limits_{k=0}^{L_w-1} g_k(n-1)x_{\mathrm{f}}^2(n-k)}, \tag{2.22}$$

where $g_l(n-1)$ is the adaptation gain factor of the lth filter coefficient as defined in (2.13).

2.3.3 Frequency domain

One of the first adaptive filtering works in frequency domain can be found in [40], where the LMS algorithm is transformed into the frequency domain, FLMS. This approach improves the computational cost when the number of taps of the adaptive filter is high, which occurs in equalization applications. This reduction in the computational cost allows to implement this algorithms in real time. Since this first contribution, different papers in frequency domain have been presented as it reduces the computational burden, but also presents a higher convergence speed using the orthogonality of the Fourier transform. All these works try to optimize one of the two indicated properties.

In order to implement real-time applications using audio devices, continuous signals have to be sampled to obtain a discrete signal, which can be processed. Sound cards allow this conversion from analog to discrete signals and vice versa, using a sampling frequency. Moreover, they operate with signals, recording and rendering them, using block of samples. Thus, block-operations are inherent to sound cards. For that reason, the developed algorithms are efficiently performed when adapted to block-update equations. That means, that the filter is updated not sample by sample, but after a block of samples.

Thus, although the different schemes presented in this thesis are handled in time domain, a transformation to the frequency domain is necessary if we want to implement this schemes with real-time performance.

There are different possibilities to implement the frequency domain Fx-LMS algorithm. As they work with block of samples, one implementation consists in updating the filter weights after receiving L samples, being L the block size. This way, the algorithm has to wait L samples before to compute the new filter. This method is called block LMS algorithm in the frequency domain, FBLMS. Moreover, the block size used in the algorithm is important for real-time applications. On the one hand, if it is too small, the time to take input samples, process them and send the output samples to the sound card would be not enough to allow all these operations to be performed. On the other hand, if the block size is too long, the latency of the algorithm could be too long for some real-time applications. Moreover, if the sound source moves or the acoustic scenario changes, the algorithm will take more time to adjust the new values or even some information could be lost. For that reason, this block size should be long enough to be able to process the AE algorithm, but small enough to reduce the latency.

The linear convolution is related with the filtering process of audio signals in time domain. The convolution theorem states that the circular convolution of two periodic sequence can be computed by using the Discrete Fourier Transform (DFT) [41]. This theorem states that sequence y is obtained as the inverse discrete Fourier transform of Y, which is obtained as the element-wise multiplication of the DFTs of x and h, X and H.

$$\begin{aligned} X &= \mathrm{DFT}(x), \\ H &= \mathrm{DFT}(h), \\ Y &= X \otimes H, \\ y &= \mathrm{iDFT}(Y) \end{aligned} \tag{2.23}$$

where $\otimes$ represents element-wise multiplication.

Convolution theorem can be also applied to linear convolution. To this end, both sequences, x and h must be zero-padded up to a size of $l_x + l_h - 1$ as a minimum. In this case, the result of the linear convolution matches with the result of the circular convolution. To work in frequency domain, the fast Fourier transform (FFT) is employed to effectively compute the discrete Fourier transform using a power-of-two size samples. The inverse FFT (iFFT) is used for the inverse Discrete Fourier Transform. If we want to apply FFT to the sequences, then the zero-padding length must be increased from $l_x + l_h - 1$ to the following power of two [41].

Furthermore, when dealing with sequences longer than the block size, as happens with AE applications, the *overlap-save* or *overlap-add* methods have to be employed to properly address the linear convolution [42]. These methods split the input signal in block sizes and operate with them, first overlapping and then discarding or adding samples, respectively. However, the overlap-save method is usually selected in real-time applications, since the overlap-add method involves an additional step of summing up the results with the previous block.

In appendix A the coefficients of the adaptive filter are updated with the LMS method in the frequency domain, [43]. Particularly, the overlap-save method has been used. Initially, the block size of the input signal has been chosen as that of the adaptive filter, since it presents a lower computational burden, [44]. Moreover, the employed overlapping factor is 50%, as it is considered the most efficient one (fast block LMS [45]).

2.4 Nonlinear distortion

In the context of audio processing, the acoustic signals can be a priori assumed to have a linear relation f in (2.24). However, due to the audio devices involve in audio processing, f can be best expressed as a nonlinear function when the output signal is not linearly related with the input signal

$$y(n) = f(x(n)). \tag{2.24}$$

The basic components of sound reproduction systems, such as digital-to-analog (D/A) and analog-to-digital (A/D) converters, amplifiers, loudspeakers and microphones, usually present linear responses, but when they are driven with large amplitude inputs, nonlinear distortions that severely degrade the audio quality can arise. Fig. 2.7 shows the acoustic equalization transmission chain including these components. For audio equalization, if the system has a linear behavior, an adaptive linear filter can properly compensate for the distortion (e.g. [46; 17]). However, if the system exhibits nonlinearities, a linear filter performs poorly and the nonlinear distortion must be accounted for in the design of the equalization system.

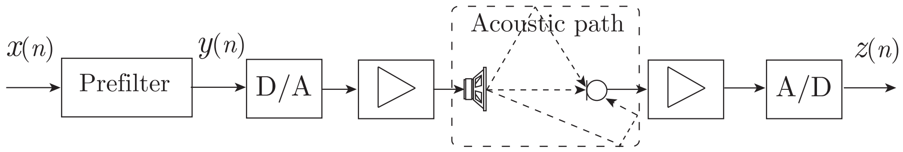


Figure 2.7. Acoustic equalization transmission chain.

The nonlinear distortion can potentially occur in each of the system components (see Fig. 2.7). Typically, the main sources of nonlinearities occur in the first blocks, that represents the D/A converter, the loudspeaker and its amplifier, and are due to the high input signal levels and the loudspeaker physical properties [47][48].

Nonlinear distortion is a topic of huge interest, that arranges from identification and elimination of nonlinearities of a loudspeaker [49] [50], to more specific applications, which require a further development to achieve a desired solution when the system exhibits nonlinearies.

In order to model the nonlinearities of a loudspeaker, some direct or adaptive methods using the measured physic (mechanical and electrical) properties of the loudspeaker have been proposed, using the Mirror filter in [47] and comparing different methods in [48].

Without the measured physic properties of the loudspeaker and with regard to audio signal processing applications, two adaptive methods to modelled nonlinearities can be highlighted: functional link artificial neural network (FLANN) filters and Volterra filters.

FLANN filter is based on a single layer of a multilayer neural network [51] and was initially proposed in [52]. This method expands the input signal $x(n)$ using a set of basis functions. Different basis functions have been employed, such as trigonometric, Chebyshev, Legrende or Lagrange. Standard FLANN filter strategy does not use products of input samples with different time shifts, thus, its performance can be deteriorated in some situations. To alleviate this problem, different modifications, such as the generalized FLANN (GFLANN) [53], the completed FLANN (CFLANN) [54] and finally the Fourier nonlinear (FN) filter [55], have been proposed with application to nonlinear active noise control. However, when used in room equalization, the nonlinear expansion of the input signal of order P produces $M = 2P+1$ functions that have to be filtered through the estimated acoustic channel for its use in the nonlinear filtered-x algorithm.

On the other hand, nonlinearities have been shown to be effectively modelled using a particular case of polynomial filters, the truncated Volterra series [56]. The output of these Volterra filters or kernels is given by the contribution of each Volterra kernel $h_q(i_1, \ldots, i_q)$

$$y(n) = \sum_{q=1}^{Q} \sum_{i_1=0}^{M_q-1} \cdots \sum_{i_q=0}^{M_q-1} h_q(i_1, \ldots, i_q) x(n-i_1) \cdot \ldots \cdot x(n-i_q), \qquad (2.25)$$

where $x(n)$ is the input signal to the nonlinear systems and M_q represents the memory length of each Volterra kernel.

The drawback of the Volterra series is the increasing number of coefficients required for higher-order kernels, which implies a huge computational cost. For that reason, only second-order and third-order Volterra filters are usually implemented [57][58].

The idea of using FLANN filters for acoustic echo cancellation (AEC) has appeared in [59][60][61] and for ANC in [62][63][54][64][65]. In [65] a comprehensive review of ANC works with emphasis on nonlinear methods is included. On the other hand, numerous authors have proposed Volterra filters for different audio applications, such as, [66][67][68] for echo cancellation and [69][70][71] for ANC. Regarding the computatuional burden of Volterra series, in [72] a simplified Volterra filters for AEC is presented by setting to zero some coefficients far from the main diagonal of the quadratic kernel. Moreover, adaptive algorithms are also proposed to compute the number of coefficients of the quadratic kernel using a combination of different sizes of the filters, estimating both the optimum length and the number of necessary diagonals of a quadratic Volterra kernel [73][74].

Concerning the general inversion problem of nonlinear transmission systems, such as communication and audio channels, different strategies have been proposed in [75][76]. As regards to the particular nonlinear room equalization problem, recently and in the best of our knowledge, no work has been reported in the literature in this context. In [77] a nonlinear signal processing method for designing equalizing filters is proposed, which uses the fuzzy c-means clustering technique. It designs minimum phase equalizing filters from the room response prototypes, and reports good results by suppressing the resonant peaks in the room transfer functions.

2.4.1 p-order method

The nonlinear acoustic system $\mathcal{H}$ in Fig.2.8, can be modeled with a linear filter $\mathbf{L}$ and nonlinear filters (denotes as N). The nonlinear inverse system employed in [76] is based on the p-order method [49], which uses a tandem connection of filters to linearize a system up to this p-order. That means, to eliminate the nonlinearities ($\mathbf{N}$) of a system up to the p-order when a p-order filter [50][58] is implemented.

This scheme adaptively identifies the linear and nonlinear components of the nonlinear system and the inverse of the linear filter. Then, it copies the nonlinear filter and the inverse of the linear filter in the preprocessor block, as can be seen in Fig. 2.8.

Adaptive linearization of loudspeakers has been widely studied using this p-order preprocessor strategy [49]. Fig. 2.8 shows the adaptive linearization scheme for a 2-order preprocessor filter.

Assuming that the following constraints are hold:

- The convolution between $\mathbf{L}$ and its inverse $\mathbf{L}^{-1}$ should correspond to a delay $\delta > 0$ so that the adaptive filter can converge.
- The assumption that the nonlinearity is weak must be fulfilled,

$$\big|x(n-\delta)\big| \gg \left|\mathbf{L}^{-1}\Big[N\Big[N\big[x(n)\big]\Big]\Big]\right|. \tag{2.26}$$

The microphone signal $z(n)$ corresponds to the input signal with a certain delay $x(n-\delta)$ filtered through the linear component of the acoustic system $\mathbf{L}$ [49]

$$z(n) = \mathbf{L}\Big[x(n-\delta)\Big]. \tag{2.27}$$

As the preprocessor filter is based on both the perfect identification of the linear and nonlinear filters and the inversion of the linear block, this

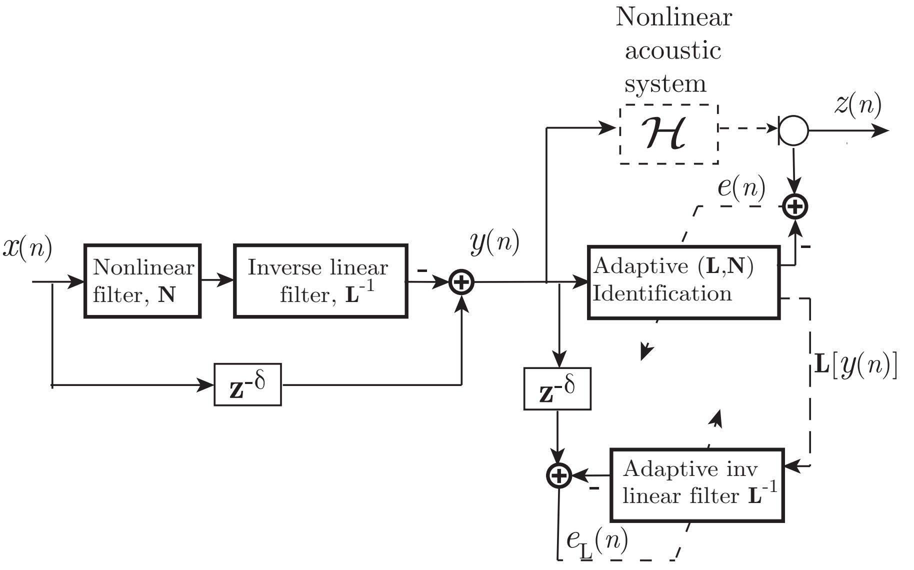


Figure 2.8. Adaptive linearization using a 2-order preprocessor filter.

method is very sensitive to channel estimation misadjustment. On the other hand, if the nonlinearities are not weak, this method produces high-order nonlinearities.

A first attempt to equalize nonlinear systems could be to use this linearization scheme with a previous filter block that includes the inverse of the linear filter [50], see Fig. 2.9, which can be called p-order equalization. If the equalization is perfectly achieved the signal at the listener point will correspond to the input signal with a delay 2δ, from (2.27) we get

$$z(n) = \mathbf{L}\Big[u(n-\delta)\Big] = x(n-2\delta). \tag{2.28}$$

This equalization method, based on the p-order strategy, has the advantage that it does not require to use the filtered-x structure. However, as stated before, the linearization or equalization of the acoustic system is not always successfully achieved. For that reason, a nonlinear filtered-x scheme is presented in Section 5 for loudspeaker and room equalization applications.

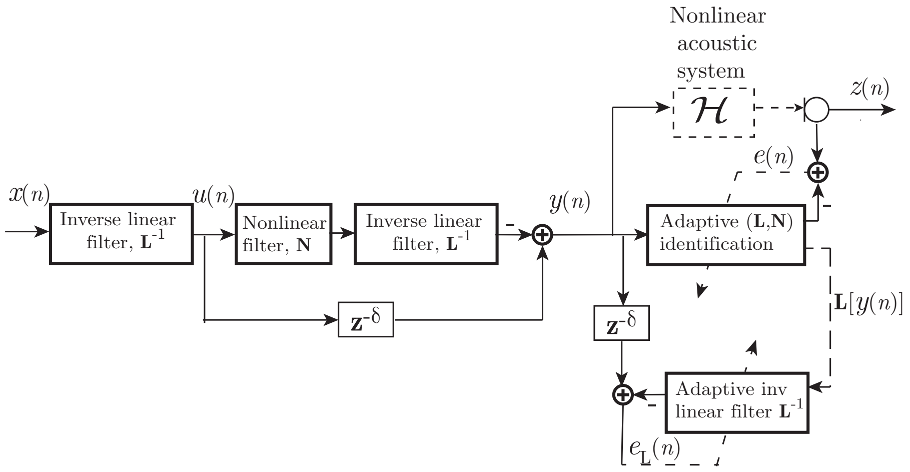

Nonlinear acoustic system
$\mathcal{H}$
$z(n)$
$e(n)$
$x(n)$
$u(n)$
$y(n)$
Inverse linear filter, $\mathbf{L}^{-1}$
Nonlinear filter, $\mathbf{N}$
Inverse linear filter, $\mathbf{L}^{-1}$
Adaptive ($\mathbf{L}$,$\mathbf{N}$) identification
$\mathbf{z}^{-\delta}$
$\mathbf{z}^{-\delta}$
$\mathbf{L}[y(n)]$
Adaptive inv linear filter $\mathbf{L}^{-1}$
$e_{\mathbf{L}}(n)$
-
-
-
-

Figure 2.9. Adaptive equalization using a 2-order preprocessor.

2.5 MIMO systems

As commented in Section 1, when the audio processing system tries to control a single point inside a listening area using a loudspeaker, we talk about a single-input single-output (SISO) system, or with multiple loudspeakers, of multiple-input single-output (MISO) systems. Whereas, the use of multiple sensors and only one loudspeaker, single-input multiple-output (SIMO) system, allows to control a larger area. The general system is that composed of multiple-input multiple-output (MIMO) system.

Thus, when a SISO or MISO systems are used, the control area is reduced to the recording sensor and a $\lambda/10$-area around this point, where $\lambda = c/f$ is the signal wavelength, being $c = 343$ m/s the speed sound at room temperature and atmospheric pressure, and f corresponds to the frequency of the signal source. On the other hand, global (MIMO) systems allow to control a wide listening area, achieving a real and immersive spatial sensation.

2.5.1 Multichannel rendering techniques

Several $3D$ audio reproduction systems are currently being used. For instance, binaural and transaural techniques are based on the human auditory system to provide all-round images and reverberation. On the other hand, multichannel audio systems, such as 5.1, 6.1 and 7.1 channel systems, try to re-create these types of three dimensional acoustics sensation. Alternatively, Ambisonics or Virtual Surround Panning are more advanced than the typical surround systems, and are adequate for less restricted areas. Ambisonics encodes sounds from all directions in terms of sound pressure and velocity components, and decodes these signals to a number of loudspeakers. Generally speaking, the solution to increase the listener's area size of these systems guides to raise the number of loudspeakers, thus increasing the complexity and difficulty. [78] presents a comprehensive description of the spatial audio properties and these multichannel rendering systems.

Nowadays, one of the most promising audio reproduction system is the Wave Field Synthesis (WFS, [79][80]), where sound field is synthesized in an wide area by means of arrays of loudspeakers. This system is based on the Huygens principle, which states that the wavefront radiated by a source behaves like a distribution of secondary sources that are placed in the wavefront up to a certain aliasing frequency. This physical principle allows WFS to synthesize an acoustic field with high quality using a high number of loudspeakers. Appendix **B** shows the synthesis operator of each loudspeaker. Whereas, a detailed description of the WFS theory can be found in [81].

2.5.2 Multichannel equalization

Some of the main problems to implement these multichannel rendering systems are related to the interaction of the loudspeakers with the listening room. The listening room distortion alters the synthesized sound field and reduces the spatial effect, and thus, the promised potentiality of these systems. Since last years, inverse filtering and equalization of multichannel acoustic systems is becoming a field of growing interest. This is mainly due to the upcoming applications of audio reproduction systems such as three-dimensional audio, or active noise control techniques, and the availability of new technology resources which allow to implement more complex signal processing algorithms. Realism of rendered sound and spatial sensation are sought.

Active multichannel equalization can be implemented applying direct solutions that solve the inverse filtering problem using a bank of multichannel filters. In the MIMO system, the input signals correspond to the loudspeakers and the output channels to the microphone or listening points, where the room responses are equalized. Various papers can be found in time and frequency domain to equalize multichannel room responses when rendering with WFS [82][83][84].

Figure 2.10 shows a diagram of a typical multichannel inversion problem. The inverse filtering problem in practical multichannel audio reproduction systems basically consists of designing a matrix **H** of digital finite duration filters (each column of **H** represents a different vector of filters for each signal to be rendered), whose convolutions with the signal transmission channels (matrix **C**), or electroacoustic system matrix, best approximates a desired response (matrix **A**).

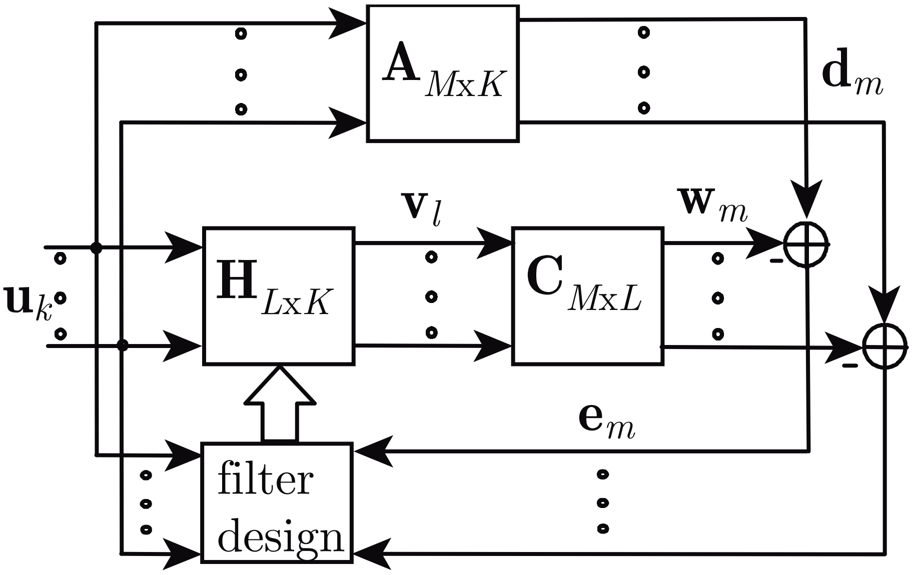


Figure 2.10. Multichannel inverse filtering problem.

Figure 2.10 illustrates a multichannel deconvolution problem where matrix **C** represents the actual transmission channels and matrix **H** is the bank of inverse filters used for deconvolution. The K source signals to be rendered are represented by $\mathbf{u}_k$, the L signals that feed the transducers are denoted as $\mathbf{v}_l$ and the K desired signals at the control points are $\mathbf{d}_m$. The difference between the received signals, represented by $\mathbf{w}_m$, and the desired signals are named error signals and denoted by $\mathbf{e}_m$. Driving signals pass through inverse filters prior to feed the transmission channels. This configuration is typical in multichannel sound reproduction systems where inverse filters are usually calculated by the least squares method in time domain [85].

This problem of computing the inverse bank of FIR filters for MIMO system in signal processing, is commonly called multichannel deconvolution. Appendix **B** presents the work carried out to develop this MIMO equalization, where the multichannel deconvolution has been developed using efficient methods to reduce the computational burden.

Due to the high computational cost of these schemes, another methods propose to achieve this multichannel active equalization from a wave domain point of view. They transform the MIMO responses from the space-time domain to the plane wave domain using a Fourier transform [31][86][32]. Thus, reducing the MIMO dimensions. These strategies also try to adaptively compute these inverse filters taking into account the time-varying acoustics conditions. As instance, a study of the time-varying room impulse response can be found in [5], when the room conditions change or even with variations of the room temperature. In [87], an adaptive method based on radiations modes is applied to both WFS and Ambisonics rendering systems.

On the other hand, adaptive multichannel equalization systems are also recently found in the time and frequency domain. For SIMO systems in [88][27] and for MIMO systems [29][17]. In [89], an equalization MIMO system is developed using Graphic Processing Units (GPUs) to parallelize the computational cost.

2.6 Performance measures

In order to measure the performance of the proposed methods employed in room equalization problems, different indexes can be used. The excess mean square error (EMSE) is very used in the context of adaptive filtering applications. For room equalization it shows the difference between the desired signal and the signal measured at the microphone. On the other hand, the normalized projection misalignment (NPM) quantifies directly how the adaptive filter tends to the inverse channel, thus its convolution with the channel response converges to the ideal response. Finally, the subjective evaluation concept is introduced.

2.6.1 Excess mean square error

The mean square error (MSE) is the mean-square value of the error signal $e(n)$, which is given in (2.5) for room equalization. The excess MSE is defined by subtracting the noise signal $r(n)$ in (2.17) from (2.5). Thus, it is defined as

$$\mathrm{EMSE}(n) = E\left\{[e(n) - r(n)]^2\right\} \tag{2.29}$$

2.6.2 Normalized projection misalignment

The normalized projection misalignment (NPM) index [90] have been adapted from blind channel identification to channel equalization and is given by

$$\mathrm{NPM}(n) = 20\mathrm{log}_{10}\left[\frac{\|(\mathbf{w}(n) * \mathbf{h}) - (\mathbf{w}_o(n) * \mathbf{h})\|_2}{\|(\mathbf{w}_o(n) * \mathbf{h})\|_2}\right], \tag{2.30}$$

being $\|\cdot\|_2$ the 2-norm. Moreover, $\mathbf{w}(n)$ is the inverse filter at time n and $\mathbf{h}$ the acoustic channel. Where $\mathbf{w}_o(n) * \mathbf{h} = \|\delta(n-\tau)\|_2 = 1$ is we consider an ideal cancellation and the optimal filter tends to the exact inverse of the room reponse.

This measure shows the difference between the inverse filter and the channel convolution with regard to the ideal response $\delta(n-\tau)$. This is, how "well" the convolution filter converges to the ideal response. The lower its value, the better its performance.

2.6.3 Subjective evaluation

These previous indexes are objective measures that give an objective value of the algorithm performance. However, the human auditory system has a complex and nonlinear behavior. As instance, [91] shows that the human ear can tolerate some kind of distortion, depending on the level and characteristics of the audio signal, ambient noise and even the listener. Thus, when the study involves rendering audio signals, also a subjective analysis has to be taken into account, since the objective measure can not provide the proper evaluation for the audio rendering immersion or sensation.

Different subjective evaluation for inverse filtering of linear systems can be found in [27][28][92][93][94].

3 Adaptive room equalization

Adaptive room equalization 3

This chapter presents different schemes to solve the room equalization problem. First and as the underlying structure, the single channel adaptive filtering algorithm introduced in Section 2.3 for room equalization applications is studied in this chapter. Moreover, an adaptive equalization scheme is presented that provides a robust performance when room rendering properties present different conditions that can impair the filter behavior. This scheme corresponds to an adaptive combination of filters, where the combined factor seeks minimizing the mean square error of the total equalizer. Furthermore, a biased equalization structure is proposed that shows a good performance under low SNR conditions.

Additionally, the adaptive room equalization problem is also extended to the multichannel case. The MIMO formulation allows to apply the previous single case to real situations, which normally imply more than one source signal and listening position.

3.1 Single channel equalization

For clarity, we will rewrite here the single channel case of the Fx-IPNLMS algorithm introduced in Section 2.3 for the adaptive equalization problem.

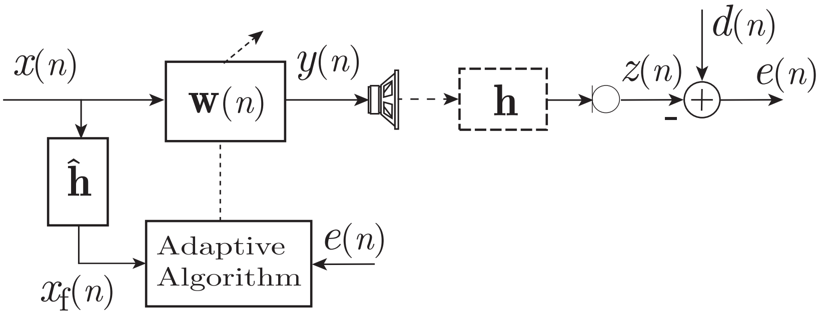


Figure 3.1. Block diagram of an acoustic audio equalization system with a filtered-x structure.

The output of the adaptive filter $y(n)$ can be expressed as:

$$y(n) = \mathbf{w}^T(n-1)\mathbf{x}(n), \tag{3.1}$$

where $\mathbf{w}(n) = [w_0(n), w_1(n), \ldots, w_{L_w-1}(n)]^T$ is the weight vector of L_w-length, and $\mathbf{x}(n) = [x(n), x(n-1), \ldots, x(n-(L_w-1))]^T$ includes the last L_w samples of the input signal $x(n)$. The signal measured at the microphone $z(n)$ is subtracted from the desired signal in order to obtain the error signal $e(n)$ which will be used to update the adaptive filter weights,

$$z(n) = \mathbf{h} * y(n) \tag{3.2}$$

$$e(n) = d(n) - z(n). \tag{3.3}$$

For a perfect cancellation of the room echoes, the desired signal $d(n)$ corresponds to the input signal with a suitable source-microphone delay,

$$d(n) = x(n-\tau). \tag{3.4}$$

The coefficients of the Fx-IPNLMS algorithm are updated at each iteration according to

$$w_l(n) = w_l(n-1) + \mu_l(n)e(n)x_{\mathrm{f}}(n-l), \text{ for } l = 0, \ldots, L_w - 1, \tag{3.5}$$

where $x_{\mathrm{f}}(n)$ is the input signal $x(n)$ filtered through the estimated impulse response $\hat{\mathbf{h}}$, which is obtained using the vector product $\mathbf{x}_{\mathbf{h}}^T(n)\hat{\mathbf{h}}$, where $\mathbf{x}_{\mathbf{h}}(n)$ is a vector column containing the last L_h samples of the input signal $x(n)$.

Furthermore, the adaptation speed for each filter weight, with μ being the step size, is computed as:

$$\mu_l(n) = \frac{\mu g_l(n-1)}{\delta + \sum_{k=0}^{L_w-1} g_k(n-1) x_{\mathrm{f}}^2(n-k)}, \tag{3.6}$$

with the adaptation gain factors given by

$$g_l(n) = (1-\kappa)\frac{1}{2L} + (1+\kappa)\frac{|w_l(n)|}{\varepsilon + 2\sum_{k=0}^{L_w-1} |w_k(n)|}, \tag{3.7}$$

where δ and ε are small constants to avoid division by zero, and $\kappa \in [-1, 1]$ arranges from an Fx-NLMS algorithm for $\kappa = -1$ to an Fx-PNLMS algorithm for $\kappa = 1$, where the adaptation is proportional to the absolute value of each filter weight.

3.2 Combined filtered-x scheme

The convex combination of two adaptive filters can combine two filters with complementary capabilities so that the overall performance of the global adaptive filter is at least as good as the performance of the component filters working separately [18].

The convex combination scheme apply to the equalization problem is illustrated in Fig. 3.2. When the two algorithms employed are the Fx-IPNLMS, the benefits of this approach are twofold. On the one hand, it allows to improve the robustness of the Fx-IPNLMS mainly in terms of convergence speed for different degrees of sparsity of the optimal filter when used with different κ values. On the other hand, and as any other gradient-base adaptive filters, it achieves a good tradeoff between convergence speed and steady-state behavior when used with different step sizes.

The output of the parallel filter $y(n)$ is obtained as the weighted sum of the single outputs $y_1(n)$ and $y_2(n)$,

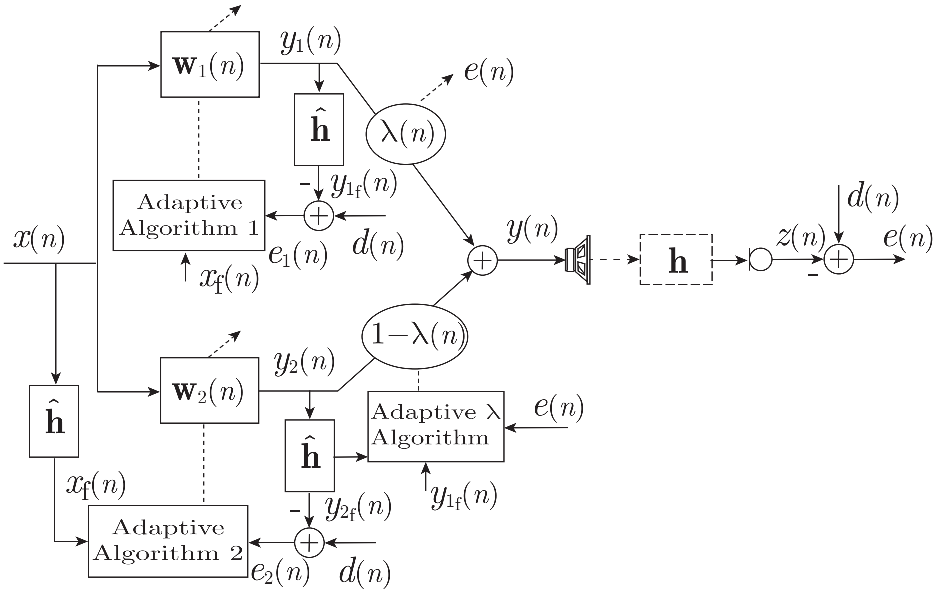


Figure 3.2. Block diagram of convex combination of filtered-x adaptive filters.

$$y(n) = \lambda(n)y_1(n) + \Big(1 - \lambda(n)\Big)y_2(n), \tag{3.8}$$

being $\lambda(n) \in [0, 1]$ the mixing parameter which is defined by using a sigmoid activation function

$$\lambda(n) = \frac{\text{sgm}[a(n)] - \text{sgm}[-a^+]}{\text{sgm}[a^+] - \text{sgm}[-a^+]}, \tag{3.9}$$

$$\text{being sgm}[a(n)] = \frac{1}{1 + e^{-a(n)}}, \tag{3.10}$$

where $a(n)$ has been restricted to the interval $[-a^+, a^+]$ [95] and it is updated in order to minimize the instantaneous square error of the overall filter, $J(n) = e(n)^2$, by using the gradient descent method. Thus, $a(n)$ is given by the following normalized LMS adaptation rule,

$$
\begin{aligned}
a(n+1) = a(n) - \frac{\overline{\mu}_a}{p(n)} \frac{\partial e^2(n)}{\partial a(n)} = \\
a(n) + \frac{\overline{\mu}_a}{p(n)} e(n) \Big[y_{1\mathrm{f}}(n) - y_{2\mathrm{f}}(n) \Big] sgm[a(n)] \Big\{ 1 - sgm[a(n)] \Big\},
\end{aligned}
\tag{3.11}
$$

being $\overline{\mu}_a$ the step size parameter for the combination factor,

$$
\overline{\mu}_a = \frac{\mu_a}{sgm[a^+] - sgm[-a^+]},
\tag{3.12}
$$

and $y_{i\mathrm{f}}(n)$ ($i = 1, 2$ for each component adaptive filter) corresponds to the filter output signal $y_i(n)$ filtered through the estimated impulse response $\hat{\mathbf{h}}$, whose substraction plays the role of the input signal to the adaptive combination factor. Moreover $p(n)$ is an estimate of its power obtained that could be obtained from

$$
p(n) = \beta p(n-1) + (1-\beta) \Big[y_{1\mathrm{f}}(n) - y_{2\mathrm{f}}(n) \Big]^2,
\tag{3.13}
$$

being β a forgetting factor defined between 0 and 1, that we have set close to one ($\beta = 0.9$) according to [96].

The error signal of each adaptive filter is obtained by subtracting from the desired signal its output filtered through the estimated channel

$$
e_i(n) = d(n) - \hat{\mathbf{h}} * y_i(n) = d(n) - y_{i\mathrm{f}}(n), \quad \text{for } i = 1, 2.
\tag{3.14}
$$

Thus, the updated filters in (3.5) can be rewritten as

$$
w_{i,l}(n) = w_{i,l}(n-1) + \mu_{i,l}(n) e_i(n) x_{\mathrm{f}}(n-l),
\tag{3.15}
$$

$$
\mu_{i,l}(n) = \frac{\mu_i g_{i,l}(n-1)}{\delta + \sum\limits_{k=0}^{L_w - 1} g_{i,k}(n-1) x_f^2(n-k)},
\tag{3.16}
$$

$$g_{i,l}(n) = (1-\kappa_i)\frac{1}{2L_w} + (1+\kappa_i)\frac{|w_{i,l}(n)|}{\varepsilon + 2\displaystyle\sum_{k=0}^{L_w-1}|w_{i,k}(n)|}, \tag{3.17}$$

for $l = 0, \dots, L_w - 1$ and $i = 1, 2$.

It has to be noticed that, for linear systems, the error combination corresponds to the overall error signal obtained from the proposed scheme as follows,

$$\begin{aligned} e(n) &= \lambda(n)e_1(n) + \Big(1-\lambda(n)\Big)e_2(n) \\ &= \lambda(n)\Big(d(n) - \hat{\mathbf{h}} * y_1(n)\Big) + \Big(1-\lambda(n)\Big)\Big(d(n) - \hat{\mathbf{h}} * y_2(n)\Big) \\ &= d(n) - \Big[\lambda(n)y_1(n) + \Big(1-\lambda(n)\Big)y_2(n)\Big] * \hat{\mathbf{h}} = d(n) - z(n). \end{aligned} \tag{3.18}$$

The combined Fx-IPNLMS (CFx-IPNLMS) algorithm is described in **Algorithm 1**, where diag$(\cdot)$ is a diagonal matrix with the elements defined in the vector $(\cdot)$.

Algorithm 1 CFx-IPNLMS algorithm.

```
Input: Reference signal $x(n)$, desired signal $d(n)$ and microphone signal $z(n)$
Output: Output of the parallel filter $y(n)$
Update the vectors $\mathbf{x}(n)$ and $\mathbf{x}_{\mathrm{h}}(n)$
$x_{\mathrm{f}}(n) = \mathbf{x}_{\mathrm{h}}^{T}(n)\widehat{\mathbf{h}}$
$y_i(n) = \mathbf{w}_i^{T}(n-1)\mathbf{x}(n), \quad i = 1, 2$
Update the vectors $\mathbf{x}_{\mathrm{f}}(n)$, $\mathbf{y}_1(n)$ and $\mathbf{y}_2(n)$
$y_{i\mathrm{f}}(n) = \mathbf{y}_i^{T}(n)\widehat{\mathbf{h}}, \quad i = 1, 2$
$e(n) = d(n) - z(n)$
$e_i(n) = d(n) - y_{i\mathrm{f}}(n), \quad i = 1, 2$
$p(n) = \beta p(n-1) + (1-\beta)[y_{1,\mathrm{f}}(n) - y_{2,\mathrm{f}}(n)]^2$
$\overline{\mu}_a = \dfrac{\mu_a}{sgm[a^+] - sgm[-a^+]}$
$a(n) = a(n-1) + \dfrac{\overline{\mu}_a}{p(n)} e(n)\,[y_{1,\mathrm{f}}(n) - y_{2,\mathrm{f}}(n)]\, sgm[a(n)]\,\{1 - sgm[a(n)]\}$
$\lambda(n) = \dfrac{\mathrm{sgm}[a(n)] - \mathrm{sgm}[-a^+]}{\mathrm{sgm}[a^+] - \mathrm{sgm}[-a^+]}$
$y(n) = \lambda(n)y_1(n) + [1 - \lambda(n)]y_2(n)$
$g_{i,l}(n-1) = (1-\kappa_i)\dfrac{1}{2L_w} + (1+\kappa_i)\dfrac{|w_{i,l}(n-1)|}{\varepsilon + 2\sum_{k=0}^{L_w-1}|w_{i,k}(n-1)|},$
    for $\quad l = 1, \ldots, L_w \quad$ and $i = 1, 2$
$\mu_{i,l}(n) = \dfrac{\mu_i g_{i,l}(n-1)}{\delta + \sum_{k=0}^{L_w-1} g_{i,k}(n-1) x_{\mathrm{f}}^2(n-k)}$, for $\quad l = 1, \ldots, L_w$
    and $i = 1, 2$
$\mathbf{G}_i(n) = \mathrm{diag}(\mu_{i,1}(n), \mu_{i,2}(n), ..., \mu_{i,L_w}(n)), \quad i = 1, 2$
$\mathbf{w}_i(n) = \mathbf{w}_i(n-1) + \mathbf{G}_i(n)\mathbf{x}_{\mathrm{f}}(n)e_i(n), \quad i = 1, 2$
```

3.3 Biased filtered-x scheme

A novel scheme that bias the weights of the adaptive filter was introduced in [97] for channel identification in order to improve the behavior of adaptive filters, mainly for low SNRs. This scheme uses an adaptive scaling factor $\alpha(n)$ to reduce the output of the filter signal when the energy of the noise is high with regard to the input signal. For equalization applications, the main difference is that the scaling factor also requires the use of a filtered-x structure.

In this section and motivated by the noisy scenarios that can be found in sound reproduction systems, we apply the idea of biasing the filter weights to the equalization context, where in case of low SNRs, the loudspeaker signal goes close to zero and can wait to better room conditions to update or render the input signal.

Fig. 3.3 shows the block diagram of a biased single channel AE system.

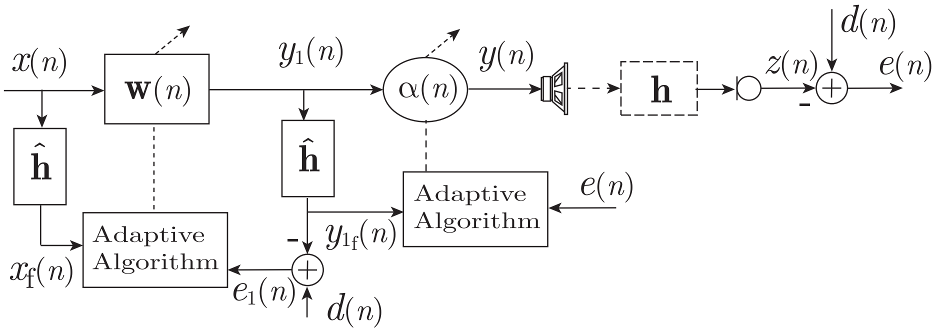


Figure 3.3. Block diagram of a biased single channel filtered-x equalization scheme.

For this scheme, we will adaptively bias the weights of the Fx-IPNLMS algorithm providing the biased Fx-IPNLMS (BFx-IPNLMS) algorithm. This strategy requires to rewrite (3.2) and (3.5) as:

$$z(n) = \mathbf{h} * \Big(\alpha(n) y_1(n)\Big). \tag{3.19}$$

$$w_l(n) = w_l(n-1) + \mu_l(n) e_1(n) x_f(n-l). \tag{3.20}$$

for $l = 0, \ldots, L_w - 1$. Where $e_1(n) = d(n) - y_{1\mathrm{f}}(n)$ is the error due to the output of the adaptive filter $y_1(n)$ filtered through $\hat{\mathbf{h}}$ (left adaptive block in Fig. 3.3). Moreover, the adaptive step size $\mu_l(n)$ is the same as in (3.16).

The biased scheme can be considered as a particular implementation of the previous combined structure, where all the filter coefficients of one of the adaptive filters are zero. Thus, only one branch of the block convex combination remains. The scaling factor of the algorithm ($\alpha(n)$) is defined by using a sigmoid activation function defined as in (3.9)-(3.13), but considering that in this case $y_2(n) = 0$.

$$\alpha(n) = \frac{sgm[a(n)] - sgm[-4]}{sgm[4] - sgm[-4]}. \tag{3.21}$$

$$\begin{aligned} a(n+1) &= a(n) - \frac{\overline{\mu}_a}{p(n)} \frac{\partial e^2(n)}{\partial a(n)} \\ &= a(n) + \frac{\overline{\mu}_a}{p(n)} e(n) y_{1\mathrm{f}}(n) sgm[a(n)] \Big\{ 1 - sgm[a(n)] \Big\}, \end{aligned} \tag{3.22}$$

being $\overline{\mu}_a$ the step size for the combination factor defined in (3.12) and $p(n)$ is estimated as

$$p(n) = \beta p(n-1) + (1-\beta) y_{1\mathrm{f}}^2(n). \tag{3.23}$$

The biased Fx-IPNLMS (BFx-IPNMLS) algorithm is described in **Algorithm 2**.

Algorithm 2 BFx-IPNLMS algorithm.

```
Input: Reference signal $x(n)$, desired signal $d(n)$ and microphone signal
    $z(n)$
Output: Output of the filter $y(n)$
Update the vectors $\mathbf{x}(n)$ and $\mathbf{x}_\mathrm{h}(n)$
$x_\mathrm{f}(n) = \mathbf{x}_\mathrm{h}^T(n)\widehat{\mathbf{h}}$
$y_1(n) = \mathbf{w}^T(n-1)\mathbf{x}(n)$
Update the vectors $\mathbf{x}_\mathrm{f}(n)$ and $\mathbf{y}_1(n)$
$y_{1\mathrm{f}}(n) = \mathbf{y}_1^T(n)\widehat{\mathbf{h}}$
$e(n) = d(n) - z(n)$
$e_1(n) = d(n) - y_{1\mathrm{f}}(n)$
$p(n) = \beta p(n-1) + (1-\beta)y_{1\mathrm{f}}^2(n)$
$\overline{\mu}_a = \frac{\mu_a}{sgm[a^+] - sgm[-a^+]}$
$a(n) = a(n-1) + \frac{\overline{\mu}_a}{p(n)} e(n) y_{1,\mathrm{f}}(n) sgm[a(n)]\left\{1 - sgm[a(n)]\right\}$
$\lambda(n) = \frac{\mathrm{sgm}[a(n)] - \mathrm{sgm}[-a^+]}{\mathrm{sgm}[a^+] - \mathrm{sgm}[-a^+]}$
$y(n) = \lambda(n) y_1(n)$
$g_l(n-1) = (1-\kappa_i)\frac{1}{2L_w} + (1+\kappa)\frac{|w_l(n-1)|}{\varepsilon + 2\sum_{k=0}^{L_w-1} |w_k(n-1)|}$,
    for $l = 1, \dots, L_w$
$\mu_l(n) = \frac{\mu g_l(n-1)}{\delta + \sum_{k=0}^{L_w-1} g_k(n-1) x_\mathrm{f}^2(n-k)}$, for $l = 1, \dots, L_w$
$\mathbf{G}(n) = \mathrm{diag}(\mu_1(n), \mu_2(n), ..., \mu_{L_w}(n))$
$\mathbf{w}(n) = \mathbf{w}(n-1) + \mathbf{G}(n)\mathbf{x}_\mathrm{f}(n) e_1(n)$
```

3.3.1 Steady-State analysis of the biased filtered-x scheme

The biased filtered-x structure is mainly proposed for low SNRs scenarios. At this point, it will be suitable to analyze the performance of this scheme to prove that it is at least as good as its unbiased version. For that purpose, we analyze the steady-state performance of the excess mean square error.

In real situations, the error measured at the microphone includes a noisy signal that depends on either the system performance or the ambient noise, $r(n)$. Moreover, the desired signal can be written as,

$$d(n) = \mathbf{w}_0^T \mathbf{x}_{\mathrm{f}}(n), \tag{3.24}$$

where $\mathbf{w}_0$ corresponds to the optimal filter solution.

For the development of this analysis, we take into account the following assumptions:

1. For low variations of the weight vector, $\mathbf{h}$ and $\mathbf{w}^T(n)$ are considered linear systems and can be interchanged.
2. $r(n)$ is an additive Gaussian noise with zero mean and variance σ_r^2 and uncorrelated with the input signal $\mathbf{x}(n)$.
3. After convergence, $\mathbf{x}_{\mathrm{f}}(n)$ and $\mathbf{w}^T(n)$ are uncorrelated.

Using assumption 1, the error signal of Fig. 3.3 can be written as,

$$\begin{aligned}
e(n) &= d(n) - z(n) + r(n) = d(n) - \alpha(n)\left[\mathbf{h}^T \mathbf{y}_1(n)\right] + r(n) \\
&= \mathbf{w}_0^T \mathbf{x}_{\mathrm{f}}(n) - \alpha(n)\left[\mathbf{w}^T(n-1)\mathbf{x}_{\mathrm{f}}(n)\right] + r(n) \\
&= \alpha(n)\left\{\left[\mathbf{w}_0^T - \mathbf{w}^T(n-1)\right]\mathbf{x}_{\mathrm{f}}(n)\right\} + [1-\alpha(n)]\left\{\mathbf{w}_0^T \mathbf{x}_{\mathrm{f}}(n)\right\} + r(n) \\
&= \alpha(n)\left\{\widetilde{\mathbf{w}}^T(n-1)\mathbf{x}_{\mathrm{f}}(n)\right\} + [1-\alpha(n)]\left\{\mathbf{w}_0^T \mathbf{x}_{\mathrm{f}}(n)\right\} + r(n) \\
&= \alpha(n)e_a(n) + [1-\alpha(n)]\left\{\mathbf{w}_0^T \mathbf{x}_{\mathrm{f}}(n)\right\} + r(n) \\
&= \left[\alpha(n)e_a(n) + [1-\alpha(n)]\,\mathbf{w}_0^T \mathbf{x}_{\mathrm{f}}(n)\right] + r(n) \\
&= e_{a_\alpha}(n) + r(n),
\end{aligned} \tag{3.25}$$

where $e_a(n)$ is the *a priori* error of the unbiased algorithm previously defined and $e_{a_\alpha}(n)$ is the *a priori* error of the biased algorithm. Moreover,

$\mathbf{y}_1(n) = [y_1(n), y_1(n-1), \ldots, y_1(n-\{L_h-1\})]^T$ includes the last L_h samples of the output of the adaptive filter $y_1(n)$.

For the sake of simplicity, we use the EMSE of the unbiased algorithm, $J_{ex}(n)$, to define the EMSE of the biased one as $J_{ex,\alpha}(n)$,

$$\begin{aligned} J_{ex,\alpha}(n) &= E\left\{|e_{a_\alpha}(n)|^2\right\} = E\left\{\left\|\alpha e_a(n) + [1-\alpha(n)]\,\mathbf{w}_0^T\mathbf{x}_\mathrm{f}(n)\right\|^2\right\} \\ &= E\left\{\alpha^2(n)\,|e_a(n)|^2\right\} + [1-\alpha(n)]^2\,E\left\{\left[\mathbf{w}_0^T\mathbf{x}_\mathrm{f}(n)\right]^2\right\} \\ &+ 2\alpha(n)\,[1-\alpha(n)]\,E\left\{e_a(n)\mathbf{w}_0^T\mathbf{x}_\mathrm{f}(n)\right\} \\ &= \alpha^2(n)J_{ex}(n) + [1-\alpha(n)]^2\,\mathbf{w}_0^T\mathbf{R}_{x_\mathrm{f}}\mathbf{w}_0 \\ &+ 2\alpha(n)\,[1-\alpha(n)]\,E\left\{\mathbf{w}_0^T\mathbf{x}_\mathrm{f}(n)\mathbf{x}_\mathrm{f}^T(n)\left[\mathbf{w}_0^T - \mathbf{w}(n-1)\right]\right\} \end{aligned} \tag{3.26}$$

where $\mathbf{R}_{x_\mathrm{f}}$ corresponds to $\mathbf{R}_{x_\mathrm{f}} = E\left\{\mathbf{x}_\mathrm{f}(n)\mathbf{x}_\mathrm{f}^T(n)\right\}$.

In steady state, the last term at the right-hand side of (3.26) can be removed using assumptions 3 and the fact that the estimation of the optimal filter tends to the optimal solution, $E\{\mathbf{w}(n)\} \rightarrow \mathbf{w}_0$ as $n \rightarrow \infty$..

Thus, the EMSE of the biased algorithm can be obtained from its unbiased version as,

$$J_{ex,\alpha}(\infty) = \alpha^2(\infty)J_{ex}(\infty) + [1-\alpha(\infty)]^2\,\mathbf{w}_0^T\mathbf{R}_{x_\mathrm{f}}\mathbf{w}_0 \tag{3.27}$$

Moreover, to obtain the optimal value in steady state of the scaling factor, the previous equation is derived with respect to α and set equal to zero,

$$\frac{\partial J_{ex,\alpha}(\infty)}{\partial\alpha(n)} = 2\alpha(\infty)J_{ex}(\infty) - 2\mathbf{w}_0^T\mathbf{R}_{x_\mathrm{f}}\mathbf{w}_0 + 2\alpha(\infty)\mathbf{w}_0^T\mathbf{R}_{x_\mathrm{f}}\mathbf{w}_0 = 0$$

$$\begin{aligned} \alpha(\infty)\left[J_{ex}(\infty) + \mathbf{w}_0^T\mathbf{R}_{x_\mathrm{f}}\mathbf{w}_0\right] &= \mathbf{w}_0^T\mathbf{R}_{x_\mathrm{f}}\mathbf{w}_0 \\ \alpha(\infty) &= \frac{\mathbf{w}_0^T\mathbf{R}_{x_\mathrm{f}}\mathbf{w}_0}{J_{ex}(\infty) + \mathbf{w}_0^T\mathbf{R}_{x_\mathrm{f}}\mathbf{w}_0} \\ \alpha(\infty) &= \frac{1}{1 + \frac{J_{ex}(\infty)}{\mathbf{w}_0^T\mathbf{R}_{x_\mathrm{f}}\mathbf{w}_0}} \end{aligned} \tag{3.28}$$

In this case, the existence of the estimated channel response $\hat{\mathbf{h}}$ in the term $\mathbf{w}_0^T \mathbf{R}_{x_\mathrm{f}} \mathbf{w}_0$ allows to suitably manipulate it as,

$$\begin{aligned}
\mathbf{w}_0^T \mathbf{R}_{x_\mathrm{f}} \mathbf{w}_0 &= \mathbf{w}_0^T E\left\{\hat{\mathbf{H}}^T \mathbf{x}_{L_{wh}}(n) \mathbf{x}_{L_{wh}}^T(n) \hat{\mathbf{H}}\right\} \mathbf{w}_0 \\
&= \left[\mathbf{w}_0^T \hat{\mathbf{H}}^T\right] E\left\{\mathbf{x}_{L_{wh}}(n) \mathbf{x}_{L_{wh}}^T(n)\right\} \left[\hat{\mathbf{H}} \mathbf{w}_0\right] \\
&= \left[\mathbf{w}_0^T \hat{\mathbf{H}}^T\right] \mathbf{R}_x \left[\hat{\mathbf{H}} \mathbf{w}_0\right],
\end{aligned} \tag{3.29}$$

where $\mathbf{x}_{L_{wh}}(n)$ is a vector with the last $L_w + L_h - 1$ samples of the input signal $x(n)$ and $\hat{\mathbf{H}}$ corresponds to the discrete linear convolution of the estimated channel response $\hat{\mathbf{h}}$, expressed in matrix form as a Toeplitz matrix of dimensions $(L_w + L_h - 1) \times L_w$,

$$\hat{\mathbf{H}} = \begin{bmatrix}
\hat{h}(0) & 0 & & 0 \\
\hat{h}(1) & \hat{h}(0) & & 0 \\
\vdots & h(1) & \ddots & 0 \\
\hat{h}(L_h - 1) & \vdots & & \hat{h}(0) \\
0 & \hat{h}(L_h - 1) & & \hat{h}(1) \\
0 & 0 & \ddots & \vdots \\
0 & 0 & 0 & \hat{h}(L_h - 1)
\end{bmatrix} \tag{3.30}$$

If the equalization works properly, the matrix vector product $\left[\hat{\mathbf{H}} \mathbf{w}_0\right]$ corresponds to a $(L_w + L_h - 1) \times 1$ column vector containing the delta function $\delta(n - \tau)$. Therefore, the term $\mathbf{w}_0^T \mathbf{R}_{x_\mathrm{f}} \mathbf{w}_0$ can be simplified as

$$\mathbf{w}_0^T \mathbf{R}_{x_\mathrm{f}} \mathbf{w}_0 = \sigma_x^2, \tag{3.31}$$

where σ_x^2 is the variance of the input signal.

Then, substituting (3.31) in both (3.27) and (3.28), we find that

$$\boxed{J_{ex,\alpha}(\infty) = \alpha^2(\infty) J_{ex}(\infty) + \left[1 - \alpha(\infty)\right]^2 \sigma_x^2,} \tag{3.32}$$

$$\alpha(\infty) = \frac{1}{1 + \frac{J_{ex}(\infty)}{\sigma_x^2}}. \tag{3.33}$$

As the scaling factor has a value between 0 and 1, the steady-state EMSE of the biased approach will always be equal to or lower than its unbiased version.

For the biased Fx-IPNLMS (BFx-IPNLMS) algorithm, the optimal value of the scaling factor in (3.28) is derived from (3.34), (3.35) and (3.36),

$$J_{ex}(\infty) = \mu\sigma_r^2(n) \sum_{l=1}^{L_w} \frac{g_l(\infty)}{2 - \mu g_l(\infty)} \tag{3.34}$$

$$g_l(\infty) = (1-\kappa)\frac{1}{2L_w} + (1+\kappa)\frac{|w_{0,l}(n)|}{\varepsilon + 2\displaystyle\sum_{k=0}^{L_w-1} |w_{0,_k}(n)|}, \tag{3.35}$$

$$\alpha(\infty) = \frac{1}{1 + \frac{J_{ex}(\infty)}{\sigma_x^2}} \tag{3.36}$$

Also it is important to notice that for this kind of application, the adaptive filter has at least a hundred of coefficients and as $\displaystyle\sum_{l=1}^{L_w} g_l(n) = 1$, the following expression can be approximated $\displaystyle\sum_{l=1}^{L_w} \frac{g_l(\infty)}{2 - \mu g_l(\infty)} = \sum_{l=1}^{L_w} \frac{1}{[2/g_l(\infty)] - \mu}$ $\cong \displaystyle\sum_{l=1}^{L_w} g_l(\infty)/2 = 1/2$, as $2/g_l(\infty) \gg \mu$. This results in a general EMSE expression independent of the κ-value

$$J_{ex}(\infty) = \frac{\mu\sigma_r^2}{2}. \tag{3.37}$$

The EMSE development of the biased scheme can also to be applied to the Fx-NLMS algorithm. Thus, (3.32) can be also used, where the EMSE value of the unbiased Fx-NLMS algorithm at steady state $J_{ex}(\infty)$ and the optimal value scaling factor $\alpha(\infty)$ are defined respectively as,

$$J_{ex}(\infty) = \frac{\mu\sigma_r^2}{2 - \mu} \tag{3.38}$$

$$\alpha(\infty) = \frac{1}{1 + \frac{\mu\sigma_r^2}{[2-\mu]\sigma_x^2}}. \tag{3.39}$$

(3.34), (3.36) and (3.39) show that when the SNR is reduced, the scaling factor decreases its value and thus it biases the final EMSE of the biased scheme.

3.4 Multichannel room equalization

A generic multichannel adaptive equalization system, with L loudspeakers and M microphones, is considered and illustrated in Fig. 3.4 to extend the single channel scheme to a MIMO system. This MIMO system presents $J \times M$ room responses, multiple error signals and multiple adaptive filters to be updated simultaneously. The management of those signals is not straightforward and becomes the main difficulty in extending the single channel adaptive equalization problem to the multichannel case.

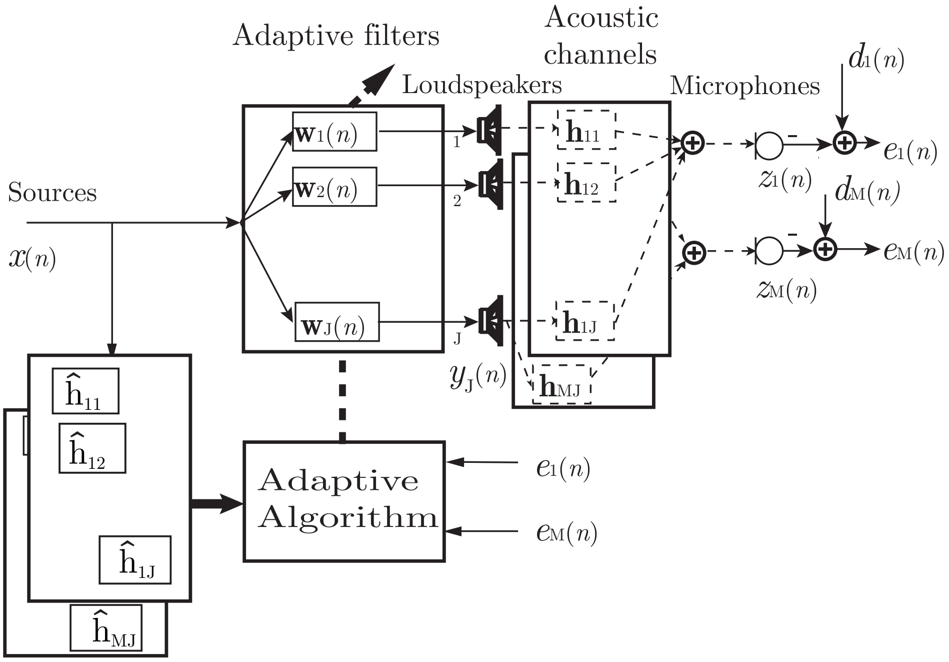


Figure 3.4. Block diagram of a multichannel AE system, with 1 primary source, L loudspeakers and M microphones.

For the specific case of the Fx-IPNLMS algorithm, the previous equations defined in Section 3.1 can be rewritten as follows. For simplicity, we are considering only a single primary input signal $x(n)$, but it can be straightforward extended to any number of input signals.

The rendering signal of each loudspeaker $y_j(n)$, for $j = 1, \ldots, J$, can be expressed as,

$$y_j(n) = \mathbf{w}_j^T(n-1)\mathbf{x}(n), \tag{3.40}$$

where $\mathbf{w}_j(n)$ denotes the coefficient vector of the j-th adaptive filter.

The signal measured at each microphone and its corresponding error are given, respectively, by (3.41) and (3.43),

$$z_m(n) = \sum_{j=1}^{J} \mathbf{h}_{m,j} * y_j(n), \quad m = 1, \ldots, M. \tag{3.41}$$

$$e_m(n) = d_m(n) - z_m(n), \tag{3.42}$$

where $\mathbf{h}_{m,l}$ is the room channel response between loudspeaker l and microphone m, and the desired signal $d_m(n)$ corresponds to the input signal with its corresponding source-microphone delay

$$d_m(n) = x(n - \tau_m). \tag{3.43}$$

Each component adaptive filter follows its own update equation, thus, among the different possibilities, we use the Fx-IPNLMS algorithm to minimize the sum of the mean square errors. The L_w weights of each adaptive filter are updated according to

$$\mathbf{w}_j(n) = \mathbf{w}_j(n-1) + \mathbf{G}_j(n) \sum_{m=1}^{M} e_m(n) \mathbf{x}_{\mathrm{f}_{(m,j)}}(n), \tag{3.44}$$

being $\mathbf{w}_j(n)$ a column vector, $\mathbf{w}_j^T(n) = [w_{j,0}(n), w_{j,1}(n), \ldots, w_{j,L_w-1}(n)]$, and $\mathbf{x}_{\mathrm{f}_{(m,j)}}(n)$ corresponds to the last L_w samples of the source signal filtered through the estimated room channel from loudspeaker j to microphone m, $\widehat{\mathbf{h}}_{(m,j)}$. Moreover, $\mathbf{G}_j(n)$ is a diagonal matrix that contains the L_w step-size parameters of every coefficient of the filter $\mathbf{w}_j(n)$.

For the Fx-IPNLMS algorithm and similarly to (3.6) and (3.7), the elements of $G_l(n)$ are obtained for $j = 1, \ldots, J$ as,

$$\mathbf{G}_j(n) = \frac{\mu \mathbf{g}_j(n-1)}{\delta + \sum_{k=0}^{L_w-1} g_{j,k}(n-1) x^2_{\mathrm{f}_{(m,j)}}(n-k)}, \tag{3.45}$$

$$\text{where} \quad \mathbf{g}_j(n) = (1-\kappa)\frac{1}{2L_w} + (1+\kappa)\frac{|\mathbf{w}_j(n)|}{\varepsilon + 2\sum_{k=0}^{L_w-1} |w_{j,k}(n)|}. \tag{3.46}$$

Algorithm 3 MFx-IPNLMS.

```
Input: Reference signal $x(n)$, desired signals $d_m(n)$ and microphone signals $z_m(n)$ for $m = 1, \ldots, M$.
Output: Output of the filter $y_j(n)$ for $j = 1, \ldots, J$.
Update the vectors $\mathbf{x}(n)$ and $\mathbf{x}_{\mathrm{h}}(n)$
$x_{\mathrm{f}_{(m,j)}}(n) = \mathbf{x}_{\mathrm{h}}^T(n)\widehat{\mathbf{h}}_{(m,j)}$ for $j = 1, \ldots, J$ and $m = 1, \ldots, M$.
$y_j(n) = \mathbf{w}_j^T(n-1)\mathbf{x}(n), \quad$ for $j = 1, \ldots, J$.
Update the vector $\mathbf{x}_{\mathrm{f}_{(m,j)}}(n)$ for $j = 1, \ldots, J$ and $m = 1, \ldots, M$.
$e_m(n) = d_m(n) - z_m(n), \quad$ for $m = 1, \ldots, M$.
$g_{j,l}(n-1) = (1-\kappa)\dfrac{1}{2L_w} + (1+\kappa)\dfrac{|w_{j,l}(n-1)|}{\varepsilon + 2\sum_{k=0}^{L_w-1} |w_{j,k}(n-1)|}$,
    for $\; j = 1, \ldots, J$ and $l = 1, \ldots, L_w$.
$\mu_{j,l}(n) = \dfrac{\mu g_{j,l}(n-1)}{\delta + \sum_{k=0}^{L_w-1} g_{j,k}(n-1)x^2_{\mathrm{f}_{(m,j)}}(n-k)}$, for $\; j = 1, \ldots, J$
    and $l = 1, \ldots, L_w$.
$\mathbf{G}_j(n) = \mathrm{diag}\Big(\mu_{j,1}(n), \mu_{j,2}(n), ..., \mu_{j,L_w}(n)\Big), \quad$ for $j = 1, \ldots, J$.
$\mathbf{w}_j(n) = \mathbf{w}_j(n-1) + \mathbf{G}_j(n)\mathbf{x}_{\mathrm{f}}(n)e_j(n), \quad$ for $j = 1, \ldots, J$
```

The multichannel Fx-IPNLMS (MFx-IPNLMS) algorithm is summarized in **Algorithm 3**.

With regard to the combined and biased schemes proposed in Section 3.2 and 3.3, they can be also extended to the multichannel case, combining or biasing each adaptive filter $\mathbf{w}_j(n)$ for $j = 1, \ldots, J$. Due to the high number of filtered-x operations that are required for the combined and biased multichannel structure and for the sake of clarity, these multichannel algorithms, the combined MFx-IPNLMS (CMFx-IPNLMS) and the biased MFx-IPNLMS (BMFx-IPNLMS) algorithm, are summarized and described in **Algorithm 4** and **Algorithm 5**, respectively.

Algorithm 4 Combined MFx-IPNLMS.

```
Input: Source signal $x(n)$, desired signals $d_m(n)$ and microphone signals
    $z_m(n)$,  for $m = 1, \dots, M$
Output: Output of the parallel filters $y_j^{'}(n)$,  for $j = 1, \dots, J$.
Update the vector $\mathbf{x}(n)$ and $\mathbf{x}_{\mathrm{h}}(n)$.
$x_{\mathrm{f}_{(m,j)}}(n) = \mathbf{x}_{\mathrm{h}}^T(n)\widehat{\mathbf{h}}_{(m,j)}$ for $j = 1, \dots, J$ and $m = 1, \dots, M$.
$y_{i,j}(n) = \mathbf{w}_{i,j}^T(n-1)\mathbf{x}(n)$,  for $i = 1, 2$ and $j = 1, \dots, J$.
Update the vector $\mathbf{x}_{\mathrm{f}_{(m,j)}}(n)$, $\mathbf{y}_{1j}$ and $\mathbf{y}_{2j}$ for $j = 1, \dots, J$
    and $m = 1, \dots, M$.
$e_m(n) = d_m(n) - z_m(n)$,  for $m = 1, \dots, M$.
$y_{i\mathrm{f}_{(m,j)}}(n) = \mathbf{y}_{ij}^T(n)\widehat{\mathbf{h}}_{(m,j)}$,  for $i = 1, 2$, $m = 1, \dots, M$ and $j = 1, \dots, J$.
$\overline{\mu}_{a_j} = \dfrac{\mu_{a_j}}{sgm[4] - sgm[-4]}$,  for $j = 1, \dots, J$
$p_j(n) = \beta p_j(n-1) + (1-\beta)\Big[y_{1\mathrm{f}_{(m,j)}}(n) - y_{2\mathrm{f}_{(m,j)}}(n)\Big]^2$,  for $j = 1, \dots, J$
    and $m = 1, \dots, M$.
$a_j(n) = a_j(n-1) + \dfrac{\overline{\mu}_{a_j}}{p_j(n)} \sum_{m=1}^{M} \Big\{ e_m(n) \Big[y_{1\mathrm{f}_{(m,j)}}(n) - y_{2\mathrm{f}_{(m,j)}}(n)\Big] \Big\}$
    $sgm[a_j(n-1)]\{1 - sgm[a_j(n-1)]\}$,  for $j = 1, \dots, J$ and $m = 1, \dots, M$.
$\lambda_j(n) = \dfrac{sgm[a_j(n)] - sgm[-4]}{sgm[4] - sgm[-4]}$,  for $j = 1, \dots, J$ and $m = 1, \dots, M$.
$y_j^{'}(n) = \lambda_j(n) y_{1,j}(n) + \Big(1 - \lambda_j(n)\Big) y_{2,j}(n)$,  for $j = 1, \dots, J$.
$e_{m,j}^{'}(n) = d_m(n) - y_{if_{(m,j)}}(n)$,  for $i = 1, 2$, $j = 1, \dots, J$
    and $m = 1, \dots, M$.
$g_{i,j,l}(n-1) = (1 - \kappa_i)\dfrac{1}{2L_w} + (1 + \kappa_i)\dfrac{|w_{i,j,l}(n-1)|}{\varepsilon + 2\sum_{k=0}^{L_w - 1} |w_{i,j,k}(n-1)|}$,
    for  $i = 1, 2$, $j = 1, \dots, J$ and $l = 1, \dots, L_w$.
$\mu_{i,j,l}(n) = \dfrac{\mu g_{i,j,l}(n-1)}{\delta + \sum_{k=0}^{L_w - 1} g_{i,j,k}(n-1) x_{\mathrm{f}_{(m,j)}}^2(n-k)}$, for  $i = 1, 2$,
    $j = 1, \dots, J$ and $l = 1, \dots, L_w$.
$\mathbf{G}_{i,j}(n) = \mathrm{diag}\Big(\mu_{i,j,1}(n), \mu_{i,j,2}(n), ..., \mu_{i,j,L_w}(n)\Big)$,  for $i = 1, 2$ and
    $j = 1, \dots, J$.
$\mathbf{w}_{i,j}(n) = \mathbf{w}_{i,j}(n-1) + \mathbf{G}i, j(n) \sum_{m=1}^{M} e_{m,j}^{'}(n) \mathbf{x}_{\mathrm{f}_{(m,j)}}(n)$,  for $i = 1, 2$ and
    $j = 1, \dots, J$.
```

Algorithm 5 Biased MFx-IPNLMS.

```
Input: Source signal $x(n)$, desired signals $d_m(n)$ and microphone signals
    $z_m(n)$, for $m = 1, \ldots, M$
Output: Output of the filters $y_j^{'}(n)$, for $j = 1, \ldots, J$.
Update the vector $\mathbf{x}(n)$ and $\mathbf{x}_{\mathrm{h}}(n)$.
$x_{\mathrm{f}_{(m,j)}}(n) = \mathbf{x}_{\mathrm{h}}^T(n)\widehat{\mathbf{h}}_{(m,j)}$ for $j = 1, \ldots, J$ and $m = 1, \ldots, M$.
$y_j(n) = \mathbf{w}_j^T(n-1)\mathbf{x}(n), \quad j = 1, \ldots, J$.
Update the vector $\mathbf{x}_{\mathrm{f}_{(m,j)}}(n)$ and $\mathbf{y}_j$ for $j = 1, \ldots, J$ and $m = 1, \ldots, M$.
$e_m(n) = d_m(n) - z_m(n), \quad$ for $m = 1, \ldots, M$.
$y_{\mathrm{f}_{m,j}}(n) = \mathbf{y}_j^T(n)\widehat{\mathbf{h}}_{(m,j)}, \quad$ for $m = 1, \ldots, M$ and $j = 1, \ldots, J$.
$\overline{\mu}_{a_j} = \dfrac{\mu_{a_j}}{sgm[4] - sgm[-4]}, \quad$ for $j = 1, \ldots, J$
$p_j(n) = \beta p_j(n-1) + (1-\beta)y^2_{\mathrm{f}_{(m,j)}}(n), \quad$ for $j = 1, \ldots, J$ and
    $m = 1, \ldots, M$.
$a_j(n) = a_j(n-1) + \dfrac{\overline{\mu}_{a_j}}{p_j(n)} \sum_{m=1}^{M} [e_m(n) y_{\mathrm{f}_{(m,j)}}(n)]$
    $sgm[a_j(n-1)]\{1 - sgm[a_j(n-1)]\}$, for $j = 1, \ldots, J$ and $m = 1, \ldots, M$.
$\alpha_j(n) = \dfrac{sgm[a_j(n)] - sgm[-4]}{sgm[4] - sgm[-4]}, \quad$ for $j = 1, \ldots, J$ and $m = 1, \ldots, M$.
$y_j^{'}(n) = \alpha_j(n) y_j(n), \quad$ for $j = 1, \ldots, J$.
$e_{m,j}(n) = d_m(n) - y_{\mathrm{f}_{(m,j)}}(n), \quad$ for $j = 1, \ldots, J$ and $m = 1, \ldots, M$.
$g_{j,l}(n-1) = (1-\kappa)\dfrac{1}{2L_w} + (1+\kappa)\dfrac{|w_{j,l}(n-1)|}{\varepsilon + 2\sum_{k=0}^{L_w-1} |w_{j,k}(n-1)|},$
    for $j = 1, \ldots, J$ and $l = 1, \ldots, L_w$.
$\mu_{j,l}(n) = \dfrac{\mu g_{j,l}(n-1)}{\delta + \sum_{k=0}^{L_w-1} g_{j,k}(n-1) x^2_{\mathrm{f}_{(m,j)}}(n-k)}$, for $j = 1, \ldots, J$
    and $l = 1, \ldots, L_w$.
$\mathbf{G}_j(n) = \mathrm{diag}\Big(\mu_{j,1}(n), \mu_{j,2}(n), ..., \mu_{j,L_w}(n)\Big), \quad$ for $j = 1, \ldots, J$.
$\mathbf{w}_j(n) = \mathbf{w}_j(n-1) + \mathbf{G}_j(n) \sum_{m=1}^{M} e_{m,j}^{'}(n) \mathbf{x}_{\mathrm{f}_{(m,j)}}(n), \quad$ for $j = 1, \ldots, J$.
```

3.5 Results

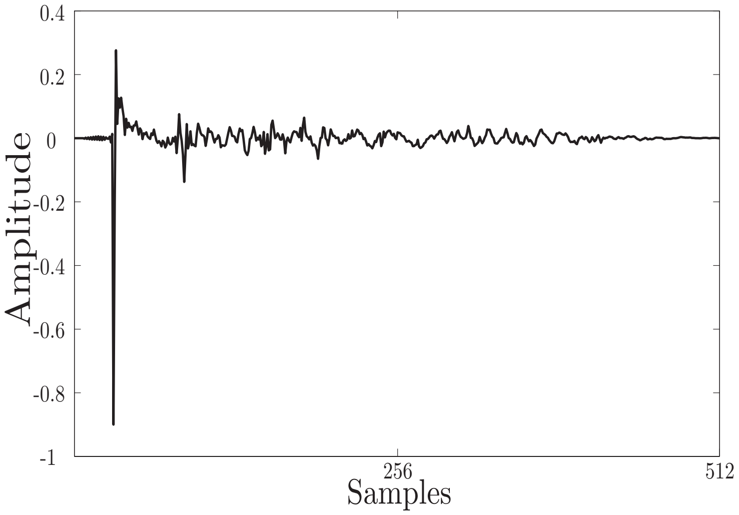
In this section we carry out three different sets of experiments to study the behavior of the algorithms proposed in this chapter. The first experiment aims at assessing the effectiveness of the combined filtered-x scheme. The second set is performed to evaluate the biased filtered-x structure in low SNRs scenarios. The last experiment includes a MIMO equalization algorithm, with 2 loudspeakers and 2 microphones, to validate the previous schemes for the multichannel case.

3.5.1 Combination scheme

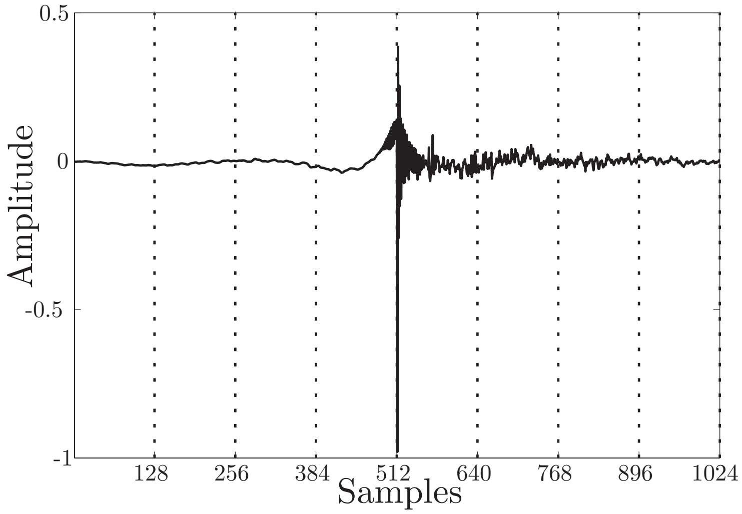
This experiment shows the performance of the convex combination of two adaptive filters of Fig. 3.2, with regard to using only one adaptive filter as in Fig. 3.1. If the component adaptive filters are set with complementary capabilities, for instance, with different step sizes, the combination scheme allows to achieve both a good convergence speed and a low steady-state EMSE. For this purpose, the combination of two filters with different step size parameters has been considered in this experiment.

The acoustic channel of Fig. 4.5 has been used. It has been measured in a room with $T_{60} = 180$ ms and a sampling frequency of 8 kHz. The input signal $x(n)$ is a white Gaussian noise with zero mean and unit variance. Moreover, an additive noise $r(n)$ has been added to the microphone signal, which is also a Gaussian noise uncorrelated with $x(n)$ and its variance is adjusted to provide an SNR of 30 dB.

Fig. 3.6 (a) shows the EMSE performance of the CFx-IPNLMS algorithm (round marker line) with $\kappa_1 = \kappa_2 = -0.5$ and different step sizes of $\mu_1 = 0.01$ and $\mu_2 = 0.1$. For the Fx-IPNLMS algorithm of the combined scheme a value of $\kappa = -0.5$ has been chosen, as recommended in [10] to achieve a good behavior. On the other hand, $\mu = 0.1$ allows a fast convergence without getting unstable, meanwhile a lower μ value of 0.01 exhibits a lower steady state. The step size of the combination factor μ_a has been set to 0.1 to allow a fast adaptation from one algorithm to the other. Also, the individual Fx-IPNLMS algorithms, with $\kappa = -0.5$ and $\mu = 0.01$ (diamond marker line) and with $\kappa = -0.5$ and $\mu = 0.1$ (square marker line), are plotted.

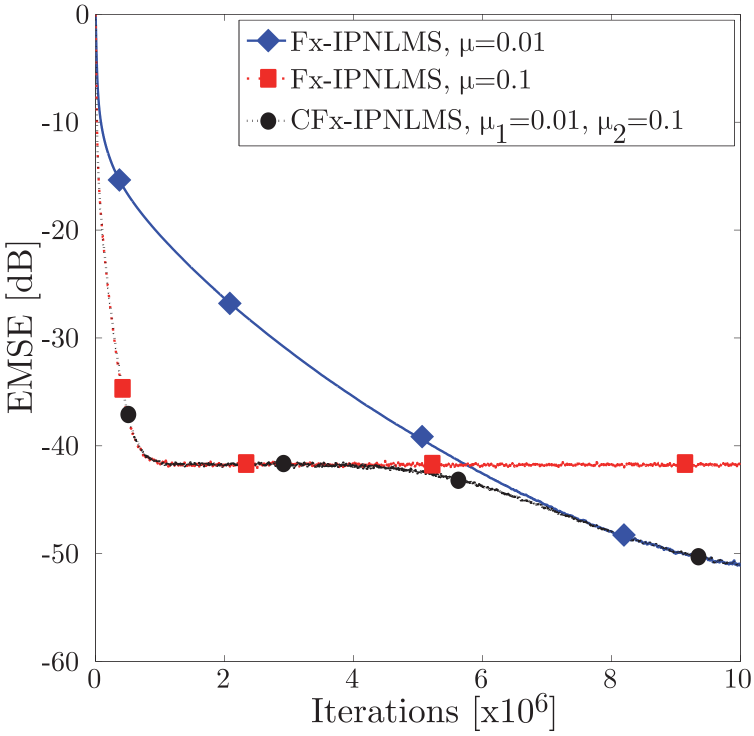


(a)

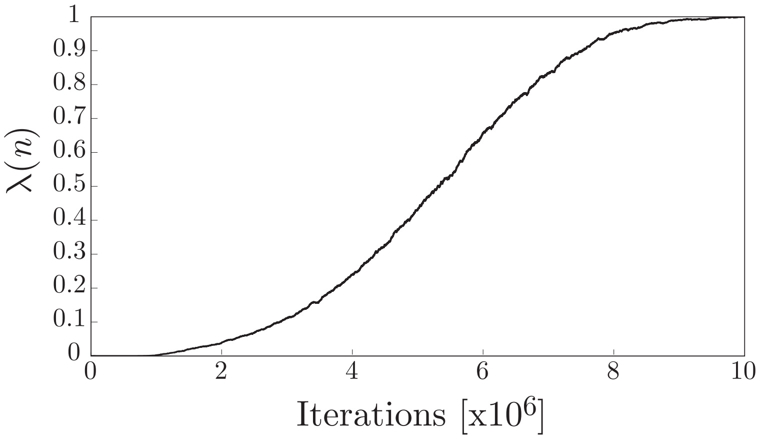


(b)

Figure 3.5. (a): Acoustic channel response and (b) its inverse filter.



(a)



(b)

Figure 3.6. (a): EMSE(n) for CFx-IPNLMS and their individual algorithms. (b): $\lambda(n)$ of the CFx-IPNLMS approach.

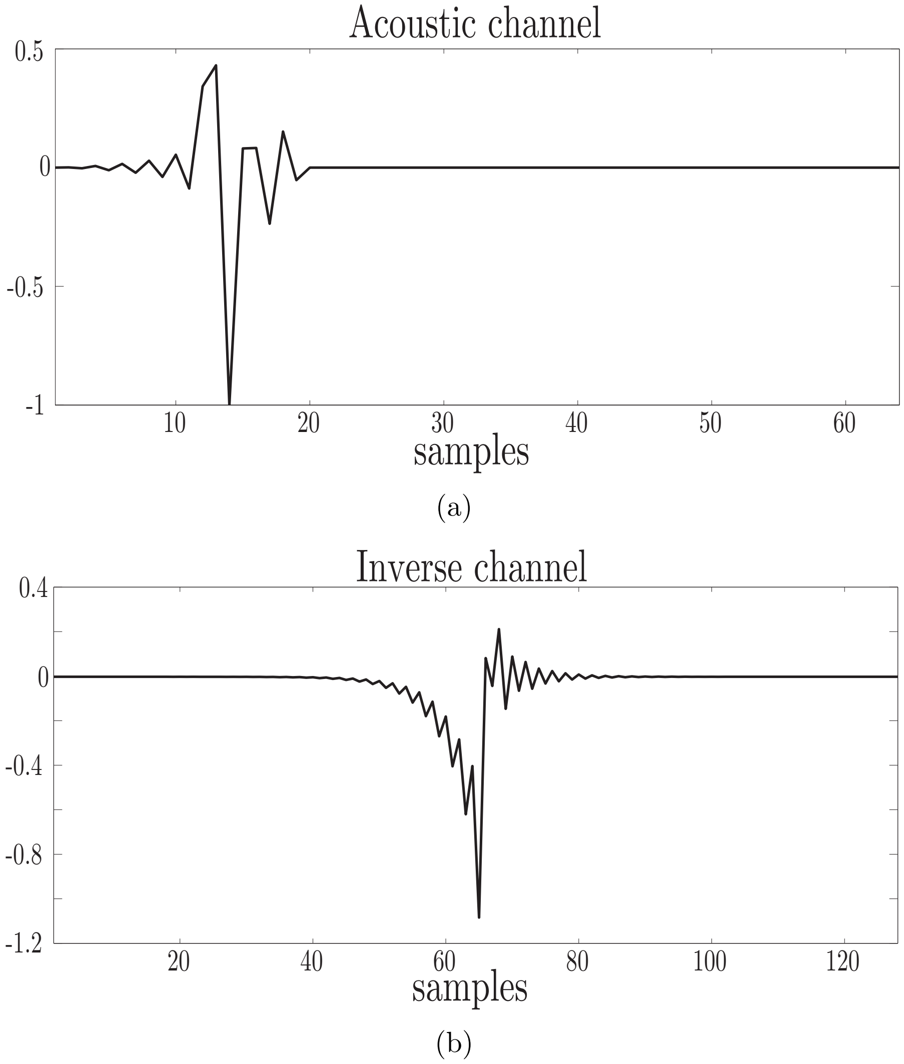
3.5.2 Biased scheme

In this section and using the analysis developed in Section 3.3.1, theoretical predicted values are compared to the averaged estimated ones for the EMSE expressions and the scaling factor $\alpha(n)$ in steady state, for both Fx-NLMS and the Fx-IPNLMS schemes and their biased versions, BFx-NLMS and BFx-IPNLMS, respectively.

Considering that the performance of the models is not dependent on the acoustic channel length considered as emerges from (3.32), a 64-samples channel is used for simplicity. This channel has been measured in a real listening room [17], but artificially modified to get a quasi-sparse inverse impulse response, see Fig. 3.7. The optimal filter $\mathbf{w}_0$ in Fig. 3.7 (b) has been computed as the inverse filter of the acoustic channel showed in Fig. 3.7 (a) by using the least squares error method (LSE) [3]. A length twice the length of the acoustic channel and a delay $\tau = 78$ samples have been considered.

Fig. 3.8 compares the theoretical results (in dashed line) for the scaling factor at steady state, $\alpha(\infty)$ in (3.33), with the averaged estimated ones (in solid line for the BFx-IPNMLS schemes and in dotted line for the BFx-NLMS ones) for different values of both SNR and μ. For the improved proportionated algorithms, $\kappa = -0.5$ has been chosen. Each curve represents a different μ value and for various SNR values along x-axis, keeping constant $\sigma_x^2 = 1$ and varying σ_r^2. Fig. 3.8 also shows that for high values of SNR the scaling factor is close to 1, and from (3.32) $J_{ex,\alpha}(\infty) = J_{ex}(\infty)$. For low SNR and high μ values $\alpha(\infty)$ tends to 0. It can be observed that the estimated results for the BFx-IPNLMS agree with the theoretical ones especially for low μ values.

As was developed in the theoretical analysis, the unbiased EMSE expression (3.34) can be approximated by (3.37) providing the same theoretical results of $J_{ex}(\infty)$ for both the Fx-IPNLMS and the Fx-NLMS algorithms. Fig. 3.9 shows the EMSE value in dB for the unbiased algorithms, $J_{ex}(\infty)$ (3.34) and (3.38). Estimated results (solid line for the Fx-IPNLMS and dotted line for the Fx-NLMS algorithms) fall close to the theoretical ones (dashed line), their values increase with σ_r^2 and μ, and are almost similar to the theoretical ones. Fig. 3.10 shows the $J_{ex,\alpha}(\infty)$ in (3.32) for the BFx-IPNLMS and the BFx-NLMS algorithms. Also the simulated results agree well with the theoretical ones, but are upper limited by 0dB as the SNR decreases. As in Fig. 3.8, a worst performance is obtained for high



(a)

(b)

Figure 3.7. (a) Acoustic channel; (b) Inverse of the channel.

μ values. Although perfect secondary path estimates have been considered in the present simulations, it has been also found that the derived models predict quite accurately the simulated results with not very high modelling errors.

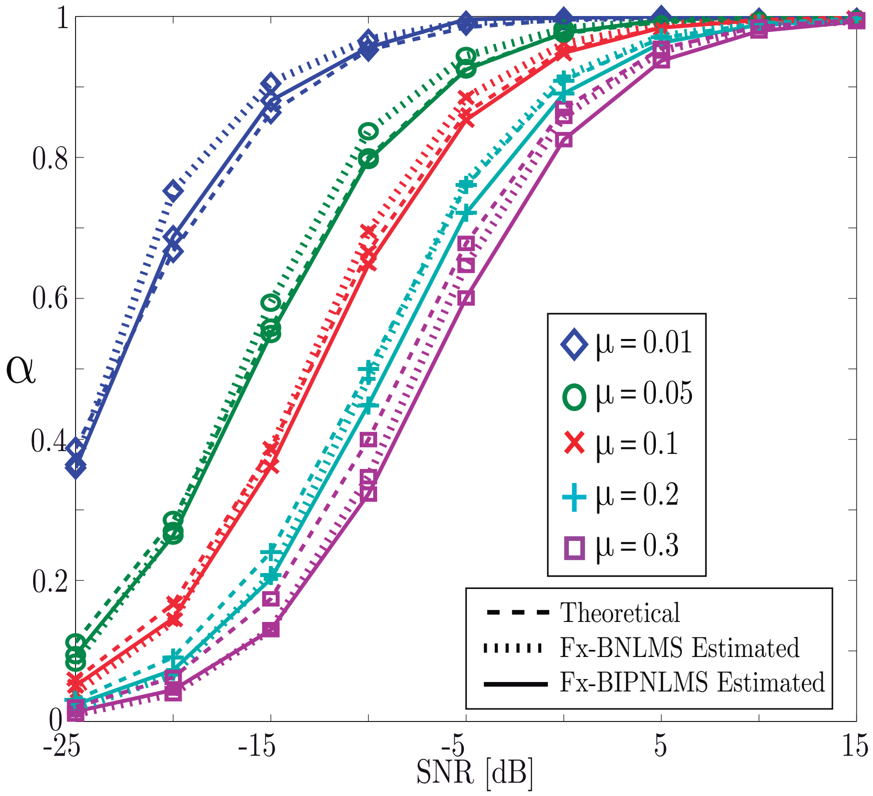


Figure 3.8. Theoretical (dashed line) and estimated (solid line for the Fx-BIPNLMS and dotted line for the Fx-BNLMS algorithms) $\alpha(\infty)$ for different μ and SNR values.

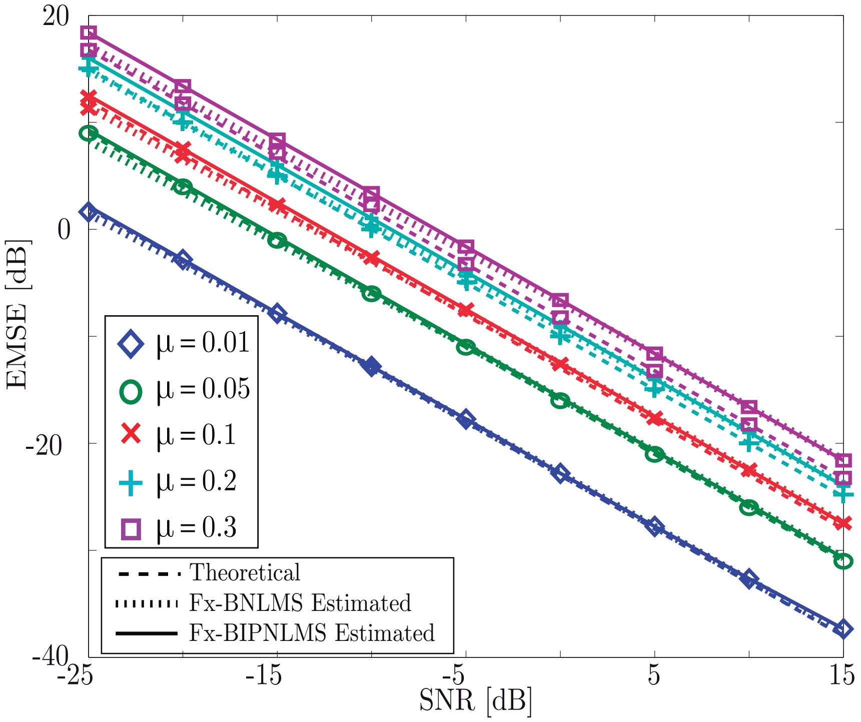


Figure 3.9. Estimated (solid line for the IP type algorithms and dotted line for the NLMS ones) and theoretical (dashed line) EMSE curves in steady state for their unbiased versions $J_{ex}(\infty)$ defined in (3.34) and (3.38).

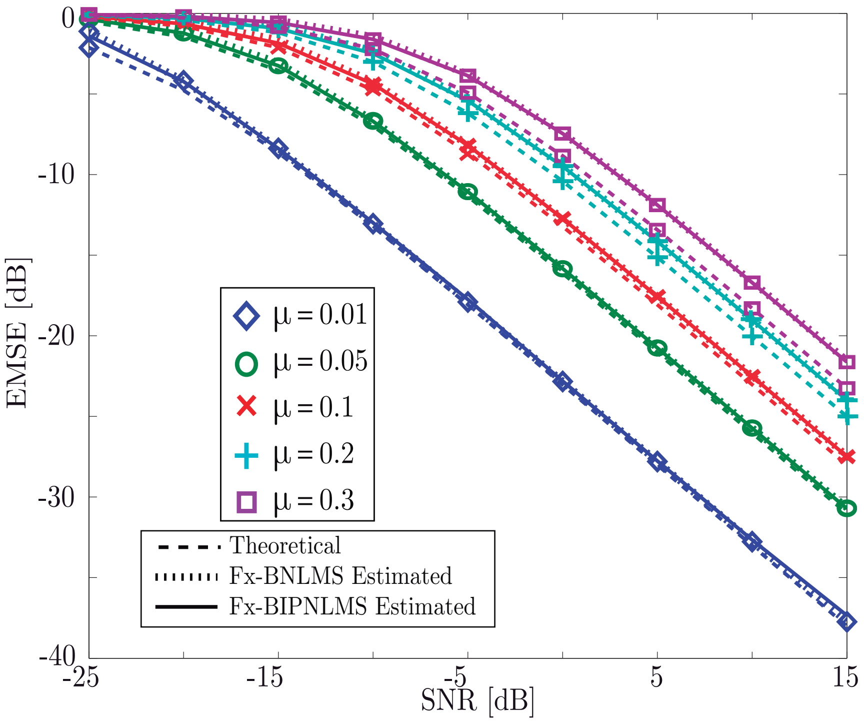


Figure 3.10. Estimated (solid line for the IP type algorithms and dotted line for the NLMS ones) and theoretical (dashed line) EMSE curves in steady state for the biased ones $J_{ex,\lambda}(\infty)$ defined in (3.32).

The second experiment shows the ability of the biased IPNLMS algorithms to improve the convergence speed performance of the normalized versions. Fig. 3.11 shows the EMSE evolution for the different algorithms with $\mu = 0.1$ and a low value of SNR$= -5$dB, thus the biased versions achieve lower EMSE values in steady state. Furthermore, after $75,000$ samples the channel $\mathbf{h}$ slightly changes and thus its optimal inverse filter $\mathbf{w}_0$, showing the ability of the adaptive algorithms to follow system variations.

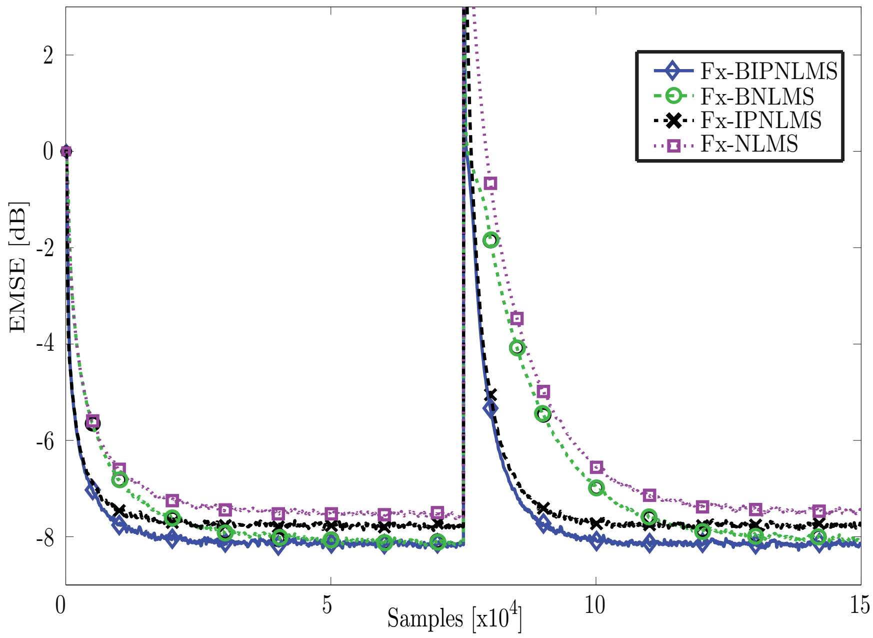


Figure 3.11. EMSE evolution with time-varying channel for the Fx-IPNLMS and Fx-NLMS algorithms and their biased versions.

3.5.3 Multichannel 1:2:2 system

In the third experiment we consider a 1:2:2 MIMO system (2 loudspeakers and 2 microphones) which involves 4 room impulse responses. A performance comparison of the MFx-IPNLMS and the BMFx-IPNLMS algorithms has been carried out. The evaluation of the algorithms behavior is based on the NPM index defined in Section 2.6.2, where lower values mean a better performance.

The robustness of the algorithms has been analyzed with different channel responses of different degrees of sparsity (see Section 2.1.2) and with time-varying SNRs. To this end, the simulations start with non-so-sparse impulse responses. These room channels have been measured in a real audio room, with a reverberation time, τ_{60}, of approximately 250ms. Fig. 3.12 shows these impulse responses (with 256 taps) of the 1:2:2 system. In the second part of the simulations, the room channels commute to sparser ones. The new impulse responses have been artificially obtained from the previous paths by taking the first 50 samples and zero-padding to length 256 (see Fig. 3.13). An uncorrelated noise signal has been added to the microphones to simulate a real scenario. The power of the additive noise has been set to get two different SNRs (5 and -5dB).

The adaptive filters have a length of $L_w = 256$ taps, the same length of the room responses. A step size $\mu = 0.2$ and an asymmetry factor $\kappa = -0.5$ have been fixed for the IPNLMS type algorithms. Other parameters for the biased method have been set to $\mu_a = 0.1$ and $\beta = 0.9$.

As expected, a behavior similar to the single-channel case is obtained. Fig. 3.14 shows the NPM evolution at the two microphones. The convergence performance of both algorithms is very similar when SNR = 5dB, independently of the degree of sparsity of the impulse responses. However, when SNR = -5dB the biased approach clearly outperforms the MFx-IPNLMS algorithm (between $n = 100,000$ and $n = 300,000$ iterations). The evolution of $\lambda(n)$ in Fig. 3.15 evidences the inclusion of a bias different to zero when the SNR is low. Moreover, the evolution of the two bias parameters is slightly different between $n = 200,000$ and $n = 300,000$, due to the different NPM performance obtained at both microphones.

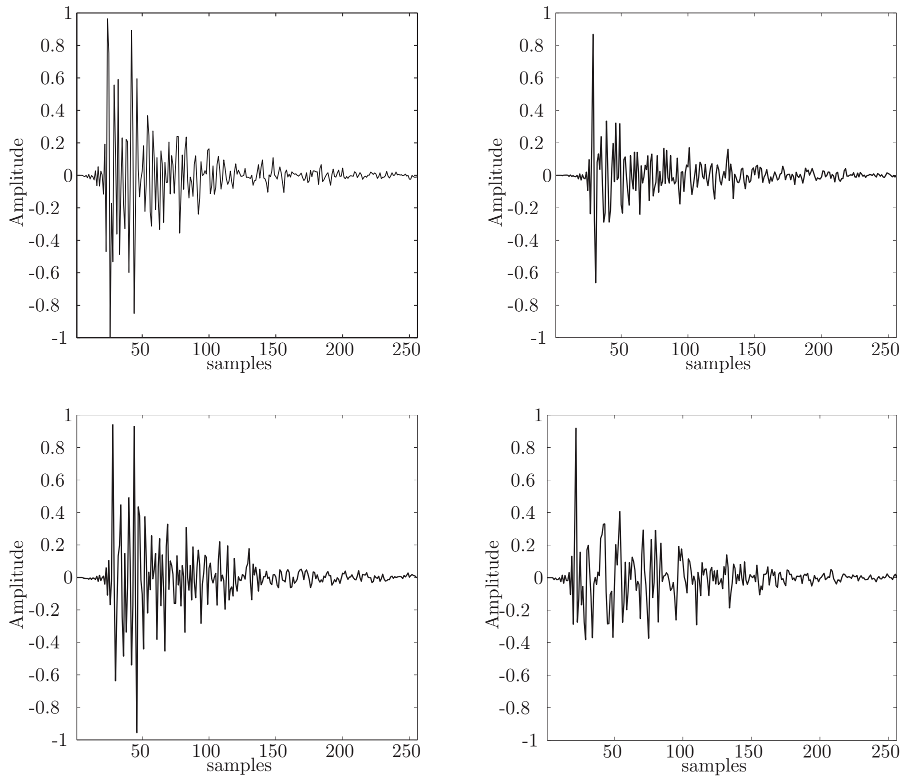


Figure 3.12. No-so-sparse room impulse responses.

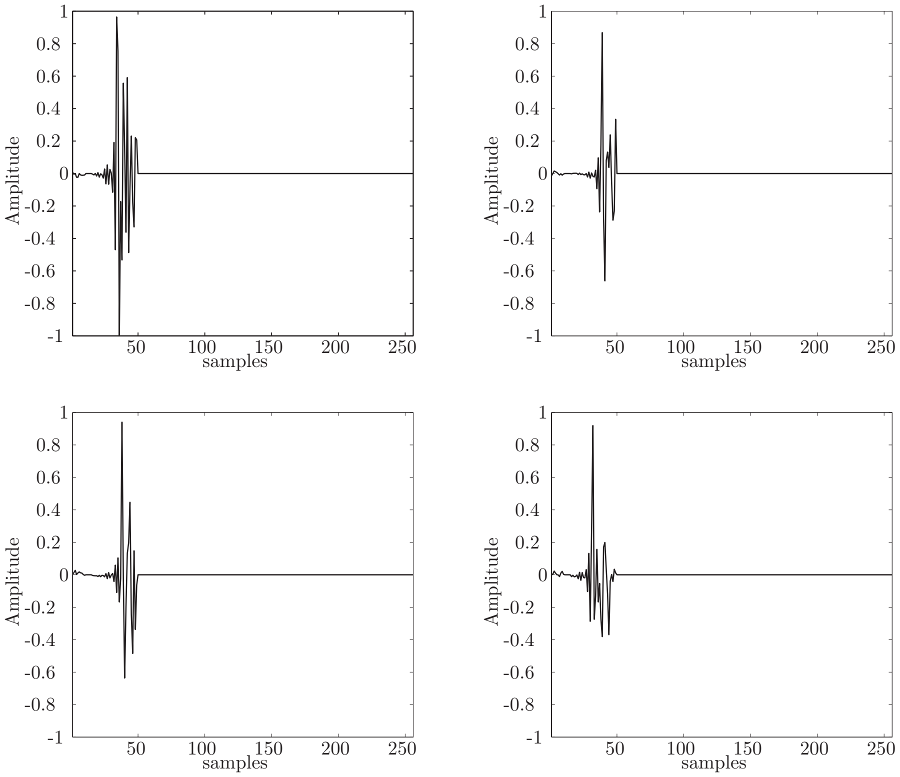
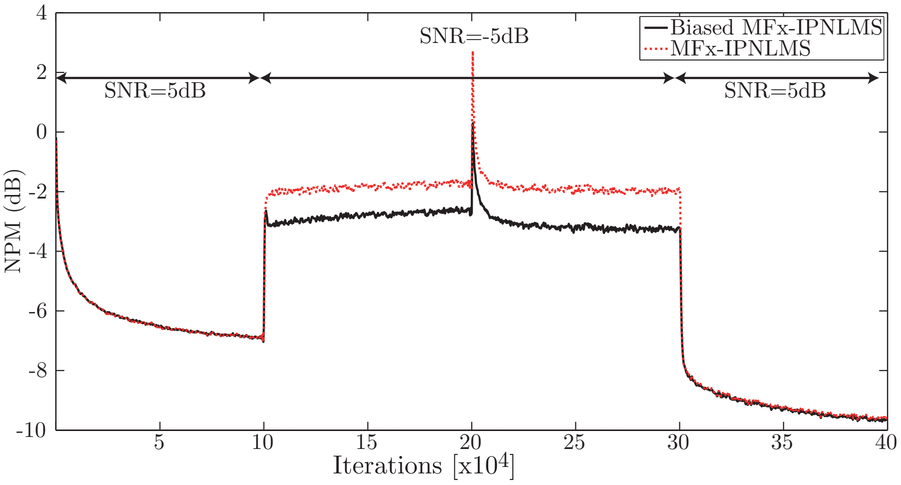
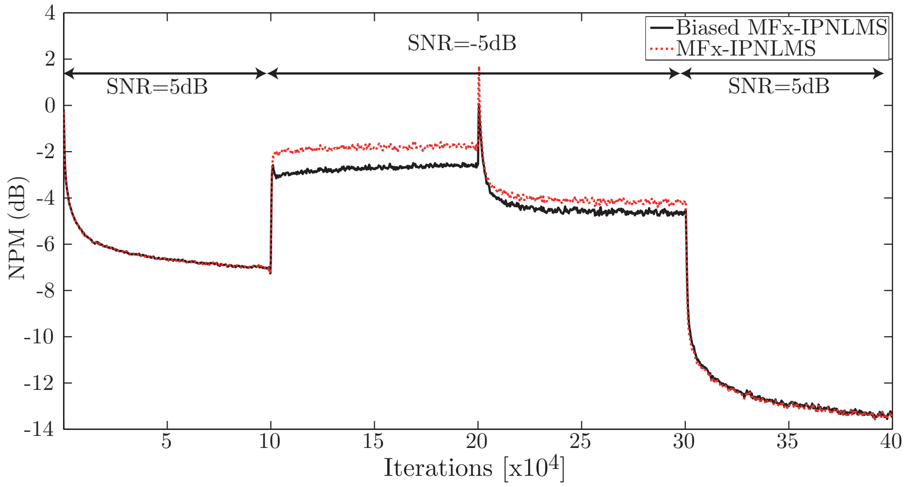


Figure 3.13. Sparse room impulse responses.



(a)



(b)

Figure 3.14. NPM evolution for the 1:2:2 configuration for: (a) microphone 1, and (b) microphone 2.

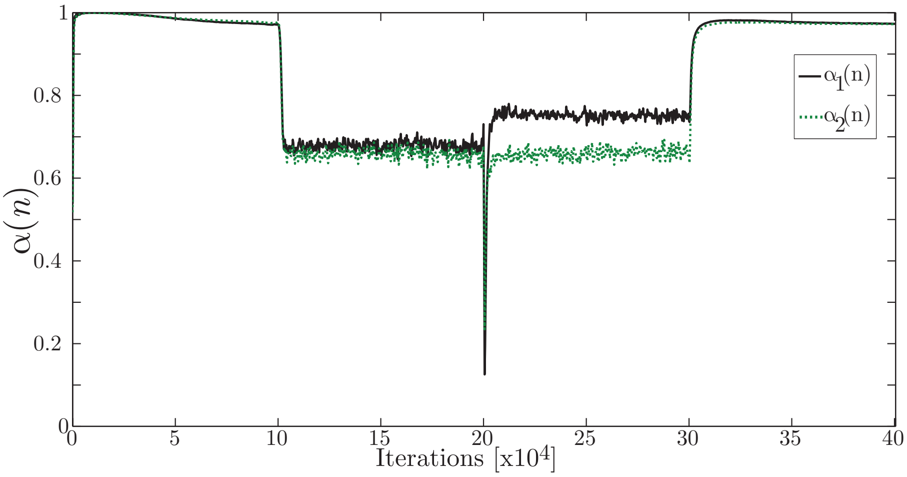


Figure 3.15. Scaling factor evolution for the 1:2:2 AE system.

3.6 Conclusions

In this chapter the adaptive filtered-x structure has been applied to the room equalization problem. Firstly, for the single channel case, the filtered-x IPNLMS algorithm has been presented. Moreover, the combination of filters has been introduced, which exhibits a robust performance in different scenarios. For low SNRs, the biased scheme has been proposed and analyzed in terms of steady state, with an improvement in the EMSE measure for low SNRs conditions. Due to the fact that the scaling factor decreases its value and thus biased the final EMSE, as the SNR is reduced.

In the following chapter and motivated by the sparseness characteristic of the acoustic channels, we use the combination and biased strategies to develop robust filtering schemes based on a block scheme.

The multichannel adaptive equalization problem has been also introduced. It requires to filter each input signal through all the acoustic channels measured between the loudspeakers and microphones. Thus, the filtered-x structure has to be applied. For acoustic applications, the estimation of the acoustic paths usually requires at least a thousand of samples. For that reason, the computational burden of the multichannel structures highly increases with the number of loudspeakers and microphones.

Keeping this in mind, it is appropriate to seek for adaptive algorithms that require a lower computational cost, the use of parallel computing to provide efficient and real-time equalization and the use of collaborative filtering techniques that allow to distribute the filtering process between different nodes.

Block-based adaptive room equalization 4

Block-based adaptive room equalization 4

In the previous chapter we have seen interesting results for the proposed algorithms for the adaptive room equalization problem. For instance, the adaptive combination of filters allows to combine the performance of different filters to improve the overall result. However, they do not always provide optimal performance for this specific application. This is due to the particular form of the inverse room response, which presents different energy distribution among its coefficients, whereas the combination factor is the same for all of them. Since working with each coefficient of the adaptive filter independently is computationally expensive, the coefficients are put together in a block-based concept.

Taking this in mind, this chapter introduces different block-based schemes for the room equalization problem.

First, in Section 4.2, the adaptive block-based convex combination of filtered-x filters is presented, where the different combination factors seek minimizing the mean square error of the block filter and thus the mean square error of the total equalizer. Moreover, the block-based biased filtered-x scheme is also proposed, which allows to biased independently each block of the adaptive filter for a given SNR.

Secondly and based on the *a priori* knowledge of the energy distribution of the optimal filter, the fitted-block schemes are presented in Section 4.3, which suitably select the coefficients of each filter block to fit the energy distribution of the adaptive filter. Although these schemes are specifically designed for adaptive equalization applications in this work, they can be successfully applied to other applications where the energy distribution of the optimal solution is roughly *a priori* known.

Particularly, the fitted combination of block-based filtered-x schemes and the fitted block-based biased filtered-x scheme are introduced.

Within the fitted-block schemes, the partitioned fitted-block scheme is also presented, which uses different parameter values for each block, depending on the desired filter performance.

However, first of all in this chapter it is necessary to introduce the block-based concept and the energy distribution of the coefficients of the inverse filter.

4.1 Block-based approach

The idea of this block distribution is based on the concept that acoustic room channel responses have room impulse responses that are normally sparse with only a small percentage of the components that have high values and the rest with values close to zero, as it has been commented in Section 2.1.2.

Nevertheless, for room equalization applications, the adaptive filter tends to the inverse of the room response which is normally not-so-sparse and even is not directly related to the sparsity of its room response. Examples of these responses and their sparsity degree are given in the result section of this chapter. For instance, for the room response and inverse filter of Fig. 4.5, the sparsity degree measured with (2.3) is $\xi(\mathbf{h}) = 0.70$ and $\xi(\mathbf{w}) = 0.30$, respectively. Despite this not-so-sparse behavior, the inverse responses still exhibit a sparse response and thus, this block-based approach can be used.

The aim of the block-based structure is to divide the adaptive filter in various blocks, so each can have a different adaptation based on the energy of this part of the inverse acoustic channel $\mathbf{w}_o$. Moreover, when used in combination of different algorithms, it will allow to select the best option for each block, improving the one-single block scheme.

For the block algorithms, the adaptive filter $\mathbf{w}(n)$ is divided in M-blocks, that is, $\mathbf{w}(n) = [\mathbf{w}_1(n), \ldots, \mathbf{w}_M(n)]^T$, where each block has $Q = L_w/M$ samples, $\mathbf{w}_m(n) = [w_{(m-1)Q+1}, w_{(m-1)Q+2}, \ldots, w_{mQ}]^T$. The update of each block is independent, allowing each adaptive block filter $\mathbf{w}_m(n)$ to converge to the block solution

$$\mathbf{w}_{o,m} = [w_{o,(m-1)Q+1}, w_{o,(m-1)Q+2}, \ldots, w_{o,mQ}]^T. \tag{4.1}$$

It has to be noticed that the equalization filters tend to obtain a stable inversion of the acoustic channel, which is introduced using a modelling delay. For that reason, the response of the inverse filter has a central response with high energy, whereas the lateral coefficients present lower values. On this basis, the fitted-block schemes tries to adjust the block position to the coefficient energy distribution. Although we are using in this chapter adjacent coefficients to form a block, the aim of fitted-blocks involves the combination of non-adjacent coefficients.

4.2 Block-based schemes

To start with, in this section we present the convex combination of two block-based adaptive filters, which allows to combine each block independently. In fact, as the optimal filter can exhibit different sparseness degree in each block, it can outperform the one-single combination scheme of the previous chapter. Secondly, the block-based biased filtered-x structure is also introduced that can provide an improved final residual error for low SNRs and a wide range of sparseness degrees, as well. Although all the proposed filtering schemes can be implemented with different kinds of adaptive filters depending on both the application and performance requirements, our work as in the previous chapter is focused on efficiently increase the robustness of the Fx-IPNLMS algorithm for adaptive equalization applications.

4.2.1 Block-based convex combination of filtered-x schemes

For the adaptive convex combination of block-based filtered-x schemes (CBFx), the structure takes at each n-sample and for a given m-block (being $m = 1, \dots, M$), the best $\mathbf{w}_{i,m}(n)$-block or a combination of both $\mathbf{w}_{i,m}(n)$-blocks (being $i = 1, 2$). These combinations allow to finally obtain the whole adaptive filter $\mathbf{w}(n)$ as a combination of each filter block.

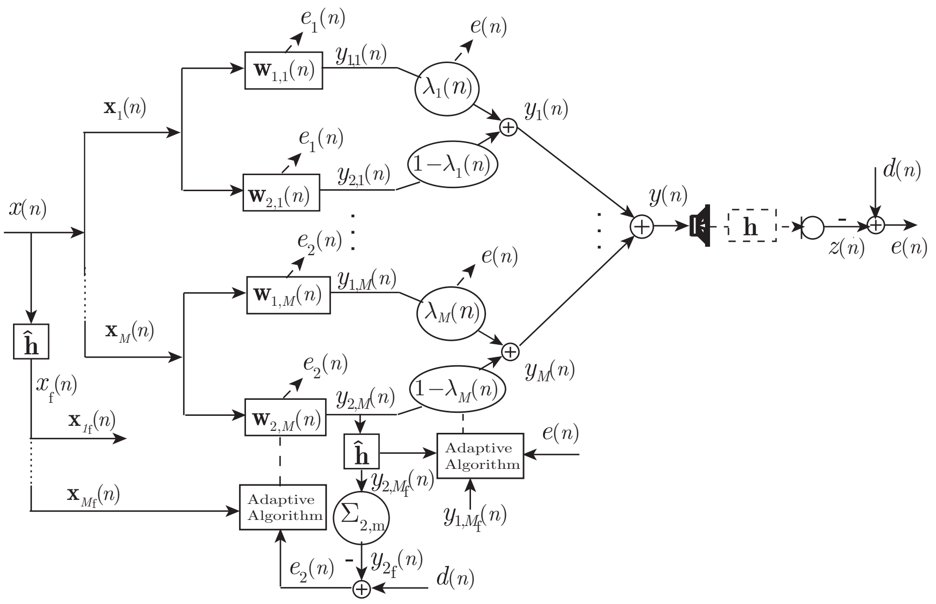


Figure 4.1. Adaptive combination of block-based adaptive filtered-x filters for equalization.

Fig.4.1 shows the different input and output signals and the errors that control the adaptive filters and combination factors of this structure. For simplicity, only the adaptation block of the $\mathbf{w}_{2,m}(n)$-block and the $\lambda_n(n)$ combined factor are drawn. The output of the adaptive filter $y(n)$ can be expressed as a weighted combination of each block filter:

$$\begin{aligned} y(n) = &\sum_{m=1}^{M} \left\{ \lambda_m(n) \left[\mathbf{w}_{1,m}^T(n-1)\mathbf{x}_m(n) \right] \right\} \\ &+ \sum_{m=1}^{M} \left\{ \left[1 - \lambda_m(n)\right] \left[\mathbf{w}_{2,m}^T(n-1)\mathbf{x}_m(n) \right] \right\}, \end{aligned} \tag{4.2}$$

where $\mathbf{w}_{1,m}(n)$ and $\mathbf{w}_{2,m}(n)$ are the adaptive weight block vectors of $Q = L_w/M$-length. Meanwhile, the $\mathbf{x}_{L_w}(n)$ vector containing the last L_w samples of the input signal $x(n)$ has been divided in M input blocks with Q samples each, $\mathbf{x}_m(n) = [x(n-[m-1]Q), x(n-[m-1]Q-1), \ldots, x(n-[m-1]Q-[Q-1])]^T$.

The final expression at the microphone signal will be

$$z(n) = \mathbf{h} * y(n) \tag{4.3}$$

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

$$e(n) = d(n) - z(n), \tag{4.4}$$

where $d(n)$ is defined as previously to allow each block to converge to a specific block solution $\mathbf{w}_{0,m}$, such that the desired signal is given by

$$d(n) = x(n-\tau) = \mathbf{w}_o^T \mathbf{x}_{\mathrm{f}}(n). \tag{4.5}$$

This error signal $e(n)$ will be used in the block case to update the M scaling factors $\lambda_m(n)$, which are defined as in the previous chapter by using a sigmoid activation function

$$\lambda_m(n) = \frac{sgm[a_m(n)] - sgm[-4]}{sgm[4] - sgm[-4]}, \quad \text{for} \quad m = 1, \ldots, M, \tag{4.6}$$

where $a_m(n)$ is updated according to the following expression

$$\begin{aligned} a_m(n+1) = a_m(n) + \frac{\overline{\mu}_{a_m}}{p_m(n)} e(n) \Big[y_{1,m\mathrm{f}}(n) - y_{2,m\mathrm{f}}(n) \Big] \cdot \\ \cdot\, sgm[a_m(n)] \left(1 - sgm[a_m(n)]\right), \end{aligned} \tag{4.7}$$

where $y_{i,m\mathrm{f}}(n)$ ($i = 1, 2$ for each branch of the convex structure for a given m-block) is the output signal $y_{i,m}(n)$ of the adaptive m-block filter $\mathbf{w}_{i,m}(n)$ filtered through the estimated impulse response $\hat{\mathbf{h}}$. Besides, $\overline{\mu}_{a_m}$ is the adaptation speed and $p_m(n)$ the normalization factor that are defined by

$$\overline{\mu}_{a_m} = \frac{\mu_{a_m}}{sgm[4] - sgm[-4]}, \tag{4.8}$$

$$p_m(n) = \beta p_m(n-1) + (1-\beta) \Big[y_{1,m\mathrm{f}}(n) - y_{2,m\mathrm{f}}(n) \Big]^2. \tag{4.9}$$

The weights of each block of the convex combination structure for the LMS-type algorithm are updated at each iteration according to

$$\mathbf{w}_{i,m}(n) = \mathbf{w}_{i,m}(n-1) + \mathbf{G}_{i,m}(n)e_i(n)\mathbf{x}_{m\mathrm{f}}(n), \quad \text{for } i = 1,2; \tag{4.10}$$

being $\mathbf{G}_{i,m}(n)$ a diagonal matrix containing in the main diagonal the adaptation speed for each filter weight $\mu_{i,m_q}(n)$, for $i = 1,2$ and $m_q = (m-1)Q, \ldots, (m-1)Q + (Q-1)$, are the Q-coefficients of the mth-block filter. $\mathbf{x}_{m\mathrm{f}}(n)$ is a vector containing Q samples of the input signal $x(n)$ filtered through the estimated impulse response $\hat{\mathbf{h}}$.

Whereas, $e_i(n)$ represents the input error to the adaptive i-block algorithm, which is computed as

$$e_i(n) = d(n) - y_{i\mathrm{f}}(n), \tag{4.11}$$

where $y_{i\mathrm{f}}(n) = \sum_{m=1}^{M} y_{i,m\mathrm{f}}(n), \quad \text{for } i = 1,2.$

Furthermore, the adaptation speed for each filter block is the same for each weight for a CBFx-NLMS algorithm $\mu_{i,m_q}(n) = \mu_{i,m}(n)$, with μ_i being a constant step size, for the ith adaptation filter of the combination scheme:

$$\mu_{i,m}(n) = \frac{\mu_i}{\delta + \sum_k x_{\mathrm{f}}^2(n-k)}, \tag{4.12}$$

for $k = (m-1)Q, \ldots, (m-1)Q + (Q-1)$, $i = 1,2$ and $m = 1, \ldots, M$. Where δ is a small positive constant to avoid division by zero.

Meanwhile, for a CBFx-IPNLMS algorithm, the adaptation speed for each filter block, for $m = 1, \ldots, M$, is different for each weight $\mu_{i,m_q}(n)$:

$$\mu_{i,m_q}(n) = \frac{\mu_i g_{i,m_q}(n-1)}{\delta + \sum_k g_{i,m_q}(n-1)x_{\mathrm{f}}^2(n-k)}, \tag{4.13}$$

for $k = (m-1)Q, \ldots, (m-1)Q + (Q-1)$, $i = 1,2$, $m = 1, \ldots, M$ and m_q refers to the qth coefficient of the m-block. Where the adaptation gain factors are given by

$$g_{i,l}(n) = (1-\kappa_{i,m})\frac{1}{2L_w} + (1+\kappa_{i,m})\frac{|w_{i,l}(n)|}{\varepsilon + 2\sum_{k=0}^{L_w-1}|w_{i,k}(n)|}, \tag{4.14}$$

where $w_{i,l}(n)$ for $l = 1, \ldots, L_w$ refers to the lth-coefficient of the $\mathbf{i}$th filter vector. δ and ε are small constants to avoid division by zero.

The computation of the mixing parameters and the update of the Q coefficients of the mth block of every component filter for the Fx-IPNLMS algorithm are illustrated in **Algorithm 6**.

Algorithm 6 CBFx-IPNLMS algorithm.

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

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

```
Update the vectors $\mathbf{x}_m(n)$ (for $m = 1, \dots, M$) and $\mathbf{x}_h(n)$
$x_{\mathrm{f}}(n) = \mathbf{x}_h^T(n)\widehat{\mathbf{h}}$
$y_{i,m}(n) = \mathbf{w}_{i,m}^T(n-1)\mathbf{x}_m(n), \quad i = 1, 2, \quad m = 1, \dots, M$
Update the vectors $\mathbf{x}_{m\mathrm{f}}(n)$, $\mathbf{y}_{1,m}(n)$ and $\mathbf{y}_{2,m}(n)$
    for $m = 1, \dots, M$
$y_{i,m\mathrm{f}}(n) = \mathbf{y}_{i,m}^T(n)\widehat{\mathbf{h}}, \quad i = 1, 2, \quad m = 1, \dots, M$
$e(n) = d(n) - z(n)$
$e_i(n) = d(n) - \sum_{m=1}^{M} y_{i,m_{\mathrm{f}}}(n), \quad i = 1, 2$
$p_m(n) = \beta p_m(n-1) + (1-\beta)[y_{1,m\mathrm{f}}(n) - y_{2,m\mathrm{f}}(n)]^2,$
    for $m = 1, \dots, M$
$a_m(n) = a_m(n-1)$
    $+\frac{\mu_{a_m}}{p_m(n)} e(n)\,[y_{1,m\mathrm{f}}(n) - y_{2,m\mathrm{f}}(n)]\, sgm[a_m(n)]\,\{1 - sgm[a_m(n)]\},$
    for $m = 1, \dots, M,$
$\lambda_m(n) = \frac{sgm[a_m(n)] - sgm[-a^+]}{sgm[a^+] - sgm[-a^+]}, \quad m = 1, \dots, M$
$y(n) = \sum_{m=1}^{M} \lambda_m(n) y_{1,m}(n) + [1 - \lambda_m(n)] y_{2,m}(n)$
$g_{i,l}(n-1) = (1-\kappa_{i,m})\frac{1}{2L_w} + (1+\kappa_{i,m})\frac{|w_{i,l}(n-1)|}{\varepsilon + 2\sum_{k=0}^{L_w-1} |w_{i,k}(n-1)|},$
    for $l = 1, \dots, L_w, \quad i = 1, 2$ and $\quad m = 1, \dots, M.$
$\mu_{i;m_q}(n) = \frac{\mu_i g_{i,m_q}(n-1)}{\delta + \sum_k g_{i,m_q}(n-1) x_{\mathrm{f}}^2(n-k)},$
    for $k = (m-1)Q, \dots, (m-1)Q + (Q-1)$, $i = 1, 2$, $m = 1, \dots, M$ and $q = 1, \dots, Q$.
$\mathbf{G}_{i,m}(n) = diag(\mu_{i,m_1}(n), \mu_{i,m_2}(n), ..., \mu_{i,m_Q}(n)),$
    for $i = 1, 2, \quad m = 1, \dots, M,$
$\mathbf{w}_{i,m}(n) = \mathbf{w}_{i,m}(n-1) + \mathbf{G}_{i,m(n)}\mathbf{x}_{m\mathrm{f}}(n) e_i(n),$
    for $i = 1, 2, \quad m = 1, \dots, M$
```

4.2.2 Block-based biased filtered-x scheme

The used of the biased scheme proposed in the previous chapter in the block-based approach allows to bias each block independently. Thus, for a given SNR, as the energy distribution of each block is different, the block-based biased scheme will provide a different scaling factor for each block output.

For the block-based biased filtered-x (BBFx-) scheme, we consider a CBFx structure where all the block filters of one branch are equal to zero.

Fig. 4.2 shows the BBFx scheme, where i-subscripts have been omitted, as they are not necessary. The previously defined expressions in 4.2.1 can be rewritten as follows,

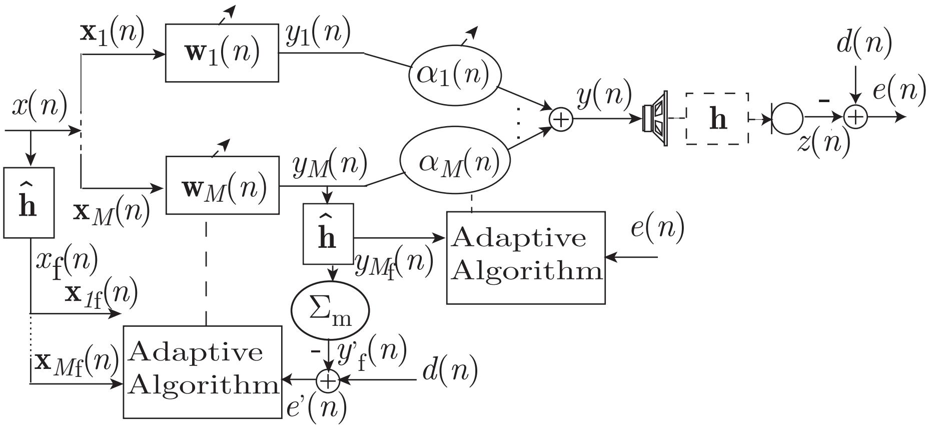


Figure 4.2. Block diagram of an adaptive block-based biased filtered-x scheme for equalization.

$$y(n) = \sum_{m=1}^{M} \left\{ \alpha_m(n) \left[\mathbf{w}_m^T(n-1)\mathbf{x}_m(n) \right] \right\}. \tag{4.15}$$

The error signal $e(n)$ defined as in (4.4) will be used in the block case to update the M scaling factors $\alpha_m(n)$, which are defined as in the previous section by using a sigmoid activation function

$$\alpha_m(n) = \frac{sgm[a_m(n)] - sgm[-4]}{sgm[4] - sgm[-4]}, \text{for } m = 1, \ldots, M, \tag{4.16}$$

where $a_m(n)$ is updated for the block-based filtered-x algorithm according to the following expression,

$$a_m(n+1) = a_m(n) + \frac{\overline{\mu}_{a_m}}{p_m(n)} e(n) y_{m\mathrm{f}}(n) sgm[a_m(n)] \left(1 - sgm[a_m(n)]\right), \tag{4.17}$$

where $y_{m\mathrm{f}}(n)$ is the output signal $y_m(n)$ of the adaptive m-block filter $\mathbf{w}_m(n)$ filtered through the estimated impulse response $\hat{\mathbf{h}}$. Besides, $\overline{\mu}_{a_m}$ is the adaptation speed defined as in (4.8) and $p_m(n)$ the normalization factor,

$$p_m(n) = \beta p_m(n-1) + (1-\beta) y^2_{m\mathrm{f}}(n). \tag{4.18}$$

The weights of each block of the biased structure are updated at each iteration according to

$$\mathbf{w}_m(n) = \mathbf{w}_m(n-1) + \mathbf{G}_m(n) e^{'}(n) \mathbf{x}_{m\mathrm{f}}(n), \tag{4.19}$$

being $\mathbf{G}_m(n)$ a diagonal matrix containing in the main diagonal the adaptation speed for each filter weight $\mu_{m_q}(n)$ defined as in (4.12) - (4.14).

Meanwhile, $e'(n)$ represents the input error to the adaptive block weight filter, which is computed as

$$e^{'}(n) = d(n) - y^{'}_{f}(n), \tag{4.20}$$

where $y^{'}_{f}(n) = \sum_{m=1}^{M} y_{m\mathrm{f}}(n)$.

The BBFx-IPNLMS algortihm is summarized in **Algorithm 7**.

Algorithm 7 BBFx-IPNLMS algorithm.

```
Input: Reference signal $x(n)$, desired signal $d(n)$ and microphone signal
    $z(n)$.
Output: Output of the parallel filter $y(n)$
Update the vectors $\mathbf{x}_m(n)$, for $m = 1, \ldots, M$ and $\mathbf{x}_h(n)$.
$x_{\mathrm{f}}(n) = \mathbf{x}_h^T(n)\widehat{\mathbf{h}}$
$y_m(n) = \mathbf{w}_m^T(n-1)\mathbf{x}_m(n), \quad m = 1, \ldots, M.$
Update the vectors $\mathbf{x}_{m\mathrm{f}}(n)$, $\mathbf{y}_m(n)$, for $m = 1, \ldots, M$.
$y_{m\mathrm{f}}(n) = \mathbf{y}_m^T(n)\widehat{\mathbf{h}}, \quad m = 1, \ldots, M$
$e(n) = d(n) - z(n)$
$e'(n) = d(n) - \sum_{m=1}^{M} y_{m\mathrm{f}}(n)$
$p_m(n) = \beta p_m(n-1) + (1-\beta)y_{m\mathrm{f}}^2(n)$, for $\quad m = 1, \ldots, M$
$a_m(n) = a_m(n-1)$
    $+\frac{\mu_{a_m}}{p_m(n)} e(n)y_{m\mathrm{f}}(n)sgm[a_m(n)]\{1 - sgm[a_m(n)]\}$, for $m = 1, \ldots, M$
$\alpha_m(n) = \frac{sgm[a_m(n)] - sgm[-a^+]}{sgm[a^+] - sgm[-a^+]}$, for $\quad m = 1, \ldots, M$
$y(n) = \sum_{m=1}^{M} \alpha_m(n)y_m(n)$
$g_l(n-1) = (1-\kappa_m)\frac{1}{2L_w} + (1+\kappa_m)\frac{|w_l(n-1)|}{\varepsilon + 2\sum_k |w_k(n-1)|}$,
    for $l = 1, \ldots, L_w$.
$\mu_{m_q}(n) = \frac{\mu g_{m_q}(n-1)}{\delta + \sum_k g_k(n-1)x_{\mathrm{f}}^2(n-k)}$,
    for $k = (m-1)Q, \ldots, (m-1)Q + (Q-1)$, $m = 1, \ldots, M$ and
    $q = 1, \ldots, Q$.
$\mathbf{G}_m(n) = diag(\mu_{m_1}, \mu_{m_2}, \ldots, \mu_{m_Q})$, for $\quad m = 1, \ldots, M$.
$\mathbf{w}_m(n) = \mathbf{w}_m(n-1) + \mathbf{G}_m(n)\mathbf{x}_{m\mathrm{f}}(n)e'(n)$, for $\quad m = 1, \ldots, M$.
```

4.2.3 Steady-state analysis of the BBFx-IPNLMS algorithm

In this section the EMSE of the BBFx scheme is computed as it was developed for the BFx scheme in Section 3.3.1.

For the sake of simplicity, we consider each coefficient separately. Thus, we use the filter coefficients $w_l(n)$ for $l = 1, \dots, L_w$, and the scaling factors that multiplies each block, $\alpha_m(n)$ for $m = 1, \dots, M$ can be rewritten as $\alpha_{\lceil \frac{l}{Q} \rceil}(n)$. That means, m is assigned to each filter coefficient $w_l(n)$ depending on the filter block it belongs to, rounding up the division i/Q towards the nearest integer. As in the one-block algorithm, we consider that for low variations of the weight vector, $\mathbf{h}$ and $\mathbf{w}^T(n)$ are considered linear systems that can be interchanged. Thus, the microphone signal $z(n)$ and the error signal of Fig. 4.2 can be written respectively as

$$z(n) = \sum_{l=1}^{L_w} \alpha_{\lceil \frac{l}{Q} \rceil}(n) w_l^T(n) x_{\mathrm{f}}(n-l). \tag{4.21}$$

$$\begin{aligned} e(n) &= \sum_{l=1}^{L_w} \left[w_{\mathrm{o},l} - \alpha_{\lceil \frac{l}{Q} \rceil}(n) w_l(n) \right]^T x_{\mathrm{f}}(n-l) + r(n) \\ &= \Psi^T(n)\mathbf{x}_{\mathrm{f}}(n) + r(n) = \Psi^T(n)\widehat{\mathbf{H}}^T \mathbf{x}_{wh}(n) = e_a(n) + r(n), \end{aligned} \tag{4.22}$$

where $r(n)$ is an additive Gaussian noise with zero mean and variance σ_r^2 and uncorrelated with the input signal $\mathbf{x}(n)$. Being $\Psi(n)$ a vector containing the $\psi_l(n)$ terms for $l = 1, \dots, L_w$, $\Psi(n) = [\psi_1(n), \dots, \psi_{L_w}(n)]^T$, where $\psi_l(n) = w_{0,l} - \alpha_{\lceil \frac{l}{Q} \rceil}(n) w_l(n)$ has been used for compactness. $\mathbf{x}_{wh}(n)$ is a column vector containing the last $L_w + L_h - 1$ samples of the input signal $x(n)$ that multiplied by the convolution matrix $\widehat{\mathbf{H}}^T$ of size $L_w \times L_w + L_h - 1$ represents the filter vector $\mathbf{x}_{\mathrm{f}}(n)$.

$$\widehat{\mathbf{H}} = \begin{bmatrix} \hat{h}(0) & 0 & \cdots & 0 \\ \hat{h}(1) & \hat{h}(0) & \cdots & 0 \\ \vdots & \hat{h}(1) & \ddots & \vdots \\ \hat{h}(L_h-1) & \vdots & \ddots & \hat{h}(0) \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \hat{h}(L_h-1) \end{bmatrix} \tag{4.23}$$

Following the analysis in [98], the EMSE of the filter is therefore given by

$$\begin{aligned} J_{ex}(n) =& \, E\{e_a^2(n)\} = E\left\{\Psi^T(n)^T\widehat{\mathbf{H}}^T\mathbf{x}_{wh}(n)\mathbf{x}_{wh}^T(n)\widehat{\mathbf{H}}\Psi(n)\right\} \\ &\approx E\left\{\Psi^T(n)\widehat{\mathbf{H}}^T E\left\{\mathbf{x}_{wh}(n)\mathbf{x}_{wh}^T(n)\right\}\widehat{\mathbf{H}}\Psi(n)\right\}, \end{aligned} \tag{4.24}$$

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

Assuming the step size is small enough to allow the filter coefficients to follow the average statistics of the input signal (see the Direct Averaging Method applied in [12]), the term $\mathbf{x}_{wh}(n)\mathbf{x}_{wh}^T(n)$ can be approximated by its expected value. If we consider the statistics of the input signal, for a white noise signal of zero mean and σ_x^2 variance, the following term in (4.24) can be simplified, $E\left\{\mathbf{x}_{wh}(n)\mathbf{x}_{wh}^T(n)\right\} = \sigma_x^2\mathbf{I}_{(L_w+L_h-1)}$, where $\mathbf{I}$ is the identity matrix. (4.24) can be rewritten as

$$J_{ex}(n) = \sigma_x^2 E\left\{\Psi^T(n)\widehat{\mathbf{H}}^T\widehat{\mathbf{H}}\Psi(n)\right\} \tag{4.25}$$

After some manipulations and using the symmetrical property of the correlation matrix, (4.25) can be rewritten as,

$$J_{ex}(n) = \sigma_x^2 E\left[\Psi^T(n)\Phi(n)\right]\mathbf{D}\widehat{\mathbf{H}}_{L_h}^T\widehat{\mathbf{h}}, \tag{4.26}$$

where matrix $\Phi(n)$ of size $L_w \times L_h$ is a convolution matrix built with the $\psi_l(n)$ coefficients, whose first column is $\Psi(n)$ and the following columns are built by shifting down this vector and zero-padding at the beginning.

$$\begin{aligned} \Psi^T(n)\Phi(n) = & \left[\begin{array}{cccc} \psi_1(n) & \psi_2(n) & \dots & \psi_{L_w}(n) \end{array}\right] \cdot \\ \cdot & \left[\begin{array}{cccc} \psi_1(n) & 0 & \dots & 0 \\ \psi_2(n) & \psi_1(n) & \dots & \vdots \\ \vdots & \vdots & \ddots & \psi_1(n) \\ \vdots & \vdots & \ddots & \vdots \\ \psi_{L_w}(n) & \psi_{L_w-1}(n) & \dots & \psi_{L_w-L_h+1}(n) \end{array}\right] \end{aligned} \tag{4.27}$$

$\mathbf{D}$ is a diagonal matrix of L_h length, whose first element is equal to 1, and the rest of the elements of the main diagonal are 2 to implement the symmetric factor. Furthermore, $\widehat{\mathbf{H}}_{L_h}^T \widehat{\mathbf{h}}$ implements the correlation of the channel response, where $\widehat{\mathbf{H}}_{L_h}^T$ is a matrix composed with the first L_h columns of $\widehat{\mathbf{H}}^T$.

For $\mathbf{h} = \delta(n-\tau)$ (propagation in free field conditions), $\widehat{\mathbf{H}}^T \widehat{\mathbf{h}} = [1, 0, \ldots, 0]^T$, and the multiplication by matrix $\mathbf{D}$ gives also the L_h length vector $[1, 0, \ldots, 0]^T$. This vector activates only the first column of $\Phi(n)$ in (4.27), and thus expression (4.26) is the same as the one obtained in [98], when the filtered-x structure is not considered.

$$J_{ex}(n) = \sigma_x^2 E\left[\Psi^T(n)\Psi(n)\right] \tag{4.28}$$

Expression (4.27) shows the influence of the channel filtering among the $\Psi_l(n)$ coefficients. Thus, (4.26) can be rewritten as

$$\begin{aligned} J_{ex}(n) &= \sigma_x^2 \|\widehat{\mathbf{h}}\|_2^2 E\left[\Psi^T(n)\Psi(n)\right] \\ &+ 2\sigma_x^2 \sum_{\tau=1}^{L_h-1} r(\tau) \sum_{l=1}^{L_w-\tau} E\left[\psi_l(n)\psi_{l+\tau}(n)\right], \end{aligned} \tag{4.29}$$

where $r(\tau) = \sum_{k=\tau}^{L_h-1} \widehat{h}(k)\widehat{h}(k-\tau)$ is the correlation of the channel response. For $\tau = 0$, $r(0) = \|\widehat{\mathbf{h}}\|_2^2$ which appears as the first term in (4.29).

After some manipulations and in steady state ($n \longrightarrow \infty$), using the coefficient weight error $\widetilde{w}_l(n)$ in $\psi_l(n) = w_{o,l} - \alpha_m(n) w_{il}(n) = \alpha_m(n)\widetilde{w}_l(n) + [1 - \alpha_m(n)] w_{o,l}$ and assuming independence of the filter coefficients and that they tend to their optimal solution, $E\{\widetilde{w}_l(n)\widetilde{w}_{l+\tau}(n)\} = E\{\widetilde{w}_l(n)\}$ $E\{\widetilde{w}_{l+\tau}(n)\} = 0$, where $E\{w_l(\infty)\} = w_{0,l}$, we get (4.30). The first term has the effect of the channel energy $\|\widehat{\mathbf{h}}\|_2^2$, and there is an additional second term dependent on the correlation of the channel response $r(\tau)$ for $\tau = 1, \ldots, L_h - 1$.

$$J_{ex}(\infty) = \sigma_x^2 \|\widehat{\mathbf{h}}\|_2^2 \left[\sum_{m=1}^{M} \alpha_m^2 E\left\{\|\widetilde{\mathbf{w}}_m(\infty)\|_2^2\right\} \right.$$
$$\left. + \sum_{m=1}^{M} (1-\alpha_m)^2 \|\mathbf{w}_{\text{o},m}\|_2^2 \right] + 2\sigma_x^2 \sum_{\tau=1}^{L_h-1} r(\tau)$$
$$\left[\sum_{l=1}^{L_w-\tau} \left(1-\alpha_{\lceil \frac{l}{Q} \rceil}\right) \left(1-\alpha_{\lceil \frac{l+\tau}{Q} \rceil}\right) w_{o,l} w_{o,l+\tau} \right]. \tag{4.30}$$

If only one block is considered ($\alpha_m = \alpha$) as in Section 3.3.1, the coefficients in (4.22) can be rewritten as $\psi_l(n) = \alpha(n)\widetilde{w}_l(n) + [1-\alpha(n)]w_{o,l}$. Leading to the EMSE in steady state for the BFx scheme,

$$J_{ex}(\infty) = \alpha^2 \sigma_x^2 \|\widehat{\mathbf{h}}\|_2^2 \sum_{l=1}^{L_w} E\left\{\|\widetilde{w}_l(\infty)\|_2^2\right\}$$
$$+\sigma_x^2 (1-\alpha)^2 \mathbf{w}_{\text{o}}^T \widehat{\mathbf{H}}^T \widehat{\mathbf{H}} \mathbf{w}_{\text{o}}, \tag{4.31}$$

where $\mathbf{w}_{\text{o}}^T \widehat{\mathbf{H}}^T \widehat{\mathbf{H}} \mathbf{w}_{\text{o}} = 1$ for a perfect equalization.

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

Following the development as in [99], the weight error coefficients can be obtained as,

$$E\left\{\|\widetilde{w}_l(\infty)\|_2^2\right\} = \frac{\mu\sigma_r^2}{\sigma_x^2 \|\widehat{\mathbf{h}}\|_2^2} \frac{g_l(\infty)}{2-\mu g_l(\infty)}, \quad l = 1, \ldots, L_w \tag{4.32}$$

$$g_l(\infty) = (1-\kappa)\frac{1}{2L_w} + (1+\kappa)\frac{|w_{0,l}|}{\varepsilon + 2\sum_k |w_{0,k}|}, \quad l = 1, \ldots, L_w \tag{4.33}$$

Finally,

$$E\left\{\|\widetilde{\mathbf{w}}_m(\infty)\|_2^2\right\} = \sum_{l=1}^{Q} E\left\{\left\|\widetilde{w}_{(m-1)Q+l}(\infty)\right\|_2^2\right\}. \tag{4.34}$$

In steady state, if the scaling factors tend to the optimal solution α_m^*, we can obtain this solution from (4.30) as $\dfrac{\partial J_{ex(\infty)}}{\partial \alpha_m} = 0$. The biased terms can be obtained as

$$\alpha_m^* = \frac{1}{1 + \frac{E\left\{\|\widetilde{\mathbf{w}}_m(\infty)\|_2^2\right\}}{\|\mathbf{w}_{\mathrm{o},m}\|_2^2}}. \tag{4.35}$$

In the result section of this chapter, this theoretical values are compared to the experimental ones to provide an insight of this analysis.

4.3 Fitted block-based schemes.

Taking advantage of the *a priori* knowledge of the inverse channel coefficient distribution, we proposed in this section two new block approaches based on the idea of adjusting the block position to the coefficient energy distribution.

Firstly, we introduce a modification of the two block-based structures of the previous section (the CBFx and BBFx schemes) when used in a room equalization context, that consists in designing the block filter location depending on the energy distribution of the adaptive filter.

Secondly, a partitioned fitted-block scheme is proposed. This new structure uses different parameters settings for each filter block depending on the goal we want to achieve. For instance, if we know *a priori* the sparsity degree of each block, we can apply a different κ value to the Fx-IPNLMS algorithm of each block.

4.3.1 Fitted CBFx and BBFx schemes.

From the CBFx and BBFx schemes of Sections 4.2.1 and 4.2.2, we derive directly the combination of fitted BFx (CFBFx) and the fitted BBFx (FBBFx) schemes. The expressions and algorithms of these fitted structures are the same as those defined in Sections 4.2.1 and 4.2.2, with the difference that the block coefficients are not define from the first coefficient to the Q-coefficient for the first block, but taking the central coefficients with more energy in the same block.

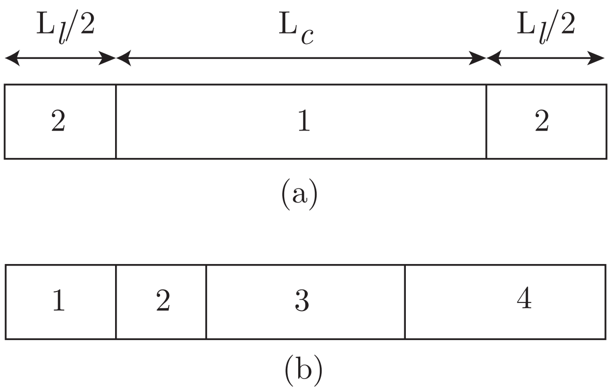


Figure 4.3. Block-coefficient distribution of a fitted-block algorithm with: (a) 2 blocks and (b) 4 blocks, with the third block containing the central coefficients.

The simplest fitted-block algorithm consists in a filter with only two blocks (see Fig. 4.3 (a)). The first block comprised of the L_c central coefficients and the second block that includes the remaining coefficients at both sides with a total length of L_l coefficients. Thus, the whole adaptive filter has a length of $L_w = L_c + L_l$ taps.

It is important to note that although we have used a fitted-block approach with two blocks, it is straightforward to extend to any number of blocks, provided that the central coefficients are located in the same block filter, as can be seen in Fig. 4.3 (b).

The question that arises is the size of the central block L_c that depends on the acoustic channel. That way, the performance of the adaptive filter can be deteriorated if the size of the central block is not properly selected. This fact suggests the use of a combined algorithm with different central-blocks sizes. Fig. 4.4 shows a combination of two FBBFx algorithms, that leads to the combined FBBFx scheme (CFBBFx). For simplicity, only two blocks are used in each algorithm, see Fig. 4.3 (a). As the central (c) and lateral (l) blocks have a different number of coefficients, the combination of both algorithms is done after adding the biased weights. For this reason, we consider different input vectors named $\mathbf{x}_{c_i}(n)$ and $\mathbf{x}_{l_i}(n)$ for $i = 1, 2$, showing that there is a different number of input signal samples for each filter block.

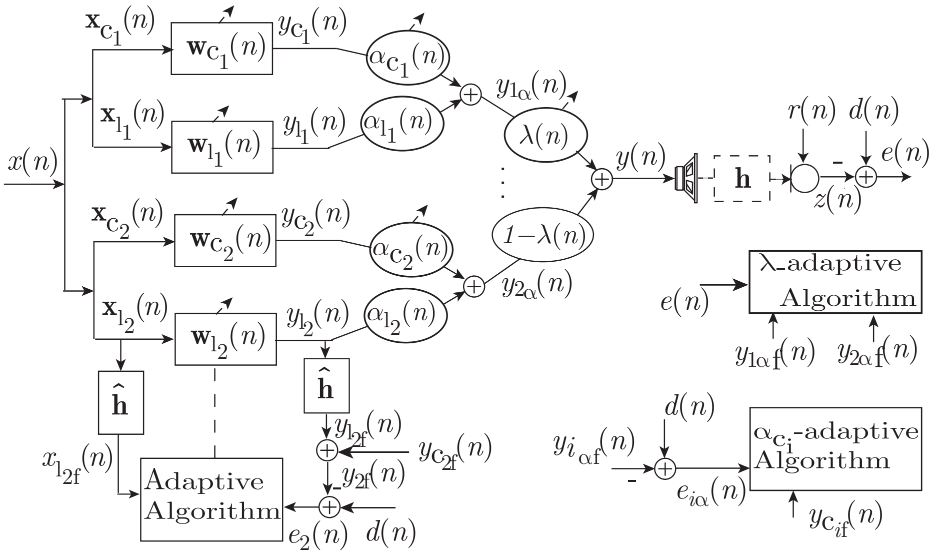


Figure 4.4. Combination of fitted 2-block biased filtered-x algorithms.

4.3.2 Partitioned fitted-block schemes.

For the partitioned FBFx (PFBFx) scheme, each block is modeled using a different parameter setting based on the algorithms goal, using the *a priori* knowledge of the coefficient energy distribution. For instance, the partitioned block algorithm presented in [100] for echo cancellation, which uses a different κ value for each block depending on the degree of sparsity of each block, can be implemented in an adaptive equalization context, where the Fx-IPNLMS algorithm with a high κ value can be used for the block with a high degree of sparsity and a Fx-NLMS algorithm can be used to estimate the coefficients of the block with a dispersive response.

For the partitioned scheme, to ensure the constraint on $\sum_{l=1}^{L_w} g_l(n) = \sum_{l_c=1}^{L_c} g_{l_c}(n) + \sum_{l_l=1}^{L_l} g_{l_l}(n) \approx 1$ to avoid problems during regularization of the PFBFx-IPNLMS algorithm, a normalized factor has to be applied to the adaptation gain factors $g_l(n)$ given in 4.14. Results in [100] showed that a good convergence speed is achieved for dispersive and sparse responses with a normalized factor (defined as $\frac{\|\mathbf{w}_c\|_1}{\|\mathbf{w}\|_1}$ and $\frac{\|\mathbf{w}_l\|_1}{\|\mathbf{w}\|_1}$) close to 0.5, where $\|\mathbf{w}_c\|_1$, $\|\mathbf{w}_l\|_1$ and $\|\mathbf{w}\|_1$ correspond to the ℓ_1-norm of the central, lateral and completed adaptive filters, respectively. Meanwhile, normalized factors close to 0 and 1 get a slower convergence speed, especially for dispersive filters.

Some experiments are carried out to study the performance improvement of the PFBFx scheme. As it will be shown experimentally, this roughly *a priori* knowledge is not so accurate to determine both the best block sizes and κ values, which depend on the inverse filter. For that reason, a combination partitioned fitted-block scheme (CPFBFx) that usually exhibits a good performance is also implemented.

4.4 Computational Cost

In this section, we analyze the computational complexity of the proposed algorithms in terms of multiplications per iteration. This study tries to bring up the number of operations that highly increase for block-based schemes when dealing with filtered-x structures. For that reason, we will also give, for comparison purpose, the computational burden of the schemes of the previous chapter, that is when only 1 block is considered.

Regarding the Fx-IPNLMS adaptive filter, it requires $6L_w+2$ multiplications more than the simplest of the LMS-type algorithm, the Fx-NLMS. In the case of the combination schemes, computation of the combination factor involves $(2L_h+11)M$ multiplications more, including also the additional filtering through the estimated channel.

For the CFBBFx algorithm, see the block diagram of Fig. 4.4, the computational cost in Table 4.1 is given for both blocks using the same parameter settings, first using two Fx-NLMS algorithms and secondly using two Fx-IPNLMS algorithms. In case the partitioned block scheme is used (as explained in Section 4.3.2) with a Fx-IPNLMS algorithm for the central block and a Fx-NLMS algorithm for the lateral blocks, the computational burden of the adaptive filter will be of $L_h+2(2L_w+6L_c+5)$ multiplications.

The computational complexity of the different schemes considered is summarized in Table 4.1, where also an example has been given at the last column for $L_h = 512$, $L_w = 1024$ and $M = 16$. As the proposed

Table 4.1. Summary of the computational complexity of the proposed schemes. Complexity is measured as the number of multiplications per iteration of the considered algorithms. Typical case: $L_h = 512$, $L_w = 1024$, $M = 16$.

Algorithm	**Multiplications**	**Typical case**
Fx-NLMS	$\mathbf{L_h + 3L_w + 2}$	3586
Fx-IPNLMS	$\mathbf{L_h + 9L_w + 4}$	9732
CFx-NLMS	$\mathbf{3L_h + 5L_w + 15}$	6671
CFx-IPNLMS	$\mathbf{3L_h + 17L_w + 19}$	18963
CBFx-NLMS	$\mathbf{L_h(2M + 1) + 5L_w + 11M + 4}$	22196
CBFx-IPNLMS	$\mathbf{L_h(2M + 1) + 17L_w + 11M + 8}$	34488
BBFx-NLMS	$\mathbf{L_h(M + 1) + 3L_w + 10M + 2}$	11938
BBFx-IPNLMS	$\mathbf{L_h(M + 1) + 9L_w + 10M + 4}$	18084
CFBBFx-NLMS	$\mathbf{7L_h + 5L_w + 54}$	8758
CFBBFx-IPNLMS	$\mathbf{7L_h + 17L_w + 59}$	21051

CFBBFx algorithm of Fig. 4.4 requires only 2 blocks ($M = 2$), it reduces the computational burden, when compared with the CBFx schemes with higher M values, although it also provides good results.

4.5 Results

In this section, we present several experiments to evaluate the behavior of the different proposed blocked-based schemes for room equalization. Moreover, the Fx-IPNLMS with different κ values will be used for comparison purposes. Hereafter, we will refer to the Fx-NLMS algorithm also with the Fx-IPNLMS algorithm with $\kappa = -1$. The evaluation of the algorithms behavior is based on the figure of merit EMSE, previously defined, that has been estimated by averaging over 100 independent runs of the algorithms.

The input signal $x(n)$ is a white Gaussian noise with zero mean and unit variance, while the additive noise $r(n)$ is also a Gaussian noise uncorrelated with $x(n)$ and its variance is adjusted to obtain different SNR. For the acoustic impulse responses, three acoustic paths with different degrees of sparsity have been considered (see Fig. 4.5 (a): $\xi = 0.89$, (c): $\xi = 0.70$ and (e): $\xi = 0.46$), all with 512 samples.

The corresponding inverse filters have been modelled with FIR filters of 1024 samples. As it has been previously commented, the sparseness of the inverse filters is lower than their respective acoustic channels ($\xi = 0.86$, $\xi = 0.30$ and $\xi = 0.38$). Note that the most dispersive acoustic response does not match with the most dispersive of the three inverse channels.

For the experiments, we also assume that the channel paths are perfectly estimated. For the combination schemes, a value of $a^{+} = 4$ has been set to update the mixing parameter [95].

4.5.1 Block-based schemes.

The inverse filter has normally nonuniform energy distribution among its coefficients. For that reason, the convex combination of blocks showed in Fig. 4.1 with different κ values will allow the CBFx-IPNLMS scheme to take advantage of the different coefficient distribution of each block. Thus, those blocks with higher energy will behave as the Fx-IPNLMS algorithm with lower κ value, meanwhile blocks with low energy will follow the Fx-IPNLMS algorithm with a higher κ, to reduce the EMSE of the CFx structure.

The acoustic channel of Fig. 4.5 (a), whose inverse channel shows a high nonuniform energy distribution (see Fig. 4.5 (b)), has been used in this section with an additive Gaussian noise with an SNR= 30 dB. A 32-block filter has been chosen for the CBFx-IPNLMS algorithm. Fig. 4.6 (a) shows the EMSE evolution of the CBFx-IPNLMS with $\kappa_1 = 0.9$, $\kappa_2 = -1$ and $\mu_1 = \mu_2 = 0.1$. The step size of the combined factors $\mu_a = 0.01$ allows a smooth transition for the combination factors. Its EMSE value tends to the lower of the individual filters and shows a better performance at steady state.

It has to be noticed that the performance of the combination scheme with complementary capabilities greatly improves with regard to each single algorithm for a given κ value. Due to the fact that, blocks with low energy slows down the convergence speed of the algorithm, whereas blocks with high energy deteriorates the final EMSE. In steady state, the combination allows to those blocks with high energy tend to the Fx-IPNLMS with $\kappa = -1$ and thus reducing its EMSE.

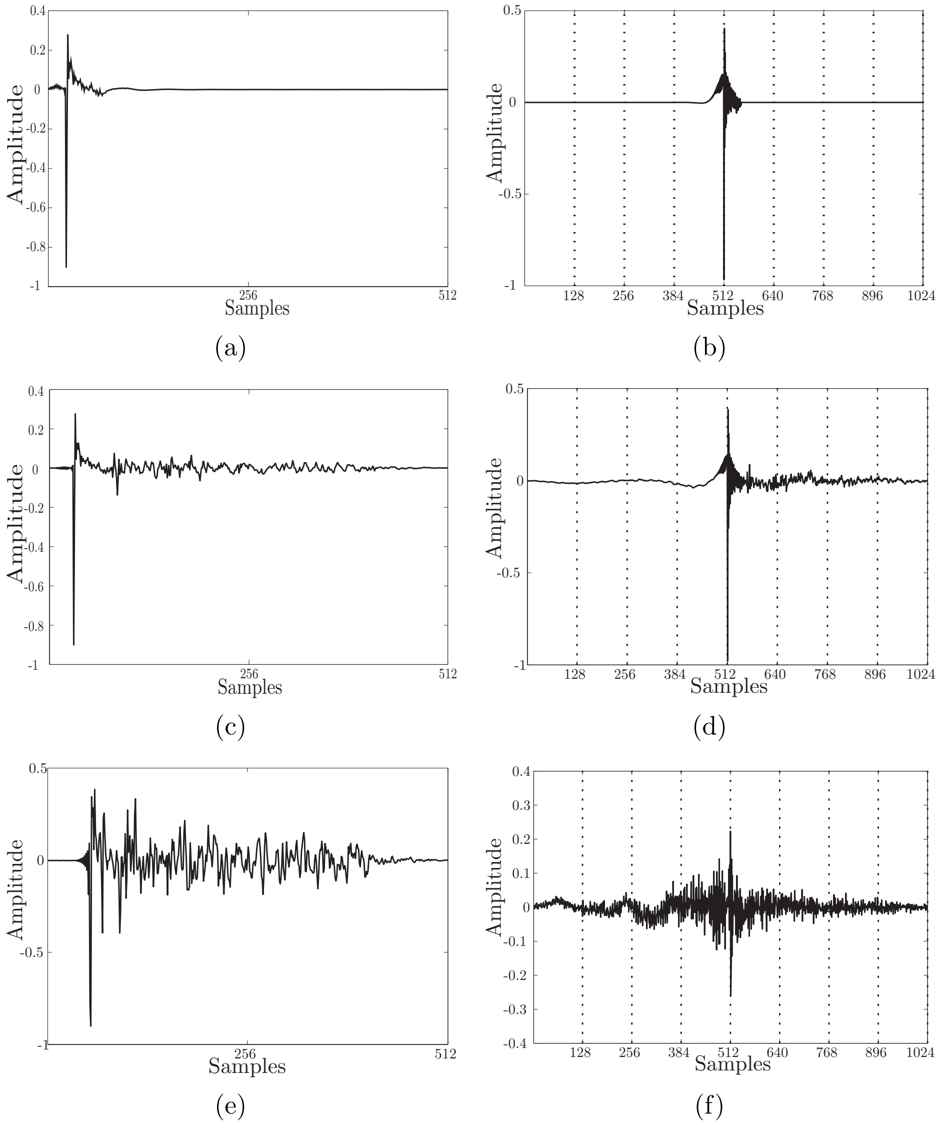
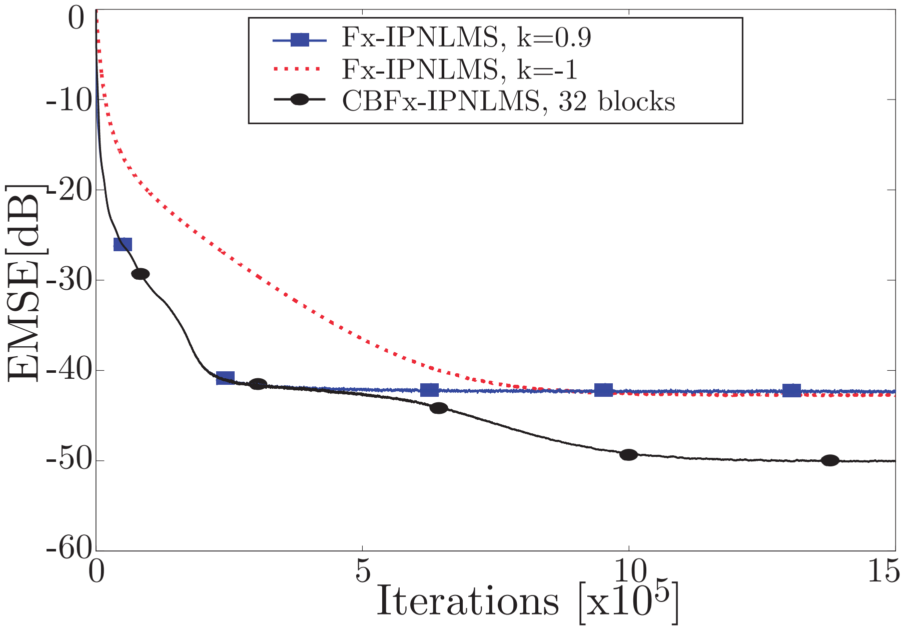
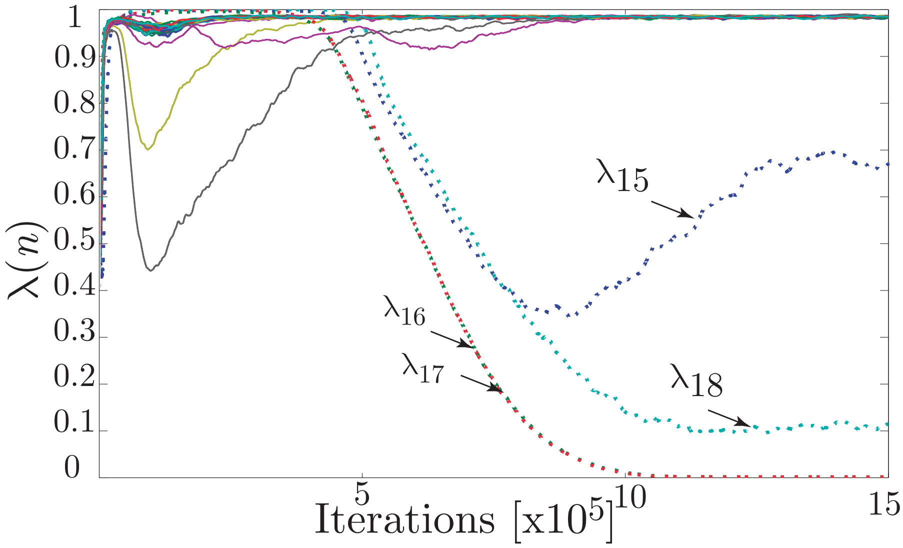


Figure 4.5. Acoustic channels (a, c and e) and their respective inverse filters (b, d and f). These responses present different degrees of sparsity. (a): $\xi = 0.89$, (b):$\xi = 0.86$, (c): $\xi = 0.70$, (d): $\xi = 0.30$, (e): $\xi = 0.46$ and (f): $\xi = 0.38$

Fig. 4.6 (b) shows the combination factors of the 32-block CBFx-IPNMLS algorithm, where a λ_m value close to 1 means a behavior similar to a block with $\kappa = 0.9$ and a value close to 0 behaves like a Fx-IPNLMS filter with $\kappa = -1$. In steady state, the central blocks (dotted line) with high energy exhibits a factor closer to the Fx-IPNLMS with $\kappa = -1$, meanwhile the blocks with lower energy behave mainly as the Fx-IPNLMS with $\kappa = 0.9$. Combination factors of blocks 15 and 18 get intermediate values as they have mixed energy values.



(a)



(b)

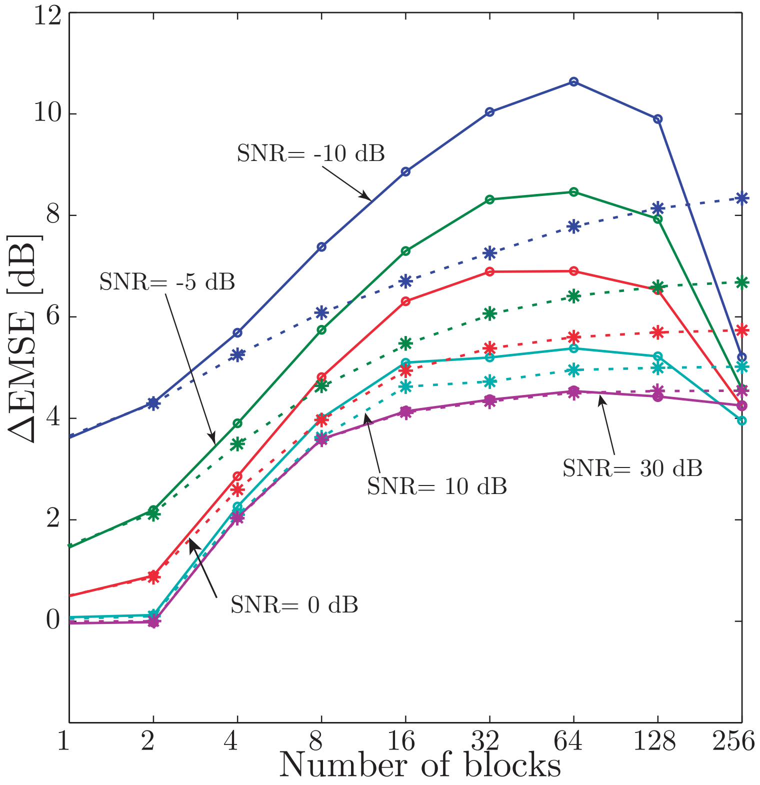
Figure 4.6. (a): EMSE(n) for CFx-IPNLMS and their individual algorithms. (b): $\lambda(n)$ of the CFx-IPNLMS approach.

4.5.2 Steady-state EMSE for unknown SNRs.

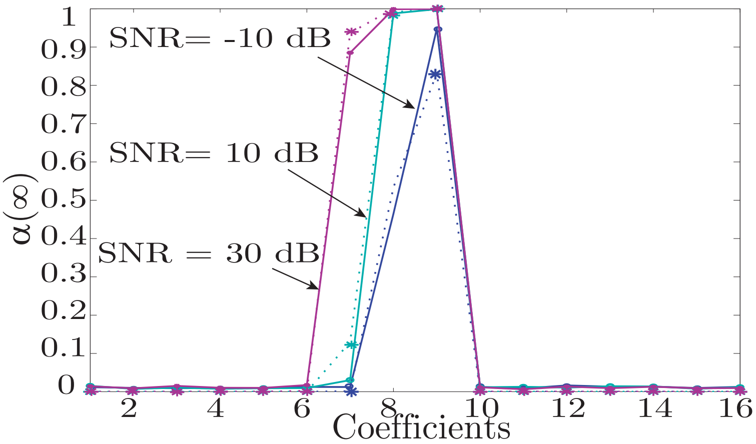
In this subsection we will illustrated how the BBFx algorithm of Fig. 4.2 outperforms the standard Fx algorithm, specially for low SNR. For this purpose, we show a comparison between the simulation results and the theoretical results obtained by means of the analysis in Section 4.2.3. We compare the EMSE with $M = 1, \ldots, L_w/4$ of the block algorithm (EMSE_M) referred to the EMSE of the standard filtered-x algorithm, $\Delta\mathrm{EMSE}(\infty) = \mathrm{EMSE}(\infty) - \mathrm{EMSE}_M(\infty)$. For inverse filters with a high degree of sparseness, the BBFx adaptive filter sketched in Fig. 4.2 allows to improve the steady-state EMSE value when increasing the number of blocks. Fig. 4.7 (a) shows with round markers the estimated $\Delta\mathrm{EMSE}(\infty)$. Meanwhile, the theoretical $\Delta\mathrm{EMSE}(\infty)$ has been represented with asterisk markers and dotted line.

The results are represented for different values of SNR and different number of blocks M, for the impulse response of Fig. 4.5 (a). For higher SNR the results are similar to that of the standard algorithm as the scaling factors are equal or close to 1. For the inverse filter of Fig. 4.5, which has a high number of coefficients close to zero, there will be scaling factors that are zero, as it is shown in Fig. 4.7 (b), leading to a performance improvement using the BBFx. Meanwhile, for lower SNR values the difference is more evident. Also, the block-based algorithm improves the EMSE results when increasing the number of blocks. However, when the block size is too small, as that for $M = 256$ and a block size of $Q = 4$, this improvement is limited to almost 4 dB for the represented SNR range. This is due to the gradient noise of the scaling factors $\alpha_m(n)$, for $m = 1, \ldots, M$ [101]. A good performance is achieved with $M = 16$, which corresponds to a block-length of 64 samples. For this number of blocks, Fig. 4.7 (b) shows the 16 steady-state scaling factors for different SNR values. For low SNR values, the factors of the adaptive blocks with low energy bias these outputs almost to 0. Meanwhile, the blocks with energy, the central ones for room equalization, exhibit a lower bias. For higher SNR values this central scaling factors get a higher value.

The theoretical values fall close to the experimental ones, but differences are more significant when the number of blocks increases. This can be due to the assumptions considered in the theoretical analysis and also to the gradient noise of the scaling factors, as can be appreciate in Fig. 4.7 (b).



(a)



(b)

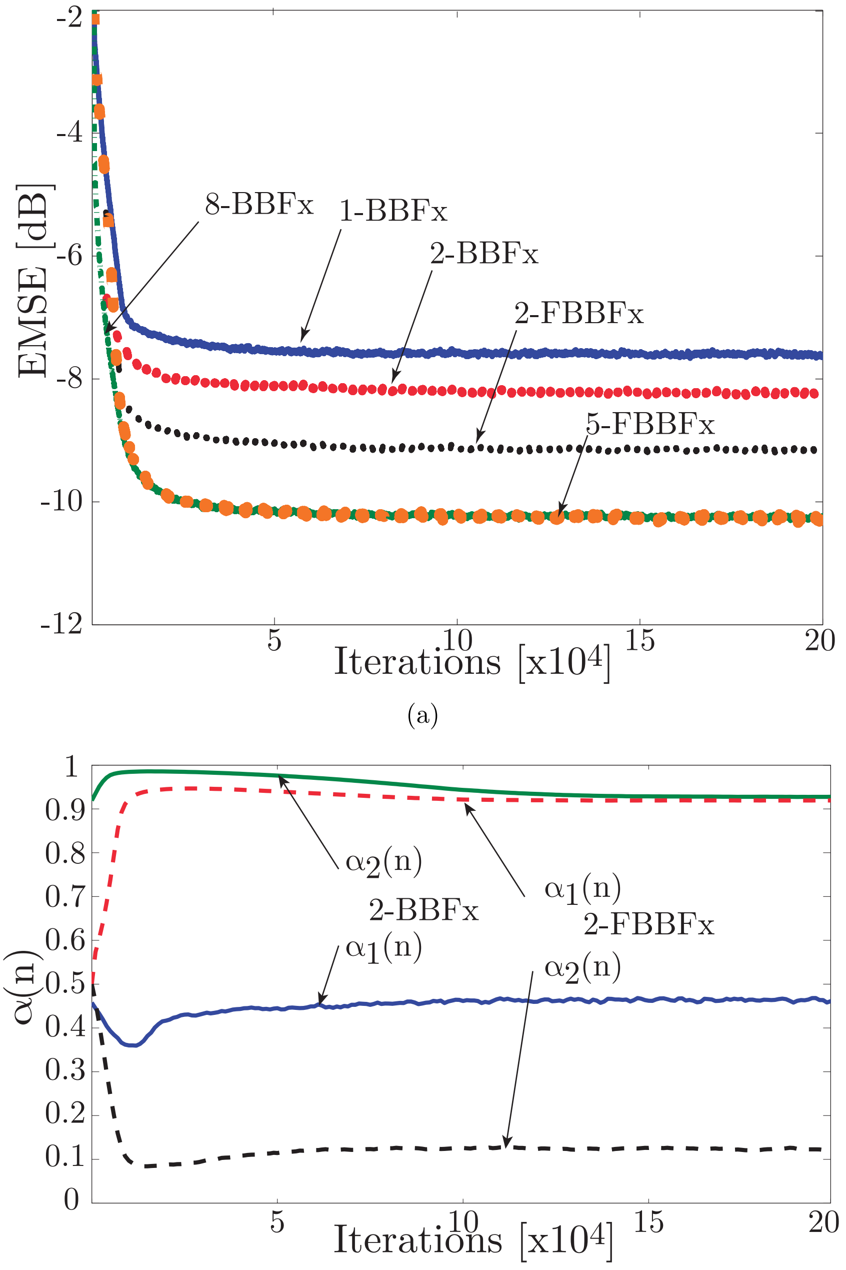
Figure 4.7. (a): $\Delta\text{EMSE}(\infty) = \text{EMSE}(\infty) - \text{EMSE}_M(\infty)$ for different number of blocks (x-axis) and SNR (curve) values. (b) Scaling factors at steady state, $\alpha_m(\infty)$, for $M = 16$ and different SNR values.

4.5.3 Fitted block-based biased filters.

As it has been showed in the previous experiments, the CBFx and the BBFx adaptive filters improve their behavior in terms of EMSE as the number of blocks increase, but with an increase in the computational load as can be seen in Table 4.1. In an AE context, the use of a fitted-block scheme, as detailed in Section 4.3.1, improves the performance of the BBFx filter with the same number of blocks. For this experiment, the fitted-block structure will consider a central block, as it will take the central coefficients with higher energy in the same block.

Fig. 4.8 shows a FBBFx-IPNLMS algorithm with a central block of 512 coefficients and another block with the 512 remaining taps (2-FBBFx), which is compared the BBFx-IPNLMS filter with $M = 1, 2$ and 8 blocks, referred in Fig. 4.8 as M-BBFx. For this experiment SNR= 0 dB has been used, although higher SNR values obtain a similar behavior provided that the scaling factors are not 1. However, lower SNR lead to more significant differences. $\mu = 0.3$ and $\kappa = -0.5$ have been chosen. Results in Fig. 4.8 (a) show that the 2-FBBFx-IPNLMS algorithm (black dotted line) outperforms the 2-BBFx-IPNLMS one (red dotted line), although it does not achieve the stationary EMSE of the 8-BBFx-IPNLMS algorithm (green solid line). In order to obtain this performance a 5-FBBFx-IPNLMS is enough (yellow dotted line). Although, the convergence speed of the FBBFx schemes is slower than those of the BBFx schemes.

Fig. 4.8 (b) shows the biased parameters of the $2-$block approaches. On the one hand, the $2-$BBFx-IPNLMS algorithm presents coefficients with energy that are distributed among the two blocks (in solid lines), due to that $\alpha_2(n) \simeq 0.9$ and $\alpha_1(n) \simeq 0.45$ have both high values for SNR= 0dB. In contrast, for the $2-$FBBFx-IPNLMS algorithm (in dashed lines), $\alpha_1(n) \simeq 0.9$ for the central coefficients with higher energy, but for the lateral coefficients with low energy $\alpha_2(n) = 0.1$, which allows to reduce the EMSE of the $2-$FBBFx-IPNLMS algorithm. The computational cost for both algorithms is the same, but if we want to achieve a similar EMSE value, a higher block number is needed for the BBFx-IPNLMS scheme, which implies a higher computational cost.



(a)

(b)

Figure 4.8. (a): EMSE evolution of the 1, 2 and 8-BBFx-IPNLMS algorithm and the 2 and 5-FBBFx-IPNLMS algorithm. (b): Scaling factors $\alpha_i(n)$ for the two-block schemes, 2-BBFx (in solid lines) and 2-FBBFx (in dashed lines).

4.5.4 EMSE behavior for different central-block lengths and sparsity degree.

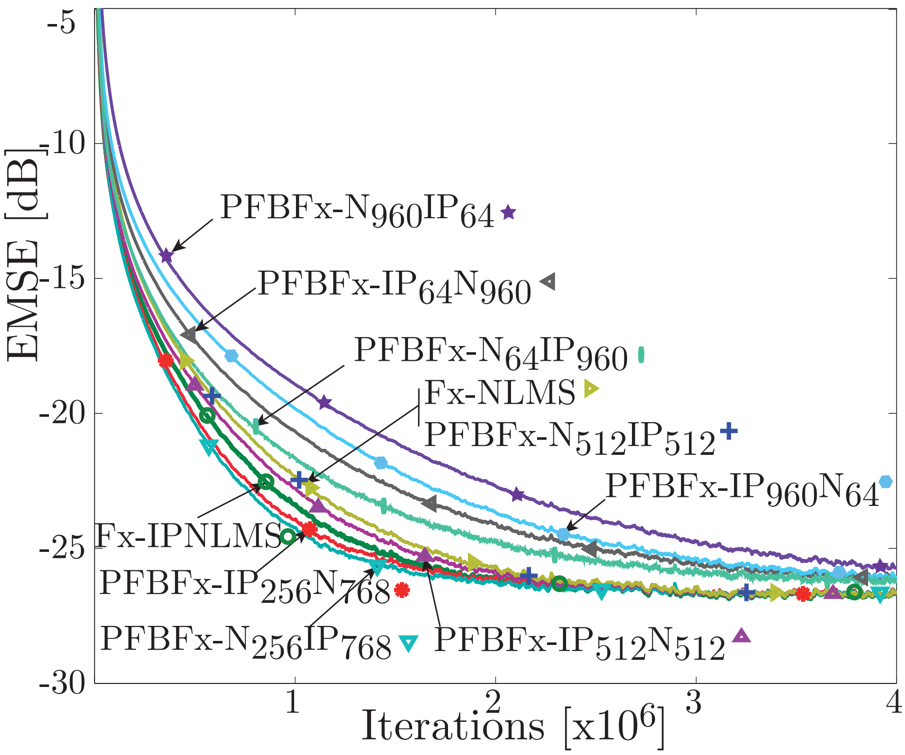
For this experiment, in order to study the influence of the central-block length, the partitioned fitted-block BFx-IPNLMS algorithm (PFBFx-IPNLMS) is used with different length. Thus, the notation PFBFx-IP$_{256}$N$_{768}$ refers to a central block with $\kappa = -0.5$ and $L_c = 256$ samples, and a lateral block with $\kappa = -1$ and $L_l = 768$ samples. To assure the constraint on $\sum_{l=1}^{L_w} g_l(n) \approx 1$, the sum of $\sum_{l_c=1}^{L_c} g_{l_c}(n)$ and $\sum_{l_l=1}^{L_l} g_{l_l}(n)$ are multiplied by 0.5.

For the experiments, $L_c = L_w/2 = 512$, $L_c = L_w/4 = 256$, $L_c = L_w/16 = 64$ and $L_c = L_w - L_w/16 = 960$ samples have been used as block sizes. Also the Fx-NLMS and Fx-IPNLMS (with $\kappa = -0.5$) are computed as reference. The other parameters are set to $\mu_a = 0.01$ and SNR= 15dB.

Fig. 4.9 corresponds to the EMSE evolution of the adaptive inverse filter of Fig. 4.5 (f) with a sparseness degree of $\xi = 0.38$. The fastest convergence is achieved with the PFBFx-IP$_{256}$N$_{768}$ and PFBFx-N$_{256}$IP$_{768}$ schemes, which corresponds to the normalized factors $\frac{\|\mathbf{w}_c\|_1}{\|\mathbf{w}\|_1} \approx \frac{\|\mathbf{w}_l\|_1}{\|\mathbf{w}\|_1} \approx 0.5$. For this block-size, $\xi_c = 0.23$ and $\xi_l = 0.27$ show a lower sparseness value for both blocks. Moreover, a narrow block size of $L_w/16 = 64$ samples for L_c or L_l results in a slower convergence. This can be due to the dispersive character of the block filters and their normalized factors close to 0 or 1.

The steady state of these algorithms is similar, but a lower value is obtained for a wider central block of $L_w/2$ and $L_w/4$, and also with the Fx-NLMS and Fx-IPNLMS algorithms.

Fig. 4.10 represents the EMSE evolution of the adaptive filter of Fig. 4.5 (b) with a sparseness degree of $\xi = 0.86$. For this inverse channel with a lot of coefficients close to zero, the normalized factors near 0.5 are obtained with $Lc = 16 = L_w/64$. For this scheme, $\xi_c = 0.24$ and $\xi_l = 0.75$, which corresponds with the results shown in Fig. 4.10 where the PFBFx-N$_{16}$IP$_{1008}$ algorithm exhibits a fast convergence. Although, the steady state of this schemes with a narrow central block is higher than the others. Furthermore, the lower results in steady state are obtained with a central block of $L_w/2$. Meanwhile, the $L_w/4$-samples, Fx-NLMS and Fx-IPNLMS schemes achieved an intermediate value. Also, the $L_c = L_w/2$ and $L_c = L_w/4$ algorithms exhibit a fast convergence, especially for the PFBFx-IP$_{256}$N$_{768}$



(a)

Figure 4.9. EMSE value for the partitioned fitted-block algorithm with different block sizes and κ values for the adaptive filter of Fig. 4.5 (f). PFBFx-IP$_i$N$_j$ refers to the partitioned fitted-block algorithm with a central block with $\kappa = -0.5$ and $L_c = i$ samples, and a lateral block with $\kappa = -1$ and $L_l = j$ samples. Also the Fx-NLMS and Fx-IPNLMS are shown as reference.

algorithm, because the filters coefficients are initialized to zero and the $L_l = 768$ samples of the lateral blocks are close to zero.

To summarize, the PFBFx-IPNLMS filter can exhibit a faster convergence speed than the Fx-IPNLMS, when the normalized factor $\dfrac{\|\mathbf{w}_c\|_1}{\|\mathbf{w}\|_1}$ and $\dfrac{\|\mathbf{w}_l\|_1}{\|\mathbf{w}\|_1}$ are close to 0.5. Meanwhile, to reduce the EMSE in steady state,

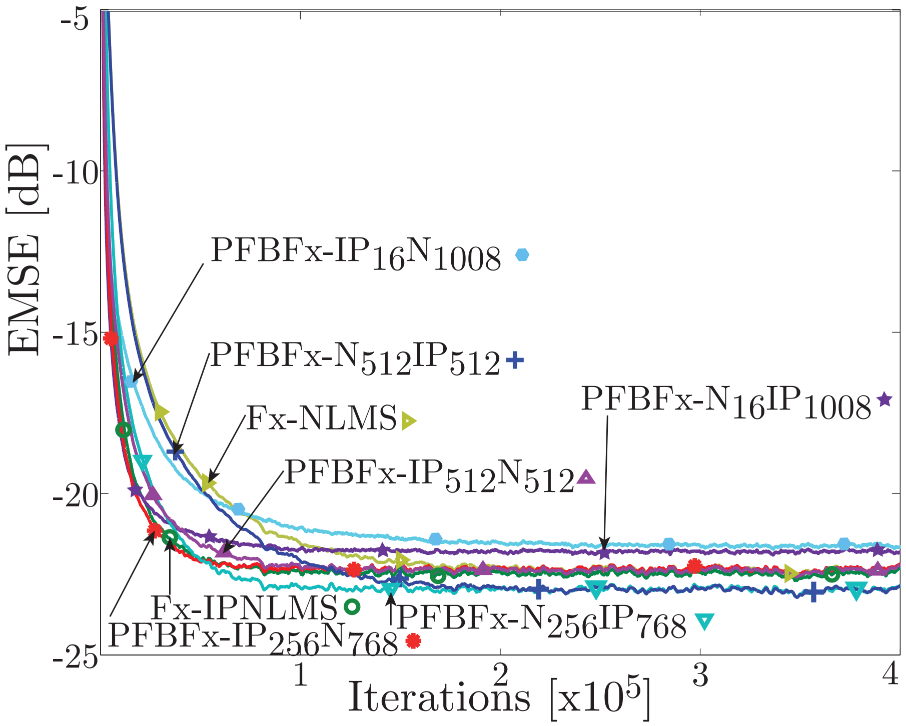


Figure 4.10. EMSE value for the partitioned fitted-block algorithm with different block sizes and κ values for the adaptive filter of Fig. 4.5 (b).

a wide center block of $L_w/4$ or $Lw/2$ with a Fx-NLMS algorithm and a Fx-IPNLMS algorithm for the lateral block will be used.

4.5.5 Convex combination of partitioned fitted-block schemes

From the results of the previous experiment, as the energy distribution of the filter coefficients is located at the center of the inverse filter, a wide center block of $L_w/2$ with a Fx-NLMS algorithm and a Fx-IPNLMS algorithm for the lateral block will allow to reduce the EMSE in steady state, although it exhibits a slow convergence. On the other hand, a narrower central block of $L_w/4$ samples with a Fx-IPNLMS algorithm has shown a good convergence speed. For that reason, a tradeoff between the final EMSE value and a fast convergence speed is to use a convex combination

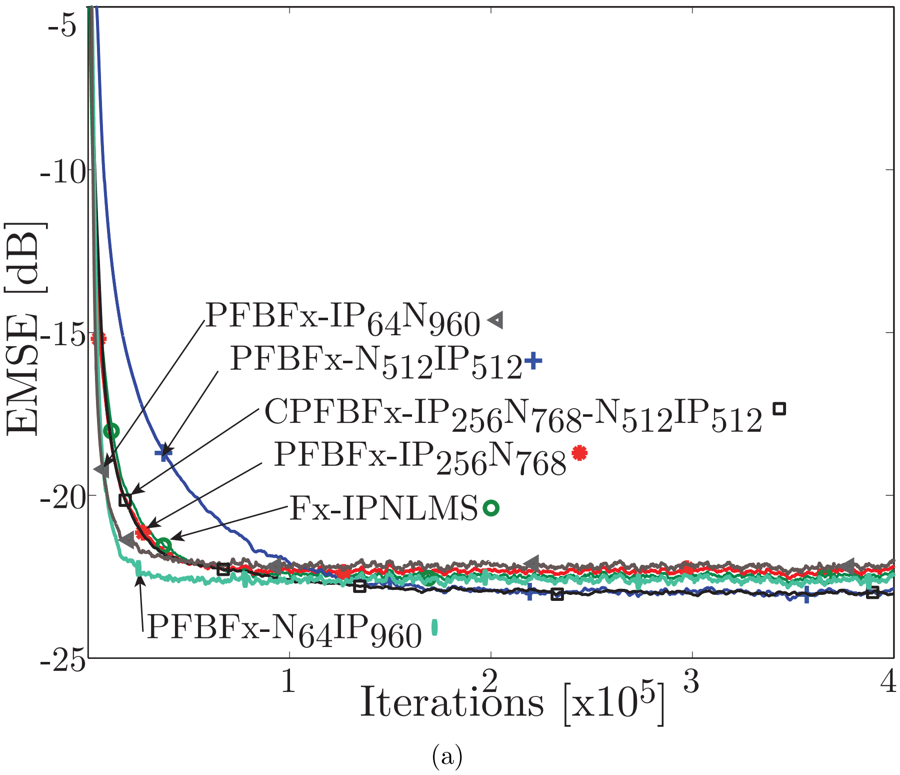
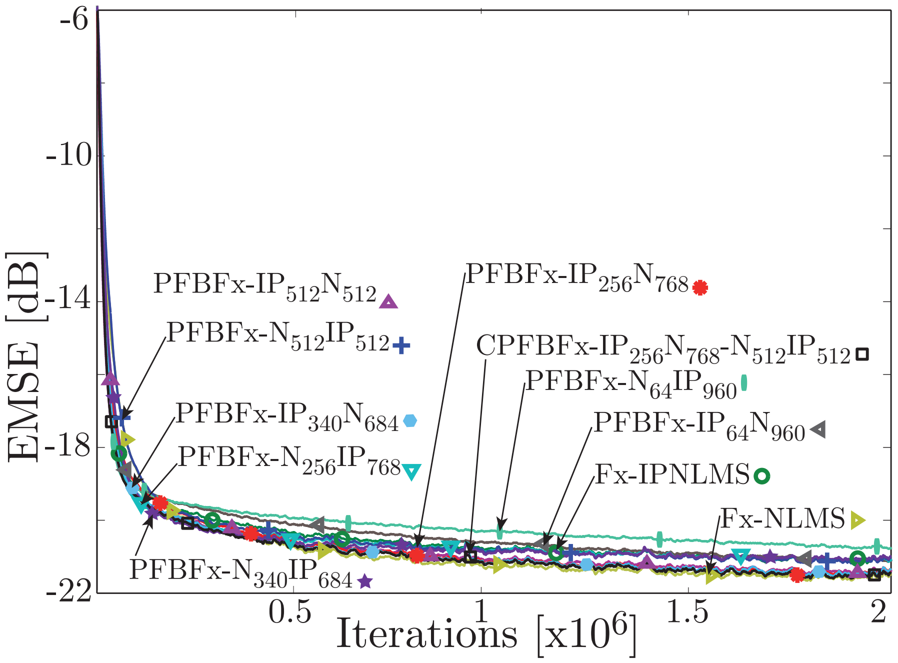


Figure 4.11. EMSE value for the combined partitioned fitted-block algorithm for the adaptive filter of Fig. 4.5 (b).

of a PFBFx-N$_{512}$IP$_{512}$ and a PFBFx-IP$_{256}$N$_{768}$ algorithms.

Fig. 4.11 illustrates the EMSE evolution of the adaptive filter of Fig. 4.5 (a). This figure is the same as Fig. 4.10, but some schemes have been removed for clarity. The CPFBFx-IP$_{256}$N$_{768}$-N$_{512}$IP$_{512}$ exhibits a fast convergence speed and the lowest steady state error. Although it is not the best solution for the convergence speed, that it is best achieved with a central block of $L_c = L_w/16$ samples.

A different channel response has been used in Fig. 4.12, which shows the EMSE evolution of the adaptive filter of Fig. 4.5 (d). Also in this case, the combination scheme exhibits a fast convergence speed and a lower steady state.



(b)

Figure 4.12. EMSE value for the combined partitioned fitted-block algorithm for the adaptive filter of Fig. 4.5 (d).

4.5.6 Convex Combination of fitted-block and BFx-NLMS Algorithms

In order to get a better steady-state performance without reducing the convergence speed, a good solution is to combine two algorithms with different μ values. The filter combination can be done between any kind of algorithms. Thus, the fitted-block scheme can be combined with an algorithm with both lower μ value and lower computational cost. For this purpose we consider the convex combination (red dotted line and round markers) of a 1-block biased Fx-NLMS algorithm with $\mu = 0.01$ (blue solid line) and a FBBFx-IPNLMS with $\mu_1 = \mu_2 = 0.3$, $\kappa_1 = \kappa_2 = -0.5$ and a central-block of 512 coefficients (green dotted line and square markers). Fig. 4.13 illustrates the EMSE values obtained for the channel response of Fig. 4.5 (c). The combination achieves both good convergence speed and low steady-state error, even outperforming the individual algorithms in some iterations. The biased version of the Fx-NLMS algorithm as been used to prevent low SNR situations (for this experiment, SNR=10dB). We can conclude that if the computational burden is not a constraint, a better performance at steady state can be achieved using the BBFx-NLMS algorithm with more than 1 block.

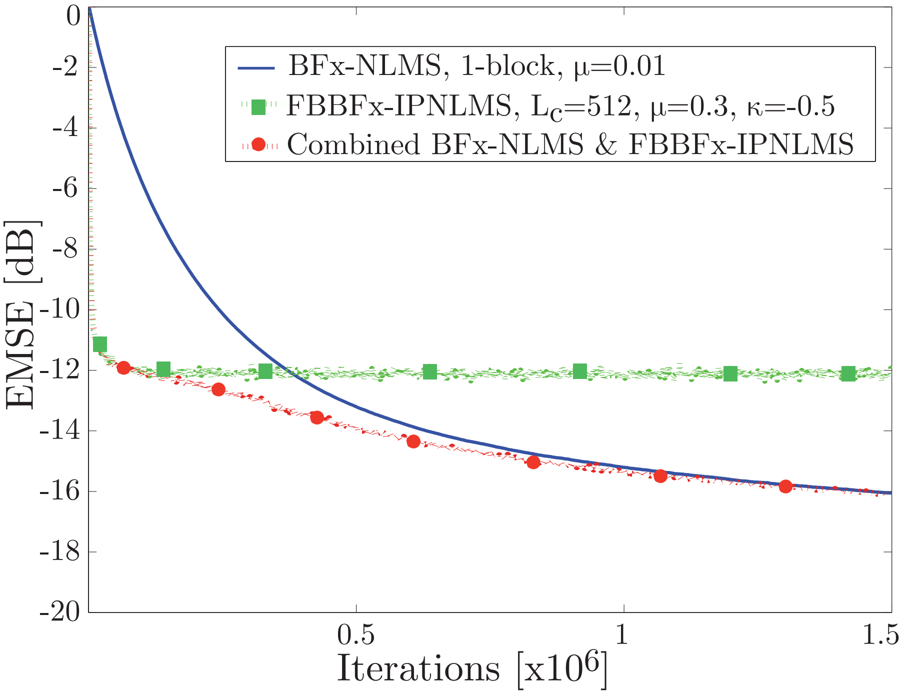


Figure 4.13. Combination of a fitted-block algorithm with a Fx-BNLMS algorithm with different μ values.

4.6 Conclusions

In this chapter different block-based schemes are proposed for room equalization applications based on the non-uniform energy distribution of the inverse adaptive filter.

On the one hand, the CFx-IPNLMS algorithm allows to combine the different blocks independently, getting a better behavior in most cases than the previous combination, but with an increase in the computational cost.

On the other hand, the BBFx-IPNLMS presents an improvement with respect to the Fx-IPNLMS algorithm, particularly for low SNR. Meanwhile, for high SNR values (with the scaling factors close to 1), the results are similar, but with an increase in the computational burden due to the update of the scaling factors, that will not be required for high SNRs.

Based on the inverse filter response, the fitted block-based schemes are developed to take advantage of this central energy distribution for the tap filters. The new approaches are derived from the previous combinations by using the fitted-block based approach showing a good performance and even better response for the same computational cost. Moreover, the partitioned fitted-block scheme allows to obtain good results, specially when the characteristics of the inverse channel are *a priori* known.

Furthermore, the computational cost of these block-based schemes has been presented to show the computational burden increase with the number of blocks for filtered-x structures. Otherwise, experimental results show that filters with block of sizes too small can impair the filter performance. Thus, a good tradeoff between computational cost and final performance is required.

5 Nonlinear adaptive equalization

Nonlinear adaptive equalization 5

This chapter introduces a new scheme for nonlinear equalization of loudspeaker and room responses which is based on Volterra filters. the aim is twofold, to identify the nonlinear system and to invert the existent linearities and nonlinearities. With this purpose, a nonlinear filtered-x (NFx) structure is employed that uses the virtual channel concept in order to obtain a good performance of the proposed scheme. For that reason, a previous identification of the nonlinear acoustic system is required. On the other hand, a simplified strategy is proposed when the nonlinear system can be considered *weakly* and only the linearities of the system are identified. Different results are presented to support these schemes.

5.1 Nonlinear system identification problem

We briefly describe the nonlinear system identification problem, which is depicted in Fig. 5.1. This scheme is also required in the nonlinear equalization system, similarly as it was addressed in Section 2.3 for linear adaptive room equalization. Moreover, this problem is inherent to other audio applications, such as nonlinear AEC and ANC, that also require to solve the identification problem.

Assuming that the nonlinear system can be modeled by a Qth-order Volterra kernel and filters of finite memory M_q. From Fig. 5.1 the output signal $d(n)$ can be expressed as

$$\begin{aligned} d(n) = \mathcal{H}[x(n)] = & \sum_{i_1=0}^{M_1-1} L(i_1)x(n-i_1) \\ + \sum_{q=2}^{Q} & \left[\sum_{i_1=0}^{M_q-1} \cdots \sum_{i_q=0}^{M_q-1} N_q(i_1,\ldots,i_q)x(n-i_1)\cdot\ldots\cdot x(n-i_q) \right], \end{aligned} \quad (5.1)$$

being $\mathcal{H}$ the nonlinear system modeled by Q parallel block filters and M_q is the memory or number of coefficients in the qth dimension of the Volterra kernel. Moreover, $L(i_1)$ is the i_1-coefficient of the first kernel and $N_q(i_1,\ldots,i_q)$ refers to the $(i_1,\ldots,i_q)$-coefficient of the qth-kernel, where a symmetric form is considered for the Volterra kernels [56]. This symmetry can be written for the second-order kernel as $N_2(i_1,i_2) = N_2(i_2,i_1)$, which can be extrapolated to any order.

For some applications, an identification of the nonlinear electroacoustic path that involves the loudspeaker-enclosure-microphone (LEM) channel is required. Fig. 5.1 shows the block diagram of an adaptive nonlinear identification problem using a second-order Volterra filter. The first kernel $\mathbf{w}_l(n)$ corresponds to the linear filter (a FIR filter with M_1 coefficients) and the second kernel $\mathbf{w}_q(n)$ to the quadratic filter (a second-order filter matrix with $M_2 \times M_2$ taps), which are are given by:

$\mathbf{w}_l^T(n) = [w_l(1;n),\ldots,w_l(M_1;n)]$ and

$$\mathbf{w}_q(n) = \begin{bmatrix} w_q(1,1;n) & w_q(1,2;n) & \cdots & w_q(1,M_2;n) \\ w_q(2,1;n) & w_q(2,2;n) & \cdots & w_q(2,M_2;n) \\ \vdots & \vdots & \ddots & \vdots \\ w_q(M_2,1;n) & w_q(M_2,2;n) & \cdots & w_q(M_2,M_2;n) \end{bmatrix} \quad (5.2)$$

The output of the adaptive filter is obtained as,

$$y(n) = \sum_{i_1=0}^{M_1-1} \mathbf{w}_l(i_i;n)x(n-i_1) + \sum_{i_1=0}^{M_2-1}\sum_{i_2=0}^{M_2-1} \mathbf{w}_q(i_1,i_2;n)x(n-i_1)x(n-i_2). \quad (5.3)$$

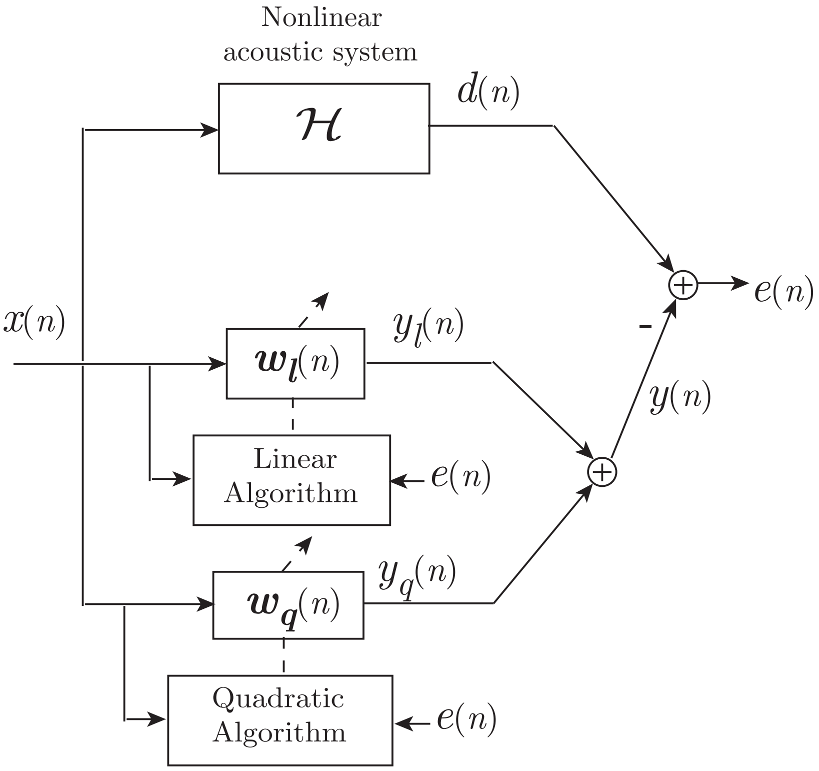


Figure 5.1. Adaptive nonlinear identification system using a second-order Volterra filter.

The error signal $e(n)$ is computed as in (2.5) and is rewritten here for clarity

$$e(n) = d(n) - y(n). \tag{5.4}$$

However, the adaptive filters $\mathbf{w}_l(n)$ and $\mathbf{w}_q(n)$ can be linearly updated as in (2.9) for the LMS-type algorithm,

$$w_l(i_1; n) = w_l(i_1; n-1) + \mu_l(i_1; n)e(n)x(n - i_1), \text{ for } i_1 = 1, \ldots, M_1 \tag{5.5}$$

$$\begin{aligned} w_q(i_1, i_2; n) = w_q(i_1, i_2; n-1) + \mu_q(i_1, i_2; n)e(n)x(n - i_1)x(n - i_2), \\ \text{for } i_1 = 1, \ldots, M_2 \text{ and } i_2 = 1, \ldots, M_2, \end{aligned} \tag{5.6}$$

being $\mu_l(i_1; n)$ and $\mu_q(i_1, i_2; n)$ the step-size factors that are computed depending on the LMS-type algorithm.

Due to the symmetry of the kernel, another usual form of representing the quadratic kernel is using the abbreviated vector form with $\frac{M_2(M_2+1)}{2}$ elements,

$$\mathbf{w}_q(n) = \mathbf{w}_q(n-1) + \boldsymbol{\mu}_q(n)e(n)\mathbf{x}_q(n), \tag{5.7}$$

where $\boldsymbol{\mu}_q(n)$ is a diagonal matrix containing the $\frac{M_2(M_2+1)}{2}$ step-size factors. $\mathbf{w}_q(n)$ corresponds to the elements of the upper part of matrix (5.2), including the main diagonal, rewritten as $\mathbf{w}_q^T(n) = \Big[w_q(1,1;n), w_q(1,2;n), \ldots, w_q(1,M_2;n), w_q(2,2;n), \ldots, w_q(M_2,M_2;n)\Big]$. Moreover, $\mathbf{x}_q^T(n) = \Big[x(n)x(n), x(n)x(n-1), \ldots, x(n)x(n-M_2), x(n-1)x(n-1), \ldots, x(n-M_2)x(n-M_2)\Big]$ represents the multiplication of the elements of the input vectors.

5.2 Nonlinear equalization problem

Similarly to linear room equalization, adaptive nonlinear room compensation requires filtered-x ad hoc treatment to avoid instability in the filter coefficient update, where the input signal $x(n)$ is filtered through the electroacoustic path. Thus, the filtering of $x(n)$ involves the use of a nonlinear filtered-x scheme as it was introduced in [75]. This structure has been previously used in ANC (using the so-called virtual secondary path) by using either adaptive Volterra filters [62][102] or FLANN filters [62][103][54]. Meanwhile in [104] and [62], a nonlinear system with a memoryless block in the secondary path is studied.

In [62], a general function expansion is used for both FLANN and Volterra filters, where the nonlinear secondary paths are modeled as nonlinear memoryless systems. However, to the best of our knowledge, none of the previous works have addressed the problem of nonlinear distortion with memory in room equalization systems.

Fig. 5.2 illustrates the nonlinear AE problem, where $\mathcal{L}$ represents a linear filter, and the nonlinear function with memory is denoted by the symbol $\mathcal{N}$. This nonlinear acoustic system can be modeled with eq. (5.1). The virtual channel has been denoted as $\widehat{\mathcal{H}}$ and is obtained in a similar way to the linear equalization problem presented in Section 2.3.2. Moreover, the coefficients of the adaptive filters are updated using this virtual channel for the filtered-x structure.

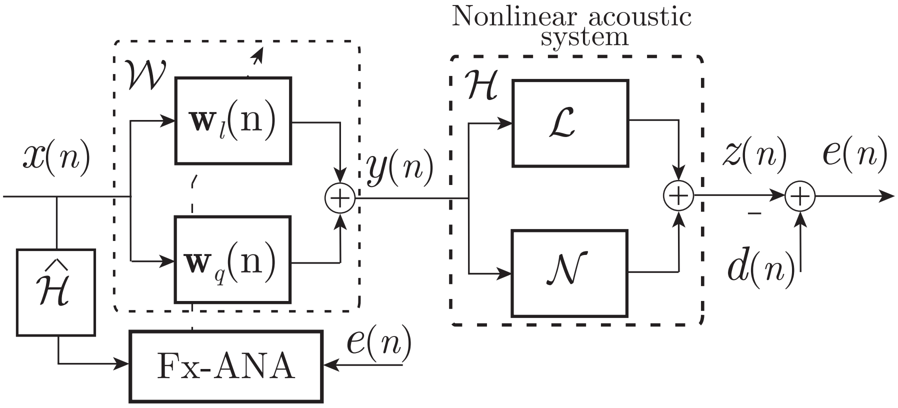


Figure 5.2. Adaptive equalization block diagram using a filtered-x adaptive nonlinear algorithm (Fx-ANA).

5.2.1 Nonlinear loudspeaker and room equalization

The block diagram of Fig. 5.3 shows an acoustic equalization system, where the inversion is applied to the specific problem of the acoustic equalization transmission chain of Fig. 2.7, which includes the LEM path.

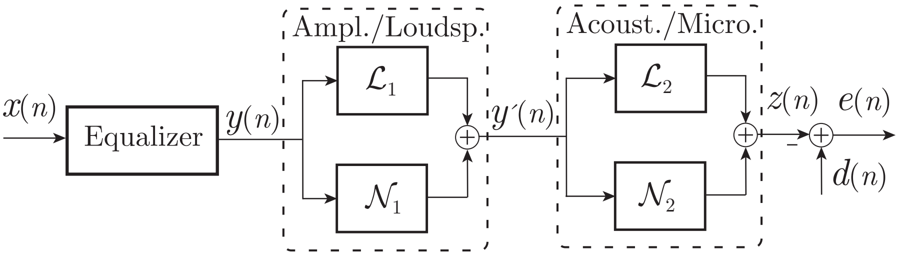


Figure 5.3. Nonlinear model of an acoustic equalization system with two nonlinear filters tandemly connected.

As commented before, the main sources of nonlinearities occur in the first filter, that corresponds to the D/A converter, the loudspeaker and its amplifier, due to the high input signal levels and the loudspeaker physical properties. While the amplifier can be modeled as a memoryless system, the loudspeaker behavior is more properly described as nonlinear distortion with memory [105]. In contrast, the propagation path between loudspeaker and microphone can be considered as a linear filter and it is reasonable to expect a linear behavior for the microphone, too. For that reason, a reasonable representation of the loudspeaker and room adaptive equalization problem can be seen in Fig. 5.4.

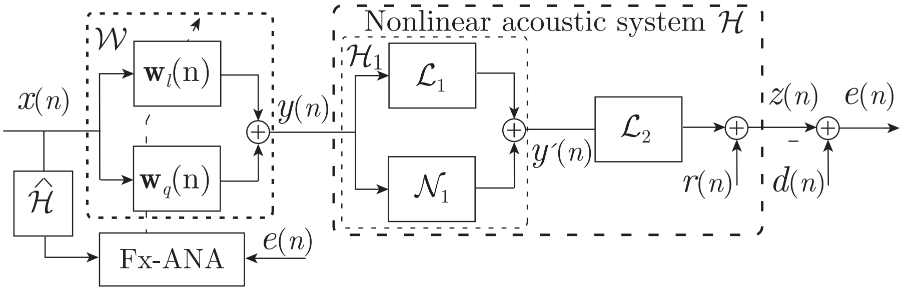


Figure 5.4. Adaptive equalization block diagram of Fig. 5.3.

The output signal $z(n)$ of the nonlinear system $\mathcal{H}$ has been modeled using a second-order Volterra filter as

$$z(n) = \mathcal{L}_2 \left\{ \mathcal{H}_1[y(n)] \right\} = \mathcal{L}_2 \Bigg\{ \sum_{i_1=0}^{M_1-1} L_1(i_1) y(n-i_1) + \sum_{i_1=0}^{M_2-1} \sum_{i_2=0}^{M_2-1} N_1(i_1,i_2) y(n-i_1) y(n-i_2) \Bigg\}, \tag{5.8}$$

being $\mathcal{H}_1$ the nonlinear system modeled with the first nonlinear filter and M_q, for $q = 1, 2$, is the memory or number of coefficients in the qth-Volterra kernel. Moreover, $L_1(i_1)$ is the i_1-coefficient of the first kernel and $N_1(i_1, i_2)$ refers to the (i_1, i_2)-coefficient of the 2th-kernel, whith a symmetric form [56].

To implement the adaptive compensation equalization filter, a second-order Volterra filter $\mathcal{W}$ has been used in Fig. 5.4 to remove nonlinearities up to the 2nd-order. The relationship between input and output of the adaptive filter is given by

$$y(n) = \mathcal{W}[x(n)] = \sum_{i_1=0}^{N_1-1} w_l(i_1; n) x(n-i_1) + \sum_{i_1=0}^{N_2-1} \sum_{i_2=0}^{N_2-1} w_q(i_1, i_2; n) x(n-i_1) x(n-i_2), \tag{5.9}$$

where N_p is the number of coefficients of the pth-Volterra kernel ($p = 1, 2$). Moreover, $w_l(i_1; n)$ and $w_q(i_1, i_2; n)$ are the specific coefficient of the linear and quadratic kernel, respectively, at time n.

The error signal $e(n)$ is computed as the difference between the signal $z(n)$ measured at the microphone (5.8) and the desired signal $d(n)$, which corresponds to the input signal with a proper time delay (τ)

$$e(n) = d(n) - z(n) = x(n-\tau) - z(n). \tag{5.10}$$

The filter coefficients are updated as in (2.4), using the LMS algorithm, that leads to

$$\begin{aligned} w_l(i_1;n) &= w_l(i_1;n-1) - \frac{\mu_l}{2}\frac{\partial e^2(n)}{\partial w_l(i_1;n)} \\ &= w_l(i_1;n-1) + \mu_l e(n)\frac{\partial e(n)}{\partial w_l(i_1;n)}, \end{aligned} \tag{5.11}$$

$$\begin{aligned} w_q(i_1,i_2;n) &= w_q(i_1,i_2;n-1) - \frac{\mu_q}{2}\frac{\partial e^2(n)}{\partial w_q(i_1,i_2;n)} \\ &= w_q(i_1,i_2;n-1) + \mu_q e(n)\frac{\partial e(n)}{\partial w_q(i_1,i_2;n)}, \end{aligned} \tag{5.12}$$

where μ_1 and μ_2 are the step size parameters.

If the system takes the structure shown in Fig. 5.4, we can derive

$$\frac{\partial e(n)}{\partial w_l(i_1;n)} = \sum_{m=0}^{M-1} \frac{\partial z(n)}{\partial y(n-m)} \cdot \frac{\partial y(n-m)}{\partial w_l(i_1;n)}, \tag{5.13}$$

$$\frac{\partial e(n)}{\partial w_q(i_1,i_2;n)} = \sum_{m=0}^{M-1} \frac{\partial z(n)}{\partial y(n-m)} \cdot \frac{\partial y(n-m)}{\partial w_q(i_1,i_2;n)}, \tag{5.14}$$

where M is the memory size of the estimated LEM system, which is given by $M = \max(M_1, M_2) + M_{L_2} - 1$, being M_{L_2} the length of the second linear block $\mathcal{L}_2$.

Moreover, when the step sizes are small enough to allow slow variations of the filter coefficients, from (5.9) it can be written

$$\frac{\partial y(n-m)}{\partial w_l(i_1;n)} \approx x(n-m-i_1), \tag{5.15}$$

$$\frac{\partial y(n-m)}{\partial w_q(i_1,i_2;n)} \approx x(n-m-i_1)x(n-m-i_2). \tag{5.16}$$

For simplicity, we are using the concept of virtual path as in [62; 75] to refer to the derivative of the nonlinear system defined in (5.8) with respect to the delayed inputs

$$\widehat{H}(m;n) = \frac{\partial z(n)}{\partial y(n-m)}, \quad m = 0, \dots, M-1. \tag{5.17}$$

Finally, combining (5.13 - 5.17) into (5.11) and (5.12), we obtain the update equations of the nonlinear Fx-LMS algorithm

$$w_l(i_1;n) = w_l(i_1;n-1) + \mu_l e(n) \sum_{m=0}^{M-1} \widehat{H}(m;n) x(n-m-i_1), \tag{5.18}$$

$$\begin{aligned} w_q(i_1,i_2;n) &= w_q(i_1,i_2;n-1) \\ &\quad + \mu_q e(n) \sum_{m=0}^{M-1} \widehat{H}(m;n) x(n-m-i_1) x(n-m-i_2). \end{aligned} \tag{5.19}$$

Notice the similarity between (5.18) and the conventional filtered-x LMS (Fx-LMS) algorithm in (2.21) used for linear applications.

Since the implementation of (5.18) and (5.19) is not straightforward, next section will provide the specific implementation of the virtual channel for this loudspeaker and listening room compensation approach.

5.2.2 Virtual channel description

The virtual channel defined in (5.17) is a time-varying filter of M-length whose coefficients depend on the input signal $y(n)$. From the block diagram depicted in Fig. 5.5, (5.17) can be expressed as

$$\begin{aligned} \widehat{N}(m;n) &= \frac{\partial z(n)}{\partial y(n-m)} = \sum_{l=0}^{M_{L_2}-1} \frac{\partial z(n)}{\partial y'(n-l)} \cdot \frac{\partial y'(n-l)}{\partial y(n-m)} \\ &= \mathbf{L}_2^T \left[\frac{\partial y'(n)}{\partial y(n-m)} \quad \frac{\partial y'(n-1)}{\partial y(n-m)} \quad \frac{\partial y'(n-M_{L_2}+1)}{\partial y(n-m)} \right]^T, \end{aligned} \tag{5.20}$$

where the derivative of the output of the nonlinear system $\mathcal{H}_1$ is filtered through the enclosure/microphone linear block, modeled as a FIR filter defined by vector $\mathbf{L}_2$.

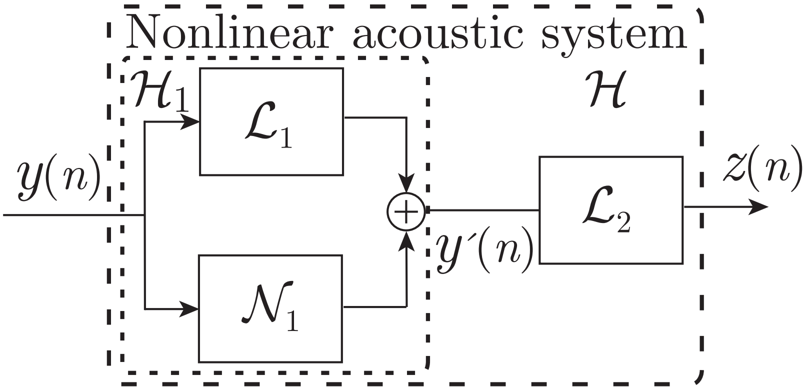


Figure 5.5. Block diagram of the loudspeaker-enclosure-microphone system.

In order to obtain (5.20) as in [75], we should also take into account the linear filter $\mathbf{L}_2$ to relate the input samples $y(n-m)$ to the loudspeaker output signal for different time delays $y^{'}(n-l)$. Thus, the coefficients can be expressed as

$$\frac{\partial y^{'}(n-l)}{\partial y(n-m)} = L_1(m-l) + 2\sum_{i=0}^{M_2-1} N_1(m-l,i)y(n-i-l), \qquad (5.21)$$

when $0 \leq (m-l) < \max(M_1, M_2)$. In other cases, the coefficients are 0. Finally, substituting (5.21) into (5.20), and (5.20) into (5.18) and (5.19), we will obtain the adaptive algorithm for the equalization filter coefficients of the system $\mathcal{W}$.

Note, that if $\mathcal{H}$ is linear ($\mathcal{N}_1 = 0$), only the linear coefficients in (5.21) are considered and the overall virtual channel is the linear convolution of the linear components ($\mathcal{L}_1$ and $\mathcal{L}_2$), which leads in (5.18) to the conventional Fx-LMS algorithm.

5.3 Results

In this section, the robustness and effectiveness of the NFx-LMS algorithm is evaluated in nonlinear equalization scenarios. A normalized adaptation of NLMS type has been used in the NFx-LMS, that leads to the NFx-NLMS algorithm.

To model the nonlinear LEM scheme, the system $\mathcal{H}$, that includes a loudspeaker, has been measured in low-reverberant conditions at a sampling frequency of 8 kHz using a second-order Volterra filter with $M_1 = M_2 = M^{'} = 64$ coefficients. The linear and quadratic kernels are shown in Fig. 5.6. As the second-order kernel presents a symmetric behavior [56], a triangular representation has been used in Fig. 5.6 (b). The impulse response of the acoustic path and the microphone ($\mathcal{L}_2$ system) has been also measured with 8 kHz and 512 samples, within a room with a reverberation time of $T_{60} = 170$ ms, see Fig. 5.7.

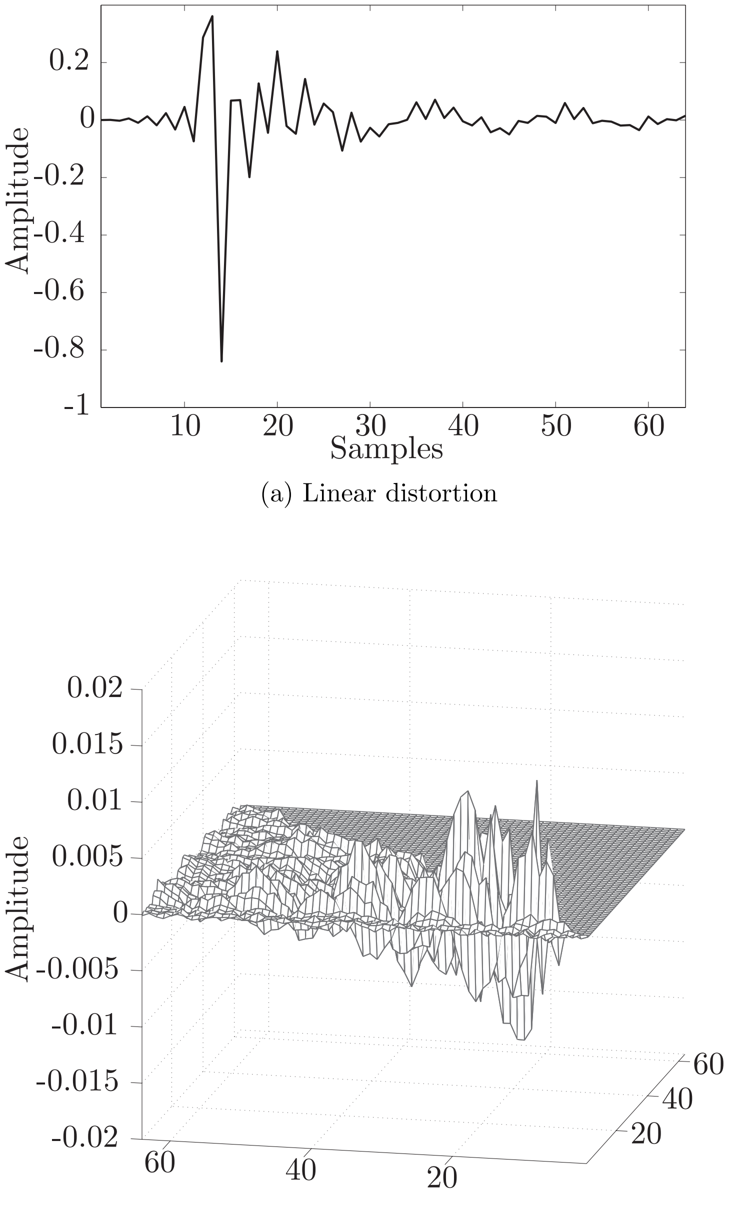
Different Linear-to-NonLinear Ratio (LNLR) setups will be considered. The LNLR level is defined as the ratio between the powers of the linear and nonlinear components ($y_l^{'}(n)$ and $y_n^{'}(n)$, respectively). To modify the LNLR value, a positive constant α has been used to control the LNLR.

$$y^{'}(n) = y_l^{'}(n) + \alpha y_n^{'}(n) = \mathbf{y}^T(n)\mathbf{L}_1 + \alpha \mathbf{y}^T(n)\mathbf{N}_{1,2}\mathbf{y}(n), \tag{5.22}$$

where $\mathbf{L}_1$ and $\mathbf{N}_{1,2}$ represent the linear and quadratic kernels of size $M^{'} \times 1$ and $M^{'} \times M^{'}$, respectively. Moreover, $\mathbf{y}(n)$ is a column vector that contains the last $M^{'}$ samples of $y(n)$.

The algorithm performance is evaluated in terms of convergence speed and steady-state error by inspecting the excess mean-square-error, EMSE $= E\left\{[e(n) - r(n)]^2\right\}$ that has been estimated by averaging over 100 independent runs of the algorithm. The input signal $x(n)$ is a white Gaussian noise with zero mean and unit variance. Moreover, an uncorrelated noise signal $r(n)$, with zero mean and SNR of 40 dB, has been added to the microphone signal. The adaptive filters have been designed to have $N_1 = 1024$ and $N_2 = 64$ coefficients for the first and second kernel, respectively. The delay of the desired signal has been chosen closed to $N_1/2$, specifically $\tau = 549$.

Fig. 5.8 illustrates the EMSE evolution of the NFx-NLMS algorithm (solid line) with $\alpha = 1$ (LNLR= 20 dB). The performance of the algorithm is compared to the linear Fx-NLMS algorithm (dotted line), which implies that the nonlinear component ($\mathcal{N}_1$) is not considered and the virtual channel is only obtained from linear component filters ($\mathcal{L}_1$ and $\mathcal{L}_2$). Also a simplified



(a) Linear distortion

(b) Nonlinear distortion

Figure 5.6. Linear (a) and quadratic (b) kernels of the system $\mathcal{H}$.

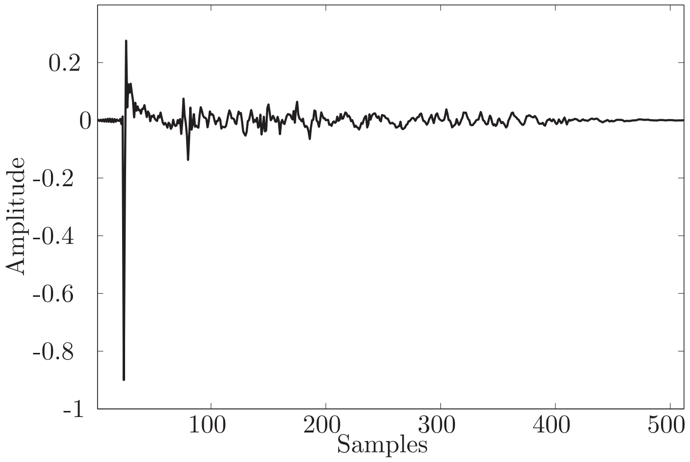


Figure 5.7. Impulse response of the acoustic path including the microphone response.

approach derived from the NFx-NLMS algorithm has been implemented (named as simplified NFx-NLMS, in dashed line and with square markers), which uses a virtual channel which depends only on these linear components to compute the 2-nd-order Volterra kernel. The step sizes are set to $\mu_1 = \mu_2 = 0.1$. Although the three adaptive filters exhibit a similar convergence rate, that is even faster for the Fx-NLMS, only the NFx-NLMS approach has a stable behavior. Thus, a suitable design of the virtual path is essential in this context.

5.3.1 Equalization with a high degree of nonlinearities

To study the behavior of the NFx-NLMS algorithm when the nonlinearities exhibit a high output with respect to the output of the linear filter, $\alpha = 7$ has been used to provide an LNLR of 0 dB. Fig. 5.9 shows the evolution of EMSE, with $\mu_1 = \mu_2 = 0.01$. As it can be observed, the NFx-NLMS filter exhibits a stable behavior reaching a steady-state EMSE of approximately -11 dB. Furthermore, the simplified and linear algorithms are also shown, which do not converge and get unstable as expected. The adaptive filter coefficients obtained for the NFx-NLMS algorithm at steady state are shown for each kernel in Fig. 5.10.

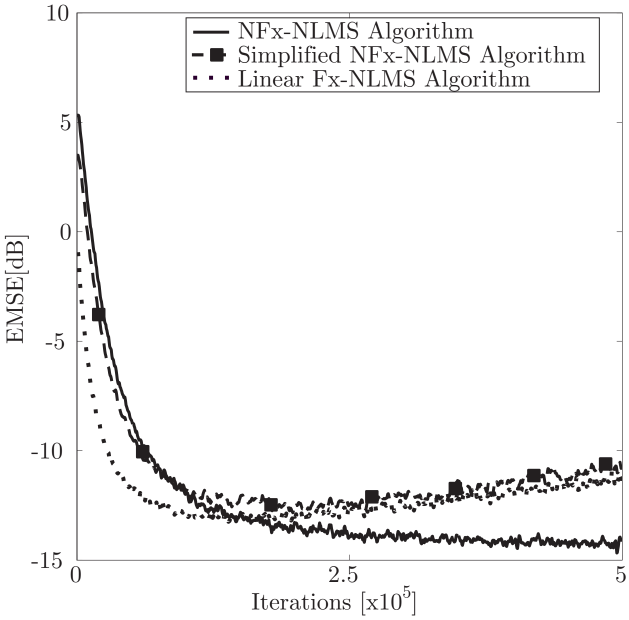


Figure 5.8. EMSE evolution of the NFx-NLMS algorithm (solid line), the simplified NFx-NLMS algorithm (dashed line and square markers) and linear Fx-NLMS algorithm (dotted line).

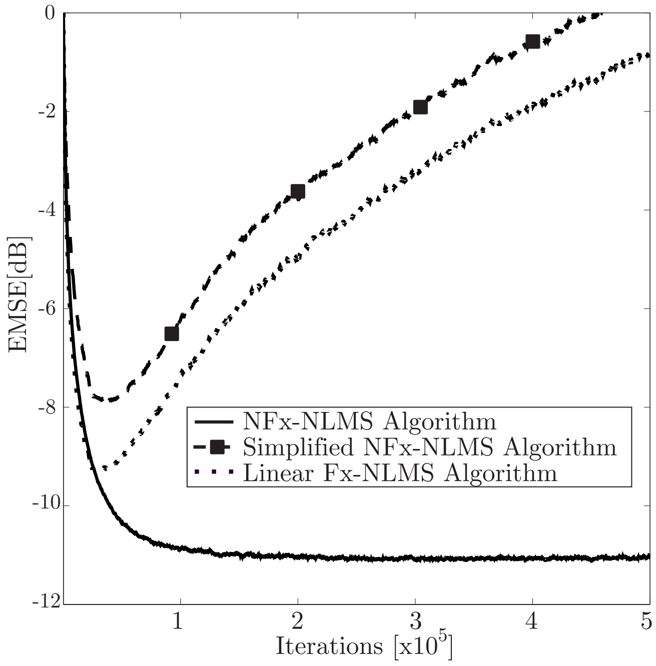


Figure 5.9. EMSE evolution of the NFx-NLMS algorithm and the simplified and linear versions, with LNLR= 0 dB ($\alpha = 7$).

5.3.2 Weakly nonlinear systems

When the electroacoustic path presents a slightly nonlinear behavior, the system is called weakly nonlinear system, [47].

$$|y_l^{'}(n)| \gg |y_n^{'}(n)|, \tag{5.23}$$

where $y_l^{'}(n)$ and $y_n^{'}(n)$ denote the outputs of the linear and nonlinear components of the nonlinear system.

In that case, the virtual channel in (5.21) of the nonlinear filtered-x structure can be approximated by the linear system components

$$\widehat{N}(m;n) = \frac{\partial z(n)}{\partial y(n-m)} = \mathbf{L}_1 * \mathbf{L}_2, \tag{5.24}$$

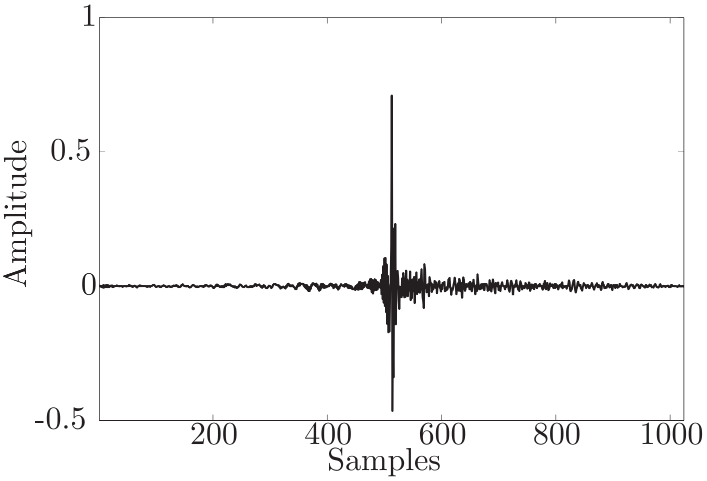
where $M = M_1 + M_{L_2} - 1$ is the memory size of the estimated LEM system, which corresponds to the linear convolution of the linear channel in Fig. 5.5.

For weakly nonlinear systems, the filtered-x equalization structure corresponds to the classic linear filtered-x scheme. Consequently, only the linear component of the acoustic channel has to be identified.

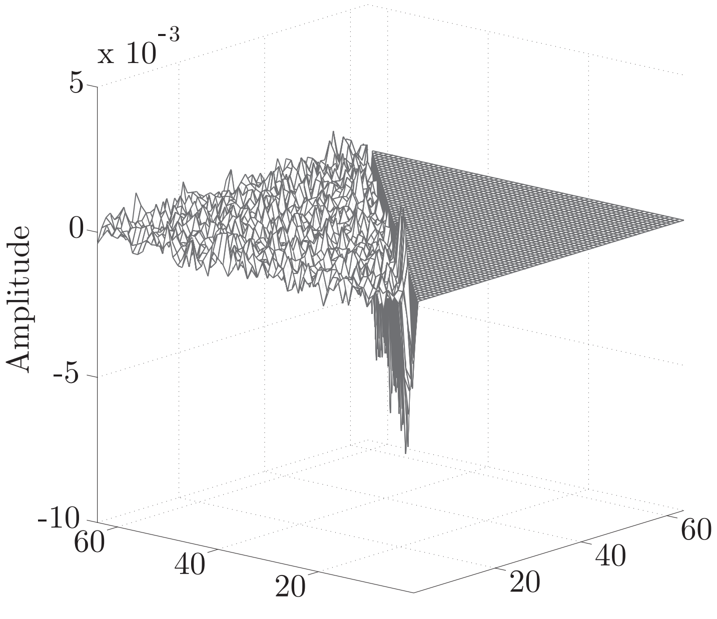
The inversion of weakly nonlinear systems has been previously presented in [106][107], where only the linear kernel is estimated for computing the nonlinear adaptive filters. In [62], the virtual secondary path is estimated with the linear component when the nonlinearities of the system are considered weakly. to do this, the authors use the phase response difference between the virtual secondary path and the linear component of the virtual secondary path. If this phase is within $\pm 90^{\text{o}}$, the virtual path can be approximated by its linear part.

Here, we use a variable $\alpha(n)$ parameter in (5.22) to provide a high LNLR and to assure a weakly system, where (5.23) can be accomplished. Also, we compute the adaptive equalization filter with a linear filter and with a 2nd-order Volterra kernel, which have been called in the previous experiment as linear Fx-NLMS and simplified NFx-NLMS algorithms.

For this simulation, the same filters of the previous experiments have been used. To assure a weakly nonlinear system, a LNLR= 25 dB has been chosen, with a variable $\alpha(n)$ that has been computed with a exponentially weighted moving average.



(a) Linear adaptive filter



(b) Triangular representation of the second-order adaptive filter

Figure 5.10. Adaptive linear and second-order Volterra filters.

Fig. 5.11 shows the evolution of the EMSE, with $\mu_1 = \mu_2 = 0.1$ and for the same input and noise signals. For comparison purpose, the curves of Fig. 5.8 are also represented, when $\alpha = 1$ for all the iterations. As it is shown, the linear Fx-NLMS and the simplified NFx-NLMS did not properly perform in that case. However, when the relation between the nonlinear output with respect to the linear output is weak, both the linear Fx-NLMS and simplified NFx-NLMS algorithms exhibit a good performance. This is expected as the nonlinearities are low, and the coefficients of the quadratic kernel have values close to 0.

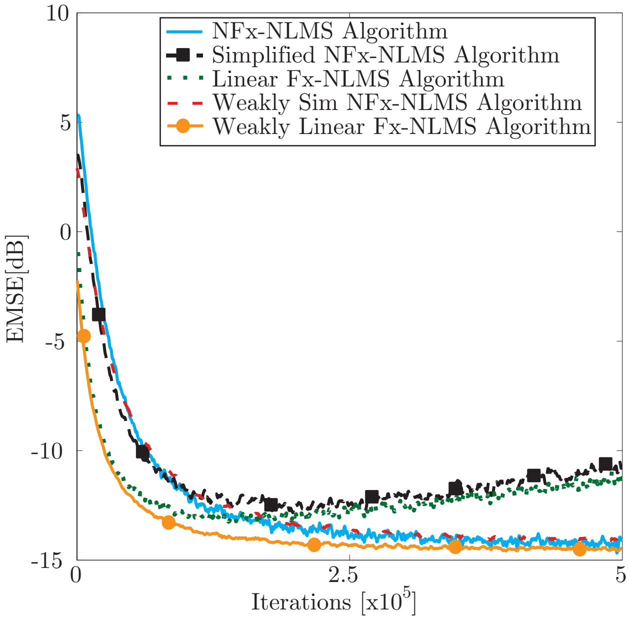


Figure 5.11. EMSE evolution of the NFx-NLMS algorithm for a weakly nonlinear system.

5.4 Conclusions

This chapter presents a nonlinear adaptive filter based on a nonlinear filtered-x structure embedded for room equalization applications. This structure is motivated by the fact that linear filtered-x schemes perform poorly when the system exhibits nonlinear distortion. For this scheme, both the nonlinear system and the adaptive compensation filters are computed using a second-order Volterra kernels.

In particular, it has been studied a novel nonlinear filtered-x scheme to compensate both room reverberation and nonlinear distortion with memory for the amplifier and loudspeaker devices. This approach is based on the development of a time-varying virtual filter that avoids problems of instability due to filter delays. The effectiveness and robustness of the NFx-NLMS algorithm has been evaluated in terms of the EMSE for different LNLR values. These results show that the proposed approach outperforms the linear Fx-NLMS type algorithm even with high LNLR conditions. Moreover, it exhibits a good performance in high nonlinear conditions, that is, for low LNLRs.

Finally, some simulations has been carried out to study the performance of the nonlinear filtered-x scheme when the LEM channel presents weakly nonlinear conditions, [106]. In that case, it has been shown that the virtual channel can be estimated using only the linear component of the LEM path and the filter performance achieves good results.

Conclusions 6

Conclusions 6

6.1 Conclusions and future research

Immersive systems are becoming a main topic for most applications, which includes audio systems to give a real and immersive sensation. To achieve this, audio rendering, recording and processing techniques are required.

The main motivation which underlie this dissertation is to accomplish this audio immersive sensation. However, the use of reproducing audio systems through rendering audio devices and inside enclosures, impairs the global audio effect and thus the 3D spatial sound.

For that purpose, an equalization system will be able to cancel these electroacoustic effects and will provide the spatial sound sensation of the desired 3D audio rendering system.

To this end, the equalization system should provide an efficient solution, but also take into account the time-varying conditions and the real-time requirements. An appropriate way to achieve this is to employ adaptive filtering algorithms.

In this dissertation, we have studied adaptive algorithms specifically designed for electroacoustic equalization. With this aim, the work is structured in three main parts.

- The first part, which is described in Chapter 3, includes different linear adaptive filtered-x schemes in order to cancel the acoustic impulse response, but also providing robust solutions for different room scenarios. With the aim of achieving a good performance, the combination of filters has been introduced. Furthermore, the biased scheme has been proposed to improve the EMSE for low SNRs. The steady-state behavior of this approach has been theoretically analyzed, with an improvement in the EMSE measure for low SNRs conditions.

  Since audio immersive systems require the use of multiple transducers, the multichannel adaptive equalization problem has been also presented in this chapter, for both the combined and the biased multichannel schemes.

  The simulation results carried out show a good performance. On the other hand, the high computational burden require for the proposed schemes has to be taken into account when these schemes are used in equalization applications. This increase in the computational cost is mainly due to the filtered-x structure that has to be applied to the input signals of the combining and scaling factors. Moreover, it is highly increased when applied to MIMO systems.

  Keeping this in mind and for future research, it is appropriate to seek for adaptive algorithms that require a lower computational cost, there should be a tradeoff between computational burden and algorithm performance. A way to solve this problem is to use parallel computing, for instance with GPUs, to provide efficient and real-time equalization. On the other hand, collaborative filtering techniques will allow to distribute the filtering processes between different nodes, thus reducing the computational time.

  With regard to adaptive combination of filters, many choices exist to pursue this research. Because only a convex combination of two filters with the same length has been considered, it would be a natural choice to study the performance of other kind of combinations with filters of different lengths and even using different updating algorithms.

- In the second part, included in Chapter 4, the block-based schemes are proposed for AE applications considering the combined and biased schemes. Motivated by the non-uniform energy distribution of the inverse adaptive filter, the approaches provide better performance, at the expense of increasing the computational burden.

Based on the optimal filter, the fitted block-based schemes have been also developed in this chapter to take advantage of the central energy distribution. Then, a class of new algorithms have been proposed to incorporate this a priori information into CBFx and BFX schemes. They show a better performance for the same computational cost. Furthermore, the proposed partitioned fitted-block schemes allow to obtain good results, specially when the characteristics of the inverse channel are *a priori* known.

Furthermore, the computational cost of these block-based schemes has been presented to put forward the increasing computational burden require with the number of blocks for filtered-x structures.

- When dealing with audio devices, the nonlinear effects caused by the introduction of loudspeaker distortion in the electroacoustic path should be taken into account. The third part of this dissertation is presented in Chapter 5. This chapter develops a nonlinear adaptive filter based on a nonlinear filtered-x structure for room equalization applications, which uses a time-varying virtual filter. This structure is required since linear filtered-x schemes perform poorly when the system exhibits nonlinear distortion. For this scheme, both the nonlinear system and the adaptive compensation filters are computed using a second-order Volterra kernels.

  A novel nonlinear filtered-x scheme has been presented to compensate both room reverberation and nonlinear distortion with memory for the amplifier and loudspeaker devices. Particularly, the nonlinear Fx-NLMS algorithm outperforms the linear Fx-NLMS type algorithm even with high LNLR conditions. Moreover, it exhibits a good performance in high nonlinear conditions, that is, for low LNLRs.

  In that chapter, the performance of the NFx scheme has been also studied when the LEM channel presents weakly nonlinear conditions, [106]. In that case, it has been shown that the virtual channel can be estimated using only the linear component of the LEM path and the filter performance achieves good results.

  Since the nonlinearities of the system are *a priori* not known and taking into account the combination schemes, a first attempt to improve nonlinear equalization systems should be the use of combination filters, as it has been previously proposed in [74][108], even combining different kinds of filters, as in [64][102].

With regard to the Volterra filter, in this work only second-order Volterra filters have been used to identify and cancel nonlinearities. However, this assumption is not always correct when dealing with loudspeakers and a third kernel of the Volterra filter may be required. Its implementation is straightforward, but it requires a higher computational cost as the number of coefficients of the adaptive filter of each kernel increases with its order.

Moreover, two appendices are included in this work. The first one gives an idea of the real-time implementation of the adaptive equalization system. For this purpose, the Fx-NLMS algorithm in frequency domain has been implemented, which shows good results, but also requires a high time to converge. With respect to the computing cost it will be easily reduced using parallel computing. Moreover, some frequency band methods are introduced in the first appendix, that will be interesting to implement as a future research.

The second appendix presents direct solutions to the equalization problem for its use in MIMO systems with a high number of loudspeakers and microphones. It includes objective and subjective measures when rendering with a WFS system.

To conclude this dissertation, it has to be noticed that all research works focus on audio applications require of a subjective analysis to give some insight of the real sensation achieved with the proposed algorithms.

To this end, some experiments, that are not described in this work, have been carried out with listeners, which show that although the cancellation of the acoustic echoes can be achieved, some other audio effects are added to the render signal, impairing the final acoustic sensation.

Thus, even if the objective measures show good results of the proposed algorithms, the subjective assessment provides results not as good as desirable. This is due to the fact that we are trying to ideally cancel the room response and it can not be always achieved and even can lead to impractical inverse responses. For that reason, the desired response should be smoothed to allow this equalization be consistent with the audio sensation.

The author recommends that further work must be undertaken to explore this.

6.2 List of publications

A list of published work produced and related to this dissertation is presented in chronological order in what follows.

- *Adaptive filtered-x algorithms for room equalization based on combined schemes: performance and theoretical analysis*, L. Fuster, M. de Diego, L. Azpicueta and M. Ferrer. IEEE Transactions on Audio, Speech, and Language Processing. Submitted for review.

- *Nonlinear filtered-x second-order adaptive Volterra filters for listening-room compensation*, L. Fuster, M. de Diego, M. Ferrer, G. Piñero and A. González. 23rd Proc. EURASIP European Signal Processing Conference (EUSIPCO). 2015.

- *Steady-state analysis of biased filtered-x algorithms for adaptive room equalization*, L. Fuster, M. de Diego, M. Ferrer and A. González. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2014.

- *A biased multichannel adaptive algorithm for room equalization*, L. Fuster, M. de Diego, M. Ferrer, A. González and G. Piñero. 20th Proc. EURASIP European Signal Processing Conference (EUSIPCO). 2012.

- *Análisis objetivo y subjetivo de la compensación de salas mediante técnicas Wave-Field Synthesis*, L. Fuster, J.J. López, A. González, M. Cobos. ITECKNE: Innovación e investigación en Ingeniería, Vol. 6, N^{o} 2, 2009.

- *Análisis subjetivo de compensación de salas mediante Wave-Field Synthesis - 1^{a} Parte*, L. Fuster, J.J. López, A. González, M. Cobos. XXIII Simposium Nacional URSI 2005, Proceedings, Madrid, 2008

- *Room Compensation using Multichannel Inverse filters for Wave Field Synthesis Systems*, L. Fuster, J.J. López, A. González, P. Zuccarello.

118th convention Audio Engineering Society (AES), Barcelona, España, 2005.

- *Time and Frequency Domain Room Compensation applied to Wave Field Synthesis*, L. Fuster, J.J. López, A. González, P. Faus. 8th International Conference on Digital Audio Effects (DAFx'05), Madrid, España, 2005.

Other publications related with this thesis, not as first author:

- *Prestaciones de un sistema de reproducción Wave Field Synthesis (WFS) utilizando unidades de procesamiento gráfico*, Jose A. Belloch, L. Fuster, E.S. Quintana, A.M. Vidal, A. González. XXV Congreso Espańol de Acústica - Tecniacústica , Murcia, Spain, 2014.

- *Multi-Speaker Localization, Separation and Resynthesis for Next Generation Videoconferencing*, M. Cobos, J.J. López, L. Fuster, E. Aguilera. WAVES. Vol. 1, N^{o} 1, pp. 53-60, 2009.

- *WFS prototypes at the Technical University of Valencia and Alicante*, J.J. López, A. González, B. Pueo, L. Fuster, J. Escolano, M. Cobos. 1st DEGA Symposium on Wave Field Synthesis, Proceedings, Illmenau, Germany, 2007.

- *Room Compensation in Wave Field Synthesis by means of Multichannel Inversion*, J.J. López, L. Fuster, A. González. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, New Paltz, Nueva York, 2005.

A

Real-time equalization

A Real-time equalization

This appendix includes some initial research to study the frequency algorithms presented in this dissertation. In particular and as a starting point, the constrained frequency domain adaptive algorithm has been implemented.

Firstly, the mathematical formulations and constraints for frequency algorithms are presented. After that, some laboratory experiments are included.

A.1 Real time room equalization in frequency domain

Fig. A.1 shows the block diagram of the AE system when implemented in frequency domain with block operations. Moreover, $\widehat{\mathbf{h}}(k)$ corresponds to the FFT of $\widehat{\mathbf{h}}$, zero-padding the vector to the FFT size. The coefficients filter update is implemented using the Fx-NLMS method in the frequency domain. For this purpose, an FFT of $2L_w$ samples has been chosen, being $L_w = 2L_h$ twice the length of the estimated channel response, L_h.

As commented before, for the adaptive filtering of Fig. A.1, the overlap-save method has been used. Initially, the block size of the input signal $(2N)$ has been chosen as the double of the adaptive filter for the FFT operation,

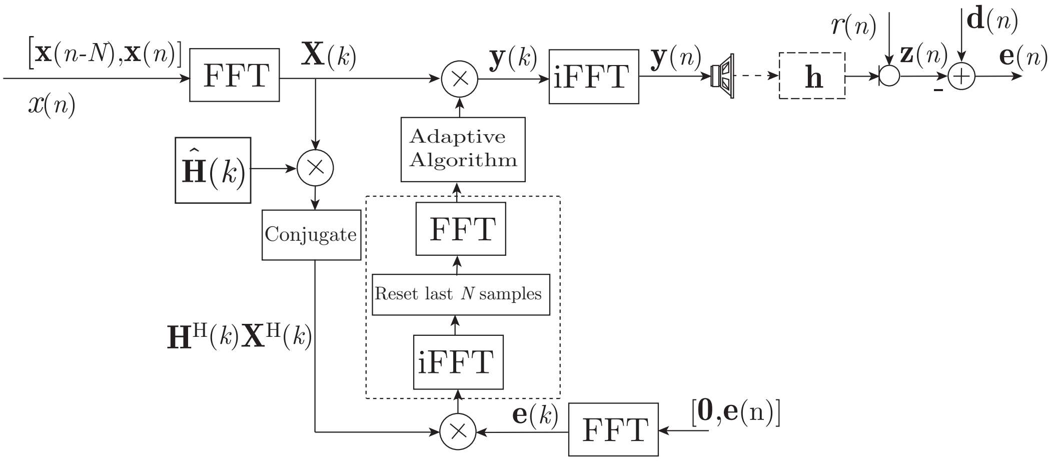


Figure A.1. Block diagram of an AE system in frequency domain with the overlap-save method.

to allow $L_w = N$, for an overlapping factor of 50%. Then, this factor uses in each iteration the last N samples of the input signal and the previous N samples, to perform the overlap operation. Thus, the FFT is computed over the last $2L_w$ input samples, $\mathbf{x}(k) = FFT\Big([\mathbf{x}(n-N), \mathbf{x}(n]\Big)$, where vectors $\mathbf{x}^T(n) = [x(n), x(n-1), \ldots, x(n-(N+1))]^T$ and $\mathbf{x}^T(n-N) = [x(n-N), x(n-N-1), \ldots, x(n-2N+1)]^T$ have $N = L_w$ samples each.

Furthermore, this frequency algorithm introduces a temporal constraint (*constrained frequency domain adaptive filter*) in order to be able to implement the linear convolution. This restriction (in dashed line of Fig. A.1) replaces the last N elements of the product vector $\hat{\mathbf{H}}^H(k)\mathbf{X}^H(k)\mathbf{e}(k)$ by zeros, which corresponds to the filtered-x structure. Using this constraint, the number of coefficients of the adaptive filter, that has to be taken into account, is N. Also for the output signal of the filter, $\mathbf{y}(n)$, N samples are useful, taking only the last N samples of the output vector $IFFT[\mathbf{y}(k)]$.

In order to compute this constraint, the matrices $\mathbf{G}$ and $\mathbf{K}$ are used,

$$\mathbf{G} = \begin{pmatrix} \mathbf{I}(N)\mathbf{0}(N) \\ \mathbf{0}(N)\mathbf{0}(N) \end{pmatrix} \tag{A.1}$$

$$\mathbf{K} = (\mathbf{0}(N)\mathbf{I}(N)), \tag{A.2}$$

where $\mathbf{I}(N)$ is the identity matrix of size $N \times N$ and $\mathbf{0}(N)$ is a zero matrix of size $N \times N$. $\mathbf{G}$ and $\mathbf{K}$ allow to simplify the previous operations taking only the desired samples of the vectors.

Another point to consider is the time domain used to compute the error $\mathbf{e}(n)$. In this case, $\mathbf{e}(n)$ is obtained subtracting from the desired vector $\mathbf{d}(n)$ the last N samples of the microphone signal, $\mathbf{z}(n)$, where the desired signal for the equalization problem will correspond to the input signal with a τ sample delay $\mathbf{d}(n) = \mathbf{x}(n-\tau)$. For this error vector and in order to compute the FFT, N zeros are added before $\mathbf{e}(n)$, to discard the first N samples of the circular convolution. With this operations, the error $\mathbf{e}(k)$ can be used to update the coefficients of the adaptive filter in frequency domain, $\mathbf{w}(k)$.

The following expressions summarized the frequency domain Fx-NLMS algorithm with the overlap-save method:

$$\mathbf{X}(k) = \mathrm{diag}\Big(\mathrm{FFT}([\mathbf{x}(n-N), \mathbf{x}(n)])\Big), \tag{A.3}$$

where diag$(\cdot)$ represents a diagonal matrix, whose main diagonal corresponds to the FFT of the last $2N$ input samples.

$$\mathbf{y}(k) = \mathbf{X}(k)\mathbf{w}(k). \tag{A.4}$$

$$\mathbf{y}(n) = \mathbf{K}\mathrm{IFFT}(\mathbf{y}(k)), \tag{A.5}$$

where $\mathbf{K}$ matrix allows to remove the first N samples and to take only the last N samples of the output $IFFT(\mathbf{y}(k))$.

To compute the desired vector $\mathbf{d}(n)$ two delays have to be considered: a delay of N samples as a result of working with blocks of size N due to the latency of the system and another delay is necessary to center the response of the inverse filter. For this last delay, a delay of $\lceil N/2 \rceil$ samples will be considered.

$$\mathbf{d}(n) = \mathbf{x}\Big(n - \Big\lceil N + \frac{N}{2} \Big\rceil\Big) \tag{A.6}$$

Thus, the error vector $\mathbf{e}(n)$ is given by,

$$\mathbf{e}(n) = \mathbf{d}(n) - \mathbf{z}(n), \tag{A.7}$$

where $\mathbf{z}(n)$ corresponds to the electroacoustical vector recorded by the microphone that includes the last N samples.

$$\mathbf{e}(k) = \text{FFT}\begin{pmatrix} \mathbf{0}(N) \\ \mathbf{e}(n) \end{pmatrix} \tag{A.8}$$

$$\phi(n) = \text{IFFT}\Big(\hat{\mathbf{H}}^H(k)\mathbf{X}^H(k)\mathbf{e}(k)\Big), \tag{A.9}$$

where $\hat{\mathbf{H}}^H(k)$ is also a diagonal matrix that includes the $\widehat{\mathbf{h}}(k)$ vector in its main diagonal.

The adaptive coefficients for the Fx-NLMS in the frequency domain are given by,

$$\mathbf{w}(k) = \mathbf{w}(k-1) - \text{diag}(\overline{\boldsymbol{\mu}}(k))\text{FFT}\begin{pmatrix} \phi_N(n) \\ \mathbf{0}(N) \end{pmatrix}, \tag{A.10}$$

where $\phi_N(n)$ corresponds to the first N samples of $\phi(n)$ and matrix $\mathbf{0}(N)$ replaces the last N samples of the IFFT in (A.9) to implement the linear convolution. $\overline{\boldsymbol{\mu}}(k)$ is a column vector containing the adaptive step size parameters of each ith frequency coefficient $\overline{\mu}_i$, for $i = 0, 1, \dots, 2N-1$, which is computed as,

$$\overline{\mu}_i = \frac{\mu}{P_i}, \tag{A.11}$$

where μ is the step size and P_i has been estimated for each frequency component $k = 0, 1, \dots, 2N-1$, of the FFT, as in [43]:

$$P_i(k) = \gamma P_i(k-1) + (1-\gamma)|x_{\text{f}}(k)|^2, \tag{A.12}$$

where $0 < \gamma < 1$ exponentially weights the actual $P_i(k)$ value respect to the previous value $P_i(k-1)$ and $|x_{\text{f}}(k)|^2$ is the power of the input signal filtered through the estimated channel response for the kth frequency component.

There is another algorithm, the *unconstrained frequency domain adaptive filter*, that has not restrictions in the adaptation algorithm (without the dashed block of Fig. A.1) and directly uses the product of the error with the filtered input signal. It computes then the circular convolution, not the linear one.

$$\mathbf{w}(k) = \mathbf{w}(k-1) - \mathrm{diag}(\overline{\boldsymbol{\mu}}(k))\hat{\mathbf{H}}^H(k)\mathbf{X}^H(k)\mathbf{e}(k) \tag{A.13}$$

This algorithm has a lower computational cost, but it doubles the number of iterations required to achieve the same final error, [109]. Also in [109], the *self-orthogonalizing frequency domain adaptive filter* is studied, which improves the convergence speed. It uses the autocorrelation matrix to compute the filter weights and the discrete cosine transform, [110], to cancel some correlation of the input signal.

The problem with these systems is the slow convergence speed of these algorithms when using correlated audio signal. Goetze et al. in [29] and [111], introduce a decoupled filtered-x LMS algorithm to improve the convergence of the algorithm. It does not employ the recorded audio signal, but a estimated signal to invert the filter channel. It is based on the concept that audio rendering signals are correlated and can not contain all frequencies, thus the convergence speed is reduced, as it depends on the direction taken to reach the minimum value ([12]) and will be not able to compensate all frequencies. With the decoupled algorithm the inverse filter is computed using a white Gaussian noise, which has a faster convergence, as it has a nondirectional convergence, and allowing the filter to compensate all frequencies. Whereas the rendering and recording are done with the desired sound.

Another approaches work with frequency bands, as in [26] and [112], where the input signal is band-pass filtered and it requires one adaptive filter with $2N$ samples for each frequency band.

A.2 Laboratory experiments

Although different algorithms in the frequency domain have been proposed in the literature, as an initial step to study the frequency algorithms presented in this dissertation, the constrained frequency domain adaptive filter has been implemented. Moreover and to ensure a good convergence of the algorithm, an input Gaussian noise signal has been used.

With regard to the equalization system, experiments with SISO systems for different loudspeaker-microphone positions and distances have been carried out.

In order to use the filtered-x structure in the equalization algorithm, a real time identification frequency domain NLMS algorithm has been previously involved to identify the impulse response.

A.2.1 Laboratory setup

The laboratory experiments have been carried out in the listening acoustic room located at the laboratories of the Audio and Communications Signal Processing Group of the Institute of Telecommunications and Multimedia Applications of the Universitat Politècnica de València.

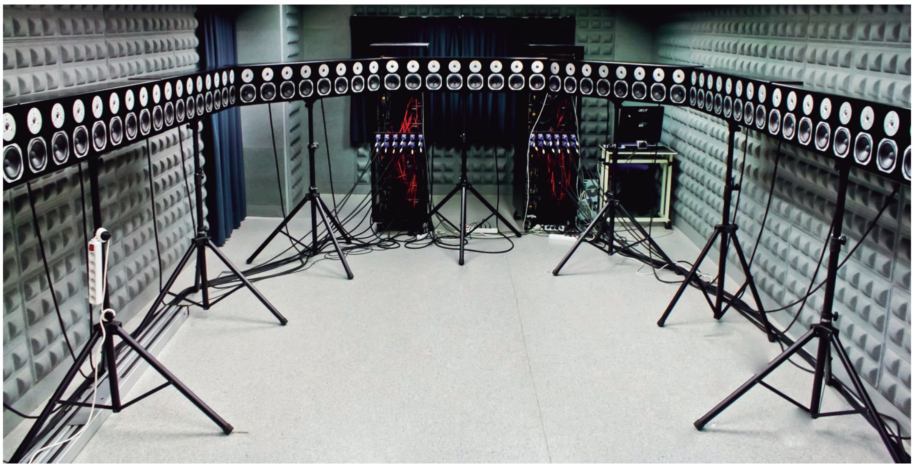


Figure A.2. Equipment and acoustic room.

The algorithms were programmed in C language on an Intel Core i7 (3.07 GHz) CPU platform that communicates with an audio card MOTU 24I/O through the ASIO (Audio Stream Input Output) drivers. The rendering loudspeaker was a Event PS6 model (Project Studio 6 Biamped System) with a frequency range from 45 Hz to 20 kHz, ±3 dB. Whereas the microphone was an omnidirectional Earthworks QTC (Quiet Time Coherent) with frequency range 4 Hz - 40 kHz ±1 dB. A sampling frequency of 44.1 kHz and a buffer size of 2048 samples have been employed. The buffer time corresponds to $2048/44100 = 46.4$ ms and has to be greater than the processing time to allow the real time implementation of the algorithm.

A.2.2 Laboratory results

For the first experiment, the microphone and loudspeaker were close located to 27 cm and at a distance of 1.25 m to the floor.

The impulse response measured with the adaptive identification algorithm is shown in Fig. A.3, where the first 2048 samples correspond to the latency of the real-time system and therefore are zero. As the impulse response is sparse, the size of the inverse filter has been computed also with 4096 samples. The step size parameter has been set to $\mu = 10^{-5}$. As can be observed in Fig. A.4 the time delay has been properly selected to obtained a centered response.

The evolution of the error signal (in linear magnitude) and the NPM parameter (in dB) can be seen in Fig. A.5 and Fig. A.6, respectively, which shows a good performance. Although the convergence speed is low. In this case, a modified version of the algorithm used will improve the convergence behavior.

Moreover, the convolution of the two responses after the final convergence is shown in Fig. A.7, which corresponds almost to the desired response with a low noise level.

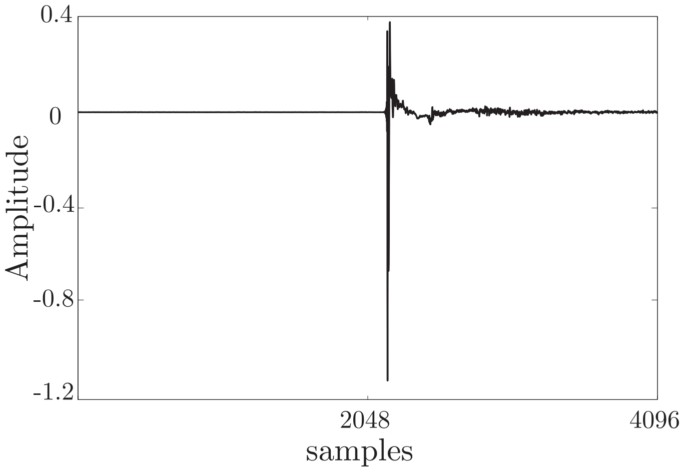


Figure A.3. Experiment 1: Room channel response measured after convergence.

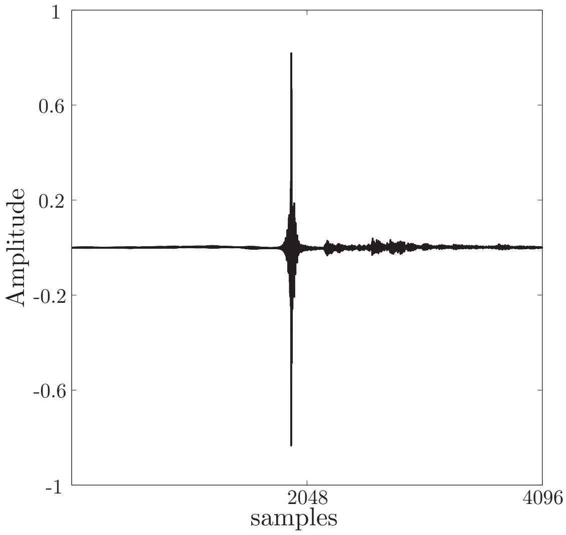


Figure A.4. Experiment 1: Inverse adaptive filter.

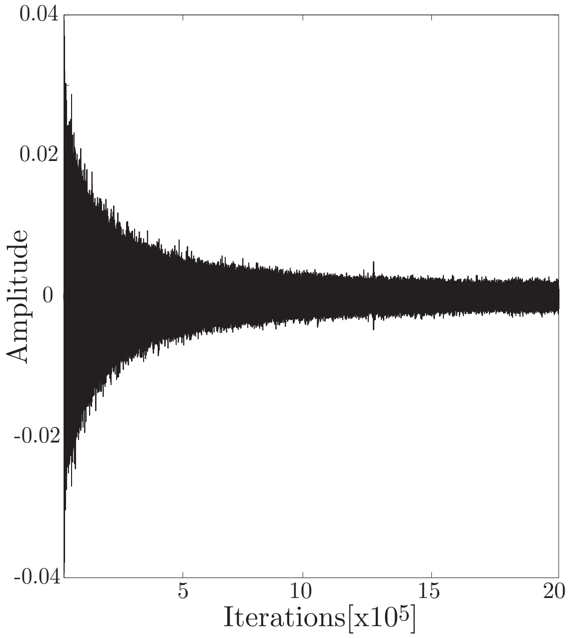


Figure A.5. Experiment 1: Evolution of the error signal $e(n)$.

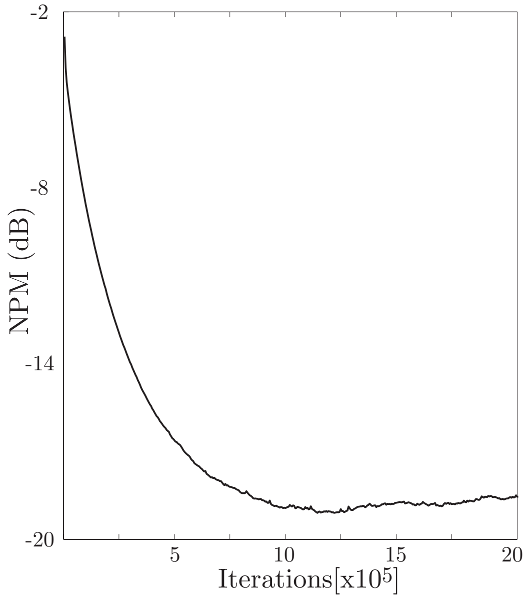


Figure A.6. Experiment 1: Evolution of the NPM parameter in dB.

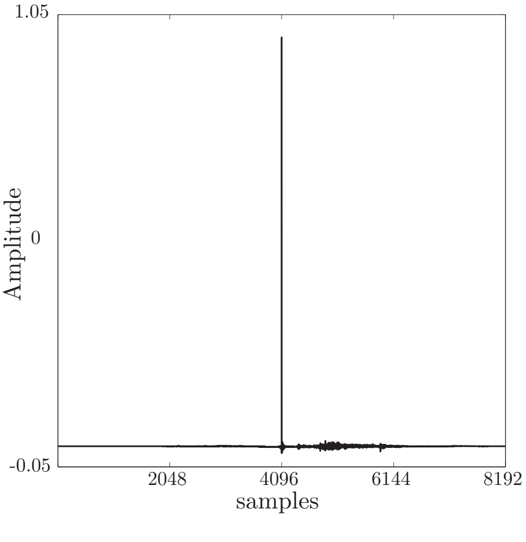


Figure A.7. Experiment 1: Convolution of the channel response and the final inverse filter.

For the second experiment, the distance between the microphone and loudspeaker was 1.7 m. Moreover, the microphone was located close to a wall to increase the echoes. In this case, the number of samples of the RIR was 10240 to assure that all echoes were included (see Fig. A.8). Also the length of L_w was 10240 samples, in Fig. A.9. For the step size parameter, μ was set to $3 \cdot 10^{-8}$.

The convergence of the algorithm has been slow down and it requires more iterations than the experiment 1 to converge. The evolution of the error signal and the NPM parameter can be seen in Fig. A.10 and Fig. A.11, respectively.

The convolution of the two responses after $8 \cdot 10^{6}$ iterations is not as good as in experiment 1 (see Fig. A.12). Moreover, the NPM evolution is depicted in Fig. A.11.

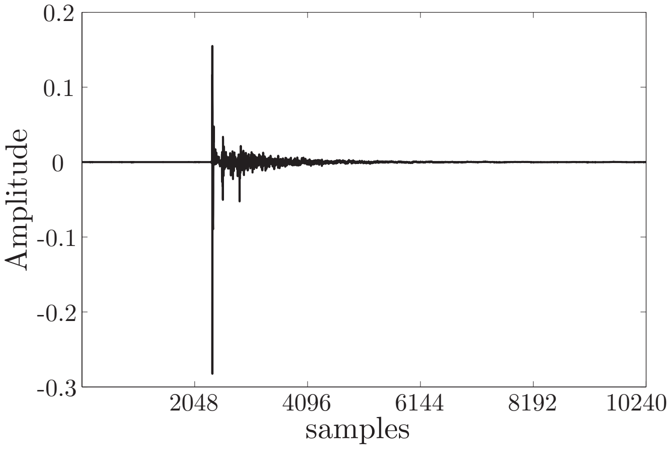


Figure A.8. Experiment 2: Room channel response measured after 40 ms.

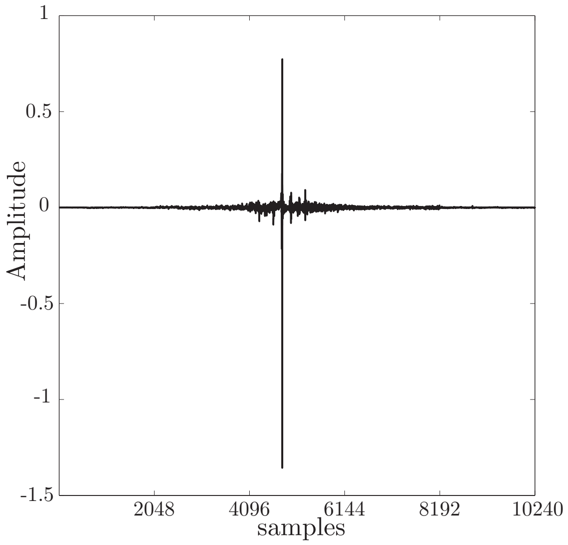


Figure A.9. Experiment 2: Inverse adaptive filter.

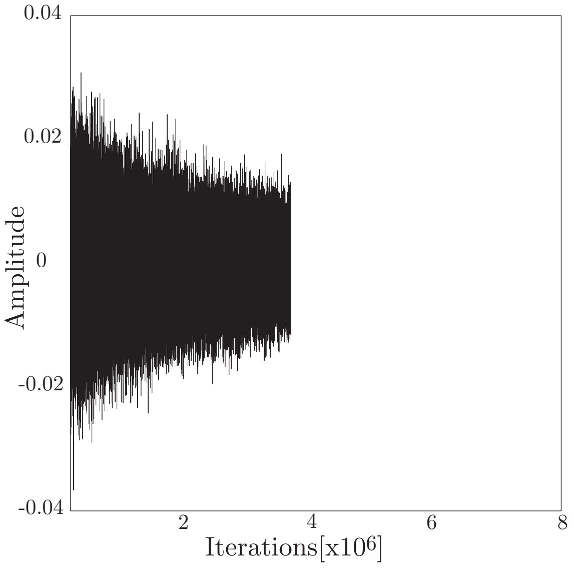


Figure A.10. Experiment 2: Evolution of the error signal $e(n)$.

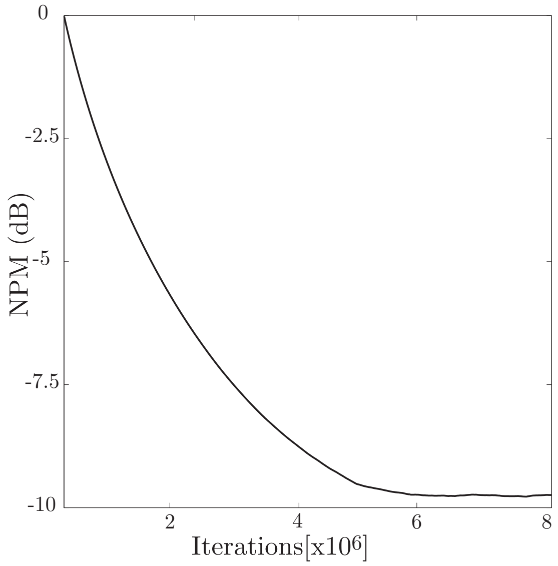


Figure A.11. Experiment 2: Evolution of the NPM parameter in dB.

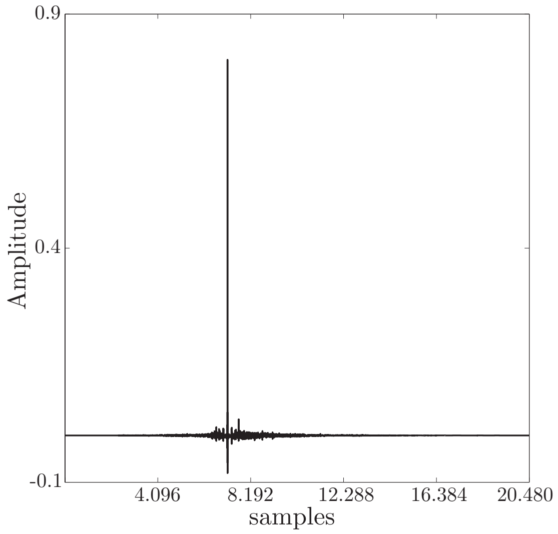


Figure A.12. Experiment 2: Convolution of the channel response and the inverse filter.

Direct multichannel equalization in a WFS system

B

Direct multichannel equalization in a WFS system

B

This appendix includes some experiments that have been carried out to show the performance of MIMO equalization systems, when dealing with audio rendering systems that require the use of a high number of loudspeakers, such as for WFS system, and also a huge amount of listening positions to achieve a global equalization.

To compute these massive MIMO inverse channels, previous algorithms that have been efficiently implemented in [4] and [113] have been used.

In the following, an introduction to massive MIMO channels is given. Details of the experiments carried out, including a subjective analysis, are also provided.

B.1 Wave Field Synthesis

Wave Field Synthesis (WFS) is a method of sound reproduction, based on fundamental acoustic principles [79],[80]. It enables the generation of sound fields with natural temporal and spatial properties within a volume or area bounded by secondary sources (arrays of loudspeakers). This method offers a large listening area with uniform and high reproduction quality.

The theoretical basis of WFS is given by the Huygens' principle. According to this, the propagation of a wave front can be described by recursively adding the contribution of a number of secondary point sources distributed along the wave front. This principle can be used to synthesize acoustic wave fronts of an arbitrary shape.

A synthesis operator for each loudspeaker can be derived using this theoretical basis. The general 3D solution can be transformed into the 2-D solution, which is sufficient for reconstructing the original sound field in the plane of listening [81],[114],[115]. For that purpose a linear array of loudspeakers is employed to generate the sound field of virtual sources.

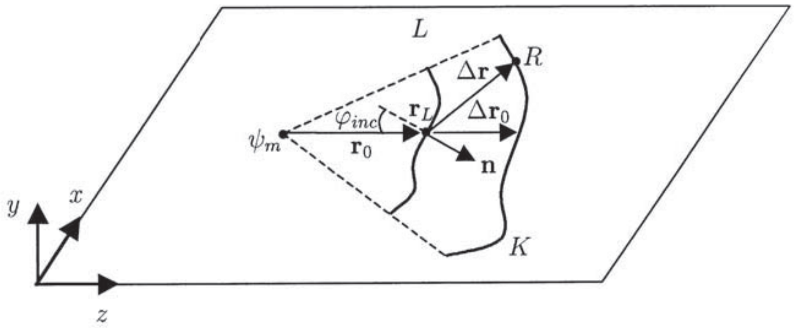


Figure B.1. Geometric representation of the primary point source ψ_m at point R using an arbitrary distribution of secondary sources over the line L.

The field rendered by a source at a point R (see Fig. B.1) within the area surrounded by the loudspeakers can be expressed as equation (B.1).

$$P(R) = \sum_{n=1}^{N} Q_n(\omega) \frac{e^{-jk\triangle r_n}}{\triangle r_n}, \tag{B.1}$$

where $\frac{e^{-jk\triangle r_n}}{\triangle r_n}$ represents the free field propagation between the nth secondary source (loudspeaker) and the point R within the listening area. $Q_n(\omega)$ corresponds to the expression of the nth loudspeaker driving signal for a rendering system of N loudspeakers. These driving signals are dependent on the virtual source, loudspeakers and listening area positions [81] and are given by

$$Q_n(\omega) = g_0(\mathbf{r}_{L,n})\sqrt{\frac{\lambda k}{2\pi j}}S(\omega)\cos\varphi_{inc,n}\frac{e^{\lambda jkr_n}}{\sqrt{r_n}}, \tag{B.2}$$

where r_n represents the distance between the primary source and the nth loudspeaker. $\varphi_{inc,n}$ is the angle between the vector $\mathbf{r}_n$ (from the primary source to the nth loudspeaker) and the normal vector $\mathbf{n}$ of the loudspeaker array at the nth loudspeaker position. Furthermore, $\lambda = 1$ for primary sources located between the loudspeakers and the listener positions, and $\lambda = -1$ for exterior primary sources. Moreover, the amplitude factor $g_0(\mathbf{r}_{L,n})$ is defined by

$$g_0(\mathbf{r}_{L,n}) = \sqrt{\frac{\Delta r_0(\mathbf{r}_{L,n})}{r_0(\mathbf{r}_{L,n}) - \lambda\Delta r_0(\mathbf{r}_{L,n})}}, \tag{B.3}$$

being $\Delta r_0(\mathbf{r}_{L,n})$ the distance between the nth loudspeaker and the point over the K line with the same stationary phase that the nth loudspeaker (see Fig. B.1), expressed as

$$\Delta r_0(\mathbf{r}_{L,n}) = |\mathbf{r}_{R,0}(\mathbf{r}_{L,n}) - \mathbf{r}_{L,n}| \tag{B.4}$$

Because of the separation between the loudspeakers, there exists an spatial aliasing frequency that allows the render system to work properly up to this frequency. This frequency is given by

$$f_{al} = \frac{c}{2\Delta x\sin\Theta_{max}} \tag{B.5}$$

being c the speed of sound, Δx represents the separation between loudspeakers and Θ_{max} corresponds to the maximum angle of incidence of the synthesized wave field relative to the loudspeaker array.

B.2 Application of MIMO room equalization to WFS reproduction systems

The multichannel inverse filtering problem introduced in Fig. 2.10 can be straightforwardly applied to a WFS system (see Fig. B.2).

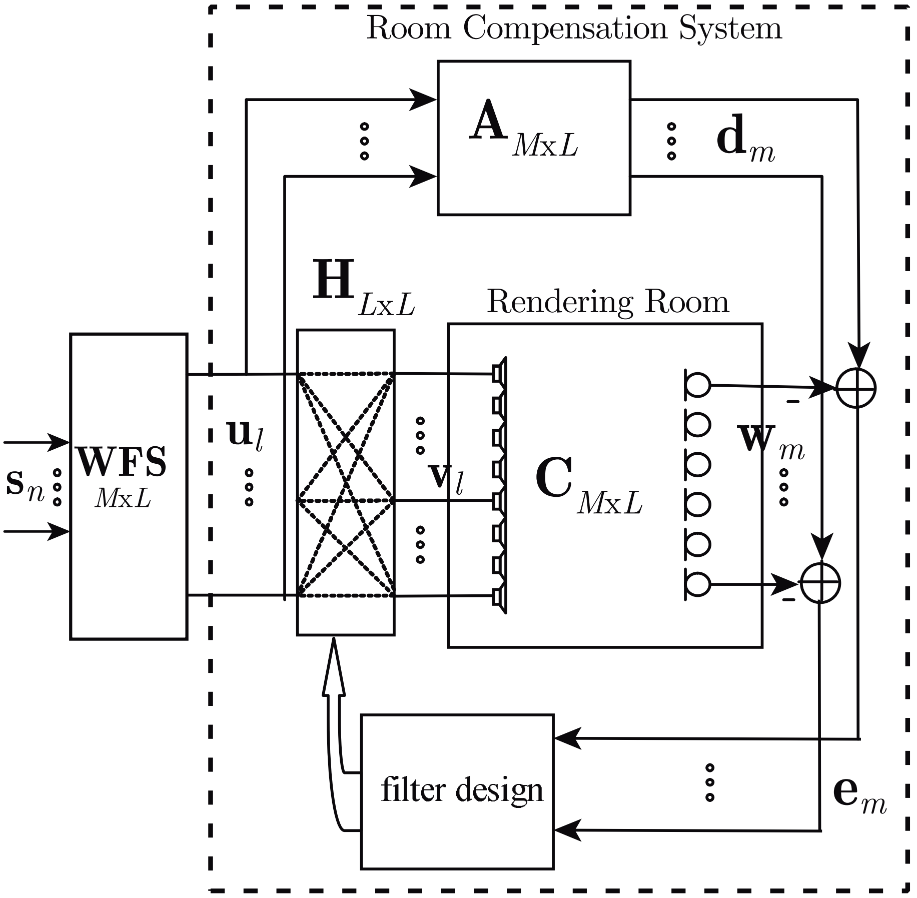


Figure B.2. Room correction system for WFS.

The filter matrix $\mathbf{H}$ is calculated using the transmission channels actual responses, which are measured *a priori* (matrix $\mathbf{C}$), and the desired signals at the listening or reproduction points. In contrast to the multichannel inverse filtering problem depicted in Fig. 2.10, the input signals to matrices $\mathbf{A}$ and $\mathbf{H}$ are not the original K sources, but the L excitation signals for the secondary sources provided by the WFS rendering algorithm. Thus, the filter matrix $\mathbf{H}$ is composed of $L \times L$ inverse filters in WFS systems.

Using this approach, as can be seen from Fig. B.2, the room compensation system is independent of the previous reproducing system that is used. In this case, the WFS, where the number of inputs to our MIMO WFS compensation system will be equal to the number of loudspeakers.

B.3 Time-direct computation of the MIMO inverse system

In order to compute a possible inverse filters bank by using a finite number of coefficients, that enables to equalize a given MIMO system, we have used the algorithm proposed in [113]. This algorithm can efficiently cope with time domain deconvolution problems where a huge set of linear equations should be solved to design optimal filters under the least squares error criterion.

The specific MIMO inverse system considered is described below. This system, depicted in Fig. B.2, can be described in the time domain by the set of equations: $\mathbf{A} = \mathbf{C}\mathbf{H}$, where $\mathbf{C}$ is composed of $M \times L$ blocks $\mathbf{C}_{ij}$ and $\mathbf{H}$ is composed of $L \times L$ vectors $\mathbf{h}_{jk}$ of n_h samples. Each block $\mathbf{C}_{ij}$ is a Toeplitz matrix that involves the convolutions with the ijth channel. Each product between matrix $\mathbf{C}_{ij}$ and vector $\mathbf{h}_{jk}$ performs the convolution between ijth transmission channel and jkth inverse filter. The bank of filters is usually calculated by the least squares method solving $\mathbf{C}^T\mathbf{C}\mathbf{H} = \mathbf{C}^T\mathbf{A}$. The symmetric matrix $\mathbf{C}^T\mathbf{C}$ is given by,

$$\mathbf{C}^T\mathbf{C} = \begin{bmatrix} \sum_{m=1}^{M} \mathbf{R}_{m1\,m1} & \cdots & \sum_{m=1}^{M} \mathbf{R}_{mL\,m1} \\ \sum_{m=1}^{M} \mathbf{R}_{m1\,m2} & \cdots & \sum_{m=1}^{M} \mathbf{R}_{mL\,m2} \\ \vdots & & \vdots \\ \sum_{m=1}^{M} \mathbf{R}_{m1\,mL} & \cdots & \sum_{m=1}^{M} \mathbf{R}_{mL\,mL} \end{bmatrix} \tag{B.6}$$

where $\mathbf{R}_{ij\,kl} = \mathbf{C}_{ij}^T\mathbf{C}_{kl}$ is a Toeplitz matrix.

Therefore matrix $\mathbf{C}^T\mathbf{C}$ is composed of $L \times L$ blocks of $n_h \times n_h$ elements. Each block has a Toeplitz structure. In the case of WFS, the multichannel system has L input signals from the WFS matrix. There will be L different sets of equations where the terms on the right-hand side, the columns of $\mathbf{C}^T\mathbf{A}$ are different, but sharing the same main matrix, $\mathbf{C}^T\mathbf{C}$, of $\mathbf{C}^T\mathbf{C}\mathbf{H} = \mathbf{C}^T\mathbf{A}$.

Furthermore, the main matrix $\mathbf{C}^T\mathbf{C}$ has a Toeplitz-block structure. This structure itself does not allow the use of efficient solution techniques. However, performing simple rows and columns permutations, a Block Toeplitz matrix can be achieved from the Toeplitz-block one. Thus a generalization of the *fast* methods used in the scalar Toeplitz case can be used for solving the Block Toeplitz case.

It is important to note that an FFT algorithm can be employed to compute the transmission channels autocorrelation and cross-correlation matrices that appear in equation (B.6).

While the solution of a general linear equations set of order n requires $O(n^3)$ operations, there exist several methods for taking advantage of the structure of a Toeplitz matrix. We refer to the well known Levinson and Trench as *fast* Toeplitz solvers because they require $O(n^2)$ arithmetic operations for the solution of an $n \times n$ Toeplitz set of equations.

In order to save computation time, we solve the generic set of equations $\mathbf{R}\mathbf{h} = [1, 0, \ldots, 0]^T$ using the Durbin's algorithm [116]. This algorithm exploits the simplified form of the right-hand side term and provides a further computational cost reduction compared to Levinson algorithm. Durbin algorithm provides first column of the inverse matrix with a computational saving of 50% compared to the general case. In order to solve the set of equations, the Gohberg-Semencul formula [117] can be used.

B.4 Frequency-direct computation of the MIMO inverse system

As an alternative to the computation of multichannel inverse filters in time domain, a fast deconvolution method was proposed in [4]. Fast deconvolution computes the inverse filters in frequency domain by using the FFT. The main benefit of the algorithm is the reduction in computation time.

In frequency domain, matrix **H** (or control filter matrix) which minimizes the quadratic error between the desired response (matrix **A**) and the room response (matrix **C**) is given by:

$$\mathbf{H}_{LSE}(z) = [\mathbf{C}^H(z^{-1})\mathbf{C}(z) + \beta\mathbf{I}]^{-1}\mathbf{C}^H(z^{-1})\mathbf{A}(z) \tag{B.7}$$

where β is the regularization parameter, which allows $\mathbf{C}^H\mathbf{C} + \beta\mathbf{I}$ to be no singular for $\beta > 0$, despite a high β implies a more biased solution to the original least squares problem.

A detailed explanation of the fast deconvolution method can be found in [4]. This method has proved to be very useful and easy to use, but can suffer from circular convolution effects when the inverse filters are not long enough compared to the duration of the responses of the transmission channels.

B.5 Experimental setup

The purpose of the laboratory experiments was to validate the multichannel inversion of MIMO systems as a possible practical solution to room compensation for WFS reproduction systems. The following laboratory experiment has been carried out. An opened U-shaped WFS array of 32 loudspeakers (4 of arrays of 8 loudspeakers each) was installed in a real room of dimensions 4 × 7 × 2.5 m. The room is acoustically conditioned with slight reflections in walls, ceiling and floor. Fig. B.3 shows the array of loudspeakers inside the listening room.

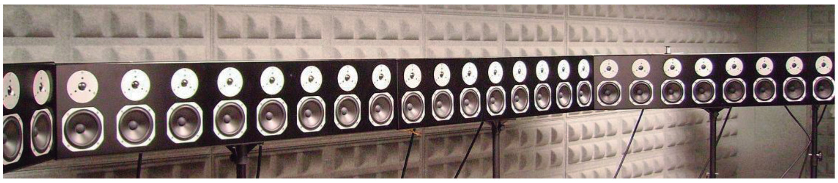


Figure B.3. Setup of WFS array and the laboratory.

For this loudspeaker distribution (18 cm separation), the spatial aliasing frequency is $f_{al} \approx 950Hz$. The room impulse responses (RIR) between each loudspeaker and 577 listening points were measured. The microphones were located in a circular distribution inside the listening area at the loudspeakers horizontal plane, as it is shown in Fig. B.4.

A set of 32 × 577 RIR were obtained using a Maximum Length Sequences (MLS) measurement method, which has been specially adapted for fast measuring of multichannel systems [118]. From these RIR, the matrix **C** containing the room responses with the direct signal and the first reflections was built. Concretely, the signals were obtained with a sampling frequency of 48 kHz, and decimated by 6 to get a sampling frequency of 8 kHz, because frequencies above 2 kHz were no considered due the aliasing frequency. Each RIR was windowed in time domain taking only the first 500 samples (about 62 ms).

Then, the bank of inverse filters **H** was computed to correct the undesired effects of the room. The algorithm previously explained was used to obtain an ideal channel up to $2kHz$ with the corresponding delay. In the algorithm, matrix **A** is designed to emulate free field conditions.

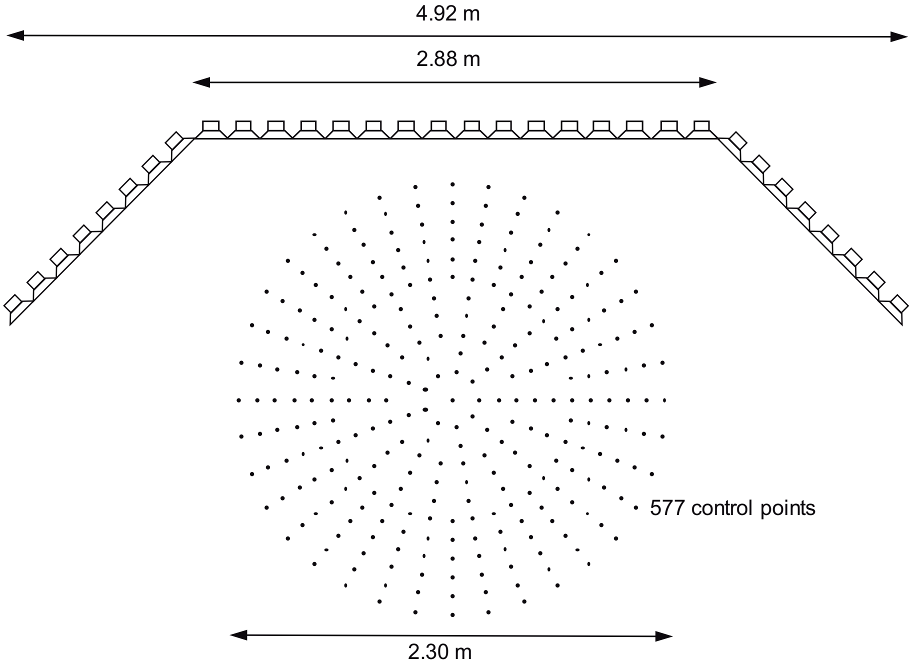


Figure B.4. Wave-Field Synthesis array of 32 loudspeakers and position of the control points.

In order to evaluate the system, measures were taken at the same listening points with the bank of inverse filters working. For every virtual source the driving signal of each loudspeaker was computed with the WFS system. These driving signals were filtered through its corresponding inverse filter bank and added at each loudspeaker before being rendered. The acoustical path was simulated by performing the convolution of the loudspeaker excitation signal with the previously measured acoustic channel. A meaningful improvement was observed at these points, obtaining better listening quality.

B.6 Results

A set of experiments have been included in this appendix to show the performance of MIMO equalization when using a WFS rendering system.

B.6.1 Time-domain vs. frequency-domain performance

In order to compare the results obtained using the Block Toeplitz solvers in time domain and the fast deconvolution algorithm in frequency domain, a comparison of the compensation achieved with both algorithms has been carried out. A randomly selected RIR between a loudspeaker and a control point has been taken for this purpose.

Fig. B.5 shows the frequency response before applying the compensation filter. Fig. B.6 (a) represents the same response after the bank of filters computed in time domain. The upper plot shows the response with a bank of inverse filters of length of 512 samples and the lower plot for a length of 1024 samples (10 dB apart). Both responses are very similar.

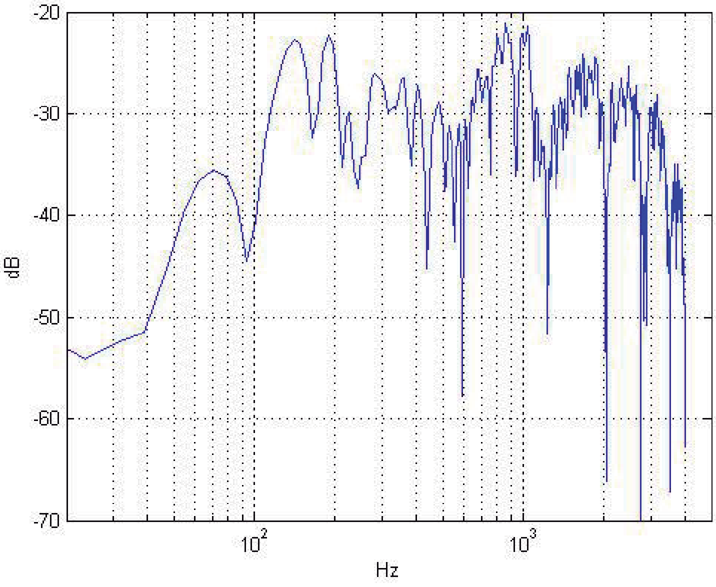
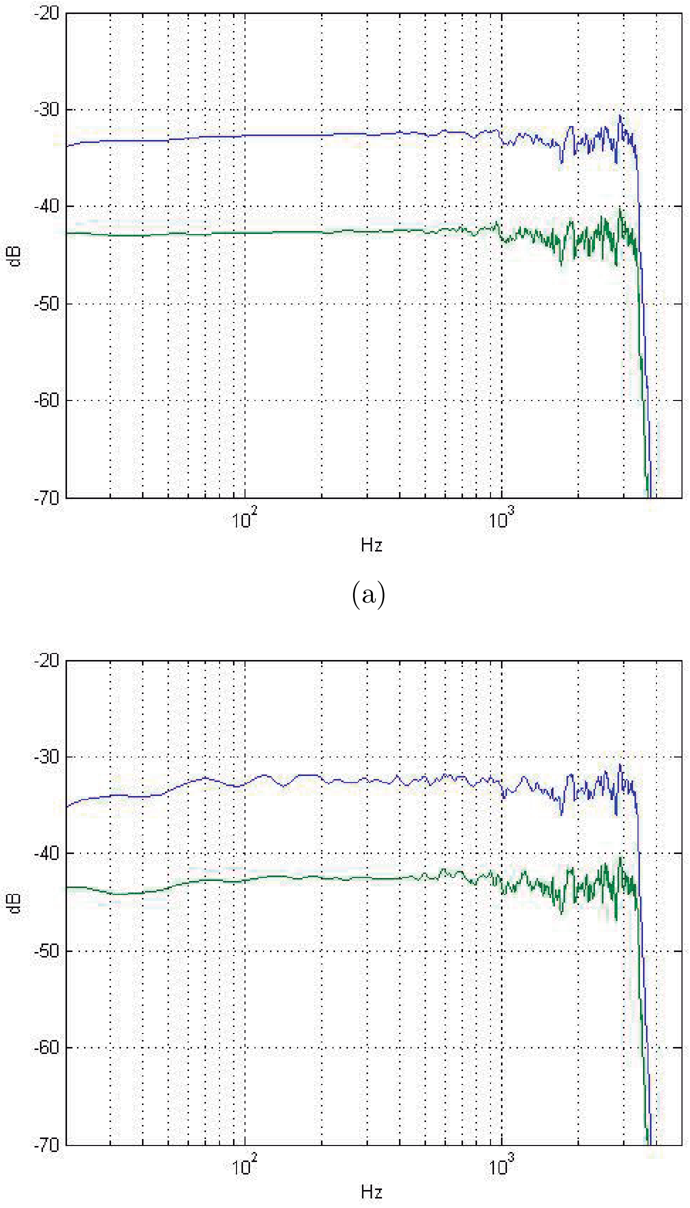


Figure B.5. Frequency response before the compensation algorithm.

In the same way, Fig. B.6 (b) shows the response for the fast deconvolution algorithm using a regularization parameter of 0.01. For a 512 length (upper plot), the filters computed in frequency domain contain peaks and deeps in the whole range of frequencies. These peaks and notches are due to the circular convolution effects. With a inverse filter length of 1024 samples, the results for the fast deconvolution algorithm has a plainer response than before, but it does not achieve the good performance exhibit by the fast Block Toeplitz solvers with 512 samples, figure B.6 (a).

B.6.2 Global inversion performance

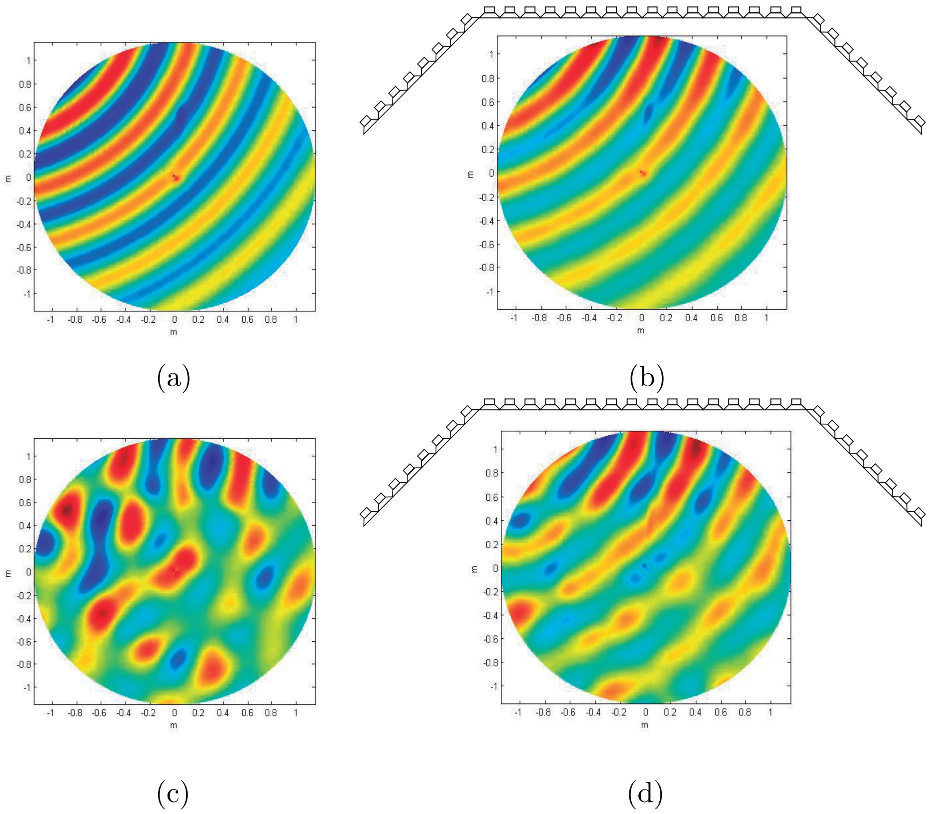
In order to evaluate the acoustic field rendered by the WFS system inside the listening room, a single source has been simulated within this area. Fig. B.7 (a) represents the field rendered by a single real source of 750 Hz. Whereas, the response of Fig. B.7 (b) simulates the field produced with the WFS array in free field, both are very similar. Fig. B.7 (c) shows the same source but into the real room, which is far from an ideal performance. The field obtained this way does not seem as perfect as the obtained in free field conditions due to the room reflections. The field rendered after applying the inverse filter bank is shown in Fig. B.7 (d). In this case, a very similar field to the original one is achieved. The improvement is clearly noticeable.



(a)

(b)

Figure B.6. Frequency response for filters with length of 512 and 1024 samples: (a) after compensation by fast Block Toeplitz solvers and (b) after the compensation by fast deconvolution algorithm.



(a)

(b)

(c)

(d)

Figure B.7. Rendered field by a source signal of 750 Hz: (a) original source in free field conditions, (b) reproduced by WFS in free field, (c) measured field in a real room and (d) after applying the inverse filter bank.

B.6.3 Subjective analysis

In order to prove the efficiency of this bank of filters not only form an objective point of view, a subjective analysis has been carried out.

The original input signal is a singing voice, which has been resampled to 8 kHz before convolving it with the WFS and the inverse filters. For the WFS system, without inversion, the input signal has been convolved with the 32 WFS filters and resampled to 44.1 kHz before rendering it through the soundcard and loudspeakers. For the equalize WFS systems, the 32 WFS filters have been first convolved with the 32×32 inverse filters, and later with the input signal, before interpolating and rendering it.

For the subjective test, a software has been implemented which renders the different audio signals for a given number of loudspeakers. This software adds no delay to the secondary sources (loudspeakers) because it is included in the WFS filters.

The test was presented to 10 spatial audio experts, who listened to the two different audio signals. First, inside the listening area of 2.3 m of diameter and later outside the controlled area, but within the 32 loudspeaker distribution. The aim is twofold: to evaluate if the location of the virtual primary source for the compensate system improves with regard to the WFS system without room compensation. And to study the effect of the equalization outside the listening area, where the rendering is uncontrolled.

For this evaluation, the parameter studied is the *locatedness*, which quantifies the ability of the listener to perceive the spatial position of a source signal, [119]. The questions of the surveys were two: In which direction is the source signal located? How clear do you perceive this location? The answers arranged from: 1-Very bad 2-Bad 3-Regular 4-Well 5-Very well.

Fig. B.8 shows the means and a 95% confidence intervals of the listener surveys for the four cases: inside the listener area with compensation, inside without compensation, outside the listener area with compensation and outside without compensation.

The results show that the perceived location of the source signal is good in all cases. Although it is slightly better for the first case, inside the listener area and with compensation, these results show that the improvement is not so significant. This can be due to the fact that the experiment was

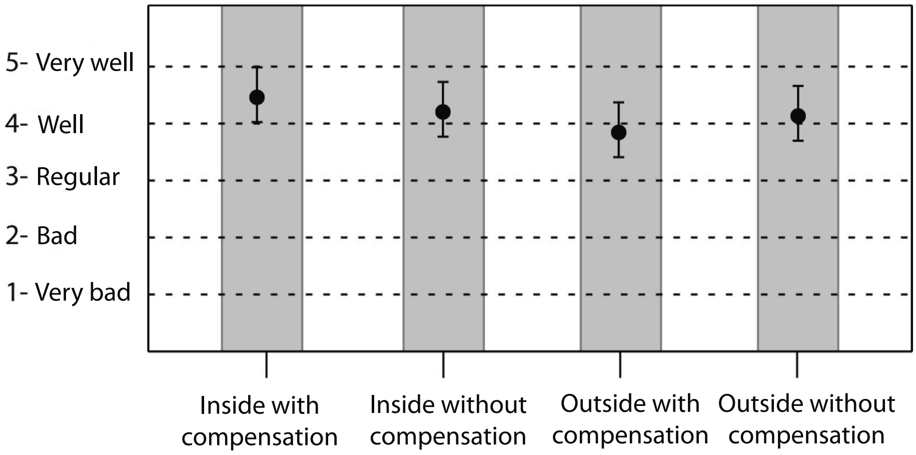


Figure B.8. Subjective analysis: means and a 95% confidence intervals.

carried out inside a room acoustically conditioned and the echoes of the room do not impair greatly the WFS rendering system.

Bibliography

[1] H.Kuttruf, *Room acoustics*, 4th ed., ser. 0-419-24580-4. London: Spon Press, 2000.

[2] L. Cremer and H. Müller, *Principles and Applications of Room Acoustics.* London: Applied Science, 1982.

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

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

[5] S. Petrausch, S. Spors, and R. Rabenstein, "Simulation and visualization of room compensation for wave field synthesis with the functional transformation method," *119th AES Convention*, 2005.

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

[7] W. C. Sabine, *Collected papers on acoustics.* Harvard university press, 1922.

[8] S. T. Neely and J. B. Allen, "Invertibility of a room impulse response," *J. of Acoustic Society of America*, vol. 66, no. 1, pp. 165–169, July 1979.

[9] B. Widrow and M. Bilello, "Adaptive inverse control," in *Intelligent Control, 1993., Proceedings of the 1993 IEEE International Symposium on.* IEEE, 1993, pp. 1–6.

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

[11] C. Paleologu, J. Benesty, and S. Ciochina, "Sparse adaptive filters for echo cancellation," *Synthesis Lectures on Speech and Audio Processing*, vol. 6, no. 1, pp. 1–124, 2010. [Online]. Available: http://dx.doi.org/10.2200/S00289ED1V01Y201006SAP006

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

[13] W. A. Murray, "Numerical methods for unconstrained optimization," 1972.

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

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

[16] J. Arenas-Garcıa, M. de Diego, L. A. Azpicueta-Ruiz, M. Ferrer, and A. González, "Combinations of proportionate adaptive filters in acoustics: An application to active noise control," in *Proc. EURASIP European Signal Processing Conference (EUSIPCO)*, 2011.

[17] L. Fuster, M. De Diego, M. Ferrer, A. González, and G. Pinero, "A biased multichannel adaptive algorithm for room equalization," in *Proc. EURASIP European Signal Processing Conference (EUSIPCO)*, 2012, pp. 1344–1348.

[18] M. Martinez-Ramon, J. Arenas-Garcia, A. Navia-Vazquez, and A. Figueiras-Vidal, "An adaptive combination of adaptive filters for

plant identification," in *Digital Signal Processing, 2002. DSP 2002. 2002 14th International Conference on*, vol. 2, 2002, pp. 1195–1198 vol.2.

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

[20] M. Ferrer, M. De Diego, A. González, and G. Pińero, "Steady-state mean square performance of the multichannel filtered-x affine projection algorithm," *IEEE Trans. Signal Process.*, vol. 60, no. 6, pp. 2771–2785, 2012.

[21] M. Ferrer, M. de Diego, A. González, and G. Pińero, "Convex combination of adaptive filters for ANC," *in Proc. 16th Int. Cong. Sound and Vibration (ICSV)*, Jul. 2009, cracow (Poland).

[22] M. T. Silva and V. H. Nascimento, "Improving the tracking capability of adaptive filters via convex combination," *Signal Processing, IEEE Transactions on*, vol. 56, no. 7, pp. 3137–3149, 2008.

[23] A. Gonzalo-Ayuso, M. Silva, V. Nascimento, and J. Arenas-Garcia, "Improving sparse echo cancellation via convex combination of two nlms filters with different lengths," in *Machine Learning for Signal Processing (MLSP), 2012 IEEE International Workshop on*, Sept 2012, pp. 1–6.

[24] Y. Zhang and J. A. Chambers, "Convex combination of adaptive filters for a variable tap-length LMS algorithm," *Signal Processing Letters, IEEE*, vol. 13, no. 10, pp. 628–631, 2006.

[25] S. Goetze, M. Kallinger, A. Mertins, and K.-D. Kammeyer, "System identification for multi-channel listening-room compensation using an acoustic echo canceller," in *Hands-Free Speech Communication and Microphone Arrays, 2008. HSCMA 2008.* IEEE, 2008, pp. 224–227.

[26] S. Cecchi, A. Primavera, F. Piazza, and A. Carini, "An adaptive multiple position room response equalizer," in *Proc. EURASIP European Signal Processing Conference (EUSIPCO)*, September 2011.

[27] S. Cecchi, A. Carini, A. Primavera, and F. Piazza, "An adaptive multiple position room response equalizer in warped domain," in *Proc. EURASIP European Signal Processing Conference (EUSIPCO)*, August 2012.

[28] S. Cecchi, L. Romoli, A. Carini, and F. Piazza, "A multichannel and multiple position adaptive room response equalizer in warped domain: Real-time implementation and performance evaluation," *Applied Acoustics*, vol. 82, pp. 28 – 37, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0003682X14000413

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

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

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

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

[33] M. Schneider and W. Kellermann, "Adaptive listening room equalization using a scalable filtering structure in thewave domain," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 2012, pp. 13–16.

[34] D. S. Talagala, W. Zhang, and T. D. Abhayapala, "Efficient multichannel adaptive room compensation for spatial soundfield reproduction using a modal decomposition," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 22, no. 10, pp. 1522–1532, 2014.

[35] E. Bjarnason, "Analysis of the filtered-x lms algorithm," *Speech and Audio Processing, IEEE Transactions on*, vol. 3, no. 6, pp. 504–514, Nov 1995.

[36] 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. accepted for publication, pp. 1–23, 2003.

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

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

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

[40] M. Dentino, J. McCool, and B. Widrow, "Adaptive filtering in the frequency domain," *Proceedings of the IEEE*, vol. 66, no. 12, pp. 1658–1659, December 1978.

[41] S. S. Soliman and M. D. Srinath, "Continuous and discrete signals and systems," *Englewood Cliffs, NJ, Prentice Hall, 1990, 523 p.*, vol. 1, 1990.

[42] E. C. Ifeachor and B. W. Jervis, *Digital signal processing: a practical approach.* Pearson Education, 2002.

[43] J. J. Shynk, "Frequency-domain multirate adaptive filtering," *IEEE Signal Processing Magazine*, vol. 9, no. 1, pp. 14–37, January 1992.

[44] G. A. Clark, S. K. Mitra, and S. R. Parker, "Block implementation of adaptive digital filters," *IEEE Transactions on Circuits and Systems*, vol. 28, no. 6, pp. 584–592, June 1981.

[45] G. A. Clark, S. R. Parkerand, and S. K. Mitra, “A unified approach to time- and frequency-domain realization of FIR adaptive digital filters,” *IEEE Transaction on Acoustics, Speech and Signal Proceesing*, vol. 31, no. 5, pp. 1073–1083, October 1983.

[46] S. J. Elliott and P. A. Nelson, “Multiple-point equalization in a room using adaptive digital filters,” *Journal of the AES*, vol. 37, no. 11, pp. 899–907, 1989.

[47] W. Klippel, “Nonlinear large-signal behavior of electrodynamic loudspeakers at low frequencies,” *Journal of the AES*, vol. 40, no. 6, pp. 483–496, 1992.

[48] F. Agerkvist, “Modelling loudspeaker non-linearities,” in *32nd Int. Conf. of AES*. AES, 2007.

[49] F. Gao and W. Snelgrove, “Adaptive linearization of a loudspeaker,” in *Int. Conf. on Acoustics, Speech, and Signal Processing*. IEEE, 1991, pp. 3589–3592.

[50] M. Tsujikawa, T. Shiozaki, Y. Kajikawa, and Y. Nomura, “Identification and elimination of second-order nonlinear distortion of loudspeaker systems using Volterra filter,” in *IEEE Int. Symposium on Circuits and Systems*, vol. 5. IEEE, 2000, pp. 249–252.

[51] J. Patra, R. Pal, B. Chatterji, and G. Panda, “Identification of nonlinear dynamic systems using functional link artificial neural networks,” *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 29, no. 2, pp. 254–262, Apr 1999.

[52] Y. Pao, “Adaptive pattern recognition and neural networks,” 1989.

[53] G. Sicuranza and A. Carini, “A generalized FLANN filter for nonlinear active noise control,” *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 19, no. 8, pp. 2412–2417, 2011.

[54] A. Carini and G. Sicuranza, “A new class of flann filters with application to nonlinear active noise control,” in *Proc. EURASIP European Signal Processing Conference (EUSIPCO)*, August 2012.

[55] A. Carini and G. Sicuranza, “Fourier nonlinear filters,” *Signal Processing*, vol. 94, no. 0, pp. 183 – 194, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0165168413002363

[56] V. Mathews and G. Sicuranza, *Polynomial signal processing.* Wiley-Interscience, 2000.

[57] L.-Z. Tan and J. Jiang, "Adaptive second-order volterra delay filter," *Electronics Letters*, vol. 32, no. 9, pp. 807–809, 1996.

[58] K. Iwai and Y. Kajikawa, "Linearization of dynamic loudspeaker system using third-order nonlinear IIR filter," in *Proceedings of EUSIPCO.* IEEE, 2012, pp. 1970–1974.

[59] D. Comminiello, M. Scarpiniti, R. Parisi, and A. Uncini, "A functional link based nonlinear echo canceller exploiting sparsity," in *Proc. Int. Workshop Acoust. Echo, Noise Control (IWAENC'10)*, 2010.

[60] D. Comminiello, M. Scarpiniti, L. Azpicueta-Ruiz, J. Arenas-Garcia, and A. Uncini, "Functional link adaptive filters for nonlinear acoustic echo cancellation," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 21, no. 7, pp. 1502–1512, July 2013.

[61] T. Sunitha and R. Malar, "Nonlinear acoustic echo cancellation using adaptive filters," in *Control, Instrumentation, Communication and Computational Technologies (ICCICCT), 2014 International Conference on.* IEEE, 2014, pp. 1400–1405.

[62] D. Zhou and V. DeBrunner, "Efficient adaptive nonlinear filters for nonlinear active noise control," *IEEE Trans. on Circuits and Systems*, vol. 54, no. 3, pp. 669–681, 2007.

[63] G. L. Sicuranza and A. Carini, "Filtered-x affine projection algorithm for multichannel active noise control using second-order volterra filters," *IEEE Signal Processing Letters*, vol. 11, no. 11, pp. 853–857, 2004.

[64] N. V. George and G. Panda, "On the development of adaptive hybrid active noise control system for effective mitigation of nonlinear noise," *Signal Processing*, vol. 92, no. 2, pp. 509–516, 2012.

[65] N. V. George and G. Panda, Ganapati, "Advances in active noise control: A survey, with emphasis on recent nonlinear techniques," *Signal processing*, vol. 93, no. 2, pp. 363–377, 2013.

[66] B. S. Nollett and D. L. Jones, "Nonlinear echo cancellation for hands-free speakerphones," *Urbana*, vol. 51, p. 61801, 1997.

[67] A. Uncini, A. Nalin, and R. Parisi, "Acoustic echo cancellation in the presence of distorting loudspeakers," in *Signal Processing Conference, 2002 11th European.* IEEE, 2002, pp. 1–4.

[68] F. Küch and W. Kellermann, "Nonlinear acoustic echo cancellation," *Topics in Acoustic Echo and Noise Control*, pp. 205–257, 2006.

[69] L. Tan and J. Jiang, "Adaptive volterra filters for active control of nonlinear noise processes," *Signal Processing, IEEE Transactions on*, vol. 49, no. 8, pp. 1667–1676, 2001.

[70] A. Carini and G. L. Sicuranza, "Filtered-x affine projection algorithms for active noise control using volterra filters," *EURASIP Journal on Applied Signal Processing*, vol. 2004, pp. 1841–1848, 2004.

[71] G. L. Sicuranza and A. Carini, "A multichannel hierarchical approach to adaptive volterra filters employing filtered-x affine projection algorithms," *Signal Processing, IEEE Transactions on*, vol. 53, no. 4, pp. 1463–1473, 2005.

[72] A. Fermo, A. Carini, and G. L. Sicuranza, "Simplified volterra filters for acoustic echo cancellation in gsm receivers," in *Signal Processing Conference, 2000 10th European.* IEEE, 2000, pp. 1–4.

[73] M. Zeller and W. Kellermann, "Evolutionary adaptive filtering based on competing filter structures," in *Signal Processing Conference, 2011 19th European.* IEEE, 2011, pp. 1264–1268.

[74] M. Zeller, L. Azpicueta-Ruiz, J. Arenas-García, W. Kellermann *et al.*, "Adaptive volterra filters with evolutionary quadratic kernels using a combination scheme for memory control," *Signal Processing, IEEE Transactions on*, vol. 59, no. 4, pp. 1449–1464, 2011.

[75] Y. Lim, Y. Cho, I. Cha, and D. Youn, "An adaptive nonlinear prefilter for compensation of distortion in nonlinear systems," *IEEE Trans. Signal Process.*, vol. 46, no. 6, pp. 1726–1730, 1998.

[76] Y. Kajikawa and Y. Nomura, "Online design of nonlinear inverse systems by adaptive volterra filter," *Electronics and Communications in Japan*, vol. 83, no. 9, pp. 46–56, 2000.

[77] S. Bharitkar and C. Kyriakakis, “New factors in room equalization using a fuzzy logic approach,” *PREPRINTS-AUDIO ENGINEERING SOCIETY*, 2001.

[78] F. Rumsey, *Spatial Audio*, ser. Music Technology Series. Focal Press, 2003.

[79] A.J.Berkhout, “A holographic approach to acoustic control,” *Journal of the Audio Engineering Society*, vol. 36, no. 12, pp. 977–995, December 1988.

[80] A. Berkhout, D. de Vries, and P. Vogel, “Acoustic control by wave field synthesis,” *J. of Acoustic Society of America*, vol. Vol. 93(5), pp. 2764–2778, 1993.

[81] E.N.G.Verheijen, “Sound reproduction by wave field synthesis,” Ph.D. dissertation, Delft Univeristy of Technology, 1997.

[82] L. Fuster, J. López, A. González, and P. Faus, “Time and frequency domain room compensation applied to wave field synthesis,” *8th Int. Conference on Digital audio Effects (DAFx’05)*, Septiembre 2005.

[83] J. J. López, A. González, and L. Fuster, “Room compensation in wave field synthesis by means of multichannel inversion,” in *Applications of Signal Processing to Audio and Acoustics, 2005. IEEE Workshop on.* IEEE, 2005, pp. 146–149.

[84] E. Corteel, “Equalization in an extended area using multichannel inversion and wave field synthesis,” *J. Audio Engineering Society*, vol. 54, no. 12, December 2006.

[85] P. A. Nelson, F. Orduna-Bustamante, and H. Hamada, “Multichannel signal processing techniques in the reproduction of sound,” *Journal of Acoustic Society of America*, vol. 44, pp. 973–989, 1996.

[86] S. Spors and R. Rabenstein, “Evaluation of the circular harmonics decomposition for wdaf-based active listening room compensation,” in *28th AES Conference: The Future of Audio Technology–Surround and Beyond*, 2006, pp. 134–149.

[87] P.-A. Gauthier and A. Berry, “Adaptive wave field synthesis with independent radiation mode control for active sound field reproduction:

Theory)," *The Journal of the Acoustical Society of America*, vol. 119, no. 5, pp. 2721–2737, 2006.

[88] A. Carini, S. Cecchi, F. Piazza, I. Omiciuolo, and G. Sicuranza, "Multiple position room response equalization in frequency domain," *IEEE Trans. Audio, Speech, and Language Process.*, vol. 20, no. 1, pp. 122–135, 2012.

[89] J. Lorente, M. Ferrer, M. de Diego, and A. Gonzalez, "Gpu based implementation of multichannel adaptive room equalization," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on.* IEEE, 2014, pp. 7535–7539.

[90] Y. Huang and J. Benesty, "A class of frequency-domain adaptive approaches to blind multichannel identification," *IEEE Trans. Signal Process.*, vol. 51, no. 1, pp. 11 – 24, Jan. 2003.

[91] W. Klippel, "Speaker auralization-subjective evaluation of nonlinear distortion," *PREPRINTS-AUDIO ENGINEERING SOCIETY*, 2001.

[92] S. G. Norcross, G. A. Soulodre, and M. C. Lavoie, "Subjective investigations of inverse filtering," *J. Audio Engineering Society*, vol. 52, no. 10, pp. 1003–1028, October 2004.

[93] L. Fúster, J. J. López, A. González, and M. Cobos, "análisis objetivo y subjetivo de la compensación de salas mediante técnicas wave-field synthesis," *ITECKNE: Innovación e Investigación en Ingeniería*, vol. 6, no. 2, pp. 11–17, 2009.

[94] S. Goetze, E. Albertin, M. Kallinger, A. Mertins, and K.-D. Kammeyer, "Quality assessment for listening-room compensation algorithms," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on.* IEEE, 2010, pp. 2450–2453.

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

[96] L. A. Azpicueta-Ruiz, A. R. Figueiras-Vidal, and J. Arenas-Garcia, "A normalized adaptation scheme for the convex combination of two

adaptive filters," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on.* IEEE, 2008, pp. 3301–3304.

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

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

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

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

[101] L. A. Azpicueta-Ruiz, M. Zeller, A. R. Figueiras-Vidal, J. Arenas-García, and W. Kellermann, "Improved acoustic echo cancellation for low snr based on blockwise combination of filters."

[102] N. George and A. González, "Convex combination of nonlinear adaptive filters for active noise control," *Applied Acoustics*, vol. 76, no. 0, pp. 157 – 161, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0003682X13001916

[103] D. Das and G. Panda, "Active mitigation of nonlinear noise processes using a novel filtered-s LMS algorithm," *IEEE Trans. on Speech and Audio Processing*, vol. 12, no. 3, pp. 313–322, 2004.

[104] G. L. Sicuranza and A. Carini, "A new recursive controller for nonlinear active noise control," in *Image and Signal Processing and Analy-*

sis (ISPA), 2013 8th International Symposium on. IEEE, 2013, pp. 633–638.

[105] C. Contan, M. Zeller, W. Kellermann, and M. Topa, "Excitation-dependent stepsize control of adaptive volterra filters for acoustic echo cancellation," in *Proceedings of EUSIPCO.* IEEE, 2012, pp. 604–608.

[106] W. J. Klippel, "Adaptive inverse control of weakly nonlinear systems," in *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, vol. 1. IEEE, 1997, pp. 355–358.

[107] V. DeBrunner and D. Zhou, "Active noise control with weak nonlinearities in the secondary path," in *Circuits and Systems, 2002. MWSCAS-2002. The 2002 45th Midwest Symposium on*, vol. 3, Aug 2002, pp. III–17–20 vol.3.

[108] L. Azpicueta-Ruiz, M. Zeller, A. R. Figueiras-Vidal, W. Kellermann, J. Arenas-Garcia *et al.*, "Enhanced adaptive volterra filtering by automatic attenuation of memory regions and its application to acoustic echo cancellation," *Signal Processing, IEEE Transactions on*, vol. 61, no. 11, pp. 2745–2750, 2013.

[109] J. C. Lee and C. K. Un, "Performance analysis of frequency-domain block LMS adaptive digital filters," *IEEE Transactions on Circuits and Systems*, vol. 36, no. 2, pp. 173–189, February 1989.

[110] M. Bouchard and B. Paillard, "A transform domain optimization to increase the convergence speed of the multichannel filtered-x least-mean-square algorithm," *J. of Acoustic Society of America*, vol. 100, no. 5, pp. 3203–3214, November 1996.

[111] S. Goetze, M. Kallinger, A. Mertins, and K.-D. Kammeyer, "A decoupled filtered-x LMS algorithm for listening-room compensation," in *Proc. Int. Workshop on Acoustic Echo and Noise Control (IWAENC)*, SEptember 2008.

[112] A. Leite and A. J. S. Ferreira, "An improved adaptive room equalization in the frequency domain," *118th AES Convention*, May 2005.

[113] J. López and A. González, “Two steps levinson algorithm for time domain multichannel deconvolution,” *Electronics Letters*, vol. 36 (7), pp. 686–688, 2000.

[114] P.Vogel, “Application of wave field synthesis in room acoustics,” Ph.D. dissertation, Delft University of Technology, 1993.

[115] E.W.Start, “Direct sound enhancement by wave field synthesis,” Ph.D. dissertation, Delft University of Technology, 1997.

[116] P. M. Clarkson, “Optimal and adaptive signal processing,” *CRC Press*, 1993.

[117] I. C. Gohberg and A. A. Semencul, “On the inversion of finite toeplitz matrices and their continuous analogs,” *Mathematics Issted.*, vol. 2, pp. 201–233, 1972.

[118] A. González, P. Zuccarello, M. de Diego, and G. P. nero, “Simultaneous measurement of multichannel acoustic systems,” *Journal of the Audio Engineering Society*, vol. 52, no. 1/2, pp. 26–42, January/February 2004.

[119] H. Wittek, F. Rumsey, and G. Theile, “Perceptual enhancement of wavefield synthesis by stereophonic means,” *Journal of the Audio Engineering Society*, vol. 55, no. 9, pp. 723–751, 2007.