SAPIENZA
UNIVERSITÀ DI ROMA

SAPIENZA UNIVERSITY OF ROME
FACULTY OF ENGINEERING

ADAPTIVE ALGORITHMS FOR INTELLIGENT ACOUSTIC INTERFACES

DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY IN INFORMATION AND COMMUNICATION ENGINEERING
XXIV CYCLE

Supervisor	Candidate
Prof. Aurelio Uncini	Dr. Danilo Comminiello

Rome, Italy
December 2011

To Alessia and my dear family

—Reading maketh a full man;
conference a ready man;
and writing an exact man.
Francis Bacon

Abstract

MODERN speech communications are evolving towards a new direction which involves users in a more perceptive way. That is the *immersive experience*, which may be considered as the "last-mile" problem of telecommunications.

One of the main feature of immersive communications is the *distant-talking*, i.e. the hands-free (in the broad sense) speech communications without body-worn or tethered microphones that takes place in a multisource environment where interfering signals may degrade the communication quality and the intelligibility of the desired speech source.

In order to preserve speech quality *intelligent acoustic interfaces* may be used. An intelligent acoustic interface may comprise multiple microphones and loudspeakers and its peculiarity is to model the acoustic channel in order to adapt to user requirements and to environment conditions. This is the reason why intelligent acoustic interfaces are based on adaptive filtering algorithms.

The acoustic path modelling entails a set of problems which have to be taken into account in designing an adaptive filtering algorithm. Such problems may be basically generated by a linear or a nonlinear process and can be tackled respectively by linear or nonlinear adaptive algorithms.

In this work we consider such modelling problems and we propose novel effective adaptive algorithms that allow acoustic interfaces to be robust against any interfering signals, thus preserving the perceived quality of desired speech signals.

As regards *linear adaptive algorithms*, a class of adaptive filters based on the sparse nature of the acoustic impulse response has been recently proposed. We adopt such class of adaptive filters, named *proportionate adaptive filters*, and derive a general framework from which it is possible to derive any linear adaptive algorithm. Using such framework we also propose some efficient proportionate adaptive algorithms, expressly designed to tackle problems of a linear nature.

On the other side, in order to address problems deriving from a nonlinear process, we propose a novel filtering model which performs a nonlinear transformations by means of *functional links*. Using such nonlinear model, we propose *functional link adaptive filters* which provide an efficient solution to the modelling of a nonlinear acoustic channel.

Finally, we introduce robust filtering architectures based on *adaptive combinations of filters* that allow acoustic interfaces to more effectively adapt to environment conditions, thus providing a powerful mean to immersive speech communications.

Acknowledgments

THE *accomplishment of the Ph.D. degree is cause of huge gratification to me so I would like to share my contentment with all the people who help me to attain this goal.*

First and foremost, I would like to express my sincere appreciation and gratitude to my advisor Prof. Aurelio Uncini for his valuable guidance and continuous support throughout the entire project. He transmitted his passion for research to me and taught me how to strive for excellence in my professional pursuits.

I am also in debt to Prof. Michele Scarpiniti for being my well-wisher. I would like to thank him for his invaluable advices, besides being an exemplary workmate he is firstly a friend.

Special thanks to Dr. Mauro Falcone, I am grateful to him for giving me an opportunity to work with him and to confront myself with an important corporate reality, such as the "Fondazione Ugo Bordoni".

My sincere thanks also go to my group member, Prof. Raffaele Parisi for the insightful discussions and help along the way.

I am also thankful with Prof. Jerónimo Arenas García who has incisively increased my professional skills during my visiting period at "Universidad Carlos III de Madrid", thus resulting decisive for my thesis work. It was a great pleasure to work with him both from a professional and a personal point of view. I would like to thank him and Dr. Luis Azpicueta Ruiz for their precious welcome to Madrid.

I would like to thank my dissertation committee members: Prof. Giovanni Iacovitti, Prof. Antonello Rizzi, Prof. Marco Listanti for taking the time to serve on my committee and for their helpful suggestions.

A fundamental role has been played by my flatmates and friends, whose presence is decisive not only in my work but above all in my life.

My best thanks are to Alessia who is continuously at my side, cheering me up in hard times and fortunately sharing with me joys and gratifications. This work is dedicated to her.

To make this acknowledgment complete, a special thanks to my family for their unconditional love and support, which added the best essence to my work.

Contents

List of Figures

List of Tables

List of Acronyms

AEC	Acoustic Echo Cancellation
AI	Artificial Intelligence
AIR	Acoustic Impulse Response
ANC	Adaptive Noise Canceller
ANN	Artificial Neural Network
APA	Affine Projection Algorithm
AZK	All-Zero Kernel
BCFLAF	Block-Based Collaborative FLAF
BM	Blocking Matrix
CANC	Combined Adaptive Noise Canceller
CFLAF	Collaborative FLAF
DSB	Delay-and-Sum Beamformer

DSP	Digital Signal Processing
DTD	Double Talk Detector
EG±	Exponentiated Gradient
EMSE	Excess Mean Square Error
ERLE	Echo Return Loss Enhancement
FEB	Functional Expansion Block
FIR	Finite Impulse Response
FLAF	Functional Link Adaptive Filter
FLANN	Functional Link Artificial Neural Network
GSC	Generalized Sidelobe Canceller
IAI	Intelligent Acoustic Interface
IIR	Infinite Impulse Response
IPAPA	Improved PAPA
IPNLMS	Improved Proportionate NLMS
ISO	International Organization for Standardization
LMS	Least Mean Square
LSI	Linear Shift-Invariant
MIMO	Multiple-Input Multiple Output
MISO	Multiple-Input Single-Output
MLP	Multi-Layer Perceptron

MMSE	Minimum Mean Square Error
MSE	Mean Square Error
NAEC	Nonlinear Acoustic Echo Cancellation
NAPA	Natural Affine Projection Algorithm
NLMS	Normalized Least Mean Square
NNG	Normalized Natural Gradient
PAPA	Proportionate APA
PBAPA	Proportionate Block APA
PNLMS	Proportionate NLMS
QAM	Quadrature Amplitude Modulation
RBF	Radial Basis Function
RLS	Recursive Least Squares
SFLAF	Split FLAF
SISO	Single-Input Single-Output
SNR	Signal to Noise Ratio
ULA	Uniform Linear Array
VAD	Voice Activity Detection
VF	Volterra Filter
VSS	Variable Step Size

Part I

INTRODUCTION

> *—My work consists of two parts: of the one which is here,*
> *and of everything which I have not written.*
> *And precisely this second part is the important one.*
> ***Ludwig Wittgenstein***

1 Introduction and Outline

Contents

1.1 MOTIVATIONS

Intelligent acoustic interfaces (IAIs) for hands-free speech communications are based on the modelling of acoustic paths and on the perception of complex sounds. In the development of such communication systems, many research areas intersect and cross-feed themselves, among which are: noise reduction, speech enhancement, acoustic echo cancellation, nonlinear channel modelling, multichannel acoustic modelling, source localization and tracking, blind source separation.

In such research context, matter of primary importance is the study of adaptive filtering algorithms and architectures [120]. Capabilities of such filtering structures to adapt to acoustic environments is that makes an acoustic interface intelligent. Moreover, adaptive filter performance bears on the quality of processed acoustic signals.

Among the foremost acoustic applications in which adaptive filtering plays a leading role are those on acoustic channel modelling, such as *acoustic echo cancellation* (AEC). The phenomenon of acoustic echo occurs when a delayed (and possibly distorted) version of the speech signal reproduced by a loudspeaker is acquired by a microphone and reflected back to remote user. An acoustic echo canceller aims at estimating the *acoustic impulse response* (AIR), i.e. modelling the acoustic path, in order to subtract the estimated echo signal from the microphone signal.

Therefore, the acoustic channel modelling represents an exhaustive issue in hands-free speech communications since it includes a set of problems common to the whole sector of acoustic scene analysis: the estimate of the impulse response, the presence of nonstationary elements in the environment, the presence of unwanted interfering signals, the presence of nonlinearities [12]. Such phenomena strongly degrade the perceived quality of the speech signal and might be tackled using signal processing techniques, that are pivotal in restoring the perceived intelligibility in a speech communication. This is the reason why the proposed research work mainly deals with applications on acoustic channel modelling, and in particular on AEC, in order to develop novel adaptive filtering techniques, which might also be used in other *distant-talking* applications.

Regarding the research in AEC, significant advances were achieved in the linear case, in which capabilities of adaptive filters have been exploited in order to model AIRs at best. In that sense significant results have been recently achieved for applications in hands-free speech communication in reverberant environments and in presence of interfering signals [12, 100], factors that cannot be neglected in immersive communications. However, similar results have not been reached yet in the nonlinear case.

The nonlinear case is characterized by the presence of distortions in the acoustic path that are funneled in the echo signal and cause a performance decrease and an even worse decrease of the perceived quality of information. Nonlinearities very often occur in acoustic applications since they are generated by loudspeakers or by the vibrations of audio devices' enclosures [147]. Therefore, nowadays, it is difficult to disregard echo cancellers that take into account nonlinearities, also due to a large spreading of low-cost audio devices, thus having low-quality electronic components and materials which may introduce even strong distortions.

Among the most popular nonlinear acoustic echo cancellers of recent years, stand out those based on adaptive Volterra filters [138, 23]. However, such nonlinear acoustic echo cancellers involve computational costs that are definitely larger than conventional echo cancellers (i.e. linear echo cancellers) and, moreover, they may provide worse performance compared to the last ones. That affects also the strategies of many companies that provide teleconferencing services, which often choose to drop the use of nonlinear echo cancellers even at the expense of communication quality. On the other side, these are also the main motivations that underpin the proposed research project.

1.2 SCOPE OF THE WORK

The development of adaptive algorithms for intelligent acoustic interfaces is based on high-complexity scenarios which take into account several phenomena that may degrade the speech intelligibility in a hands-free speech communication. We start from an analysis of such interfering phenomena that may be essentially labelled as linear or nonlinear events. Such division allows to design *ad hoc* adaptive algorithms, thus making acoustic intelligent interfaces robust against interfering signals.

Regarding the acoustic channel modelling in the linear case it is sufficient to investigate about adaptive models that are statistically robust. However, in order to recreate accurately an acoustic scene free from any interference, noise and unwanted signals, it is advisable to perform a nonlinear processing of acquired information that is able to learn from the environment in a supervised or unsupervised way. In both the cases, linear and nonlinear, automatic learning and continuous adaptivity are fundamental elements to satisfy quality requirements of speech communication [12].

In order to tackle linear interfering signals, we deal with a recently proposed filtering technique that is based on *proportionate adaptive filters* [100]. This family of algorithms exploits sparsity constraints that are typical of AIRs, thus yielding a performance improvement which is able to reduce the limits posed by acoustic environments. The investigation about such family of algorithms bears to the formulation of a framework for the derivation of (linear) adaptive filters and to the development of efficient proportionate adaptive algorithms for immersive speech communication.

On the other side, in order to tackle nonlinearities in acoustic channel modelling, we propose a novel nonlinear filtering model based on *functional links*. From such nonlinear model we develop some algorithms and architectures on purpose of *nonlinear acoustic echo cancellation* (NAEC). The main idea which underpins such *functional link adaptive filters* is that of estimating and modelling nonlinearities introduced in the echo path by the environment and interfering sources, and then cancelling them, thus improving the perceived quality of acoustic information.

Moreover, both in linear and nonlinear cases, the proposed adaptive algorithms are used to form more complex filtering architectures based on the *adaptive combination of filters*. Such architectures result more robust against several kinds of adverse environment conditions compared to conventional filtering techniques.

1.3 ORGANIZATION

The proposed research project is structured in three main parts: the first one dealing with linear adaptive algorithms, the second one with nonlinear adaptive algorithms and the last one dealing with robust filtering architectures. An introducing part is also added at the beginning of the work, as well as a conclusive part is added at the end. In detail, this dissertation is organized as follows:

Part I introduces some preliminary basics.

Chapter 1 describes the motivation and the scope of our proposal.

Chapter 2 introduces intelligent acoustic interfaces and their application in immersive speech communications.

Chapter 3 explains the formulation of main problems in hands-free speech communications that we aim at tackling with adaptive algorithms.

Part II deals with adaptive algorithms designed to address those problems classified as linear.

Chapter 4 introduces a brief view on the theory of adaptive filtering.

Chapter 5 introduces proportionate adaptive algorithms according to the proposed general framework.

Chapter 6 describes by means of simulations the most important features of the proportionate adaptive algorithms introduced in the previous chapter.

Part III deals with adaptive algorithms designed to tackle the presence of nonlinearities in the acoustic channel.

Chapter 7 formulates the problem of nonlinearities which cause an important limitation to the achievable speech quality.

Chapter 8 introduces a new class of nonlinear algorithms, the functional link adaptive filters, whose structure is based on Hammerstein model.

Chapter 9 describes some variants of functional link adaptive filters properly designed for nonlinear acoustic echo cancellation.

Part IV introduces more complex architectures based on adaptive combination of filters to increase robustness against adverse acoustic environments.

Chapter 10 introduces intelligent circuits based on the adaptive combination of filters.

Chapter 11 describes combined architectures for speech enhancement in multisource environments.

Chapter 12 describes collaborative architectures for nonlinear acoustic echo cancellation.

Part V draws our conclusions.

Chapter 13 concludes the work and introduces possible future perspectives.

1.4 NOTATION

In this dissertation, matrices are represented by boldface capital letters and vectors are denoted by boldface lowercase letters. Time-varying vectors and matrices show discrete-time index as a subscript index, while in time-varying scalar elements the time index is denoted in square brackets. A regression vector is represented as $\mathbf{x}_n \in \mathbb{R}^M = \begin{bmatrix} x[n] & x[n-1] & \ldots & x[n-M+1] \end{bmatrix}^T$, where M is the overall vector length and $x[n-i]$ is individual entry at the generic time instant $n-i$. On the other side, a snap-shot vector, which includes a number Q of different contributions at n-th time instant, is represented as

$\mathbf{y}[n] \in \mathbb{R}^Q = \begin{bmatrix} y_0[n] & y_1[n] & \ldots & y_{Q-1}[n] \end{bmatrix}^T$. However, a generic coefficient vector, in which all elements depend on the same time instant, is denoted as $\mathbf{w}_n \in \mathbb{R}^M = \begin{bmatrix} w_0[n] & w_1[n] & \ldots & w_{M-1}[n] \end{bmatrix}^T$, where $w_i[n]$ is the generic i-th individual entry at n-th time instant. When the coefficient vector is a realization of a time-invariant process the time index is omitted. All vectors are represented as column vectors.

2

Intelligent Acoustic Interfaces

Contents

2.1 WHAT IS AN INTELLIGENT ACOUSTIC INTERFACE?

In order to formulate a comprehensive definition for the term "intelligent acoustic interface" (IAI), it is necessary first to know what an acoustic interface is and what makes an acoustic interface.

2.1.1 Acoustic interfaces

An *acoustic interface* provides a means to exchange acoustic information between two or more entities through an acoustic signal processing. More exactly, an acoustic interface is the front-end of a processing system of audio and speech signals aiming at the extraction and the reproduction of acoustic information. An acoustic interface is generally composed of a microphone array and one or more loudspeakers, as depicted in Fig. 2.1.

To control noise, reverberation, and competing speech, microphone array systems are generally more powerful than a single microphone [45]. Based on how the microphones are arranged, these systems have two basic forms: organized and distributed arrays [66]. In an *organized array*, the sensors are arranged to form a particular geometry (such as a line, a circle, or a sphere) in which each sensor's position with reference to a common point is known. These sensors spatially sample the sound field and are required to have the same sensitivity. In comparison, a *distributed array* consists of randomly placed microphones. It offers the advantage of logistic convenience during installation and later operations. Typically, distributed arrays have a large number of elements forming a large sensor network. The microphone positions and the pattern of the array are usually not known, and a uniform response among the microphones cannot be presumed beforehand.

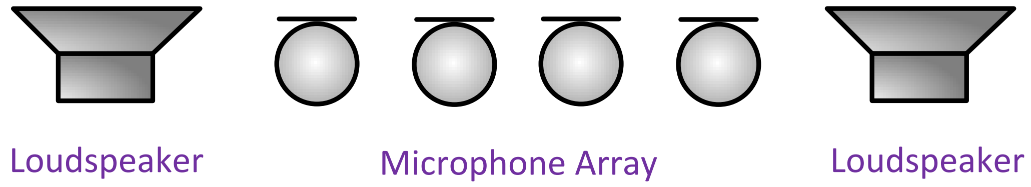


Fig. 2.1: *An acoustic interface.*

2.1.2 The "intelligence" in interfaces

"Intelligence" is not an easy term to define. What makes a system intelligent? In intelligent interfaces, the "intelligence" might be in predicting what the user wants to do, and presenting information with this prediction in mind [61]. Intelligent interfaces can also make doing a task more intuitive and helpful. Instead of trudging along a task in the mire of an inefficient and clumsy interface, the user might find a helpful and information-using interface to be more intelligent. Thus, "intelligence" does not actually mean cognition in this context; instead, it means using information in an appropriate manner [61, 87].

"Intelligence" in interfacing is a subjective term. One person may look at a system with context-sensitive help and say that the system seems smart; another person might look at the same system and see nothing special about it. In a sense, "intelligence" in interfaces might be defined as "the next best thing" [61]. Once we have a system which one would say is intelligent, the novelty of the system wears off, and people are in search for more intelligent interfaces. "Intelligence" is that goal which is always one step ahead of us; once we conquer it, it is no longer intelligence.

Interfaces can be intelligent about the user. Through the use of a user model, the system can tailor communication (both input and output) to the user [61]. Examples of tailored communications include methods of communicating (voice, visual, tactile) and way of presenting data (graph, chart, multimedia messages). The interface can also be sensitive to the wants and needs of the user. This ties closely with the user model, but it deals more with interface adaptability than outright use of models.

2.1.3 Human-machine interaction by intelligent acoustic interfaces

The *Association for Computing Machinery* defines *human-machine interaction* as “a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them” [62]. An important role in human-machine interaction is played by *intelligent acoustic interfaces* (IAIs). An IAI translates acoustic information from user to computer, and *vice versa*, in order to allow an homogeneous interaction between parties. From the user point of view, an IAI should be as invisible and intuitive as possible: working with and understanding an IAI should not be a task so that the user should be able to concentrate on the task which he is to perform.

An IAI must be able to adapt to user, to acquire and process information from user, to understand user requirements, to give user an answer satisfying his demands disguised as natural language or multimedia message. Moreover, once information has been acquired, an IAI must be able to autonomously decide whether the user needs an answer or not. In many cases, an IAI must learn user behaviour, mood and personality in order to yield an answer being as compliant as possible to user needs.

2.1.4 Applications using intelligent acoustic interfaces

IAIs are widely used in several fields of application, as also confirmed by the scientific and technological state of the art.

In the multimedia sector it is possible to think to applications such that: speech/audio real-time interaction [68]; speech automatic analysis, automatic music composition and transcription [15]; automatic genre and context recognition in broadcast programs [145]; high-interactivity entertainment [31], a sector that has viewed a growing interest also due to emerging videogame technologies.

In domotics IAIs may be employed in the following applications: the development of “intelligent rooms”, in which speakers and speech commands must be recognized [142]; advanced anthropomorphic robotics [133]; integration with videosurveillance systems [33], that provide, in an automatic way, event identification, audio/video zoom in the region where the event is detected, and the consequent activation of an alarm or any other action related to the identified event.

Moreover, it is possible to exploits IAIs to develop aid systems for disabled people, that may be hearing aids or even devices able to provide an accurate reconstruction of an acoustic environment [122, 89].

2.2 SCIENCE AND TECHNOLOGY OF INTELLIGENT ACOUSTIC INTERFACES

2.2.1 Historical background on speech communications

Before the invention of electromagnetic telephones, there were mechanical devices for transmitting spoken words over a greater distance than that of normal speech. The very earliest mechanical telephones were based on sound transmission through *pipes* or other physical media (see Fig. 2.2). *Speaking tubes* long remained common, including a lengthy history of use aboard ships, and can still be found today.

The telephone emerged from the creation of, and successive improvements to the *electrical telegraph*. In 1804 Catalan polymath and scientist Francisco Salvá i Campillo constructed an electrochemical telegraph. An electromagnetic telegraph was created by Baron Schilling in 1832.

The first commercial electrical telegraph was constructed by Sir William Fothergill Cooke and entered use on the Great Western Railway in England. It ran for 13 miles from Paddington station to West Drayton and came into operation on April 9, 1839.

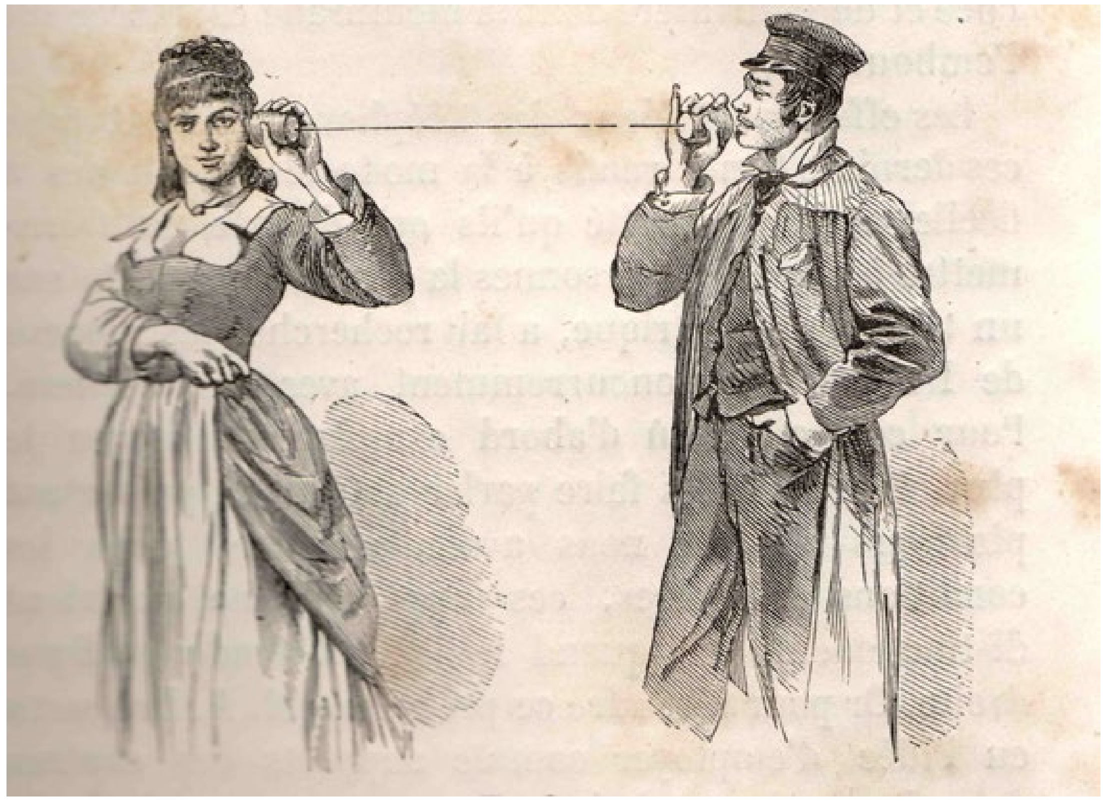


Fig. 2.2: *The tin can telephone, or also known as lover's phone, connected two diaphragms with a taut string or wire, which transmitted sound by mechanical vibrations from one to the other along the wire.*

During the second half of the 19th century inventors tried to find ways of sending multiple telegraph messages simultaneously over a single telegraph wire by using different modulated audio frequencies for each message. These inventors included Charles Bourseul, Thomas Edison, Elisha Gray, and Alexander Graham Bell. Their efforts to develop *acoustic telegraphy* in order to significantly reduce the cost of telegraph messages led directly to the invention of the *telephone*, or *the speaking telegraph*.

The commercial use of the telephone started in 1876 [57]. This was one year after Alexander Graham Bell filed a patent application for a telephone apparatus. Efforts to transmit voice by electric circuits, however, date further back. Already in 1854 Charles Bourseul described a transmission method. Antonio Meucci set up a telephone system in his home in 1855. Philipp Reis demonstrated his telephone in 1861. One has also to mention Elisha Gray who tragically filed his patent application for a telephone just 2 hours later than Alexander Graham Bell.

In the very early days of the telephone conducting a phone call meant to have both hands busy; one was occupied to hold the loudspeaker close to the ear and the other hand to position the microphone in front of the mouth. This troublesome way of operation was due to the lack of efficient electroacoustic converters and amplifiers. The inconvenience, however, guaranteed optimal conditions: a high signal-to-(environmental) noise ratio at the microphone input, a perfect coupling between loudspeaker and the ear of the listener, and - last but not least - a high attenuation between the loudspeaker and the microphone. The designers of modern speech communication systems still dream of getting back those conditions [57].

In a first step one hand had been freed by mounting the telephone device, including the microphone at a wall; further on, only one hand was busy holding the loudspeaker. In a next step the microphone and the loudspeaker were combined in a handset. Thus, still one hand was engaged. This basically remained the state of the art until today [57].

Early attempts to allow telephone calls with a loudspeaker and a microphone at some distance in front of the user had to use analog circuits. In 1957 Bell System introduced a so called *speakerphone*. At the same time, however, the telephone connection degraded to a half-duplex loop making natural conversations difficult. The introduction of a "center clipper" may be considered as a last step along this line [16]. This nonlinear device suppresses small amplitudes. Thus, it extinguishes small echoes. Moreover, small speech signals are erased, as well [57].

The invention of the least mean square algorithm in 1960 [153], the application of adaptive transversal filters [79, 130] and the availability of digital circuits with increasing processing power opened new paths to acoustic echo and noise control [57]. It took at least two more decades of breathtaking progress in digital technology until commercial applications of adaptive filters for acoustic echo and noise control became feasible.

Modern technologies are evolving towards new directions which take into account the *distant-talking*, i.e. the hands-free speech communication using intelligent acoustic interfaces. However, this change involves new challenging problems to address, as we see throughout this dissertation.

2.2.2 Philosophical background on intelligent interfaces

All along the human-interface interaction has aroused the interest of researchers, philosophers and cognitive scientists, which attempt to answer to questions about *artificial intelligence* (AI), such as "Can a machine display intelligence?".

The basic position of most AI researchers is summed up in this statement, which appeared in the proposal for the Dartmouth Conferences of 1956 [88]:

> "*Every aspect of learning or any other feature of intelligence can be so precisely described that a machine can be made to simulate it*."

The first step to answering those questions is to clearly define "intelligence". Alan Turing, in a famous and seminal 1950 paper [144], reduced the problem of defining intelligence to a simple question about conversation. He suggests that:

> "*If a machine can answer any question put to it, using the same words that an ordinary person would, then we may call that machine intelligent*."

Recent AI research defines intelligence in terms of "intelligent agents", that is more close to our definition of "intelligent interfaces". An "agent" is something which perceives and acts in an environment; a "performance measure" defines what counts as success for the agent [117]:

> "*If an agent acts so as maximize the expected value of a performance measure based on past experience and knowledge then it is intelligent*."

In 1963, Allen Newell and Herbert Simon proposed that "symbol manipulation" was the essence of both human and machine intelligence. They wrote [95]:

> "*A physical symbol system has the necessary and sufficient means of general intelligent action*."

This claim is very strong: it implies both that human thinking is a kind of symbol manipulation (because a symbol system is necessary for intelligence) and that machines can be intelligent (because a symbol system is sufficient for intelligence). Another version of this position was described by philosopher Hubert Dreyfus, who called it "the psychological assumption" [38]:

> "*The mind can be viewed as a device operating on bits of information according to formal rules*."

A distinction is usually made between the kind of high level symbols that directly correspond with objects in the world and the more complex "symbols" that are present in a machine like a neural network. Moreover, Dreyfus argued that human intelligence and expertise depended primarily on unconscious instincts rather than conscious symbolic manipulation, and argued that these unconscious skills would never be captured in formal rules [38, 117].

Russell and Norvig point out [117] that, in the years since Dreyfus published his critique, progress has been made towards discovering the "rules" that govern unconscious reasoning. The situated movement in robotics research attempts to capture our unconscious skills at perception and attention [21]. Computational intelligence paradigms, such as *neural networks*, *evolutionary algorithms* and so on are mostly directed at simulated unconscious reasoning and learning.

Probably IAIs will never able to solve any problem that a person would solve by thinking, however, they may help users to enjoy an immersive communication.

2.3 INTELLIGENT ACOUSTIC INTERFACES FOR IMMERSIVE COMMUNICATIONS

After years of extraordinary technological advances in telecommunications, new requirements are demanded by users which are no longer satisfied with talking to someone over a long distance and in real time, but they want to collaborate through communication in a more productive way with the feeling of being together and sharing the same environment. That gives rise to *immersive communication*. Such immersive communication is yet to become a reality supported by modern communication technologies. A person's sense of acoustic immersion is formed by his or her sensory response to the auditory stimuli that exist in the ambiance of their environment [66].

Immersive communications take place in multisource environments, as depicted in Fig. 2.3 where interfering signals may degrade quality and intelligibility of the desired speech source. Therefore, acquisition of desired signals with high quality is far more difficult and challenging for immersive communications than in the classical telephony environment where the microphone is close to the user. In immersive communications, it is more likely that multiple parties will be involved and conferencing is a more common mode of operation than point-to-point calling. In conferencing, one may hear the unwanted interfering signals from every other participant and therefore the level of the perceived noise can grow with the number of participants. When the number is large and if interfering sources are not well controlled, the perceived noise can reach a level such that speech is overwhelmed. So interfering sources become a more quality-threatening problem for immersive voice communication [65, 66].

Immersive communication offers great opportunities for acoustic and speech signal processing and implies the use of IAIs. Voice is by far the dominant media in the exchange of conference content. In fact, a teleconferencing session can still go on when the video link is broken, but it has to stop if the audio link is disrupted. So in addition to the pursuit of multimodal capabilities, we should never forget the importance of *speech quality* (including intelligibility and naturalness) and *intermodal synergy*. Moreover, there are great potentials to improve these two factors in an immersive teleconference with multiple parties being involved since binaural hearing is now allowed and can be fully exploited. This is an imperative step towards immersive communication. With both ears being kept busy, our auditory system can more easily extract a single talker's speech among multiple conversations and background noise, and can more seamlessly work together with the visual system in an adverse acoustic environment for speech perception (e.g., lip-reading).

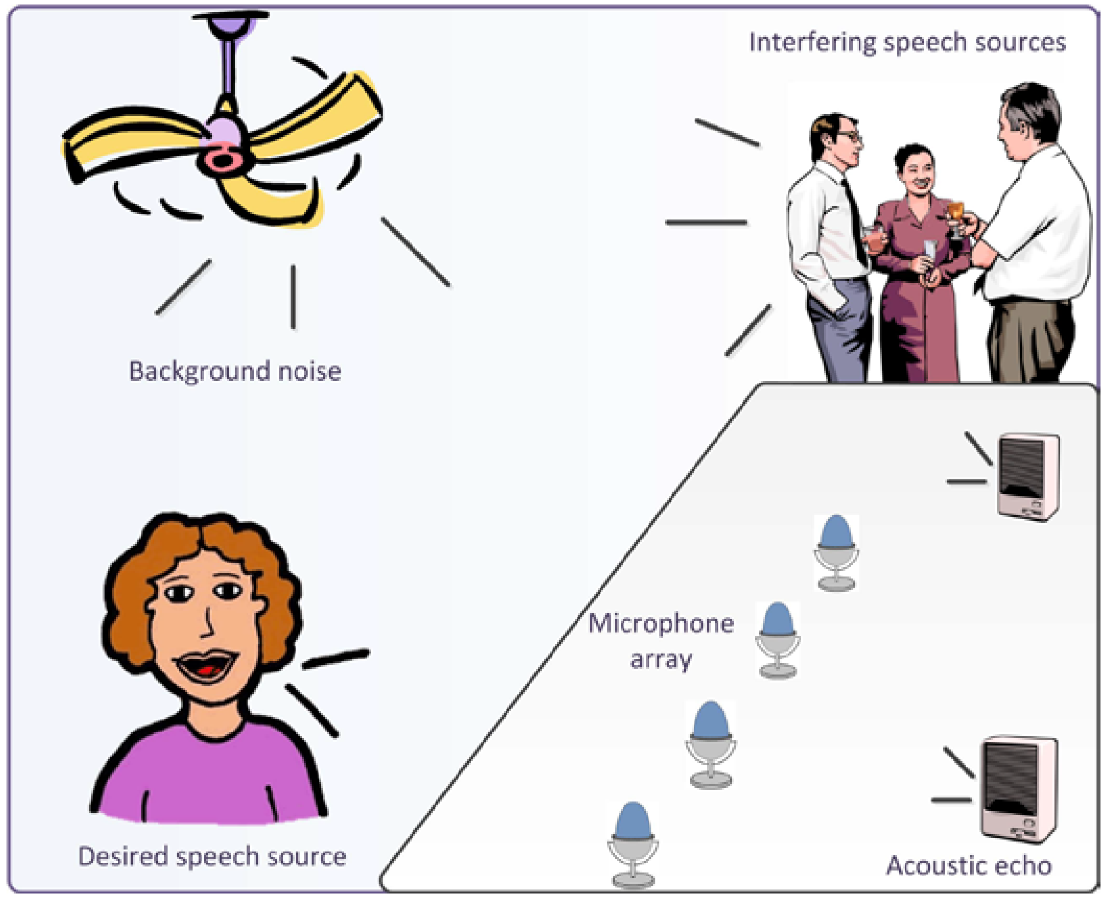


Fig. 2.3: *Immersive speech communication in multisource environment.*

An IAI for immersive communication aims at extracting, from audio signals, useful informations for computational or human purpose, such as analysis or synthesis of audio signals. This feature is also known as *machine listening*. At the same time, an IAI has to reproduce desired acoustic information taking into account that the listener would hear the sound exactly as in the original sound field. This feature indeed is known as *spatial sound reproduction*. To these ends, an IAI needs to replicate four attributes of face-to-face communication [65, 66]:

1. full-duplex exchange;
2. freedom of movement without body-worn or tethered microphones (i.e., hands-free in the broad sense);
3. high-quality speech signals captured from a distance;
4. spatial realism of sound rendering.

These requirements imply that multiple microphones and loudspeakers would be used and the entire voice communication infrastructure might need to be renovated. However, the scope of this thesis mainly concerns with the machine listening feature, since we deal with adaptive algorithms which have to process the acoustic signals acquired by a microphone interface.

3

Problem Formulations in Acoustic Modelling

Contents

IMMERSIVE speech communications often take place in multisource reverberant environments where interfering signals may deteriorate the speech intelligibility. In order to tackle such limitations, IAIs aims at modelling the acoustic channel by means of adaptive filtering algorithms. In this chapter we introduce a set of problems which limit the achievable communication quality, and how to address these problems using adaptive filtering algorithms. Moreover, we briefly describe some of the main acoustic applications in which it is possible to employ IAIs based on adaptive algorithms.

3.1 MAIN CHARACTERISTICS OF ACOUSTIC CHANNELS

The problems to address in the modelling of acoustic channels are substantially different from those occurring in other communication channels, such as wireless or fibre channels. This is due to the fact that acoustic channels possess distinctive characteristics that set them apart from other kinds of transmission channels and focus attention on the development of more effective algorithms for IAIs. In the following we summarize some of the main characteristics of acoustic channels that must be taken into account in designing adaptive algorithms for IAIs.

3.1.1 Linearity and shift-invariance

An acoustic channel can be definitely labelled as a *linear shift-invariant* (LSI) system [65]. Linearity and shift-invariance are the two most important properties for simplifying the analysis and design of discrete-time system and often such characteristics do not belong to other communication channels. A linear system ought to satisfy the rules of *homogeneity* and *additivity* which are

the basis of the *principle of superposition*. For a homogeneous system, scaling the input by a constant results in the output being scaled by the same constant. For an additive system, the response of the system to a sum of two signals is the sum of the two responses. A system is shift-invariant when a time shift in its input leads to the same shift in its output. Therefore, taking into account these properties, an LSI system can be easily characterized by its impulse response. Once the impulse response is known, it is possible to foresee the response of the LSI system to any possible input stimuli.

3.1.2 Modelling by FIR filters

The AIR is usually very long. However, *finite impulse response* (FIR) filters are more frequently used than *infinite impulse response* (IIR) filters in acoustic applications. This choice is justified by the fact that the stability of FIR filters is easily controllable; moreover, there are a large number of adaptive algorithms providing good performance for FIR filters, thus allowing an accurate modelling of the acoustic channel [65, 120].

3.1.3 Time-varying AIR

Like many other communication channels with different physical medium, acoustic channels are inherently *time-varying* systems. In immersive speech communications sound sources are free to move in the environment. Moreover, even a change of atmospheric conditions in the environment may cause a variation of the AIR. However, this time-varying property usually does not prevent the use of FIR filters to model acoustic channels since acoustic systems generally change slowly compared to the length of their AIR [65]. Therefore, dividing time into periods, it is possible to assume that in each period the acoustic channel is stationary and can be modelled by means of an FIR filter.

3.1.4 Frequency selectivity

Acoustic waves are pressure disturbances propagating in the air. With spherical radiation and spreading, the inverse-square law rules and the sound level falls off as a function of distance from the sound source. As a rule of thumb, the loss is 6 dB for every doubling of distance. But when acoustic sound propagates over a long distance (usually greater than 30 m), an excess attenuation of the high-frequency components can often be observed in addition to the normal inverse-square losses, which indicates that the acoustical channel is *frequency selective* [65]. The level of this high-frequency excess attenuation is highly dependent on the air humidity and other atmospheric conditions.

The inverse-square law governs free-space propagation of sound. But in such enclosures as offices, conference rooms, and cars, acoustic waveforms might be reflected many times by the enclosure surfaces before they reach a microphone. The attenuation to the reflection is generally frequency-dependent. However, for audio signals this dependency is usually not significant, unlike radio-frequency signals in indoor wireless communication. For acoustic channels in these environments, it is the aspect of multipath propagation that leads to frequency-selective characteristics. Frequency-selective fading is viewed in the frequency domain. In the time domain, it is called *multipath delay spread* and induces sound reverberation analogous to inter-symbol interference observed in data communications.

3.1.5 Reverberation time

Room reverberation is usually regarded as destructive since sound in reverberant environments is subject to temporal and spectral smearing, which results in distortion in both the envelope and fine structure of the acoustic signal [65]. If the sound is speech, then speech intelligibility will be impaired. However, room reverberation is not always detrimental. Although it may not be realized consciously, reverberation is one of many cues used by a listener for sound source localization and orientation in a given space. In addition, reverberation adds “warmth” to sound due to the colorization effect, which is very important to musical quality. The balance between sound clarity and spaciousness is the key to the design of attractive acoustic spaces and audio instruments, while the balance is achieved controlling the level of reverberation.

The level of reverberation is typically measured by the *reverberation time*, T_{60}, which was introduced by Sabine [118] and is now a part of the ISO (*International Organization for Standardization*) reverberation measurement procedure. The reverberation time is defined as the length of time that it takes the reverberation to decay 60 dB from the level of the original sound. The most widely used method for measuring the sound decay curves is to employ an excitation signal and record the acoustic channel’s response with a microphone.

3.1.6 Channel invertibility and minimum-phase

The *invertibility* of an acoustic channel is of particular interest in many acoustic applications such as speech enhancement and dereverberation. A system is invertible if the input to the system can be uniquely determined by processing the output with a stable filter [65]. In other words, there exists a stable inverse filter that exactly compensates the effect of the invertible system. A stable, causal, rational system requires that its poles be inside the unit circle. Therefore, a stable, causal system has a stable and causal inverse only if both its poles and zeros are inside the unit circle. Such a system is commonly referred to as a *minimum-phase* system [65].

Unfortunately AIRs are almost never minimum-phase [94]. This implies that perfect deconvolution of an acoustic channel can be accomplished only with an “acausal” filter. This may not be a serious problem for off-line processing since we can incorporate an overall time delay in the inverse filter and make it causal. But the delay is usually quite long for acoustic channels and the idea is difficult to implement with real-time systems.

3.1.7 Multichannel diversity

In *multiple-input multiple-output* (MIMO) systems, one of the most important feature is the *channel diversity*, which implies that different channels of a MIMO system would have no modes in common [65]. If the channels are modelled as FIR filters, channel diversity means that their transfer functions share no common zeros, or in other words, they are co-prime polynomials.

However, in this dissertation we deal with adaptive algorithms for *single-input single-output* (SISO) systems; therefore, for possible future extension of such algorithms in the multichannel domain, the characteristic of multichannel diversity will have to be take into account.

3.1.8 Sparse acoustic impulse response

Recently, it has been recognized that most AIRs are sparse in their nature, i.e., only a small percentage of the impulse response components have a significant magnitude while the rest are zero or small [40]. This characteristic can be exploited by a class of adaptive algorithms, named *proportionate adaptive filters* [40, 13, 100], in order to improve their performance in terms of initial convergence and tracking. Proportionate adaptive algorithms will be extensively discuss in Part II of this dissertation.

3.2 LIMITATIONS AND PROBLEMS IN ACOUSTIC PATH MODELLING

As previously said, adaptive filtering algorithms in IAIs aim at modelling an acoustic channel through the estimate of the AIR generated by the acoustic coupling between a loudspeaker and a microphone. However, the AIR estimate becomes more critical when the acoustic path is affected by adverse conditions of the environment. The design of an adaptive algorithm has to take into account such problems in order to provide anyway an accurate estimate of the AIR that allows to preserve the quality of an immersive speech communication.

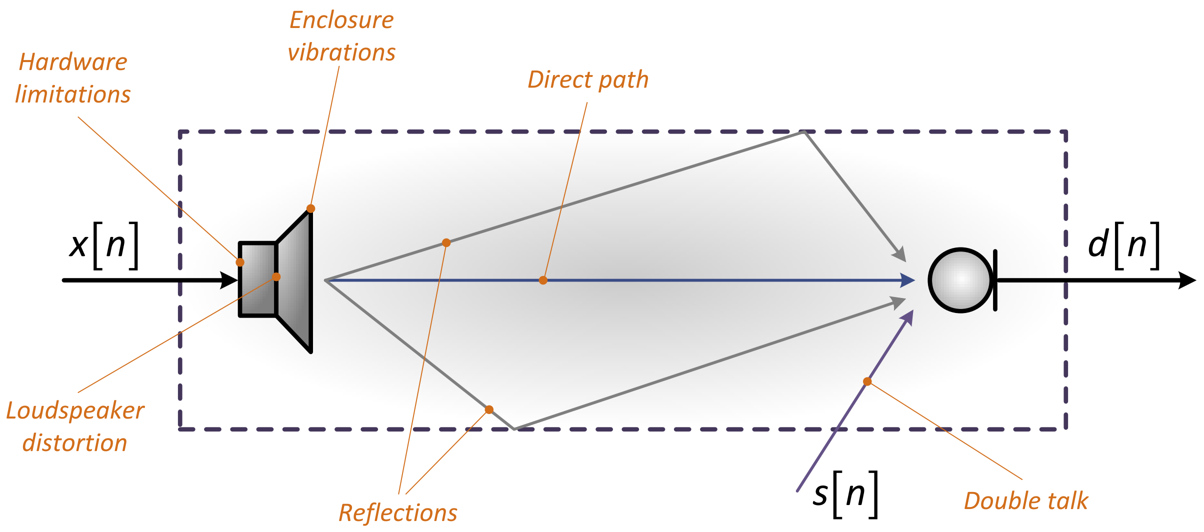


Fig. 3.1: *An acoustic interface.*

In this section we introduce a brief overview on such problems which limit the performance of an AIR modelling; they may be essentially labelled as linear or nonlinear events and are depicted in Fig. 3.1.

3.2.1 Linear limitations

Hardware limitations

Hardware limitations include thermal and impulsive circuit noise from amplifiers, and DSP related noise such as truncation, finite word lengths and characteristics of the particular algorithm being used [18]. These limitations are often caused by low-quality electronic components used in low-cost acoustic interfaces. This kind of problem essentially affects the step size value of the adaptive algorithm which may need to be very small, thus leading to a decrease of convergence performance at steady-state. Therefore, this limitation requires a good trade-off between convergence rate and precision.

Under-modelling of the AIR

As said in Par. 3.1.2, the modelling of the AIR is usually performed by means of FIR filters. However, this entails some difficulties in designing the filter, and the first and foremost one is the choice of the filter length. Indeed, it is very difficult to *a priori* know the *exact* length of the AIR, and, anyway, it usually requires a large number of filter coefficients, that is unpracticable for a real-time implementation. This is the reason why the habit is to choose a filter length smaller than the actual length of the AIR, thus leading to an *under-modelling* of the AIR. The remaining unmodelled tail portion of the AIR manifests itself as a finite error at the output of the processor. However, blindly increasing the number of taps results in added complexity, greater algorithmic noise and slower convergence. Therefore, this limitation requires a proper setting of the step size value in order to avoid this further error contribution at the output of the modelling system.

Nonstationary environment

The initial convergence of a particular algorithm identifies the room configuration, however as objects move and the input characteristics become nonstationary, the tracking ability of the algorithm becomes important. For example, although Hessian-based algorithms, such as the *recursive least squares* (RLS) algorithm, have fast convergence, it has been found that algorithms based on instantaneous gradient estimates, like the *normalized least mean square* (NLMS), actually outperform Hessian-based algorithms when nonstationarities occur [120, 18].

Double talk

The *double talk* event occurs when an interfering speech signal is present and is superimposed over the acoustic path to model. In order to solve this problem a *double talk detector* (DTD) is usually adopted [57], which stops the filter adaptation in presence of double talk in order to preserve the desired

speech. A DTD is a good mean to meet the contradictory requirement of low divergence rate and fast convergence in acoustic channel modelling. However, not ever a DTD provides desired performance, since an optimal DTD is difficult to realize and may be even very expensive from a computational point of view.

3.2.2 Nonlinear limitations

Loudspeaker distortions

Generated mainly in the loudspeaker, *nonlinear distortions* effectively put a limit on the achievable quality of algorithms based on linear mechanics [147, 18]. In addition to the direct loudspeaker effects, secondary nonlinear effects such as *rattling* can be considered nonlinear in nature. Rattling is very difficult, if not impossible to model. However, the loudspeaker nonlinearity is weak and may therefore be modelled accurately with nonlinear models. Loudspeaker distortions represent a very difficult problem to solve since they may be highly time-varying, thus leading to a kind of nonlinearity with memory.

Enclosure vibrations

A major part of the AIR is due to loudspeaker/microphone/enclosure coupling which is stationary in nature and larger in amplitude than a speech signal. The particular adaptive algorithm used will devote a portion of its computation to adapt these AIR coefficients which may be better modelled by another method. *Whistling* can occur in small orifices in sealed enclosures. This whistling is essentially chaotic in nature and can be a problem if it occurs close to the microphone [18]. Such vibrations, especially in the lower voice frequencies, causes significant nonlinearities which may seriously impair the intelligibility of a hands-free speech communication.

3.3 ACOUSTIC ECHO CANCELLATION

A typical application of acoustic channel modelling is definitely the *acoustic echo cancellation* (AEC). Acoustic echo in a hands-free voice communication system is produced by the acoustic coupling between a loudspeaker and a microphone, as depicted in Fig. 3.2. The perception of an echo depends on not only its level but also its delay [66]. Through long-distance transmission, the echo features a long delay time and would significantly reduce the quality of voice communication. When the delay approaches a quarter of a second, most people find it difficult to carry on a normal conversation. Full-duplex voice telecommunication was implausible, if not impossible, before the echo cancellation theory was developed by Bell Labs researchers in the 1960s [132]. For an immersive audio system with several microphones and loudspeakers, multiple echo paths need to be identified. Regardless of how many microphones there are, AEC is always carried out individually with respect to each of them. But the number of loudspeakers present in the system draws a theoretical difference between monophonic (one loudspeaker) and multichannel (multiple loudspeakers) echo cancellations in the difficulty of tracking the echo paths [66].

In echo cancellation, the source (loudspeaker) signals are known. So echo control is theoretically a well-posed problem [66], and its practical applications have been relatively more successful than the control of the other types of noise (such as additive noise, reverberation and unwanted speech) in which blind or semiblind methods have to be incorporated.

Historically, the study of acoustic echo cancellation substantially enriched the adaptive filtering and system identification literature. Indeed, an adaptive filter plays a central role in a monophonic echo cancellation system. It attempts to dynamically identify the acoustic echo path. As long as the channel impulse response of the echo path can be quickly and accurately determined, it is then straightforward to generate a good estimate of the echo and subtract it from the microphone signal. Since the loudspeaker signal as the reference is available, numerous nonblind adaptive filtering methods for system identification are applicable for solving this problem [131, 12, 66].

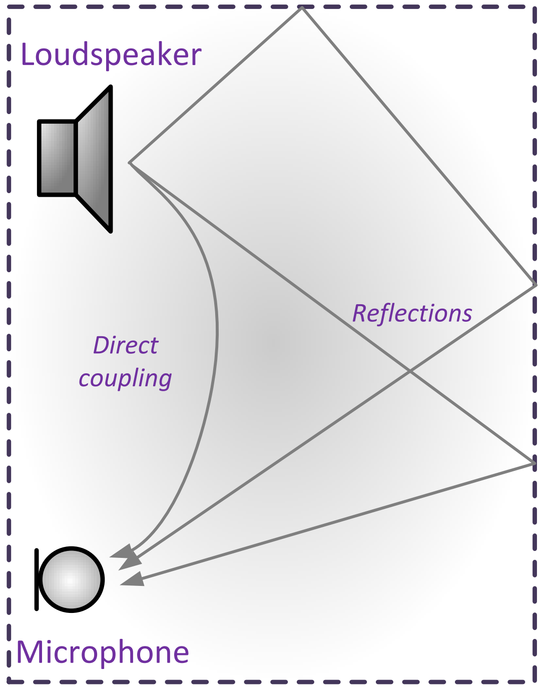


Fig. 3.2: *Microphone-loudspeaker acoustic coupling.*

In order to better comprehend AEC application, let us introduce a brief description of the processing performed by an *acoustic echo canceller* in the context of a teleconferencing communication between two (or more) users located in different environments. As it is possible to notice from the scheme in Fig. 3.3, at n-th time instant, the speech signal coming from the remote user, also known as *far-end*, and denoted as $x[n]$, arrives at the other side of communication and is reproduced by the loudspeaker. During the reproduction the far-end signal may result distorted by loudspeaker nonlinearities. Moreover, being the speech communication *immersive*, the far-end signal reproduced by the loudspeaker is acquired by the microphone(s) of the acoustic interface used by the local user, or also said *near-end*. The acoustic coupling between the microphone and the loudspeaker is characterized by an acoustic path which contains information about the environment reverberations. The signal

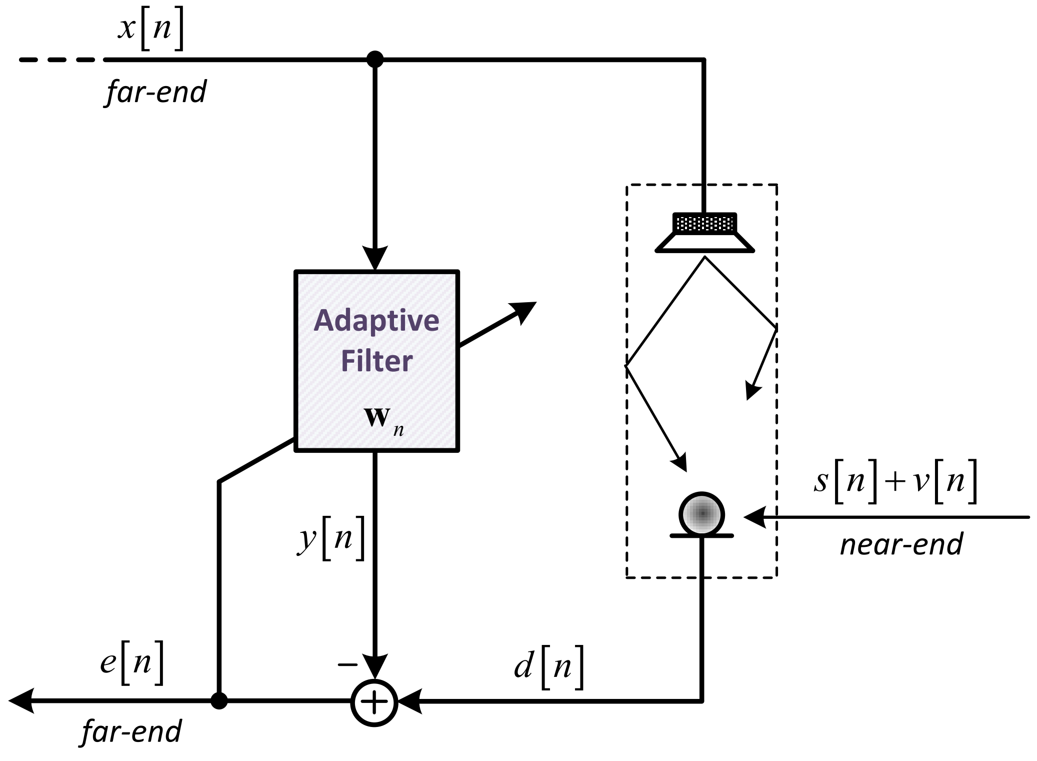


Fig. 3.3: *Processing scheme of an acoustic echo canceller.*

emitted by the loudspeaker and acquired by the microphone represents the *echo signal*, which may be possibly superimposed on the near-end contribution that is the desired information for the far-end user. The near-end signal is composed of the near-end speech signal $s\left[n\right]$ with the addition of background noise $v\left[n\right]$. In literature, the overall microphone signal is usually named as *desired signal* and it is denoted with $d\left[n\right]$. At the same time, the far-end signal $x\left[n\right]$ is processed by the acoustic echo canceller in order to estimate the AIR between microphone and loudspeaker. The output signal of this filtering process, $y\left[n\right]$, represents the estimated echo signal which is then subtracted by the microphone signal $d\left[n\right]$, preserving the near-end information, to the end of generating the *error signal* $e\left[n\right]$ that is sent to the far-end user.

AEC represents an exhaustive application in hands-free speech communications since it includes a set of problems common to the whole sector of acoustic scene analysis: the estimate of the impulse response, the presence of nonstationary elements in the environment, the presence of unwanted interfering signals, the presence of nonlinearities [12]. Moreover, AEC allows to obtain a complete evaluation of the adaptive filtering algorithms that may be used afterwards also in other acoustic applications, such as adaptive beamforming, noise reduction, speech dereverberation, speech enhancement, etc.

3.4 PERFORMANCE MEASURE

In order to evaluate performance of adaptive filtering algorithm in AEC applications two measures are usually computed: the echo return loss enhancement and the normalized misalignment.

3.4.1 Echo return loss enhancement

The *echo return loss enhancement* (ERLE) is defined by G.168 as "the attenuation of the echo signal as it passes through the send path of an echo canceller". The ERLE results from the ratio in dB between the instantaneous power of the desired signal $d\left[n\right]$, i.e. the microphone signal, and the instantaneous power of the residual echo signal $e\left[n\right]$ [57]:

$$\mathrm{ERLE}\left[n\right] = 10\log\frac{\mathrm{E}\left\{d^2\left[n\right]\right\}}{\mathrm{E}\left\{e^2\left[n\right]\right\}} \tag{3.1}$$

A large value of the ERLE denotes a good performance of the acoustic echo canceller, while a small value of the ERLE denotes a significant presence of the echo signal in the processed signal.

In Fig. 3.4 the limitation effects on the maximum achievable ERLE is represented. It is possible to see that a first important limit is posed by the acoustic environment due above all to reflections and nonstationary signals. However, more important limits are generated by the presence of nonlinearities in the

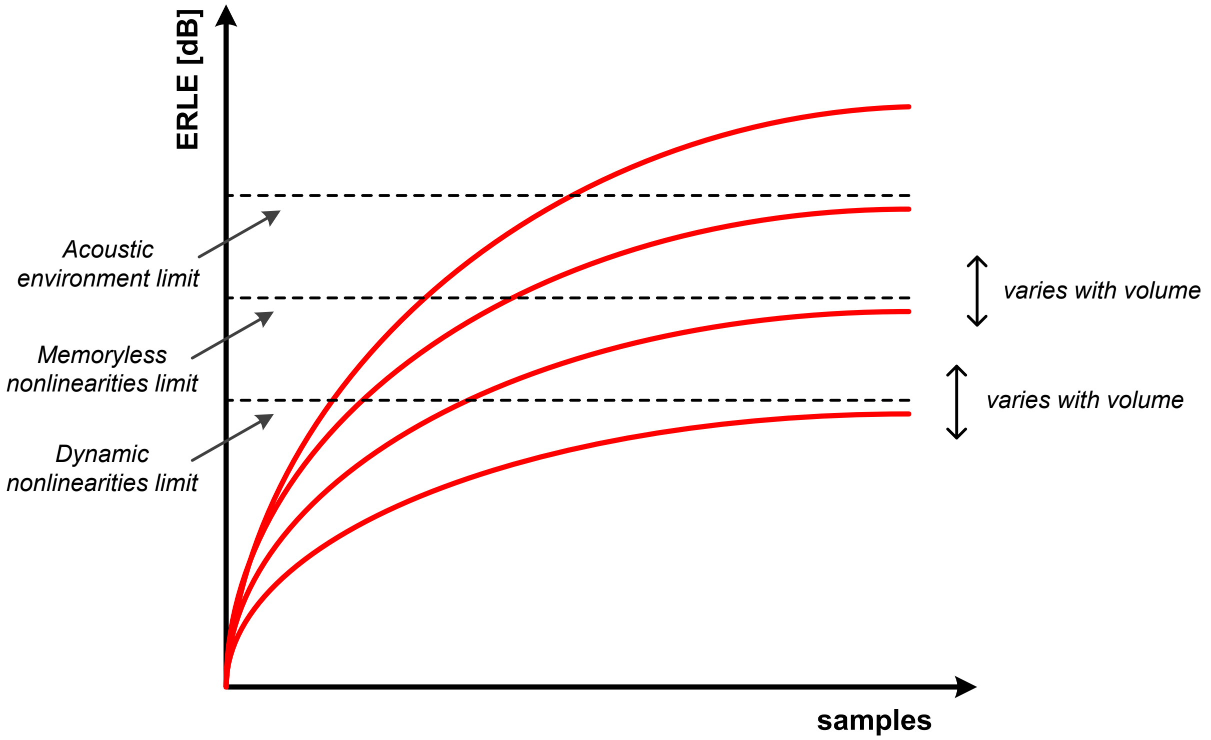

ERLE [dB]
Acoustic environment limit
Memoryless nonlinearities limit
Dynamic nonlinearities limit
varies with volume
varies with volume
samples

Fig. 3.4: *Limitation effects on the achievable ERLE.*

echo path, and in particular by nonlinearities with memory, i.e. those nonlinearities which are originated by dynamic systems. These limits posed by nonlinearities also depends on volume and frequency variations and may be particularly harmful to speech quality when *intermodulation distortions* occur at low frequencies.

As it is possible to notice from equation (3.1), the ERLE is a measure that depends on the minimization of the error signal. This allows to use the ERLE in the evaluation of both linear and nonlinear echo cancellers. However, the ERLE does not highlight sufficiently small variations of the adaptive algorithm; moreover, a large value of the ERLE does not guarantee as much large degree of speech quality. Due to these reasons, according to our opinion, the ERLE is not always the best performance measure to adopt in order to evaluate an adaptive filter in AEC applications; however, in literature the ERLE remains the most used performance measure to evaluate echo cancellers.

3.4.2 Normalized misalignment

Another important performance measure is the *normalized misalignment* which quantifies how "well" an adaptive filter converges to the impulse response of the system that needs to be identified [12]. It is defined in dB as:

$$\mathcal{M} = 20\log_{10}\left(\frac{\left\|\mathbf{w}^{\mathrm{opt}} - \hat{\mathbf{w}}_n\right\|_2}{\left\|\mathbf{w}^{\mathrm{opt}}\right\|_2}\right) \tag{3.2}$$

where $\mathbf{w}^{\mathrm{opt}}$ is the optimal solution to estimate, i.e. the AIR, and $\hat{\mathbf{w}}_n$ is the filter estimate by the adaptive filter.

Unlike the ERLE, the normalized misalignment depends on the coefficients of the adaptive filter instead of the error signal, thus leading to some advantages and drawbacks. The most significant drawback is the fact that the normalized misalignment cannot be used to evaluate adaptive filters in presence of nonlinearities. This is due to the fact that nonlinearities are not taken into account in the optimal solution while they affect the filter estimate, thus the normalized misalignment does not have sense in this case. However, the normalized misalignment, unlike the ERLE, allows to have a complete evaluation of a linear adaptive algorithm in terms of convergence rate, tracking, and accuracy of the solution at steady-state. Moreover, the behaviour of the normalized misalignment also reflects the perceived quality of the processed speech signal. In fact, when the normalized misalignment shows a jumpy behaviour usually the processed signal may display some musical noise.

Such analysis focus the attention on the evaluation of the performance of adaptive filters in the nonlinear case, in which it is not possible to exploit a such important measure as the normalized misalignment. This might be definitely matter of future research.

Part II

LINEAR ADAPTIVE ALGORITHMS

—Playing chess is about the dumbest question you can ask.
But, if you want, maybe can make money that way, or something.
Noam Chomsky

4

Fundamentals of Adaptive Filtering

Contents

4.1 INTRODUCTION TO ADAPTIVE FILTERS

In studying *digital signal processing* (DSP) techniques, the term “adaptive” is used when a (digital or analog) system is able to automatically “adjust” its parameters in response to input stimuli in order to achieve a processing goal [146].

An *adaptive filter* is defined as a self-designing system that relies for its operation on a recursive algorithm, which makes it possible for the filter to perform satisfactorily in an environment where knowledge of the relevant statistics is not available [59]. In that context, an adaptive filter can be viewed as an "intelligent circuit" able to adapt according to a predetermined law [146].

The ability of an adaptive filter to carry out a certain target is usually expressed through a criterion that minimizes a given *cost function*, often denoted as $J(\cdot)$, which is a function of filter parameters. The procedure which determines the variation law of the filter parameters, according to a given cost function, is also known as *adaptive algorithm*, or in same cases *learning algorithm*.

Usability of adaptive filtering techniques for the solution of real problems is widely stretched as much as fields of their applications. Adaptive filters are extensively used in many DSP areas, such as: modelling, estimate, localization, source separation, etc. Due to the rise of *neural networks*, which may be considered as a particular nonlinear class of adaptive filters, the field of interest has been further extended, thus intersecting *artificial intelligence* methods in order to provide consistent solutions even for the so-called *ill-posed* problems [146]. Recently such methods have merged into an infant subject named *computational intelligence*.

4.1.1 Classification of adaptive filters

There are a lot of way of classifying adaptive filters [146], however, the most popular classification may be carried out based on the learning algorithm and on the input-output relation.

A first subdivision concerns the adopted learning algorithm, i.e. the modality with which it is possible to adapt the filter parameters. In particular, adaptive filters can be classified into:

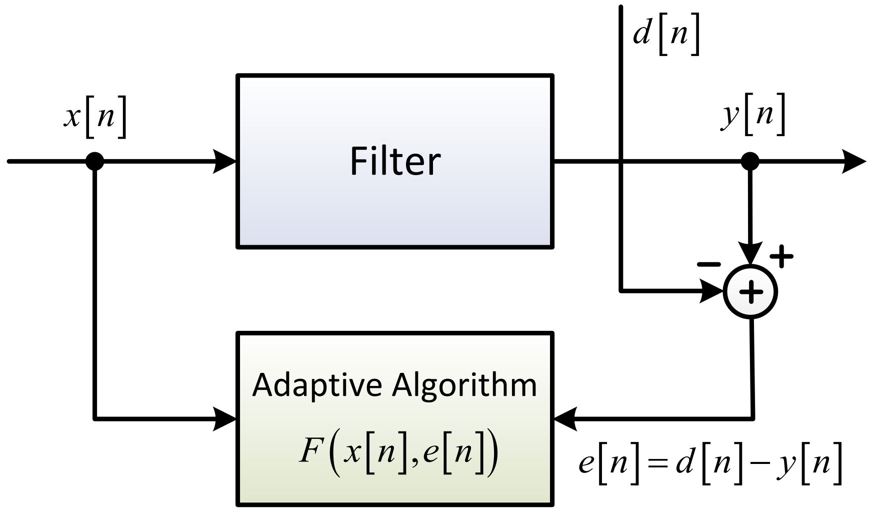


Fig. 4.1: *Scheme of a supervised adaptive filter.*

- *supervised adaptive filters* - require the availability of a training sequence that provides different realizations of a desired response for a specified input signal vector. The desired response is compared against the actual response of the filter due to the input signal vector, and the resulting error signal is used to adjust the free parameters of the filter. The process of parameter adjustments is continued in a step-by-step fashion until a steady-state condition is established. A representation of a supervised adaptive filter is depicted in Fig. 4.1;

- *unsupervised adaptive filters* - perform adjustments of its free parameters without the need for a desired response. For the filter to perform its function, its design includes a set of rules that enable it to compute an input-output mapping with specific desirable properties. In the signal-processing literature, unsupervised adaptive filtering is often referred to as blind deconvolution or blind adaptation [59]. However, in this dissertation we essentially deal with supervised adaptive filters.

Adaptive filters may also be classified according to an input-output relation. Denoting with $\mathbf{w}_n$ the time-varying vector of filter coefficients (i.e. filter parameters), it is possible to classify an adaptive filter according to the properties of an operator $T\left\{\cdot\right\}$ which defines the relation between the input of the filter $x\left[n\right]$ and its output $y\left[n\right]$:

$$y\left[n\right] = T\left\{x\left[n\right], \mathbf{w}_n\right\}. \tag{4.1}$$

On this basis, two main groups of adaptive filters can be characterized:

- *linear adaptive filters* - for the operator $T\left\{\cdot\right\}$ the *superposition principle* holds. Linear adaptive filters compute an estimate of a desired response by using a linear combination of the available set of observables applied to the input of the filter [59, 146];
- *nonlinear adaptive filters* - for the operator $T\left\{\cdot\right\}$ the *superposition principle* is not valid anymore [59]. In this case it is usually necessary to define further sub-labels due to the nature of the nonlinearity, that can be monodrome, invertible, uninvertible, static, dynamic, etc. [146].

Therefore, sub-labels for linear and nonlinear adaptive filters, always taking into account the input-output relation, may be the following ones:

- *static* - the output at time instant n only depends on the input at time instant n; in this case the operator $T\left\{\cdot\right\}$ has the same properties of a function;
- *dynamic with finite memory* or *FIR* - the output at time instant n depends on the input samples according to instants $n, n-1, \ldots, n-M+1$ of a time window, i.e.:

$$y\left[n\right] = T\left\{x\left[n\right], x\left[n-1\right], \ldots, x\left[n-M+1\right], \mathbf{w}_n\right\} \tag{4.2}$$

where M is the length of the time window, i.e. the *filter length*;

- *dynamic with infinite memory* or *IIR* - the output at time instant n depends on the input at time instants $n, n-1, \ldots, n-M+1$, and on past output samples, i.e.:

$$\begin{aligned} y\left[n\right] =& T\left\{x\left[n\right], x\left[n-1\right], \ldots, x\left[n-M+1\right],\right. \\ & \left. y\left[n-1\right], \ldots, y\left[n-M+1\right], \mathbf{w}_n\right\}. \end{aligned} \tag{4.3}$$

A possible classification of adaptive filters [146] based on the input-output relation (restricted to the dynamic case), is depicted in Fig. 4.2.

4.2 LINEAR OPTIMUM FILTERING

Linear optimum discrete-time filters are also known as *Wiener filters*, which are an extremely useful tool since its invention in the early 30's by Norbert Wiener [156]. Wiener was one of the first researchers to treat the filtering problem of estimating a process corrupted by additive noise. The optimum estimate that he derived required the solution of an integral equation known as the *Wiener-Hopf equation* [158]. Soon after he published his work, Levinson formulated the same problem in discrete time [77]. Levinson's contribution has had a great impact on the field of adaptive signal processing. Indeed, thanks to him, Wiener's ideas have become more accessible to many engineers [65]. Wiener theory plays a fundamental role in acoustic applications in which the AIR between a loudspeaker and a microphone needs to be identified. Thanks to many adaptive algorithms directly derived from the Wiener-Hopf equations, this task is now rather easy.

With the Wiener theory, it is possible to identify an unknown system, that in the acoustic case is the AIR. Given the input signal $x\left[n\right]$ and the desired signal $d\left[n\right]$ it is possible to define the error signal $e\left[n\right]$:

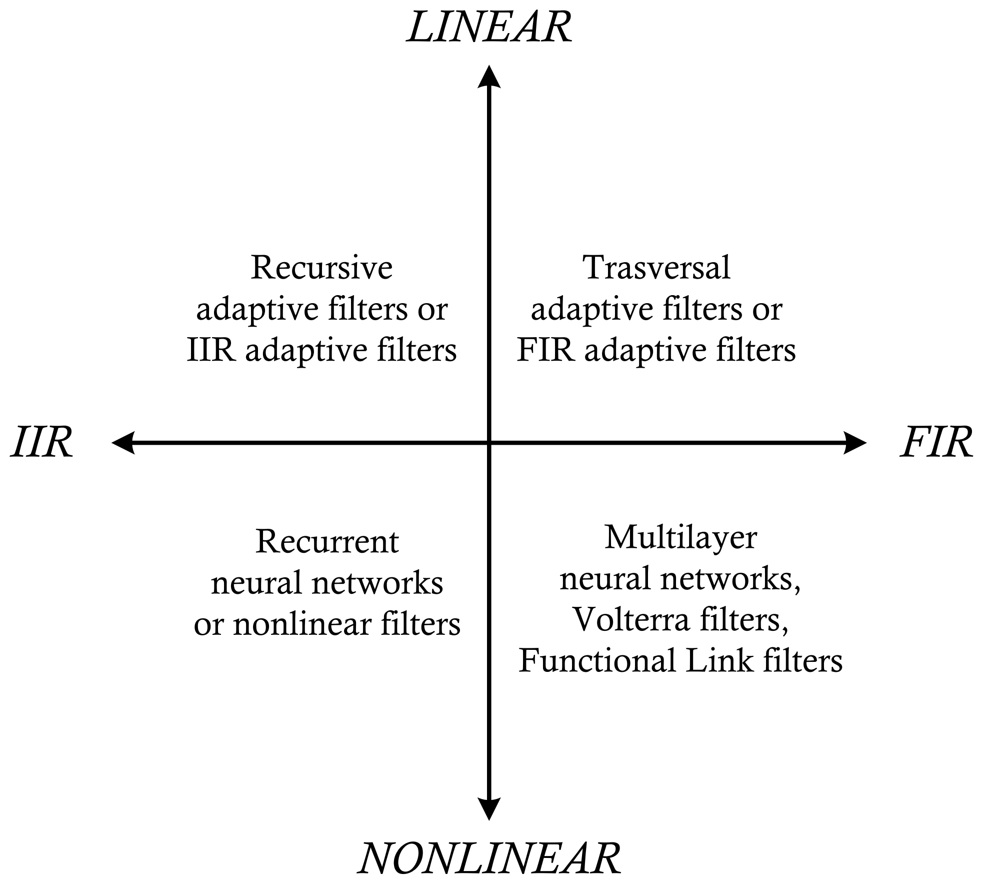

LINEAR
Recursive adaptive filters or IIR adaptive filters
Trasversal adaptive filters or FIR adaptive filters
IIR
FIR
Recurrent neural networks or nonlinear filters
Multilayer neural networks, Volterra filters, Functional Link filters
NONLINEAR

Fig. 4.2: *Classification of adaptive filters based on the input-output relation.*

$$\begin{aligned} e[n] &= d[n] - y[n] \\ &= d[n] - \mathbf{x}_n^T \mathbf{w}_{n-1} \end{aligned} \tag{4.4}$$

where $y[n]$ is the filter output and vector $\mathbf{w}_n \in \mathbb{R}^M = \left[\begin{array}{cccc} w_0[n] & w_1[n] & \dots & w_{M-1}[n] \end{array}\right]^T$ is an estimate of the AIR to identify. We suppose that the AIR and the vector $\mathbf{w}_n$ have the same length M.

To find the optimal filter, we need to minimize a cost function which is always built around the error signal (4.3) [59, 65]. The usual choice for this

criterion is the *mean square error* (MSE) [59]:

$$\begin{aligned} J(\mathbf{w}_n) =& \mathrm{E}\left\{e^2[n]\right\} \\ =& \mathrm{E}\left\{d^2[n]\right\} - \mathrm{E}\left\{\mathbf{w}_n^T \mathbf{x}_n d[n]\right\} - \mathrm{E}\left\{\mathbf{x}_n \mathbf{w}_n^T d[n]\right\} \\ &- \mathrm{E}\left\{\mathbf{w}_n^T \mathbf{x}_n \mathbf{x}_n^T \mathbf{w}_n\right\}. \end{aligned} \tag{4.5}$$

Let us remember that, for definition: $\sigma_d^2 = \mathrm{E}\left\{d^2[n]\right\}$ is the variance of the signal $d[n]$; $\mathbf{g}_n = \mathrm{E}\left\{\mathbf{x}_n d[n]\right\} \in \mathbb{R}^M$ is the cross-correlation between the input $\mathbf{x}_n$ and the desired signal $d[n]$; and, finally, $\mathbf{R}_n = \mathrm{E}\left\{\mathbf{x}_n \mathbf{x}_n^T\right\} \in \mathbb{R}^{M \times M}$ is the autocorrelation matrix. Equation (4.4) can be brought back in the following quadratic form [59, 146]:

$$J(\mathbf{w}_n) = \sigma_d^2 - \mathbf{w}_n^T \mathbf{g}_n - \mathbf{g}_n^T \mathbf{w}_n + \mathbf{w}_n^T \mathbf{R}_n \mathbf{w}_n \tag{4.6}$$

The optimal Wiener filter, that we denote as $\mathbf{w}^{\mathrm{opt}}$, is the one that cancels the gradient of $J(\mathbf{w}_n)$ with respect to $\mathbf{w}_n$, i.e.:

$$\nabla J(\mathbf{w}_n) = \frac{\partial J(\mathbf{w}_n)}{\partial \mathbf{w}_n} = \mathbf{0} \tag{4.7}$$

where the operator ∇ denotes the gradient. We have:

$$\begin{aligned} \nabla J(\mathbf{w}_n) &= 2\mathrm{E}\left\{e[n] \frac{\partial e[n]}{\partial \mathbf{w}_n}\right\} \\ &= -2\mathrm{E}\left\{e[n]\, \mathbf{x}_n\right\}. \end{aligned} \tag{4.8}$$

Therefore, taking into account (4.5) and (4.7), at the optimum we have:

$$\begin{aligned} \nabla J(\mathbf{w}_n) &= \frac{\partial\left(\sigma_d^2 - \mathbf{w}_n^T \mathbf{g}_n - \mathbf{g}_n^T \mathbf{w} + \mathbf{w}_n^T \mathbf{R}_n \mathbf{w}_n\right)}{\partial \mathbf{w}_n} \\ &= 2\left(\mathbf{R}_n \mathbf{w}_n - \mathbf{g}_n\right). \end{aligned} \tag{4.9}$$

Therefore, the solving system results:

$$\mathbf{R}_n \mathbf{w}_n = \mathbf{g}_n \tag{4.10}$$

which corresponds to a linear system of equations, also known as *Wiener-Hopf normal equations* [156]. The solution to (4.9), also known as *Widrow-Hopf equation* [155, 153], can be written as:

$$\mathbf{w}^{\mathrm{opt}} = \mathbf{R}_n^{-1} \mathbf{g}_n \tag{4.11}$$

Linear optimum filtering provides minimum MSE and therefore helps to estimate accurately the unknown AIR.

4.3 GRADIENT ADAPTATION

The optimal solution to (4.9) can be obtained employing a *gradient descent* optimization procedure.

4.3.1 The steepest descent method

The method of *steepest descent gradient*, as the name implies, relies on the slope at any point on the error performance surface to provide the best direction in which to move. The steepest descent direction gives the greatest change in elevation of the surface of the cost function for a given step laterally. The steepest descent procedure uses the knowledge of this direction to move to a lower point on the surface and find the bottom of the surface in an iterative manner.

The steepest descent method is based on an iterative approach for finding the parameter value associated with the minimum of the cost function: simply move the current parameter value in the direction opposite to that of the slope of the cost function at the current parameter value. Furthermore, if we make the magnitude of the change in the parameter value proportional to the magnitude of the slope of the cost function, the algorithm will make large adjustments of the parameter value when its value is far from the optimum value and will make smaller adjustments to the parameter value when the value is close to the optimum value [85]. This approach is the essence of the steepest descent algorithm.

The *steepest descent algorithm* can be defined considering a recursive solution to Wiener normal equations (4.10). The algorithm can be represented by its general form:

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \frac{1}{2}\mu\left(-\nabla J\left(\mathbf{w}_{n-1}\right)\right) \tag{4.12}$$

where the value $1/2$ is just a proportionality constant and the parameter μ is termed the *step size* of the algorithm. Note that for the steepest descent algorithms n is an iteration index and does not coincide with the time instant. Denoting $J\left(\mathbf{w}_n\right) = \mathrm{E}\left\{e^2\left[n\right]\right\}$, the explicit expression of the gradient $\nabla J\left(\mathbf{w}_n\right)$ can be easily derived from (4.6), thus resulting in (4.9). Therefore, replacing (4.9), evaluated at iteration index $n-1$, in (4.12), the explicit form of the steepest descent algorithm results:

$$\begin{aligned} \mathbf{w}_n &= \mathbf{w}_{n-1} - \mu\left(\mathbf{R}_{n-1}\mathbf{w}_{n-1} - \mathbf{g}_{n-1}\right) \\ &= \left(\mathbf{I} - \mu\mathbf{R}_{n-1}\right)\mathbf{w}_{n-1} + \mu\mathbf{g}_{n-1} \end{aligned} \tag{4.13}$$

where $\mathbf{I} \in \mathbb{R}^{M\times M}$ is an identity matrix (therefore it does not require any iteration index). Equation (4.13) is a recursive, multidimensional, finite different equation in the index n, with initial condition (i.c.) $\mathbf{w}_{-1}$ [146, 59].

4.3.2 Convergence of the steepest descent algorithm

Given that the stationary point of the steepest descent algorithm is the optimum *minimum mean square error* (MMSE) solution, a second, equally-important consideration is whether the algorithm converges at all. In order to analyze the convergence properties of the steepest descent algorithm, let us consider the misalignment vector of the filter, denoted as $\mathbf{u}_n = \mathbf{w}_n - \mathbf{w}^{\mathrm{opt}}$. Remembering (4.10), after few passages [59, 146], it results from from (4.13) that:

$$\mathbf{u}_n = \left(\mathbf{I} - \mu \mathbf{R}_{n-1}\right) \mathbf{u}_{n-1} \tag{4.14}$$

Applying the *unitary similarity transformation* [53] on the correlation matrix $\mathbf{R}_n$, it is possible to obtain:

$$\mathbf{R}_n = \mathbf{Q}_n \boldsymbol{\Lambda} \mathbf{Q}_n^T = \sum_{i=0}^{M-1} \lambda_i \mathbf{q}_{n,i} \mathbf{q}_{n,i}^T \tag{4.15}$$

where $\boldsymbol{\Lambda} = \mathrm{diag}\left(\lambda_0, \lambda_1, \ldots, \lambda_{M-1}\right)$, also known as *spectral matrix*, is the diagonal matrix containing the *eigenvalues* λ_i, with $i = 0, \ldots, M-1$, of the correlation matrix $\mathbf{R}_n$. Matrix $\mathbf{Q}_n$, defined as $\mathbf{Q}_n = \left[\begin{array}{cccc} \mathbf{q}_{n,0} & \mathbf{q}_{n,1} & \cdots & \mathbf{q}_{n,M-1} \end{array}\right]$, is known as *modal matrix* and it is composed of a set of orthogonal vectors $\mathbf{q}_{n,i}$ having unitary length, defined as *eigenvectors* of matrix $\mathbf{R}_n$. Matrix $\mathbf{Q}_n$ is orthonormal (such that $\mathbf{Q}_n^T \mathbf{Q}_n = \mathbf{I}$, i.e. $\mathbf{Q}_n^{-1} = \mathbf{Q}_n^T$).

Taking into account the decomposition (4.15), it is possible to rewrite (4.14) as:

$$\mathbf{u}_n = \left(\mathbf{I} - \mu \mathbf{Q}_{n-1} \boldsymbol{\Lambda} \mathbf{Q}_{n-1}^T\right) \mathbf{u}_{n-1}, \tag{4.16}$$

and setting $\widehat{\mathbf{u}}_n = \mathbf{Q}_n^T \mathbf{u}_n$, where $\widehat{\mathbf{u}}_n$ represents the rotated vector $\mathbf{u}_n$, it follows that:

$$\widehat{\mathbf{u}}_n = \left(\mathbf{I} - \mu \boldsymbol{\Lambda}\right) \widehat{\mathbf{u}}_{n-1} \tag{4.17}$$

Therefore, equation (4.17) consists of a set of M decoupled difference equations of the first order, such as:

$$\widehat{u}_i\left[n\right] = \left(1 - \mu \lambda_i\right) \widehat{u}_i\left[n-1\right] \tag{4.18}$$

where $n > 0$ and $i = 0, \ldots, M-1$. This last equation describe all the M *natural modes* of the steepest descent algorithm. The solution to (4.18) can be determined starting from the i.c. $\widehat{u}_i\left[-1\right]$, such that, with a backward substitution, it is possible to write:

$$\widehat{u}_i\left[n\right] = \left(1 - \mu \lambda_i\right)^n \widehat{u}_i\left[n\right]. \tag{4.19}$$

Necessary condition so that the algorithm does not diverge, and therefore for the stability of the algorithm, is that the argument of the exponent is $\left|1 - \mu \lambda_i\right| < 1$, or, equivalently:

$$0 < \mu < \frac{2}{\lambda_i}. \tag{4.20}$$

This proves that, with an appropriate choice of the step size μ satisfying (4.20), $\widehat{u}_i\left[n\right]$ tends to zero for $n \to \infty$. This implies that:

$$\lim_{n \to \infty} \mathbf{w}_n = \mathbf{w}^{\mathrm{opt}}, \qquad \forall \mathbf{w}_{-1} \ \ (\text{i.c.}). \tag{4.21}$$

It follows that the vector $\mathbf{w}_n$ converges exponentially and exactly to the optimum.

4.4 STOCHASTIC GRADIENT ADAPTIVE ALGORITHMS

The method of steepest descent can be used to find the optimum MMSE estimate of $\mathbf{w}^{\mathrm{opt}}$ in an iterative fashion. However, this procedure uses the statistics of the input and desired response signals and not on the actual measured signals. In practice, the input signal statistics are not known *a priori*. Moreover, if these statistics were known and if the autocorrelation matrix $\mathbf{R}_n$ was invertible, we could find the optimum solution given in (4.11) directly in one step! However, implementing this procedure exactly requires knowledge of the input signal statistics, which are almost always unknown for real-world problems. Therefore, an approximate version of the gradient descent procedure can be applied to adjust the adaptive filter coefficients using only the measured signals [59, 120, 85, 146].

More precisely, from equations (4.12) and (4.8), it is possible to see that the steepest descent method depends on the input data and desired response signal statistics through the expectation operation that is performed on the product $e[n]\,\mathbf{x}_n$. This product is the gradient of the squared error function $\left(e^2[n]\right)/2$ with respect to the coefficient vector $\mathbf{w}_n$. We can consider the vector $e[n]\,\mathbf{x}_n$ as an approximation of the true gradient of the MSE estimation surface. This approximation is known as the *instantaneous gradient* of the MSE surface. In order to develop a useful and realizable adaptive algorithm it is possible to replace the gradient vector $\mathrm{E}\{e[n]\,\mathbf{x}_n\}$ in the steepest descent update in (4.8) by its instantaneous approximation $e[n]\,\mathbf{x}_n$. Adaptive filters that are based on the instantaneous gradient approximation are known as *stochastic gradient adaptive filters* [59, 120, 85, 146].

4.4.1 The Least Mean Square Algorithm

The *least mean square* (LMS) algorithm is the most popular memoryless stochastic gradient algorithm. Introduced by Widrow-Hoff in 1960 [153], it consists of simply considering the instantaneous squared error $e^2[n]$ instead of its expectation. The LMS algorithm can be viewed as a stochastic approximation of the steepest descent algorithm. Another important aspect concerns with the iteration index $n$of the algorithm that, in this case, coincides with the time index [146].

Denoting with $\nabla\widehat{J}(\mathbf{w}_{n-1}) \approx \nabla J(\mathbf{w}_{n-1})$ the gradient vector estimate, the general expression of the adaptation, similarly to (4.12), turns to be:

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \frac{1}{2}\mu\left(-\nabla\widehat{J}(\mathbf{w}_{n-1})\right) \tag{4.22}$$

with an *a priori error* [59, 120], or simply named *error*, defined as:

$$\begin{aligned} e[n] &= d[n] - y[n] \\ &= d[n] - \mathbf{x}_n^T\mathbf{w}_{n-1} \end{aligned} \tag{4.23}$$

The explicit expression of the gradient vector $\nabla\widehat{J}(\mathbf{w}_{n-1})$:

$$\begin{aligned} \nabla\widehat{J}(\mathbf{w}_{n-1}) &= \frac{\partial e^2[n]}{\partial \mathbf{w}_{n-1}} = 2e[n]\,\frac{\partial e[n]}{\partial \mathbf{w}_{n-1}} \\ &= 2e[n]\,\frac{\partial\left(d[n] - \mathbf{x}_n^T\mathbf{w}_{n-1}\right)}{\partial \mathbf{w}_{n-1}} = -2e[n]\,\mathbf{x}_n \end{aligned} \tag{4.24}$$

such that the adaptation equation (4.22) simply becomes:

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \mu e[n]\,\mathbf{x}_n. \tag{4.25}$$

The algorithm is adjusted by the step size μ, which in this basis formulation is kept constant. Similarly to what done in the previous section for the steepest descent algorithm, it is possible to prove that the algorithm converges when:

$$0 < \mu < 2/\mu_{\max} \tag{4.26}$$

where $\lambda_{\max}$ represents the larger eigenvalue of the autocorrelation matrix of the input signal.

4.4.2 The Normalized Least Mean Square Algorithm

The *normalized least mean square* (NLMS) algorithm is structurally the same as the LMS, but it differs in the way that the filter coefficients are updated. In the LMS algorithm the weight adjustment is directly proportional to the amplitude of input vector samples according to (4.25). Therefore, when the vector $\mathbf{x}_n$ is large, the LMS suffers from a gradient noise amplification problem.

To overcome this problem, the adjustment applied to the weight vector at each iteration is normalized with respect to the squared Euclidean norm of $\mathbf{x}_n$ [59, 120], thus the updating rule results:

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \mu \frac{e\left[n\right]\mathbf{x}_n}{\delta_{\text{NLMS}} + \mathbf{x}_n^T\mathbf{x}_n} \tag{4.27}$$

with $0 < \mu \leq 2$: $\delta_{\text{NLMS}} > 0$ is the *regularization parameter* which prevents division by zero during initialization when $\mathbf{x}_n = \mathbf{0}$.

4.4.3 The Recursive Least Squares Algorithm

Least squares algorithms aim at the minimization of the sum of the squares of the difference between the desired signal and the model filter output [59, 120]. When new samples of the incoming signals are received at every iteration, the solution for the least squares problem can be computed in recursive form resulting in the *recursive least squares* (RLS) algorithms.

The RLS algorithm is known to pursue fast convergence even when the eigenvalue spread of the input signal correlation matrix is large. This algorithm has excellent performance when working in time-varying environments. All these advantages come with the cost of an increased computational complexity and some stability problems, which are not as critical in LMS-based algorithms [59, 120, 36]. The RLS can be classified as a Hessian-based algorithm, thus resulting an algorithm with *memory* [146, 42].

The cost function for this class of algorithms has the following expression:

$$\begin{aligned}\widehat{J}\left(\mathbf{w}_{n-1}\right) &= \sum_{i=0}^{n} \beta^{n-i}\left|e\left[n\right]\right|^2 \\ &= \sum_{i=0}^{n} \beta^{n-i}\left|d\left[i\right] - \mathbf{x}_n^T\mathbf{w}_{n-1}\right|^2\end{aligned} \tag{4.28}$$

where the constant $0 < \beta \leq 1$, defined as *forgetting factor*, takes into account the memory of the algorithm. Therefore, the cost function depends on the actual

instantaneous error and on error samples evaluated in the past iterations with a weight continuously smaller. Note that when $\beta = 1$ the RLS consider the same weight for all the past samples; in that case the algorithm has a *growing memory*.

Let us take into account the following *sequential regression notation* with an input data matrix $\mathbf{X}_n \in \mathbb{R}^{N\times M}$, where N is the length of the analysis window, defined as:

$$\mathbf{X}_n = \begin{bmatrix} \mathbf{x}_n^T \\ \mathbf{x}_{n-1}^T \\ \cdots \\ \mathbf{x}_{n-N+1}^T \end{bmatrix}^T \tag{4.29}$$

$$= \begin{bmatrix} x\left[n\right] & x\left[n-1\right] & \cdots & x\left[n-M+1\right] \\ x\left[n-1\right] & x\left[n-2\right] & \cdots & x\left[n-M\right] \\ \vdots & \vdots & \ddots & \vdots \\ x\left[n-N+1\right] & x\left[n-N\right] & \cdots & x\left[n-N-M+2\right] \end{bmatrix}$$

As a consequence the error vector and the desired signal vector are respectively defined as:

$$\begin{aligned}\mathbf{e}_n \in \mathbb{R}^N &= \begin{bmatrix} e\left[n\right] & e\left[n-1\right] & e\left[n-N+1\right] \end{bmatrix} \\ \mathbf{d}_n \in \mathbb{R}^N &= \begin{bmatrix} d\left[n\right] & d\left[n-1\right] & d\left[n-N+1\right] \end{bmatrix}\end{aligned} \tag{4.30}$$

Therefore, equation (4.28) can be expressed in the regression notation [146] as:

$$\widehat{J}\left(\mathbf{w}_{n-1}\right) = \mathbf{e}_n^T\mathbf{B}_n\mathbf{e}_n = \mathbf{B}_n\left\|\mathbf{d}_n - \mathbf{X}_n\mathbf{w}_{n-1}\right\|_2^2. \tag{4.31}$$

where $\mathbf{B}_n$ represents a weighted matrix:

$$\mathbf{B}_n = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & \ddots & \cdots & 0 \\ \vdots & \vdots & \beta^{n-1} & \vdots \\ 0 & 0 & \cdots & \beta^n \end{bmatrix}. \tag{4.32}$$

Solving for the cost function (4.31), after few passages [59, 120, 146], it is possible to achieve the following *regression equation*:

$$\mathbf{X}_n^T \mathbf{B}_n \mathbf{X}_n \mathbf{w}_{n-1} = \mathbf{X}_n^T \mathbf{B}_n \mathbf{d}_n \tag{4.33}$$

Denoting the correlation estimates as:

$$\mathbf{R}_{xx,n} = \mathbf{X}_n^T \mathbf{B}_n \mathbf{X}_n \mathbf{w}_{n-1} \quad \text{and} \quad \mathbf{R}_{xd,n} = \mathbf{X}_n^T \mathbf{B}_n \mathbf{d}_n \tag{4.34}$$

that can be also expressed as:

$$\mathbf{R}_{xx,n} = \sum_{i=0}^{n} \beta^{n-1} \mathbf{x}_i \mathbf{x}_i^T = \beta \mathbf{R}_{xx,n-1} + \mathbf{x}_i \mathbf{x}_i^T \tag{4.35}$$

$$\mathbf{R}_{xd,n} = \sum_{i=0}^{n} \beta^{n-1} \mathbf{x}_i d\left[i\right] = \beta \mathbf{R}_{xd,n-1} + \mathbf{x}_i d\left[i\right] \tag{4.36}$$

such that the correlations can be computed in a recursive way updating the estimate carried out at the past iteration with new available information. The solution of the sequential regression (4.33) at n-th time instant can be written as:

$$\mathbf{R}_{xx,n} \mathbf{w}_{n-1} = \mathbf{R}_{xd,n} \tag{4.37}$$

Applying the *matrix inversion lemma* [53, 59, 146] to the matrix (4.35) and setting $\mathbf{P}_n = \mathbf{R}_{xx,n}^{-1}$, it is possible to achieve:

$$\mathbf{P}_n = \beta^{-1} \mathbf{P}_{n-1} - \frac{\beta^{-1} \mathbf{P}_{n-1} \mathbf{x}_n \beta^{-1} \mathbf{x}_n^T \mathbf{P}_{n-1}}{1 + \beta^{-1} \mathbf{x}_n^T \mathbf{P}_{n-1} \mathbf{x}_n} \tag{4.38}$$

where for computational convenience it is usual to define the vector:

$$\mathbf{k}_n = \frac{\beta^{-1} \mathbf{P}_{n-1} \mathbf{x}_n}{1 + \beta^{-1} \mathbf{x}_n^T \mathbf{P}_{n-1} \mathbf{x}_n} \tag{4.39}$$

also known as *Kalman gain*, so that the recursion (4.38) can be written as:

$$\mathbf{P}_n = \beta^{-1} \mathbf{P}_{n-1} - \beta^{-1} \mathbf{k}_n \mathbf{x}_n^T \mathbf{P}_{n-1} \tag{4.40}$$

also known as *Riccati equation*.

The main drawback of the RLS algorithm is its computational cost, thus LMS based algorithms, while they do not perform as well as RLS, are more favourable in practical situations.

4.4.4 The Affine Projection Algorithm

The *affine projection algorithm* (APA) can be interpreted as a generalization of the NLMS algorithm. The main advantage of the APA over the NLMS algorithm consists of a superior convergence rate, especially for correlated inputs, like speech. For this reason, the APA and different versions of it were found to be very attractive choices for acoustic applications, such as AEC.

The APA, originally proposed in [98], was derived as a generalization of the NLMS algorithm, in the sense that a filter vector of the NLMS may be viewed as a one dimensional affine projection, while in the APA the projections are made in multiple dimensions. When the projection dimension increases, the convergence rate of the filter vector also increases. However, this also leads to an increased computational complexity. The APA, like the RLS is a Hessian-based algorithm, however it is not an "exact" second order adaptive algorithm since its adaptation uses an estimate of the correlation matrix $\mathbf{R}_{xx,n}$ "projected" over a subspace with appropriate dimension [146].

In order to derive the classical APA equations, let us consider an FIR adaptive filter of length M, defined by the coefficients vector $\mathbf{w}_n$, and an input data matrix defined similarly to (4.29) but using a window length N equal to $K > 0$, which is also defined as *projection order*. Therefore, the input data matrix is defined as $\mathbf{X}_n \in \mathbb{R}^{K\times M}$, while the error signal and the desired signal are respectively $\mathbf{e}_n, \mathbf{d}_n \in \mathbf{R}^K$, similarly to (4.30). This corresponds to take into account the last K samples of the input sequence. When $K = 1$ the adaptation becomes one dimensional and thus the APA turns to be an NLMS algorithm. Therefore, the equations that define the classical APA are [98]:

$$\mathbf{e}_n = \mathbf{d}_n - \mathbf{X}_n \mathbf{w}_{n-1} \tag{4.41}$$

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \mu \mathbf{X}_n^T \left(\delta_{\text{APA}} \mathbf{I} + \mathbf{X}_n \mathbf{X}_n^T\right)^{-1} \mathbf{e}_n \tag{4.42}$$

where δ_{APA} is the *regularization factor* of the APA and $\mathbf{I} \in \mathbb{R}^{K\times K}$ is an identity matrix.

We will see in the next chapter a general framework for the derivation of adaptive algorithms, both for these classical stochastic gradient algorithms and for the proportionate algorithm that will be introduced in the next chapter.

5 Proportionate Adaptive Algorithms

Contents

In the recent past, a family of proportionate adaptive filters has been proposed for use in network telephony and acoustic applications. Proportionate algorithms offer better convergence and tracking performances than standard stochastic algorithms when the echo path is sparse. In this chapter, we describe proportionate algorithms introducing an alternative perspective on proportionate adaptive filters.

5.1 INTRODUCTION

Nowadays, *acoustic echo cancellation* (AEC) is a key application in modern speech communication systems. Echo phenomena are generated in speech devices by a microphone-loudspeaker coupling, such as a far-end signal is sent out by a loudspeaker and crosses an echo path before being acquired by the microphone. Therefore, the acquired signal contains an echo contribution which may be cancelled by means of an acoustic echo canceller. The main component of an echo canceller is the adaptive filter which aims at estimating the *acoustic impulse response* (AIR). Such applications require adaptive filters with hundreds or even thousands of taps and their success depends on the nature of the AIR [57]. Often enough the impulse response is time-varying and it is affected by echo path changes, different degrees of sparseness, double-talk events and under-modelling noise [57, 12].

Classic algorithms based on stochastic gradient, such as *least mean square* (LMS) and *normalized LMS* (NLMS), distribute the adaptation energy among all filter coefficients causing a very slow convergence for long filters [120, 59]. As a result, the application of these filtering algorithms to acoustic applications becomes unpractical. In order to address this problem, in the last years it has been conceived to act on the nature of AIRs. In fact, for both network and acoustic scenarios, echo path have a specific property, which can be used in order to help the adaptation process. Indeed, these systems are sparse in nature, i.e., only a small percentage of the impulse response components have a significant magnitude while the rest are zero or small [40].

The "sparseness" character of the echo paths inspired the idea to "proportionate" the algorithm behaviour, i.e., to update each coefficient of the filter independently of the others, by adjusting the adaptation step size in proportion to the magnitude of the estimated filter coefficient. In this manner, the adaptation gain is "proportionately" redistributed among all the coefficients, emphasizing the large ones in order to speed up their convergence, and consequently to increase the overall convergence rate. This means that the region with higher energy of the sparse impulse response is adapted faster than the tail of the AIR. An example of sparse AIR can be found in Fig. 5.1, in which the difference between the early reflections and the tail of the AIR is quite clear.

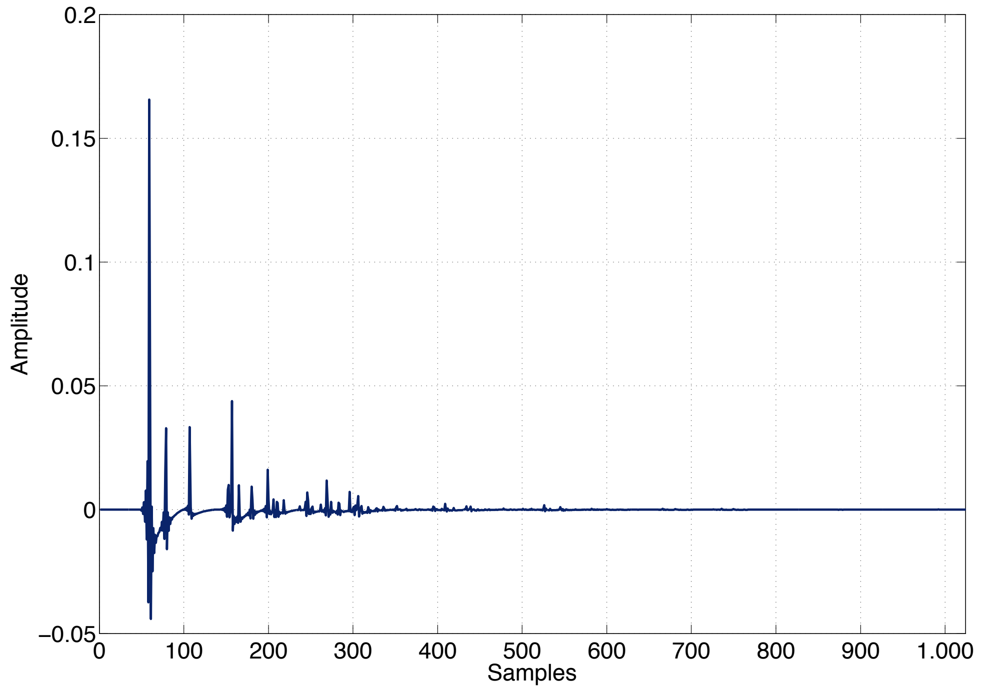


Fig. 5.1: *A sparse acoustic impulse response.*

The first proportionate algorithm was proposed by Duttweiler [40]; he defined the *Proportionate NLMS Algorithm* (PNLMS) algorithm, whose idea was to make the step size of each tap proportional to current absolute value of the estimated weight. PNLMS converges and tracks much faster than the NLMS algorithm when the impulse response that we need to identify is sparse. However, its behaviour degrades significantly when the impulse response is dispersive. PNLMS++ algorithm [50] partially solves the above mentioned problem by alternating the update process between NLMS and PNLMS. PNLMS++ seems a little bit less sensitive to the assumption of a sparse impulse response than PNLMS, so it is far from the optimal solution. In [13], the *improved PNLMS* (IPNLMS) was proposed where each step size shows a better balance between the fixed step size of NLMS and the large amount of proportionality in PNLMS. As a result, IPNLMS always converges and tracks better than NLMS and PNLMS, no matter how sparse the impulse response is.

Another filter that unevenly weights the adaptation of the different taps of the filter is the *EG± algorithm* [71], based on the exponentiated gradient adaptation. Nevertheless, it has been proved [14, 93] that IPNLMS is a very good approximation of the EG± algorithm, while being more convenient from a practical point of view. Unfortunately, as any other gradient-based adaptive filter, IPNLMS is subject to some compromises due to the selection of its parameters. As a matter of fact, a large step size results in faster convergence, while the residual misalignment is reduced for small step sizes. Moreover, the choice of the proportionality factor imposes a behaviour trade-off for channels with different degrees of sparseness [13].

In order to achieve faster convergence for a wide range of echo paths, it is possible to combine the ideas of proportionate algorithms with the general *affine projection algorithm* (APA). In [48], it is shown that a robust *proportionate affine projection algorithm* (PAPA) converge faster than NLMS and performs significantly better even during a double-talk situation. Moreover, in [119], it is proved that an *improved PAPA* (IPAPA) easily outperforms all the above mentioned proportionate algorithms and its performance does not depend on the type of the impulse response. Furthermore, the choice of a proper value for the proportionate factor has no any significant impaction on the IPAPA tracking properties comparing to the IPNLMS. Moreover, in the last years proportionate APAs have been improved [64, 152, 149, 78] until coming to an *efficient proportionate APA* [102], which takes into account the “history” of the proportionate factors.

Proportionate algorithms improve adaptive filtering performance when the AIR is sparse; however, even in proportionate algorithms some problems may occur in the choice of the parameters. A key parameter in adaptive echo cancellation is the *step size* which governs the stability and the adaptation speed of the filtering algorithm. The choice of the step size sets the trade-off between convergence, tracking ability and steady state misalignment. In order to achieve the best trade-off, several *variable step size* (VSS) algorithms have been proposed [58, 80, 124, 99]. In general, classic algorithms assume an exact modelling situation, i.e. the length of the adaptive filter is equal to the length of the system that has to be modeled. Since echo paths are extremely long, under-modelling situations, in which the length of the adaptive filter is shorter than the length of the echo path, often occur in echo cancelling applications. The residual echo due to the unmodelled part of the impulse response can be viewed as additional noise, also named under-modelling noise, that affects the performance of the algorithm. In [101], the under-modelling case has been considered.

In this chapter, we derive a novel perspective on proportionate algorithms and then we define a new block-based proportionate APA and a variation of it based on the recursive update of the covariance matrix. Furthermore, we investigate the introduction of a variable step size. The chapter is organized as follows: in Section 5.2 a new framework for the derivation of proportionate algorithms is derived. Section 5.3 introduces the derivation of algorithms using the new framework while the analytical description of the proposed *proportionate block APA* is introduced in Section 5.4. In Section 5.5, variable step size based proportionate algorithms are investigated.

5.2 AN ALTERNATIVE PERSPECTIVE ON PROPORTIONATE ADAPTIVE FILTERS

In order to give an overall description of the proportionate algorithms, we derive a general framework based on a novel perspective on the proportionate algorithms using a *natural gradient* adaptive rule [2] and employing the least perturbation property [120] by means of which we suggest the family of proportionate APA filters.

5.2.1 General properties of adaptive algorithms

Adaptive algorithms are usually introduced as an approximate iterative solution of a *global optimization problem* as they are derived, in the steepest descent implementation, by replacing the actual gradient vector with an instantaneous approximation of it (see Section 4.3). It turns out that, starting from an energy point of view and some general properties of the adaptive algorithms, it is possible to define a class of algorithms that can be seen as an *exact*, i.e. non-approximate, solution of a *local optimization problem* [120].

For this purpose, let us consider the regression vector $\mathbf{d}_n \in \mathbb{R}^K$ containing the K more recent samples of the observed desired signal:

$$\mathbf{d}_n = \begin{bmatrix} d[n] & d[n-1] & \dots & d[n-K+1] \end{bmatrix}^T \tag{5.1}$$

where K is known as *projection order* (see Section 4.4). Similarly, the data matrix of the input signal $\mathbf{X}_n \in \mathbb{R}^{K\times M}$ can be expressed as:

$$\mathbf{X}_n = \begin{bmatrix} \mathbf{x}_n^T \\ \mathbf{x}_{n-1}^T \\ \cdots \\ \mathbf{x}_{n-K+1}^T \end{bmatrix}^T \tag{5.2}$$

$$= \begin{bmatrix} x[n] & x[n-1] & \cdots & x[n-M+1] \\ x[n-1] & x[n-2] & \cdots & x[n-M] \\ \vdots & \vdots & \ddots & \vdots \\ x[n-K+1] & x[n-K] & \cdots & x[n-K-M+2] \end{bmatrix}$$

Moreover, let us assume to dispose, at n-th time instant, of some weight estimate of the previous iteration, $\mathbf{w}_{n-1}$, so that it is possible to define the *a priori* error signal:

$$\mathbf{e}_n = \mathbf{d}_n - \mathbf{X}_n \mathbf{w}_{n-1}, \tag{5.3}$$

and the *a posteriori* error signal:

$$\boldsymbol{\varepsilon}_n = \mathbf{d}_n - \mathbf{X}_n \mathbf{w}_n. \tag{5.4}$$

Introducing the step size parameter $\mu[n]$ in its general time-varying form and denoting with $\boldsymbol{\alpha}_n \in \mathbb{R}^K = \mathrm{diag}(\mu_0[n], \dots, \mu_{K-1}[n])$ the corresponding diagonal matrix, it is possible to write the relation between the *a posteriori* and the *a priori* error signals:

$$\boldsymbol{\varepsilon}_n = (\mathbf{I} - \boldsymbol{\alpha}_n)\,\mathbf{e}_n \tag{5.5}$$

It can be notice that in case of constant step size value the diagonal matrix can be written omitting the time index as $\boldsymbol{\alpha} = \mu\mathbf{I}$, where μ is the fixed step size value.

The relation (5.5), in which $\mathbf{0} < \boldsymbol{\alpha}_n < \mathbf{I}$, expresses an energy constraint between *a priori* and *a posteriori* errors, thus entailing the *passivity* of the corresponding adaptive circuit scheme.

Taking into account equation (5.5) and denoting with:

$$\widetilde{\mathbf{w}}_n = \mathbf{w}_n - \mathbf{w}_{n-1} \tag{5.6}$$

the vector that adjust the coefficients of the estimated filter, we can define a cost function as:

$$J(\mathbf{w}_n) = \|\widetilde{\mathbf{w}}_n\|_2^2 \tag{5.7}$$

Due to the fact that the filter weights at steady state no longer change during adaptation, it follows that any adaptive algorithm that minimized $J(\mathbf{w}_n)$ can be expressed as an *exact* method of local minimization, which is a constrained optimization problem:

$$\begin{aligned} \mathbf{w}^{\mathrm{opt}} &= \arg\min_{\mathbf{w}_n} \|\widetilde{\mathbf{w}}_n\|_2^2 \\ \text{subject to} \quad \boldsymbol{\varepsilon}_n &= (\mathbf{I}-\boldsymbol{\alpha}_n)\,\mathbf{e}_n \end{aligned} \tag{5.8}$$

Such optimization problem describes the steepest descent adaptation process. This process continues iteratively until the value of $J(\mathbf{w}_n)$ reaches a suitably-small value; at that point $\mathbf{w}_n$ is close to $\mathbf{w}^{\mathrm{opt}}$. With a proper selection of $\mu[n]$, the steepest descent method adjusts $\mathbf{w}_n$ in a way that $\lim_{n\to\infty} \mathbf{w}_n = \mathbf{w}^{\mathrm{opt}}$. Such an algorithm allows $\mathbf{w}_n$ to converge to $\mathbf{w}^{\mathrm{opt}}$.

Equation (5.8) represents the so-called *least perturbation property* and it is equivalent to seek a solution $\mathbf{w}_n$ that is closest to $\mathbf{w}_{n-1}$ in the Euclidean norm sense, under an equality constraint between $\mathbf{e}_n$ and ε_n. The constraint is most relevant when $\mu[n]$ is a small value, such that $(\mathbf{I}-\boldsymbol{\alpha}_n) < \mathbf{I}$, because, when the step size $\mu[n]$ is small enough, the magnitude of the *a posteriori* error $\boldsymbol{\varepsilon}_n$ will always be less than that of the *a priori* error $\mathbf{e}_n$, i.e.:

$$|\boldsymbol{\varepsilon}_n| < |\mathbf{e}_n| \tag{5.9}$$

An important consequence of the least perturbation property is that *a priori* and *a posteriori* errors tend to zero at steady state. In other words, as explained in [71], an adaptive algorithm should be characterized by a reasonable balance between the conservative (keep information gained in previous iterations) and corrective requirements (ensure that any new information gained increases the result accuracy).

Therefore, in conclusion, any adaptive algorithm can be derived and characterized taking into account the following *general properties*:

(a) the magnitude of the *a posteriori* error is always less than the *a priori* error, i.e. $|\boldsymbol{\varepsilon}_n| < |\mathbf{e}_n|$;

(b) at steady state, for $n \to \infty$, the weights no longer change during adaptation (*least perturbation property*);

(c) at steady state, for $n \to \infty$, *a priori* and *a posteriori* errors tend to zero.

5.2.2 Natural gradient adaptation

In order to take advantage from these properties, instead of the steepest descent method we may adopt a different procedure to construct the coefficient updates that takes into account the “non-isotropic nature” of the parameter space. *Natural gradient adaptation* [2], [37] is a modified gradient search that changes the standard gradient update procedure according to the non-Euclidean nature of the parameter space [51]. The resulting updates are based on a “non-straight-line” distance metric that is defined by the Riemannian geometry of the parameter space [3], [3]. According to the natural gradient procedure, the cost function in (5.7) can be rewritten as:

$$\begin{aligned} J(\mathbf{w}_n) &= \|\widetilde{\mathbf{w}}_n\|_{\mathbf{G}_n}^2 \\ &= \widetilde{\mathbf{w}}_n^T \mathbf{G}_n \widetilde{\mathbf{w}}_n \end{aligned} \tag{5.10}$$

where $\mathbf{G}_n \in \mathbb{R}^{M\times M}$ is a *Riemannian metric tensor*, which is a positive-definite matrix, whose entries at n-th time instant depend on the coefficients of the filter at time instant $n-1$. The Riemannian metric tensor characterizes the intrinsic curvature of a particular manifold in M-dimensional space. In the case of the Euclidean space the Riemannian tensor is the identity matrix $\mathbf{G}_n = \mathbf{I}$, such that (5.10) reduces to (5.7).

Before recasting the least perturbation property with the use of the Riemannian metric tensor, let us consider the following aspect. The formalization in (5.8) of the least perturbation property has merely theoretical significance as it is based on the knowledge of *a priori* and *a posteriori* errors. For a more constructive use of the general properties (a)-(c), it is necessary to define the energy constraint as function of the only *a priori* error. Left multiplying both sides of (5.6) with $\mathbf{G}_n\mathbf{X}_n$ and then adding and subtracting the desired signal vector $\mathbf{d}_n$ defined in (5.1), it is possible to express the energy constraint in (5.5) just as a function of the *a priori* error. That is:

$$\begin{aligned}\mathbf{G}_n\mathbf{X}_n\widetilde{\mathbf{w}}_n &= \mathbf{G}_n\mathbf{X}_n\mathbf{w}_n - \mathbf{G}_n\mathbf{X}_n\mathbf{w}_{n-1} \\ &= \mathbf{G}_n\left[-\left(\mathbf{d}_n - \mathbf{X}_n\mathbf{w}_n\right) + \left(\mathbf{d}_n - \mathbf{X}_n\mathbf{w}_{n-1}\right)\right] \\ &= \mathbf{G}_n\left(-\boldsymbol{\varepsilon}_n + \mathbf{e}_n\right) \\ &= \mathbf{G}_n\boldsymbol{\alpha}_n\mathbf{e}_n.\end{aligned} \tag{5.11}$$

Hence, we can formally rewrite the least perturbation property (5.8) as:

$$\begin{aligned}\mathbf{w}^{\text{opt}} &= \arg\min_{\mathbf{w}_n}\|\widetilde{\mathbf{w}}_n\|^2_{\mathbf{G}_n} \\ \text{subject to}\quad \mathbf{G}_n\mathbf{X}_n\widetilde{\mathbf{w}}_n &= \mathbf{G}_n\boldsymbol{\alpha}_n\mathbf{e}_n.\end{aligned} \tag{5.12}$$

The update equation can be straightly derived solving the system relative to the constraint (5.11). Thus, it results:

$$\widetilde{\mathbf{w}}_n = \left(\mathbf{G}_n\mathbf{X}_n\right)^{\#}\boldsymbol{\alpha}_n\mathbf{e}_n \tag{5.13}$$

where $(\mathbf{G}_n\mathbf{X}_n)^{\#}$ is a pseudo-inverse matrix. Expliciting $\widetilde{\mathbf{w}}_n$ we can write:

$$\mathbf{w}_n - \mathbf{w}_{n-1} = \left(\mathbf{G}_n\mathbf{X}_n\right)^T\left(\mathbf{X}_n\mathbf{G}_n\mathbf{X}_n^T\right)^{-1}\boldsymbol{\alpha}_n\mathbf{e}_n. \tag{5.14}$$

Inserting the regularization parameter δ, we achieve the general update equation of the family of *normalized natural gradient* (NNG) algorithms:

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \left(\mathbf{G}_n\mathbf{X}_n\right)^T\left(\delta\mathbf{I} + \mathbf{X}_n\mathbf{G}_n\mathbf{X}_n^T\right)^{-1}\boldsymbol{\alpha}_n\mathbf{e}_n. \tag{5.15}$$

In case of Euclidean space, when $\mathbf{G}_n = \mathbf{I}$, for a unitary projection order, i.e. $K = 1$, and a fixed step size, i.e. each diagonal element of $\boldsymbol{\alpha}$ is equal to a fixed scalar value μ, the update equation (5.15) describes the *normalized least mean square* (NLMS) algorithm:

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \mu\frac{\mathbf{x}_n e\left[n\right]}{\mathbf{x}_n^T\mathbf{x}_n + \delta_{\text{NLMS}}} \tag{5.16}$$

On the other hand, when the projection order is $K > 1$, equation (5.15) yields the *affine projection algorithm* (APA) in its standard form [98] in case of Euclidean space, or the *natural APA* (NAPA) [65], in case of Riemannian space.

5.3 DERIVATION OF PROPORTIONATE ALGORITHMS

Starting from equation (5.15), it is possible to derive a complete formulation of the class of proportionate algorithms. Different proportionate algorithms can be obtained simply changing the projection order K and the Riemannian tensor $\mathbf{G}_n$. In particular, in proportionate algorithms, the Riemannian tensor is consider as a full-blown sparseness constraint which weight the input signal; this is why $\mathbf{G}_n$ is called *proportionate matrix*.

The simplest proportionate algorithm is the *proportionate normalized least mean squares* in its improved version (IPNLMS) [13], whose derivation can be achieved choosing a unitary projection order $K = 1$ and a diagonal proportionate matrix $\mathbf{G}_n \in \mathbb{R}^{M\times M}$ built up in order to adjust the step sizes of the individual taps of the filter in a way that each step size turns out to be proportional to the corresponding filter coefficient:

$$\mathbf{G}_n = \mathit{diag}\left\{g_0\left[n\right], \ldots, g_{M-1}\left[n\right]\right\} \tag{5.17}$$

The diagonal elements at n-th time instant are computed from the estimate of the filter coefficients at time instant $n-1$ in such a way that a larger coefficient receives a larger increment, thus increasing the convergence rate of the coefficient. The result is that active coefficients are adjusted faster than non-active coefficients. Hence, proportionate algorithms converge much faster than classic algorithms for sparse impulse responses.

The choice of diagonal elements differentiates proportionate NLMS algorithms proposed in literature [40, 50, 13]. However, the most efficient choice, which exploits the "proportionate" idea better than other PNLMS algorithms, is the one proposed in the IPNLMS [13]. According to that, diagonal elements are:

$$g_l\left[n\right] = \frac{1-\alpha_{\mathrm{p}}}{2M} + \left(1+\alpha_{\mathrm{p}}\right)\frac{\left|w_l\left[n-1\right]\right|}{2\left\|\mathbf{w}_{n-1}\right\|_1+\xi} \tag{5.18}$$

where:

$$\left\|\mathbf{w}_{n-1}\right\|_1 = \sum_{l=0}^{M-1}\left|w_l\left[n-1\right]\right| \tag{5.19}$$

In (5.18), the coefficient index $l = 0,\dots,M-1$ and ξ is a small positive number which avoids divisions by zero; the *proportionality factor* α_{p} balances the proportionality and its recommended value is 0 or -0.5 [13]. For $\alpha_{\mathrm{p}} = -1$, the IPNLMS is equal to NLMS. For α_{p} close to 1, the IPNLMS behaves like the PNLMS. The regularization parameter δ_{p} in IPNLMS is chosen as:

$$\delta_{\mathrm{p}} = \frac{1-\alpha_{\mathrm{p}}}{2M}\delta_{\mathrm{NLMS}}. \tag{5.20}$$

Similarly to the development of PNLMS and IPNLMS, if we consider a projection order $K > 1$, we can derive the *proportionate affine projection algorithm* (PAPA) [48] and the *improved PAPA* (IPAPA) [64, 119]. However, we describe an efficient version of proportionate APA which considers the "history" of the proportionate factors [102]. Besides the projection order, the relevant difference of the proportionate APA compared to IPNLMS is the construction of $\mathbf{G}_n$. In fact, the proportionate matrix for $K > 1$ can be built up as a rectangular matrix, that we denote as $\mathbf{G'}_n \in \mathbb{R}^{K\times M}$ to distinguish from (5.17), in which the first row contains the proportionate weight computed at n-th time instant, $\mathbf{g}_n \in \mathbb{R}^M = \begin{bmatrix} g_0\left[n\right] & \dots & g_{M-1}\left[n\right]\end{bmatrix}$, while the other $K-1$ rows contain the previous $K-1$ realizations of $\mathbf{g}_n$:

$$\mathbf{G'}_n = \begin{bmatrix} \mathbf{g}_n^T \\ \mathbf{g}_{n-1}^T \\ \dots \\ \mathbf{g}_{n-K+1}^T \end{bmatrix}. \tag{5.21}$$

The matrix product in (5.15) can be written in this case as a Hadamard product:

$$\begin{aligned}\mathbf{\Gamma}_n &= \mathbf{G'}_n \odot \mathbf{X}_n \\ &= \begin{bmatrix} \mathbf{g}_n^T \odot \mathbf{x}_n^T \\ \mathbf{g}_{n-1}^T \odot \mathbf{x}_{n-1}^T \\ \dots \\ \mathbf{g}_{n-K+1}^T \odot \mathbf{x}_{n-K+1}^T \end{bmatrix}\end{aligned} \tag{5.22}$$

where the operator $\odot$ denotes the Hadamard product, i.e. $\mathbf{a} \odot \mathbf{b} = \begin{bmatrix} a_0b_0 & a_1b_1 & \dots & a_{M-1}b_{M-1}\end{bmatrix}^T$, being $\mathbf{a}$ and $\mathbf{b}$ two vectors of length M. Therefore, using (5.22), the update equation of (5.15) can be rewritten in case of PAPA algorithms as:

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \boldsymbol{\alpha}\mathbf{\Gamma}_n^T\left(\delta_{\mathrm{p}}\mathbf{I} + \mathbf{\Gamma}_n\mathbf{X}_n^T\right)^{-1}\mathbf{e}_n. \tag{5.23}$$

Due to the fact that equation (5.21) takes into account the past $K-1$ realization of the proportionate elements, the PAPA described in (5.23) can be considered as an efficient algorithm since this "proportionate memory" increases its performance [102].

Another advantage of the PAPA in (5.23) is the lower computational complexity compared with the classical proportionate-type APA, such as [48, 64, 119]. This is because the matrix $\mathbf{\Gamma}_n$ in (5.22) can be realized recursively, since it contains $K-1$ rows, whose products are computed in previous iterations. Thus, the rows from 1 to $K-1$ of the matrix $\mathbf{\Gamma}_{n-1}$ can be used directly for computing the matrix $\mathbf{\Gamma}_n$, i.e. they become the rows from 2 to K of $\mathbf{\Gamma}_n$.

This is not the case of the classical proportionate-type APA, where all the rows of $\mathbf{\Gamma}_n$ have to be evaluated at each iteration, because all of them are multiplied with the same vector $\mathbf{g}_n$. Concluding, the evaluation of $\mathbf{\Gamma}_n$ in the classical proportionate APAs needs KM multiplications, while the evaluation of $\mathbf{\Gamma}_n$ from (5.22), i.e. considering the "proportionate memory", requires only M multiplications. This advantage becomes more apparent when the projection order increases. Moreover, the fact that $\mathbf{\Gamma}_n$ has the time-shift property, like the data matrix $\mathbf{X}_n$, could be a possible opportunity to establish a link with the *fast APA* [52, 140]. It is also likely possible to derive efficient ways to compute the linear system involved in (5.23). This point in particular will address in the next section.

5.4 PROPORTIONATE BLOCK APA

In this section we propose a variation of the PAPA described in (5.23) based on the block processing of the input signal [141, 11, 1, 115]. Block processing is an effective approach to reduce the computational complexity, however in proportionate case it may assume a further sense due to the time-shift properties of the proportionate input matrix. In fact, in sample-by-sample PAPA the time-shift property of the input matrix is the same of the proportionate matrix allowing a computational saving; however, in *proportionate block APA* (PBAPA), the proportionate matrix is still subjected to the same shifting of PAPA, while the input matrix is subjected to a shift equal to the length of block. This may initially appear as a drawback since at each iteration the whole data matrix has to be weighted by the whole proportionate matrix, thus loosing the computational advantage of PAPA. Moreover, PBAPA may show a slower convergence rate due to less frequent updating of the adaptive filter. However, choosing a block length equal to the projection order, the computation cost remains the same of PAPA due to the fact that the block processing requires $1/K$ of the iterations compared to PAPA, thus the computational cost results $KM/K = M$. Moreover, the different time-shifting properties of $\mathbf{X}_n$ and

$\mathbf{G}_n$ can be seen as an interpolation of the proportionate weighting over input blocks, and this may improve the steady-state behaviour of the filter compared to PAPA.

The update equation of PBAPA is similar to (5.23); however, at each iteration the input data matrix (5.2) does not receive just a sample but a block of K samples. Moreover, a further correction term can be introduced to narrow the convergence gap with the sample-by-sample PAPA and to develop a *fast* version of PBAPA. In addition, it can be also possible to derive efficient ways to compute the inversion of the covariance matrix by means of recursive techniques, following what done in [141, 134].

5.5 VARIABLE STEP SIZE PROPORTIONATE ALGORITHMS

The overall performance of proportionate algorithms is governed by the step size parameter, which controls the filter trade-off between convergence, tracking ability and steady state misalignment. A constant value of the step size can set *a priori* performance compromise, however, it is not an optimal solution and in many cases it can produce not satisfying performance. In particular, this may occur in acoustic applications, in which nonstationary signals, such as speech, may alter initial conditions. In order to address this compromise, a *variable step size* (VSS) may be adopted. Therefore, even for proportionate algorithms, a performance improvement may be expected using a variable step size. Considering a variable step size, each element of the diagonal matrix $\boldsymbol{\alpha}_n$ in (5.15) may be different from the others, being time-varying.

In this section we derive the overall formulation of VSS-based proportionate algorithms, starting from equation (5.15), from which it is possible to derive the VSS-IPNLMS or the VSS-PAPA in its several versions. We generalize the proportionate algorithms in order to achieve a better robustness also in nonstationary conditions, double-talk events, path changes and under-modelling situations of the impulse response. For this purpose we introduce the *generalized variable step size proportionate algorithm for under-modelling scenarios*.

Under-modelling situations occur when the length of the adaptive filter is shorter than the length of the echo path, and this is often the rule in acoustic applications where AIRs are extremely long for a real-time adaptation. Under-modelling an AIR may introduce an additional noise to the near-end signal, generated by the part of the system that cannot be modelled. The power of the under-modelling noise cannot be estimated in a direct way due to the fact that it is not available in a real scenario. Therefore, its contribution cannot be evaluated.

Denoting with M_{A} the length of the acoustic impulse response $\mathbf{w}_0$, let us consider an under-modelling situation in which $M < M_{\mathrm{A}}$; it is possible to break up the data input matrix in the following way:

$$\mathbf{X}_{\mathrm{UM},n} = \begin{bmatrix} \mathbf{X}_n & \mathbf{X}_{\mathrm{A},n} \end{bmatrix} \tag{5.24}$$

where $\mathbf{X}_n$ is defined as (5.2), and $\mathbf{X}_{\mathrm{A},n} \in \mathbb{R}^{K\times(M_A-M)}$ is the data matrix referred to the under-modelled part of the AIR:

$$\mathbf{X}_{\mathrm{A},n} = \begin{bmatrix} \mathbf{x}_{\mathrm{A},n}^T \\ \mathbf{x}_{\mathrm{A},n-1}^T \\ \cdots \\ \mathbf{x}_{\mathrm{A},n-K+1}^T \end{bmatrix}^T \tag{5.25}$$

$$= \begin{bmatrix} x[n-M] & x[n-M-1] & \cdots & x[n-M_{\mathrm{A}}+1] \\ x[n-M-1] & x[n-M-2] & \cdots & x[n-M_{\mathrm{A}}] \\ \vdots & \vdots & \ddots & \vdots \\ x[n-M-K+1] & x[n-M-K] & \cdots & x[n-K-M_{\mathrm{A}}+2] \end{bmatrix}$$

Similarly, in an under-modelling scenario, we can split the AIR in two parts, a modelled part and an unmodelled one:

$$\mathbf{w}_{0,\mathrm{UM}} = \begin{bmatrix} \mathbf{w}_0 & \mathbf{w}_{0,\mathrm{A}} \end{bmatrix} \tag{5.26}$$

where:

$$\mathbf{w}_0 = \begin{bmatrix} w_{0,0} & w_{0,1} & \ldots & w_{0,M-1} \end{bmatrix} \tag{5.27}$$

and:

$$\mathbf{w}_{0\mathrm{A}} = \begin{bmatrix} w_{0,M} & w_{0,M+1} & \ldots & w_{0,M_A-1} \end{bmatrix}. \tag{5.28}$$

Let us note that the AIR vectors do not have any time index since they are assumed to be time invariant.

As a consequence, taking into account K subsequent realizations, the resulting echo path in under-modelling case, that we denote as $\overline{\mathbf{x}}_{\mathrm{UM},n} \in \mathbb{R}^K$, can be decomposed in a modelled term $\overline{\mathbf{x}}_n$ and an unmodelled term $\overline{\mathbf{x}}_{\mathrm{A},n}$, which represents the under-modelling noise:

$$\begin{aligned} \overline{\mathbf{x}}_{\mathrm{UM},n} &= \overline{\mathbf{x}}_n + \overline{\mathbf{x}}_{\mathrm{A},n} \\ &= \mathbf{X}_n\mathbf{w}_0 + \mathbf{X}_{\mathrm{A},n}\mathbf{w}_{0\mathrm{A}} \end{aligned} \tag{5.29}$$

The term $\overline{\mathbf{x}}_{\mathrm{A},n}$ acts like an additional noise for the adaptive process, so that the desired signal in under-modelling case can be rewritten as:

$$\mathbf{d}_{\mathrm{UM},n} = \overline{\mathbf{x}}_n + \overline{\mathbf{x}}_{\mathrm{A},n} + \mathbf{q}_n \tag{5.30}$$

where $\mathbf{q}_n$ is the near-end contribution which can be composed of a near-end speech signal $\mathbf{s}_n$ and a near-end background noise $\mathbf{v}_n$. In (5.30) we assume that $\overline{\mathbf{x}}_n$ and $\overline{\mathbf{x}}_{\mathrm{A},n}$ are uncorrelated. Now, squaring and then taking the expectations of both sides of (5.30) results in:

$$\mathrm{E}\left\{\mathbf{d}^2_{\mathrm{UM},n}\right\} = \mathrm{E}\left\{\overline{\mathbf{x}}^2_n\right\} + \mathrm{E}\left\{\overline{\mathbf{x}}^2_{\mathrm{A},n}\right\} + \mathrm{E}\left\{\mathbf{q}^2_n\right\} \tag{5.31}$$

Moreover, according to the least perturbation property (5.8), we assume that filter coefficients converge at steady-state, thus:

$$\mathrm{E}\left\{\overline{\mathbf{x}}^2_n\right\} \approx \mathrm{E}\left\{\mathbf{y}^2_n\right\} \tag{5.32}$$

where $\mathbf{y}_n = \mathbf{X}_n \mathbf{w}_{n-1}$ is adaptive filter output signal. As a consequence,

$$\mathrm{E}\left\{\overline{\mathbf{x}}^2_{\mathrm{A},n}\right\} + \mathrm{E}\left\{\mathbf{q}^2_n\right\} = \mathrm{E}\left\{\mathbf{d}^2_{\mathrm{UM},n}\right\} - \mathrm{E}\left\{\mathbf{y}^2_n\right\}. \tag{5.33}$$

Moreover, it is possible to assume that at steady-state the noise contributions converge to the *a posteriori* error, defined in (5.4), so taking into account the energy relation (5.5) it is possible to write:

$$\begin{aligned}\mathrm{E}\left\{\overline{\mathbf{x}}^2_{\mathrm{A},n}\right\} + \mathrm{E}\left\{\mathbf{q}^2_n\right\} &\approx \mathrm{E}\left\{\boldsymbol{\varepsilon}^2_n\right\} \\ &= \left(\mathbf{I} - \boldsymbol{\alpha}_n\right) \mathrm{E}\left\{\mathbf{e}^2_n\right\}.\end{aligned} \tag{5.34}$$

Therefore, replacing (5.34) in (5.33), it is possible to derive an expression of the variable step size parameter vector:

$$\boldsymbol{\alpha}_n = \mathbf{I} - \sqrt{\frac{\mathrm{E}\left\{\mathbf{d}^2_{\mathrm{UM},n}\right\} - \mathrm{E}\left\{\mathbf{y}^2_n\right\}}{\mathrm{E}\left\{\mathbf{e}^2_n\right\}}}. \tag{5.35}$$

From a practical point of view, we evaluate the expectations in terms of power estimates, thus each diagonal element of $\boldsymbol{\alpha}_n$ can be written as:

$$\mu_l\left[n\right] = \left|1 - \frac{\sqrt{\left|\widehat{\sigma}^2_{\mathrm{d}}\left[n-l\right] - \widehat{\sigma}^2_{\mathrm{y}}\left[n-l\right]\right|}}{\widehat{\sigma}^2_{\mathrm{e}}\left[n-l\right] + \zeta}\right| \tag{5.36}$$

where $l = 0, \ldots, K-1$. Let us note that in order to make the reading clearer, in (5.36) and in the following we omit the subscript "UM" for the desired signal.

The general parameter $\widehat{\sigma}^2_\theta\left[n\right]$ represents the power estimate of the sequence $\theta\left[n\right]$, considering $\theta = \{d, y, e\}$ and can be computed as:

$$\widehat{\sigma}^2_\theta\left[n\right] = \beta\widehat{\sigma}^2_\theta\left[n-1\right] + \left(1-\beta\right)\theta^2\left[n\right] \tag{5.37}$$

where β is a forgetting factor chosen as $\beta = 1 - 1/\left(QM\right)$, with $Q > 1$. The initial value is $\widehat{\sigma}^2_\theta\left[0\right] = 0$. Furthermore, a small positive number ζ should be added in (5.37) to avoid division by zero. In order to satisfy the steady-state approximation (5.34), as suggested in [101], the process starts using a fixed step size value for the first M iterations when the estimate of the coefficients may be influenced only by the system noise $v\left[n\right]$. However, even if we do not consider this "trick", the experimental results will prove that performance degradation is not very significant, especially when the value of the projection order is increased [100]. Another practical consideration is that the computation of the power estimates in (5.36) could lead to minor deviations from the previous theoretical conditions; this is the reason why in (5.36) we consider the absolute value of the step size parameter. Nevertheless, when echo path changes occur, the power of the estimate of the echo signal $\widehat{\sigma}^2_y\left[n\right]$ may be larger than the power of the desired signal $\widehat{\sigma}^2_d\left[n\right]$. This is the reason why, in order to avoid complex values, in (5.36) we take also the absolute value of the difference under the square root.

6

Acoustic Interfaces Exploiting Sparsity Constraints: An Experimental Study

Contents

THE objective of this chapter is to present by means of simulations the most important features of the proportionate adaptive algorithms described in the previous chapter. In particular, we analyzed the behaviour of those algorithms in several conditions and we investigate the performance of the proposed variations in order to give an overall description

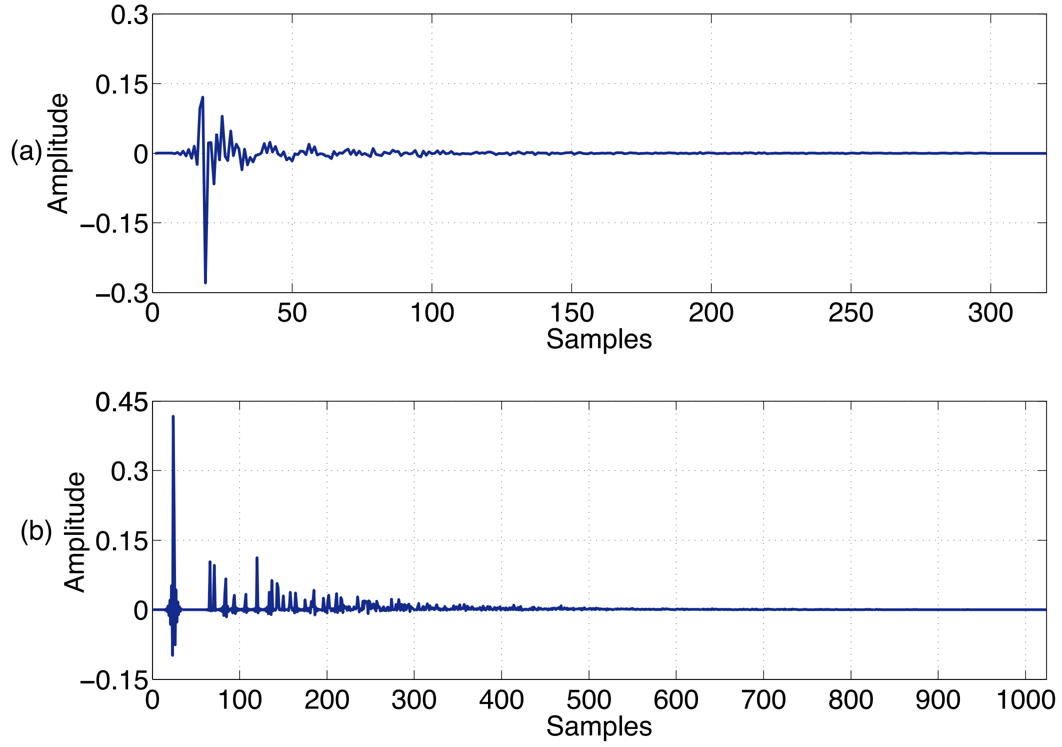


Fig. 6.1: *Acoustic impulse responses used in simulations. (a) Real AIR measured in a low reverberant room. (b) Simulated AIR with a reverberation time of 130 ms.*

of the effectiveness of proportionate algorithms. Most of the experiments are conducted in acoustic echo cancellation scenarios, which allows to better comprehend the capabilities of the algorithms.

6.1 AEC EXPERIMENTAL CONDITIONS

In this first part of the chapter we show experiments conducted in the context of echo cancellation since it is the best acoustic application to evaluate the effectiveness of an adaptive algorithm.

Experimental simulations in an exact modelling case were performed using a real echo path measured using a low-cost loudspeaker inside a room with short reverberation time. This AIR is composed of 320 coefficients and

it is depicted in Fig. 6.1 (a). When we have considered an under-modelling scenario, experiments have been conducted using a different AIR, simulated by means of a Matlab tool, *Roomsim* [24], and is measured by using an 8 kHz sampling rate. This simulated AIR has been achieved considering a $(10 \times 6, 6 \times 3)$ m room with a reverberation time of $T_{60} \approx 130$ ms. It consists of 1024 coefficients; however when we consider an under-modelling filter we truncate it after the first 512 coefficients. The simulated AIR is depicted in its total length in Fig. 6.1 (b).

The far-end signal, i.e. the input signal, is either a white Gaussian noise signal or a female speech signal. The output of the echo path is corrupted by an independent white Gaussian noise (which simulates the near-end background noise) providing a *signal-to-noise ratio* (SNR) of 20 dB. All the signals are evaluated over a length of 10 seconds. Most of the simulations are conducted in a single-talk case, i.e. in absence of near-end speech input; however, we also use a double-talk scenario to evaluate VSS-based algorithms.

In addition, we want to prove the effectiveness of the algorithms even in adverse environment conditions, in which the acoustic environment changes due to a nonstationary source or to an alteration in the environmental conditions. In order to introduce an abrupt change in the acoustic environment we shift the AIR circularly to the right by 20 samples, 5 seconds after the start of the adaptive process.

In order to have a fair comparison we use, where possible, the same parameter setting for all the algorithms. Performance are evaluated in terms of *normalized misalignment* and in many cases also in terms of *ERLE* (see Section 3.4).

6.2 PERFORMANCE ADVANTAGES OF PROPORTIONATE FILTERS

6.2.1 Simplest scenario: exact path modelling in absence of near-end speech

In the first set of experiments, we evaluate the performance of proportionate algorithms with respect to the correspondent classic ones. We start our analysis taking into account the simplest algorithms (having unitary projection order) introduced in Chapter 5, i.e. the *normalized least mean squares* (NLMS) (5.16) and its proportionate version that we denote as IPNLMS, as its original indication [13]. We consider an exact modelling scenario in absence of near-end speech; the AIR used for these simulations is the one represented in Fig. 6.1 (a). We use the same parameter setting for both the algorithms: a step size value $\mu = 0.2$ and a proportionality factor of $\alpha = 0$; in addition, we choose a regularization parameter of $\delta_{\text{NLMS}} = 30\sigma_x^2$ for the NLMS, where σ_x^2 is the input signal variance, and a regularization parameter for IPNLMS δ_{p} according to (5.20). When the far-end signal is white Gaussian noise it is simple to certify a performance improvement of IPNLMS compared to NLMS in terms of convergence rate, as it is possible to see from the behaviour of the normalized misalignment in Fig. 6.2. The difference between NLMS and IPNLMS is more evident when the far-end signal is a speech input. Performance of IPNLMS are clearly improved in terms of filter misalignment, depicted in Fig. 6.3; moreover, an evident advantage results in the quantity of cancelled echo, i.e. in terms of ERLE, as it is possible to see in Fig. 6.4.

In Fig. 6.5 we evaluate the misalignment performance of a selection of PAPA algorithms with different projection order in case of speech input.

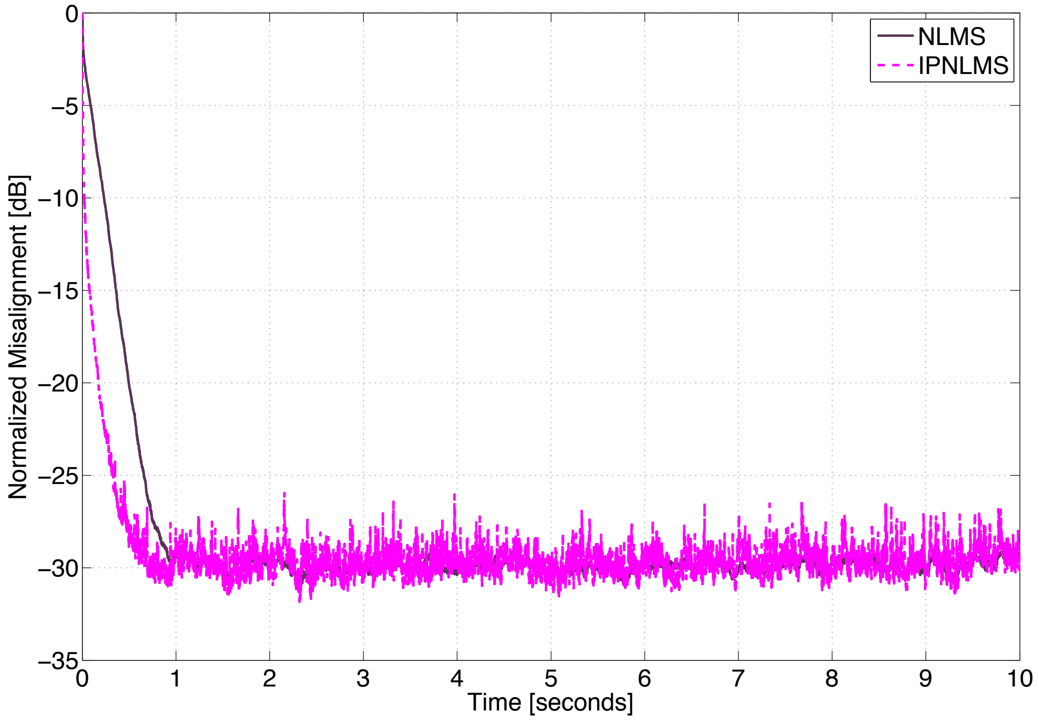


Fig. 6.2: *Misalignment of NLMS and IPNLMS algorithms with a white Gaussian noise input.*

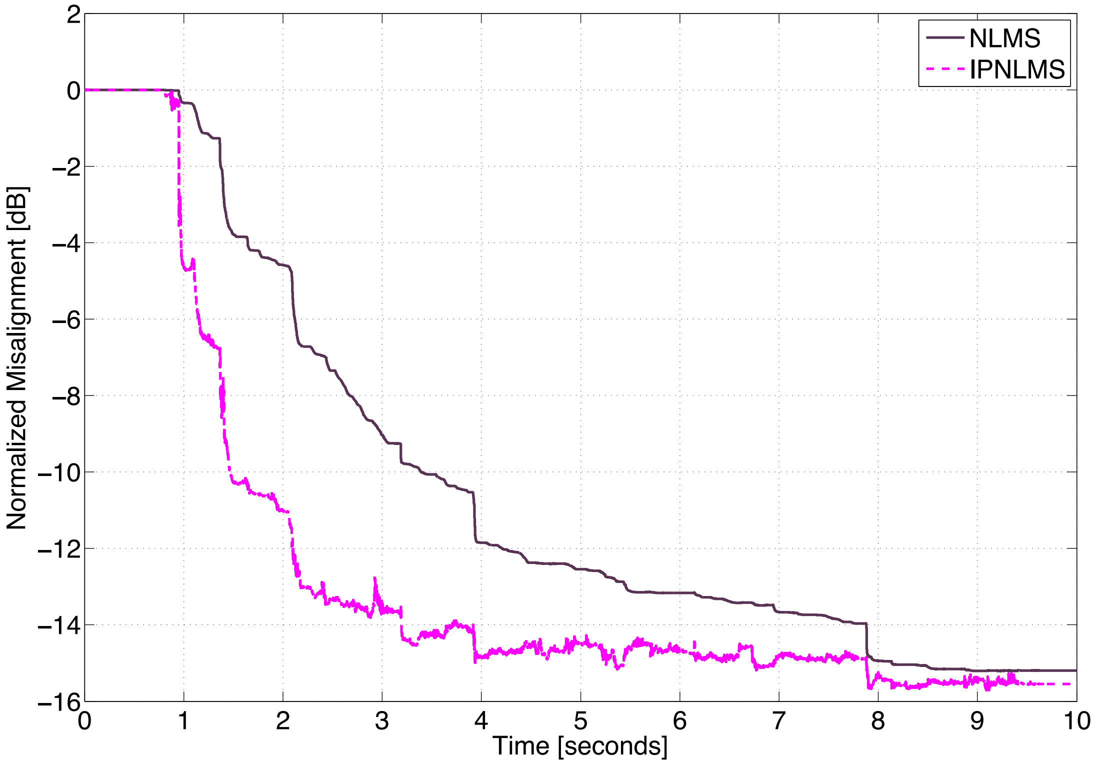


Fig. 6.3: *Misalignment of NLMS and IPNLMS algorithms with a female speech input.*

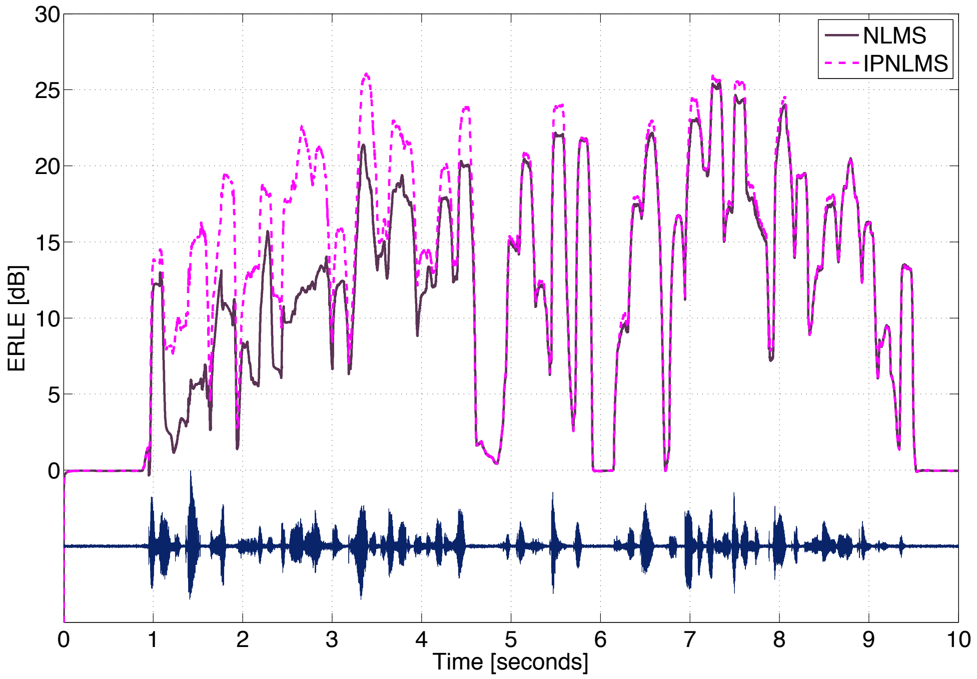


Fig. 6.4: *ERLE of NLMS and IPNLMS algorithms with a female speech input. The speech signal is reported for clearness.*

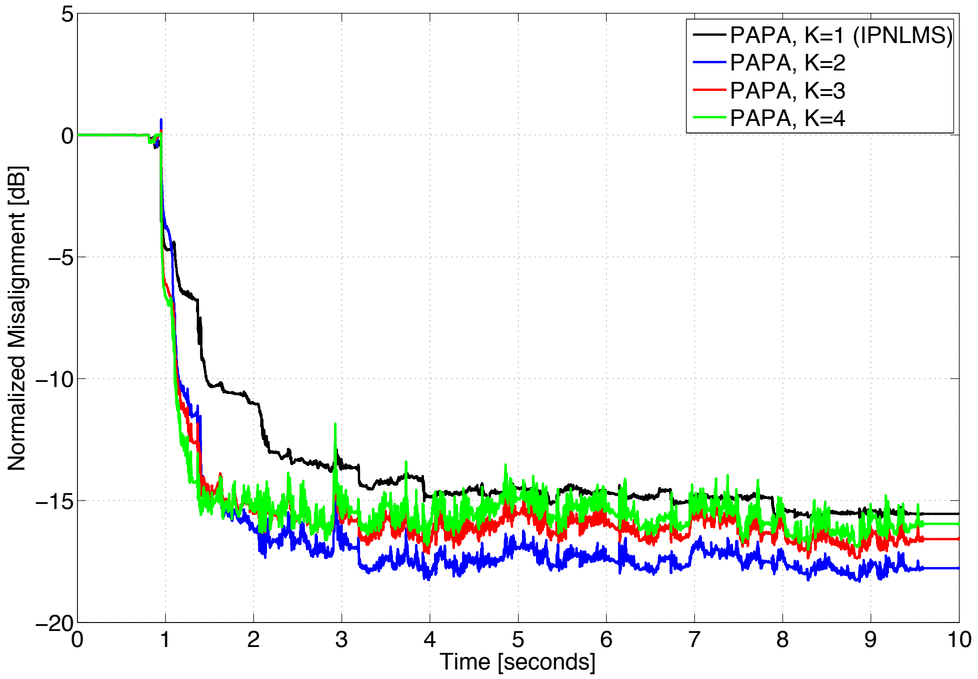


Fig. 6.5: *Misalignment comparison of PAPA algorithms with different projection order in case of female speech input.*

Let us note that in this case we evaluate only the speech input since the whitening capabilities of APA algorithms are obviously not evident when the input signal is already a white signal. From Fig. 6.5 we gather that satisfactory results can be obtained with a projection order equal to $K = 2$, or $K = 3$ at most.

We have also investigated the behaviour of the PBAPA (see Section 5.4). We report the comparison between PAPA and PBAPA in Fig. 6.6 in terms of filter misalignment when the input signal is speech. For both the algorithms we use a projection order of $K = 2$.

The behaviour of PBAPA misalignment confirms as said in Section 5.4: due to its structure the PBAPA overcomes PAPA misalignment at steady-state while showing poorer convergence performance. Due to this result we can say that PBAPA could be suited for applications with quite stationary conditions; however, if we consider AEC scenarios with adverse environment conditions

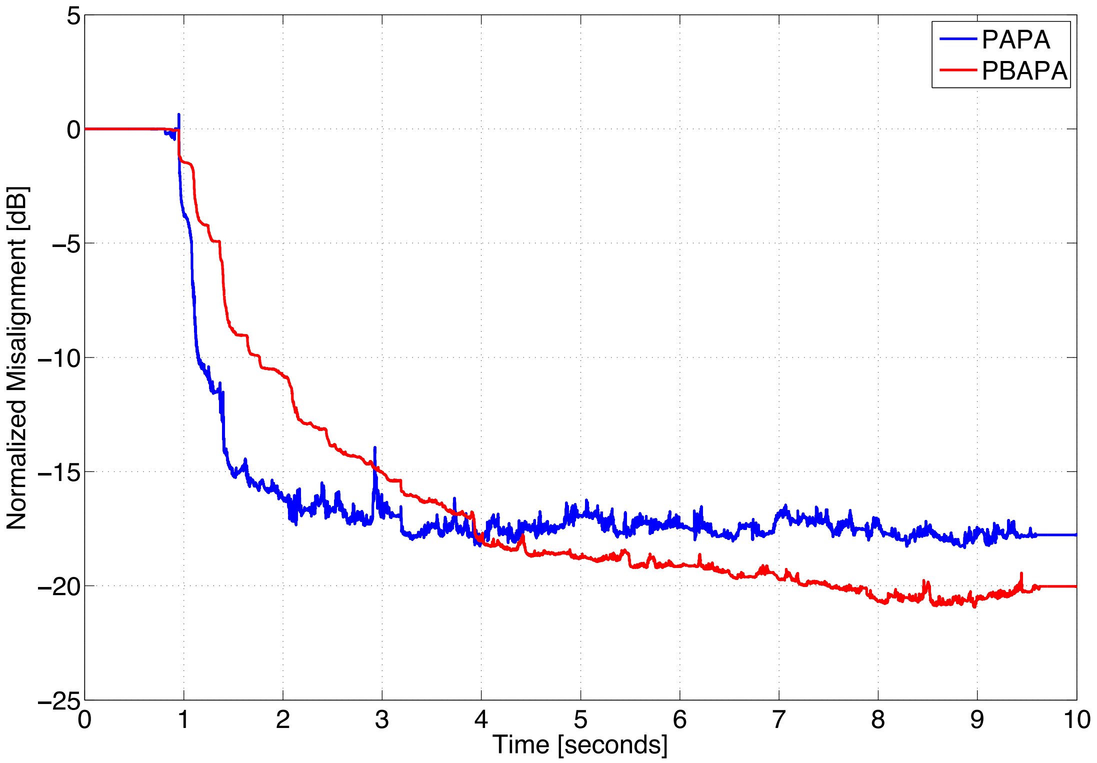


Fig. 6.6: *Misalignment comparison between PAPA and PBAPA algorithms.*

we still prefer the PAPA.

6.2.2 Exact modelling scenario in adverse environment

In this set of experiments we consider worse environment conditions respect to experiments conducted in the previous section. In a real AEC scenario several factors can be involved, thus altering the environment conditions, a source position change rather than an alteration of the environment temperature or the sudden presence of a new interfering source. When such an alteration occurs, the filter has to be readapted, so in order to achieve performance improvements an adaptive algorithm must have good tracking capabilities, i.e. a faster convergence rate in readapting.

We repeat some of the previous most representative experiments only changing the environment conditions, and in particular introducing a path change, due to an alteration in the environment, which occurs 5 seconds after the start of the adaptive process. In case of speech input it is possible to see in Fig. 6.7 that misalignment performance improvement of IPNLMS results more evident in adverse environment conditions compared to the simpler scenario in Fig 6.3. Comparing Fig. 6.4 and Fig. 6.8, when a path change occurs improvements even increase in terms of ERLE, since the behaviour of IPNLMS always keeps an advantage margin with respect to NLMS. It can be notice in Fig. 6.8 that the ERLE improvement (in dB) is directly proportional to the convergence rate; in fact, just after seconds 0 and 5, i.e. in transient state, the ERLE improvement is small due to a filter adaptation, while a larger improvement is achieved in steady-state, i.e. in time intervals $1.5 - 5$ and $6.5 - 10$ seconds.

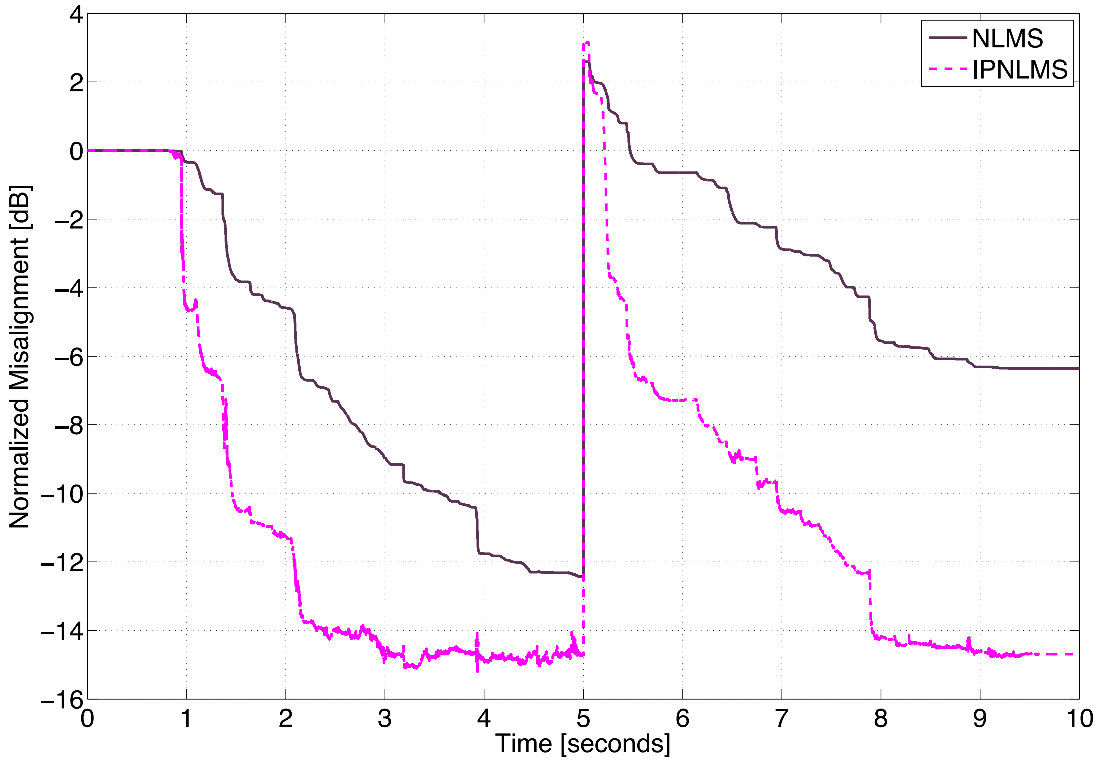


Fig. 6.7: *Misalignment comparison between NLMS and IPNLMS algorithms when a path change occurs. The far-end input is a female speech signal.*

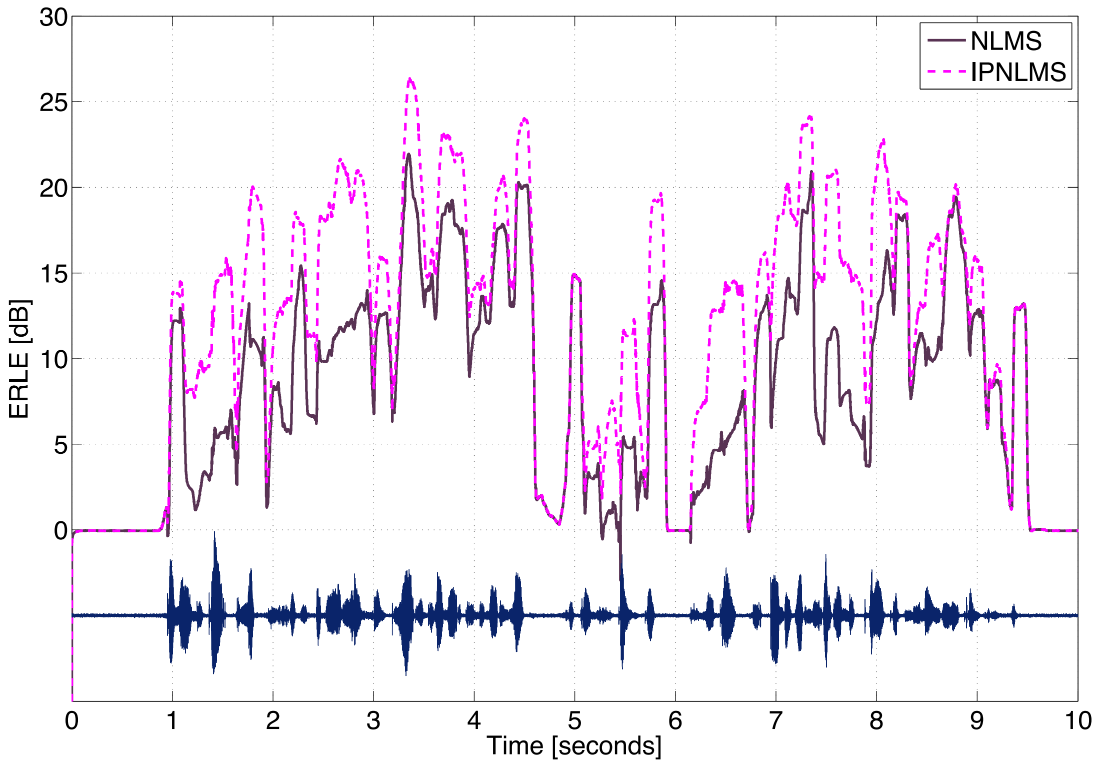


Fig. 6.8: *ERLE of NLMS and IPNLMS algorithms with a female speech input. The AIR changes at fifth second.*

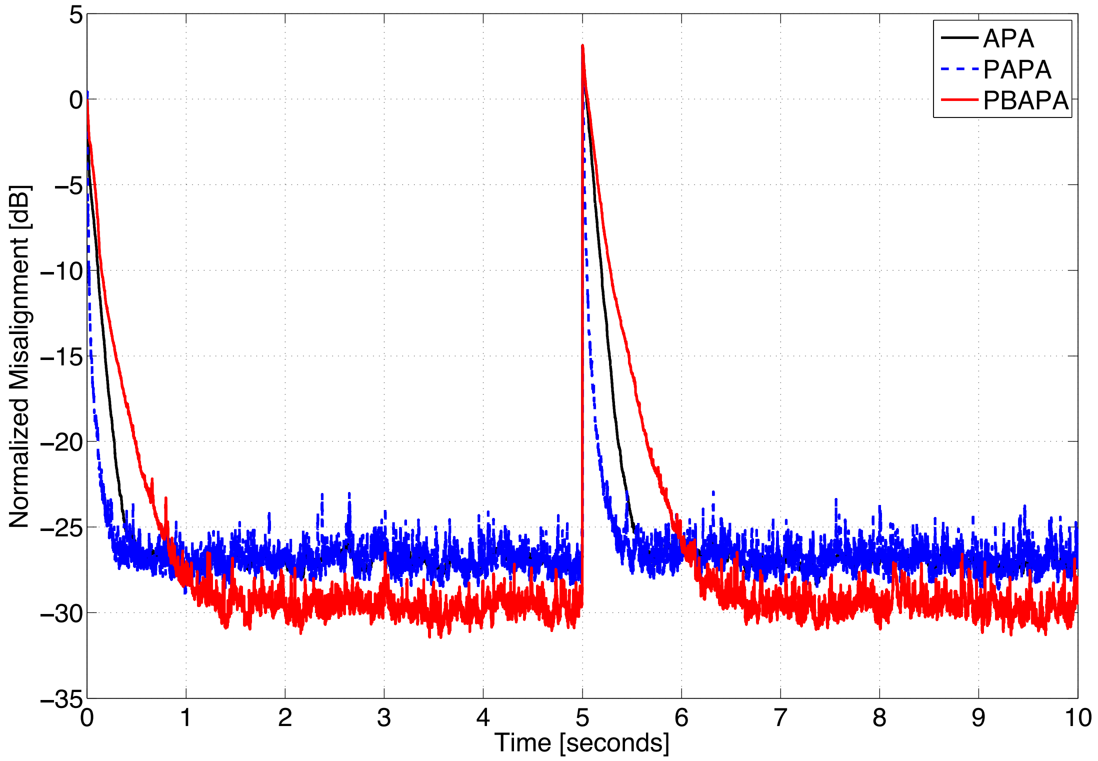


Fig. 6.9: *Misalignment comparison between PAPA and PBAPA algorithms with a white Gaussian noise input when the echo path changes. PBAPA shows better performance in steady-state; however, its tracking performance is poorer compared to IPAPA.*

We also investigates the behaviour of PAPA algorithms, including the PBAPA, when the echo path changes. Misalignment performance, depicted in Fig. 6.9, confirms the analysis done in the previous subsection, i.e. the PBAPBA provides the best steady-state behaviour while the PAPA shows the best tracking performance.

6.3 PERFORMANCE ANALYSIS OF VSS PROPORTIONATE FILTERS

Variable step size algorithms can bring significant improvement according to the environment conditions. In fact, due to their nature, VSS algorithms provides tracking performance improvements [124, 99] and this is the reason

why VSS algorithms are well suited for AEC scenarios with adverse environment conditions and in presence of double talk. Moreover, VSS algorithms do not suffer from any under-modelling noise (see Section 5.5) and this allows to estimate the AIR with shorter length than the exact AIR length.

The variable step size approach introduced in Section 5.5 can be applied to any proportionate algorithm; however, for a performance analysis purpose we evaluate the behaviour of VSS-PAPA with a projection order of $K = 2$. For the set of experiments conducted in this section we use the AIR simulated in typical office room and depicted in Fig. 6.1 (b), whose length is $M_{\mathrm{A}} = 1024$.

6.3.1 Under-modelling the acoustic impulse response

In case of exact modelling scenario we set the filter length $M = M_{\mathrm{A}}$ while in under-modelling scenario we halve the exact length, so $M = 512$. In addition,

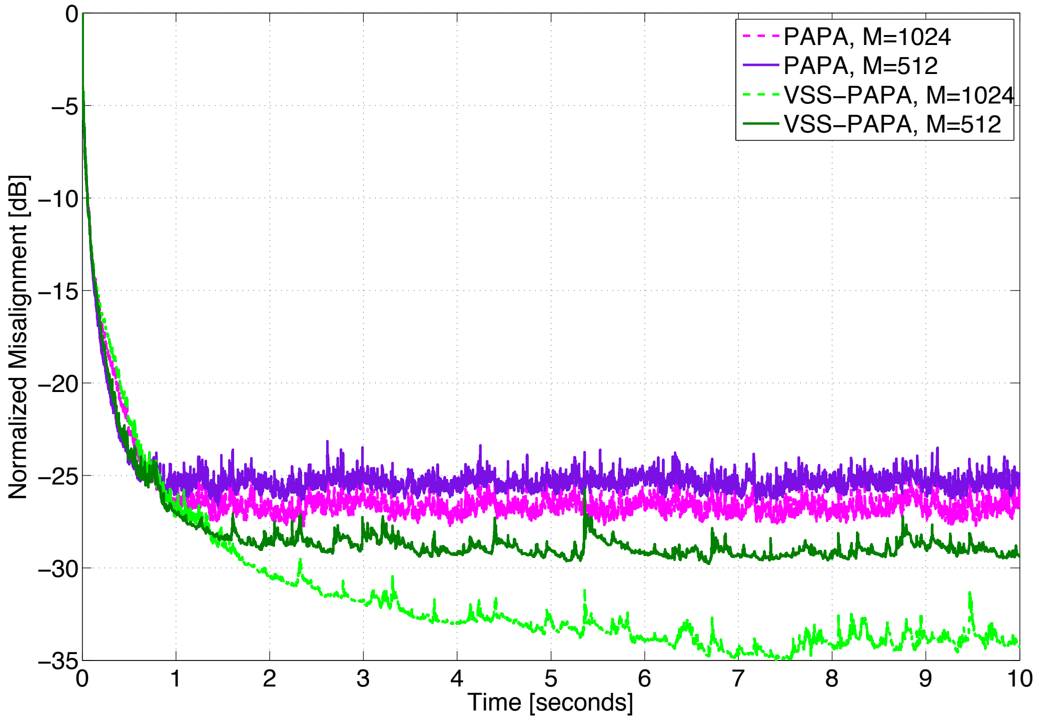


Fig. 6.10: *Misalignment comparison between PAPA and VSS-PAPA algorithms with a white Gaussian noise input. Both algorithms are evaluated in either an exact and an under-modelling scenario.*

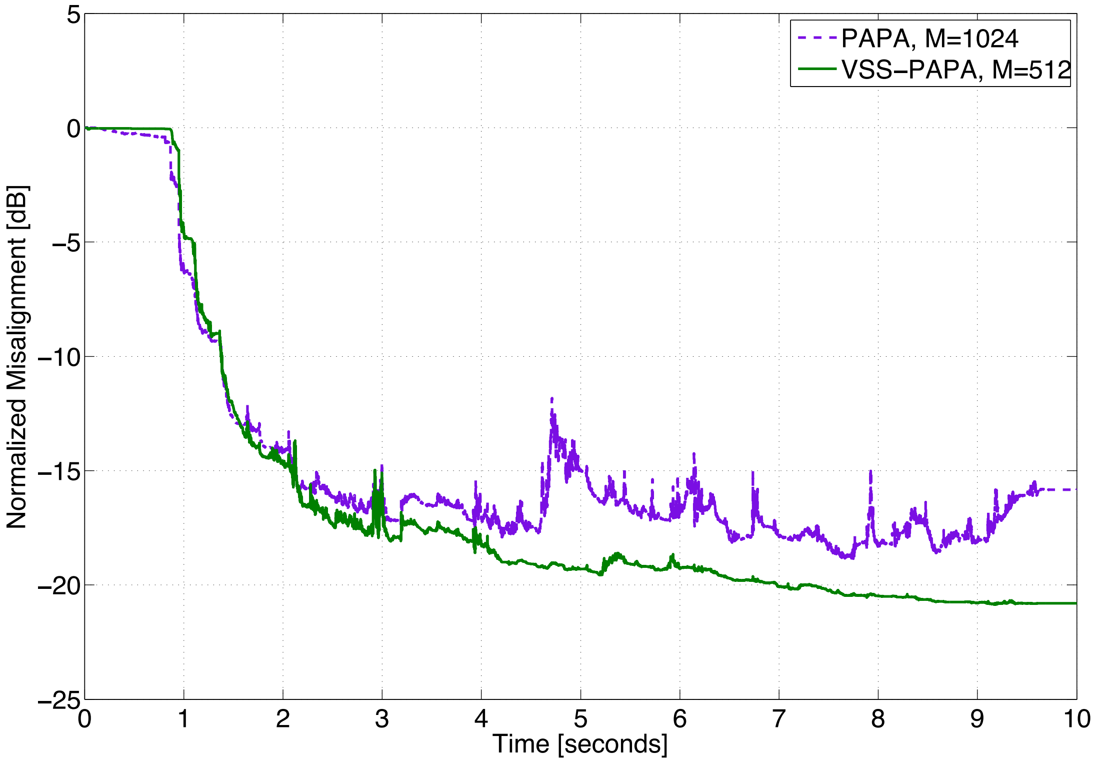


Fig. 6.11: *Misalignment comparison between PAPA and VSS-PAPA algorithms with a female input. The PAPA is evaluated in exact modelling while the VSS-PAPA in under-modelling.*

for the computation of the forgetting factor β in (5.37) we choose $Q = 6$ for white Gaussian noise input and $Q = 20$ for speech input.

In Fig. 6.10 we compare the misalignment performance of PAPA and its VSS version both in exact modelling and under-modelling scenarios using a white Gaussian noise input. It can be notice that even with a strong under-modelling of the AIR the VSS-PAPA achieves better performance compared to PAPAs. In Fig. 6.11 the misalignment comparison is reported in case of speech input using an exact modelling PAPA filter and an under-modelling VSS-PAPA filter; also in this case the VSS-PAPA still outperforms the PAPA. On the other side, not significant improvement is obtained in terms of ERLE, as depicted in Fig. 6.12, however, for an equivalent ERLE the misalignment improvement still represents an advantage since it implies a higher quality of the processed signal in perceptive terms.

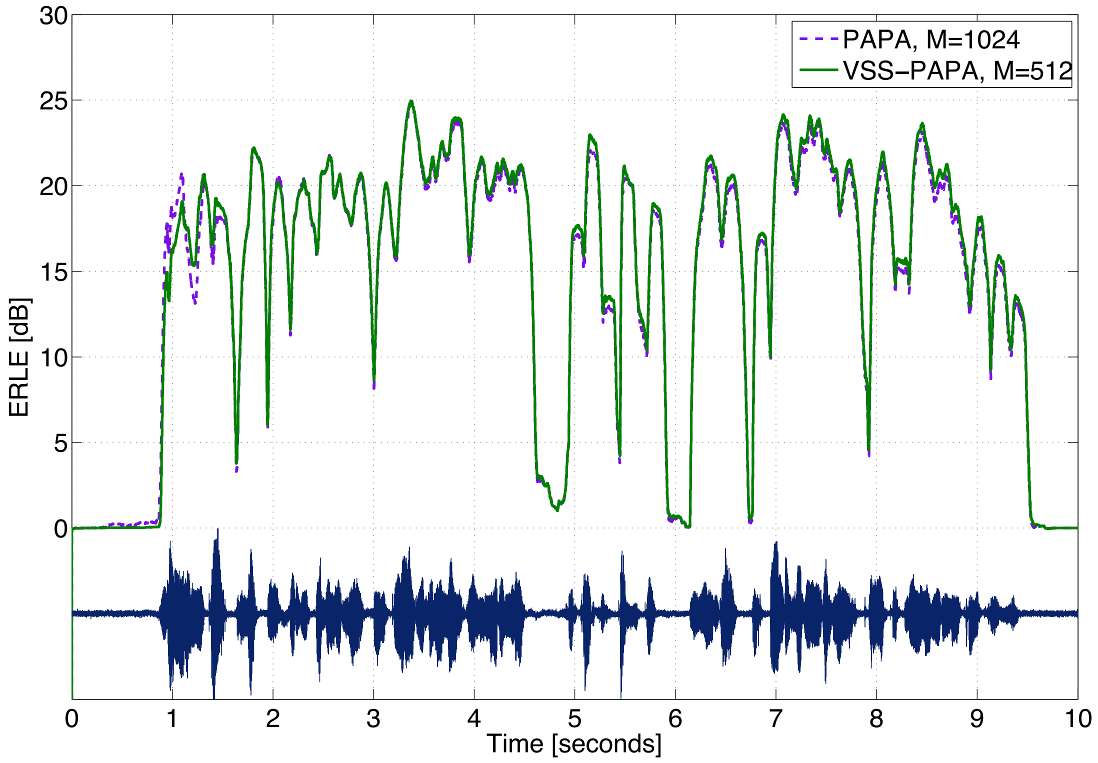


Fig. 6.12: *ERLE comparison between PAPA and VSS-PAPA algorithms with a female input. The PAPA is evaluated in exact modelling while the VSS-PAPA in under-modelling.*

6.3.2 Robustness against double talk

Another situation in which the VSS algorithms result effective is in presence of *double talk*, i.e. when a near-end speech is present and is superimposed over the echo path. In fact in this case it results very difficult to cancel the echo contribution without eating away at near-end speech. The performance of an echo canceller during double talk is an important measurement because near-end speech often causes divergence, especially at high convergence rate. In order to solve this problem a *double talk detector* (DTD) is usually adopted [57], which stops the filter adaptation in presence of double talk in order to preserve the near-end speech. A DTD is a good method to meet the contradictory requirement of low divergence rate and fast convergence in echo cancellation.

DTDs can mostly be classified into energy-based or correlation-based techniques. The most popular representative of energy-based DTDs is the Geigel algorithm [39]. It is based on an observation that the energy of echo is typically much smaller than the energy of far-end speech. Therefore, if the near-end speech is present, the energy of the desired signal increases. The Geigel DTD detects the near-end signals by comparing the magnitude of current far-end sample and the maximum magnitude of the recent past samples of the near-end signals, which means declaring double talk when:

$$|d\,[n]| = \tau \max\left\{|x\,[n]|\,,\ldots,|x\,[n-M+1]|\right\} \tag{6.1}$$

The parameter τ is a threshold usually set to 0.5 based on the assumption of 6 dB hybrid attenuation. Once the double talk is declared, the updates is inhibited for some hangover time in order to reduce the miss of detection.

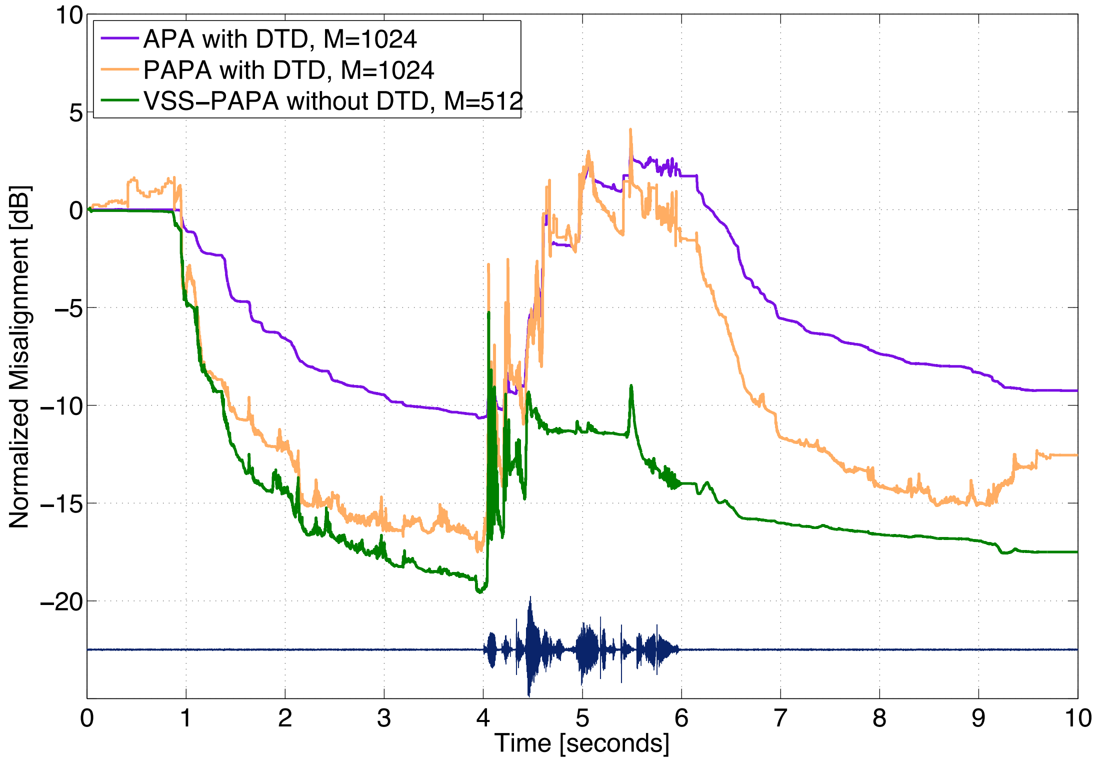


Fig. 6.13: *Misalignment comparison between APA, PAPA and VSS-PAPA algorithms in presence of double talk. APA and PAPA use a Geigel DTD, unlike the VSS-PAPA, which also considers an under-modelling of the AIR. The near-end speech is reported for clearness.*

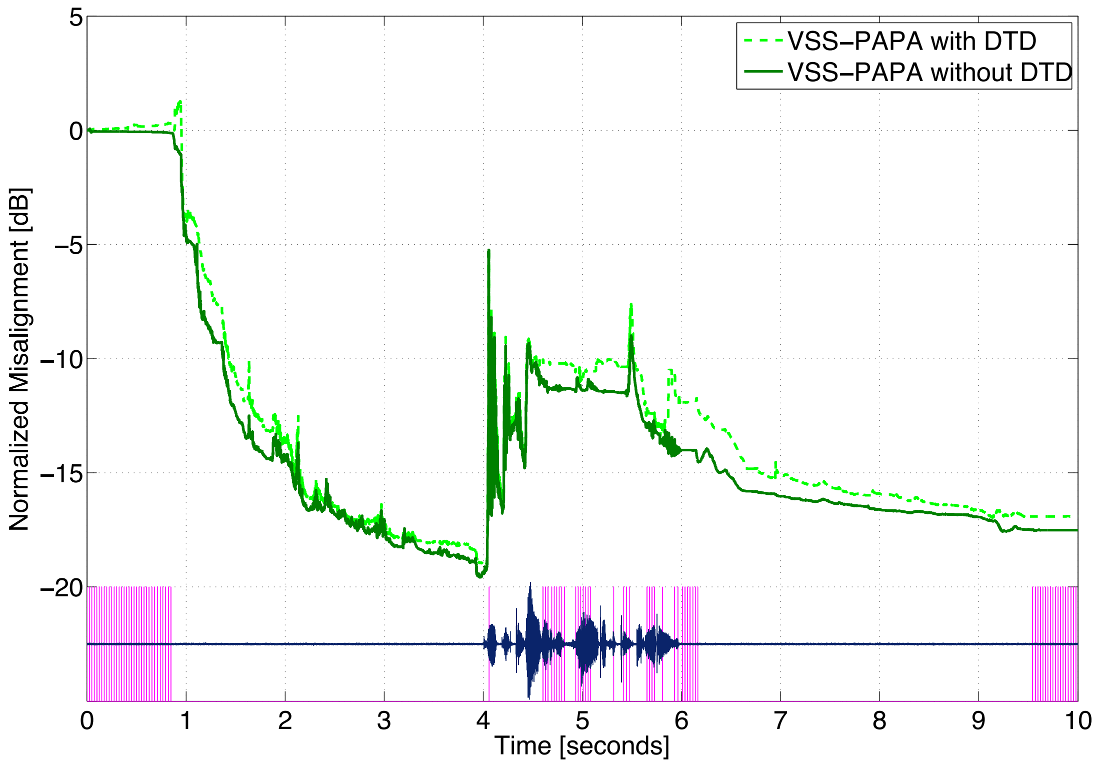


Fig. 6.14: *Misalignment comparison between VSS-PAPA algorithms with and without a DTD in presence of double talk. In both the cases an under-modelling of the AIR is considered. The near-end speech and the double talk detections are reported for clearness.*

However, a DTD is not always a good solution and often it is necessary a strong DTD to preserve the intelligibility of the near-end speech. The strength of the VSS is that it is able to govern the adaptation when a double talk occur, so there is no further need of using any DTD.

Here we consider the same scenario of the previous set of experiments just adding a near-end speech contribution in the time interval $4-6$ seconds in order to simulate a double talk situation. We compare APA, PAPA and VSS-PAPA algorithms in presence of double talk. For APA and PAPA, we use a Geigel DTD with $\tau = 0.5$ and a hangover time equal to 200 samples; on the other side, we use a VSS-PAPA without any DTD and moreover in an under-modelling of the AIR. In Fig. 6.13 it is possible to see that, despite VSS-PAPA is without DTD, it achieves the best misalignment performance compared to other algorithms. In Fig. 6.14, it is possible to verify that a VSS-PAPA without DTD achieves almost the same performance of a VSS-PAPA with DTD, or rather better performance due to the fact that sometimes the DTD may detect a false alarm, so the algorithm stops the adaptation when it should not.

Part III

NONLINEAR ADAPTIVE ALGORITHMS

> *—The best material model of a cat*
> *is another, or preferably the same, cat.*
> ***Norbert Wiener***

7

Consequence of Nonlinearities on Hands-free Acoustic Applications

Contents

One of the most important limitations of acoustic interfaces in hands-free environments is their inability to effectively cancel or reduce nonlinear interfering signals which impair the speech intelligibility. Nonlinearities in acoustic applications are mainly caused by loudspeakers during large signal peaks; this is the reason why, in this chapter and in the following ones, we focus on applications of nonlinear acoustic echo cancellation where the loudspeaker distortions may affect the echo signal. In this chapter we introduce the problem of nonlinearities and how to address it in acoustic echo cancellation.

7.1 LIMITATIONS OF ACOUSTIC INTERFACES DUE TO NONLINEAR INTERFERING SIGNALS

As said in Chapter 3, the limitations of acoustic interfaces for hands-free applications include circuit and DSP noise, acoustic reverberation, nonstationary signal sources, under-modelling of the AIR, double talk, and in Chapters 5-6 we investigates some algorithms able to tackle these limitations. However, another important limitation is caused by nonlinear interfering sources which draws a significant line at the achievable sound quality. Nonlinearities can be generated by loudspeakers during large signal peaks or by the vibration of the loudspeaker shell which often may be a plastic enclosure; this is the reason why the acoustic application most subjected to nonlinearities is the acoustic echo cancellation due to the acoustic coupling between a microphone and a loudspeaker.

The presence of nonlinearities in acoustic echo paths affects the performance of a conventional AEC compromising the quality requirements of speech communications. In recent years, this topic has become even more sensible matter of interest, due to the growing spread of low-cost commercial hands-free systems, which are often composed of poor quality elements, most of all electronic components, such amplifiers and loudspeakers, and covering materials, such as plastic shells. These devices may cause significant nonlinearities in AIRs leading to perceptual quality degradation of speech [18, 147]. In order to tackle this problem, nonlinear acoustic echo cancellers (NAECs) are employed, thus resulting in nonlinear path modelling and speech enhancement.

In recent years, different structures have been investigated in order to

model the nonlinearities rebounding on acoustic echo paths. A prevalent technique is based on the use of nonlinear transformations, able to compensate different kinds of distortions [63, 46, 106, 147]. A *raised-cosine function* is used in [63] to model both *soft-clipping* and *hard-clipping* nonlinearities. In [46], a two-parameter *sigmoid function* is proposed, whose slope and amplitude can be updated during the learning process. Another adaptive sigmoid function is used in [106] to evaluate NAEC performance as reverberation time changes. A more flexible solution is proposed in [147] by using *spline functions*, that are smooth parametric curves defined by interpolation of properly control points collected in a look-up table [148]. Block-based *Wiener-Hammerstein models* using nonlinear functions are also investigated [32, 123, 121].

Even if NAECs using nonlinear functions provide good performance, the most popular nonlinear model for echo cancelling applications is based on *adaptive Volterra filters* (VFs). The generic structure of VFs derives from the well-known Taylor series, and it can be considered as a straightforward generalization of linear adaptive filters [86]. Thus, due to its nature, VFs can model a large range of nonlinearities, both with memory and memoryless [137, 56]. However, acoustic echo cancellation, as well as other hands-free applications, requires large adaptive filter order to model the AIR [120]. Therefore, since computational complexity is proportional to the number of filter coefficients, the adaptation of VFs can become prohibitively expansive, compromising real-time implementation. Moreover, the limitation of Volterra series expansion are similar to those of the Taylor series expansion, thus some types of nonlinearities cannot be modelled by Volterra series, e.g. hard clipping nonlinearities. In recent years Volterra models with reduced computational complexity have been investigated to make real-time implementation possible [43, 44, 138, 47, 10]. However, even in this case an expansion order larger than two has been hardly adopted, due to the complications in adapting such systems and controlling learning rates.

7.2 NONLINEARITY EFFECT ON THE PERFORMANCE OF AN AEC

Before introducing some nonlinear models it is convenient to investigate and analyse the consequence of nonlinearities from a performance perspective. We analyse a loudspeaker model using a real loudspeaker as a case study. Then we show how a nonlinearity of such loudspeaker deteriorates the performance of a conventional AEC.

7.2.1 Nonlinearities in the echo transmission chain

In studying the effects of distortion caused by a loudspeaker, many authors adopted a simplified circuit model of the electro-mechanical-acoustic transducer [70, 19, 56, 136, 137]. An approximation of the loudspeaker model can be justified analysing the kind of nonlinearities involved in the transmission chain, depicted in Fig. 7.1, as done in [138].

The main source of nonlinearities is found in part B (see Fig. 7.1), since the loudspeaker and the power amplifier are operated at the highest signal level of the transmission chain. This part of the system is assumed to be weakly time-variant, e.g. due to temperature drift. The acoustic echo path C is known to be linear and time-variant, while the microphone and the amplifier C can be modeled as linear shift-invariant (LSI) systems (see Paragraph 3.1.1) because of their low signal amplitudes. Also the nonlinear quantization of the A/D and D/A converters can be neglected in this context. If nonlinear distortions are mainly caused by an overdriven amplifier, they are approximately memoryless and can be modeled by a saturation curve [136, 96]. In particular, in [96], parts A and C of Fig. 7.1 are modelled with adaptive FIR filters and part B is realized by a saturation curve with one adaptive parameter. However, the adaptation of the whole system results computationally very demanding. On the other side, in [136], a system with non-adaptive nonlinearity models part A in Fig. 7.1 as a delay, part B by a 7-th order polynomial, and part C as a classical NLMS adaptive filter. With negligible additional effort an ERLE improvement is obtained, without affecting convergence properties of the adaptive filter. However, experiments in [138] show, that both systems obtain their good results only if the major cause of nonlinearities is a clipping amplifier. In many non-portable applications, like smartphones, the power amplifier is not necessarily overdriven, but it is still desiderata to operate a small, cheap speaker at its power limit. With such an echo path the systems in [136] and [96] do not achieve remarkable ERLE improvements. This shows the need to develop another kind of nonlinear echo canceller which is appropriate for systems with loudspeaker nonlinearities.

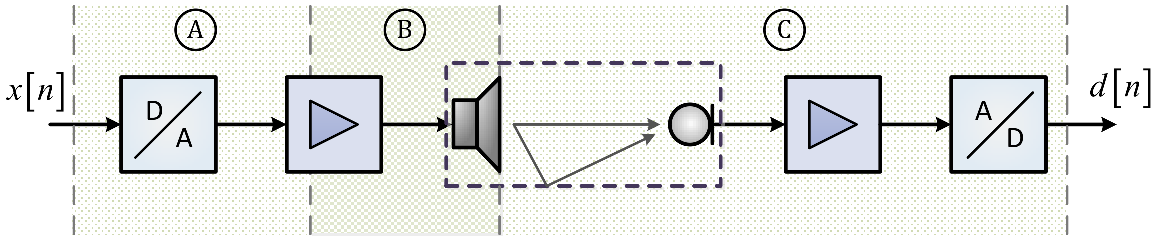


Fig. 7.1: *Echo transmission chain.*

This kind of nonlinearity is caused by the loudspeaker [49], especially when it is operated at its power limit. Due to the long time constants of the electro-mechanical system, the memory of this nonlinear behaviour cannot be neglected, as confirmed in [138]. To combat this type of nonlinearity, adaptive systems with memory are required. A *time-delay neural network*, being such a system, is proposed in [19]. With a cascade of a time-delay neural network and an adaptive FIR filter, considerable improvement of nonlinear echo reduction is achieved. A disadvantage is the need for a second reference microphone to provide an error signal for the adaptive neural network. In [143], adaptive VFs have been proposed for line echo cancelling. However, due to their high numerical complexity they have not been used in practical systems yet. In [138], an acoustic echo canceller with a second order adaptive Volterra filter has been developed and a method that keeps the computational complexity modest is proposed. From then on, other works have been proposed using Volterra models, as previously said in Section 7.1.

7.2.2 Loudspeaker identification by means of a neural network

In order to prove to evaluate nonlinear models in presence of distortions caused by a loudspeaker, we exploit the generalization capabilities of an artificial neural networks (ANN) [60] to obtain a functional model of a loudspeaker. In order to obtain adequate examples for the training of an ANN, we use data collected in a thesis work [112]. Data consists of 11 signals with linearly increasing amplitude including sinusoidal sweeps with frequency rate from 10 to 500 Hz in 16 bit wave form with a sample rate of 48 kHz. These signals

Electrical resistance [Ω]	11.06
Mechanical compliance of driven suspension	0.14E-0.3
Loudspeaker resonance frequency [Hz]	77.19
Equivalent acoustic volume	64.9E-03
Mechanical stiffness of driver suspension [N/m]	4.08
Force factor [N/A]	14.8
Electric Q factor	0.73
Sound Pressure Level	99.649
Total Q factor	0.62
Efficiency	3.95%
Equivalent inductance [mH]	1.15
Equivalent piston area [m^2]	56.8E-03
Nominal impedance [Ω]	16
Mechanical mass of the driven diaphragm [g]	29.7

Table 7.1: *Technical description of the loudspeaker model APW300, S.I.P.E. S.P.A. Electroacoustics.*

are used to excite a commercial loudspeaker, model APW300, produced by S.I.P.E. S.P.A. Electroacustics in Chiaravalle (AN), Italy, whose technical data are reposted in Table 7.1 and whose frequency response is depicted in Fig. 7.2. Measurements are conducted in an anechoic room in order to avoid any reverberations; all the data are finally decimate at 2 kHz.

We use a dynamic ANN with 20 inputs (10 MA and 10 AR), with 12 spline

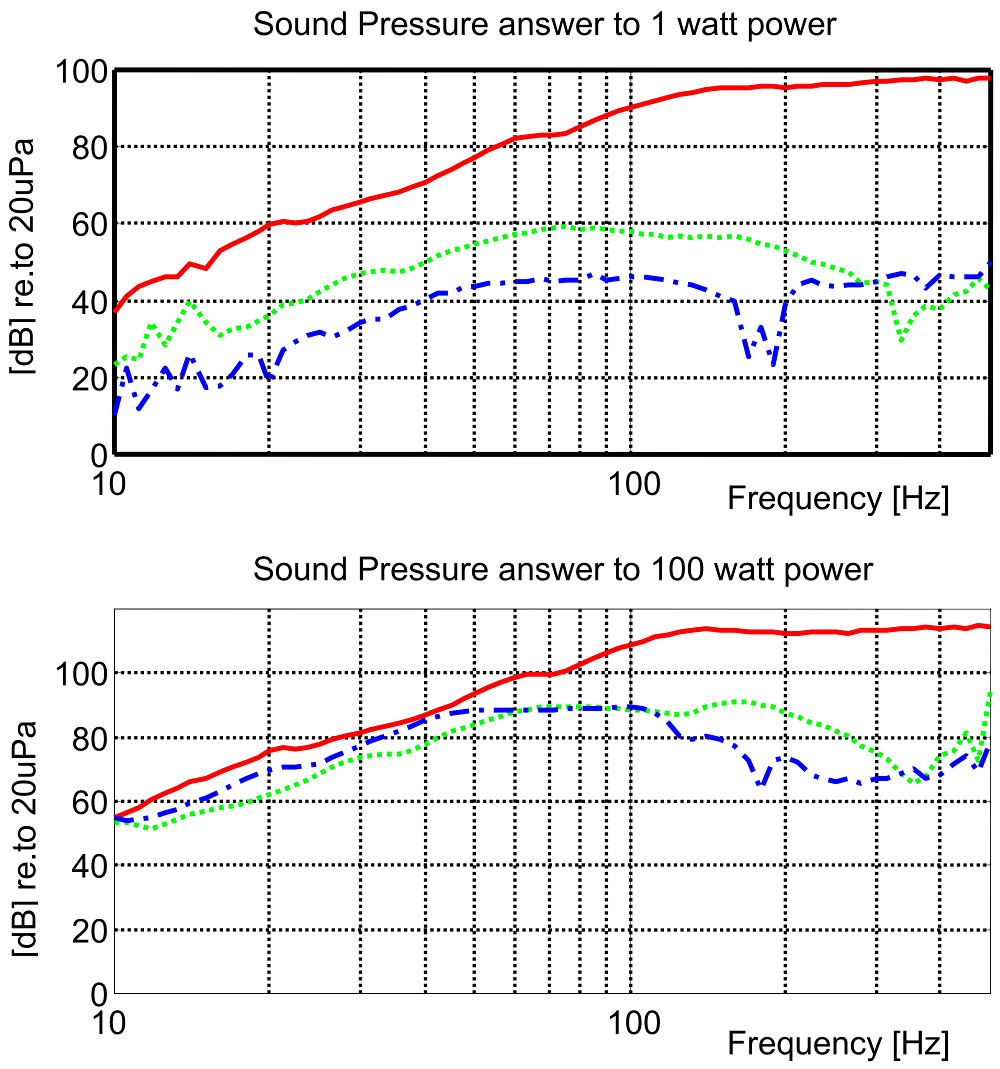
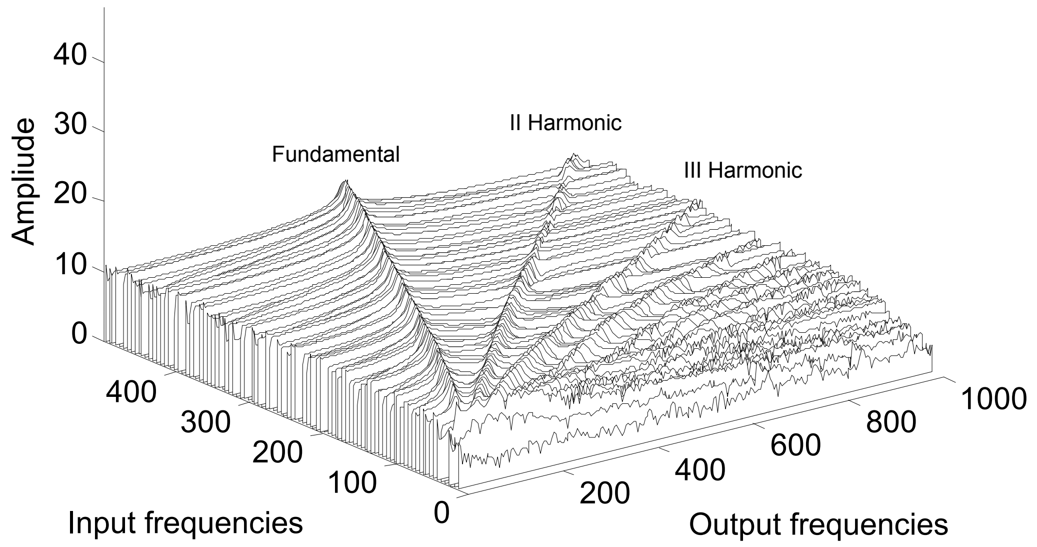
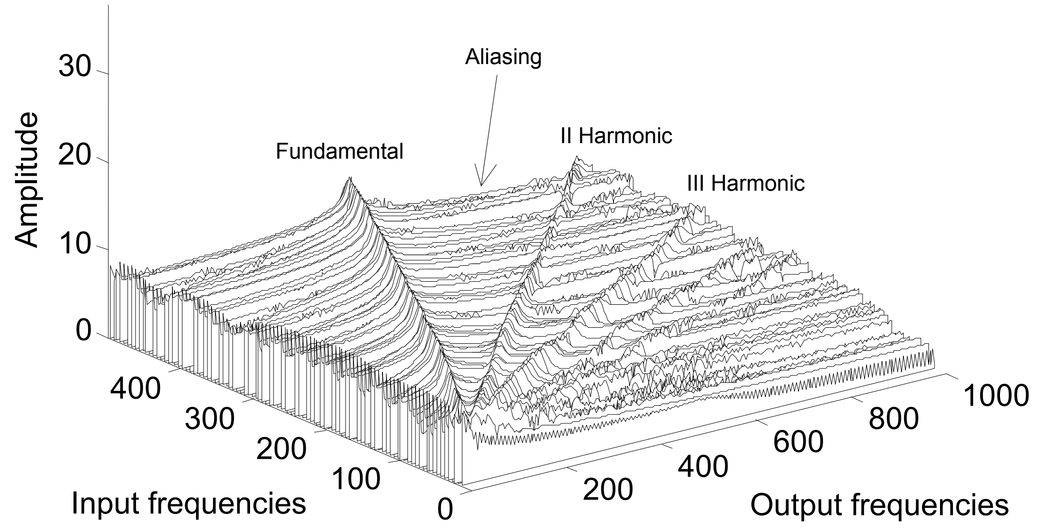


Fig. 7.2: *Frequency response of the loudspeaker APW300 at 1 W and 100 W. The red line is the fundamental harmonic, the green line is the second harmonic and the blue line is the third harmonic.*

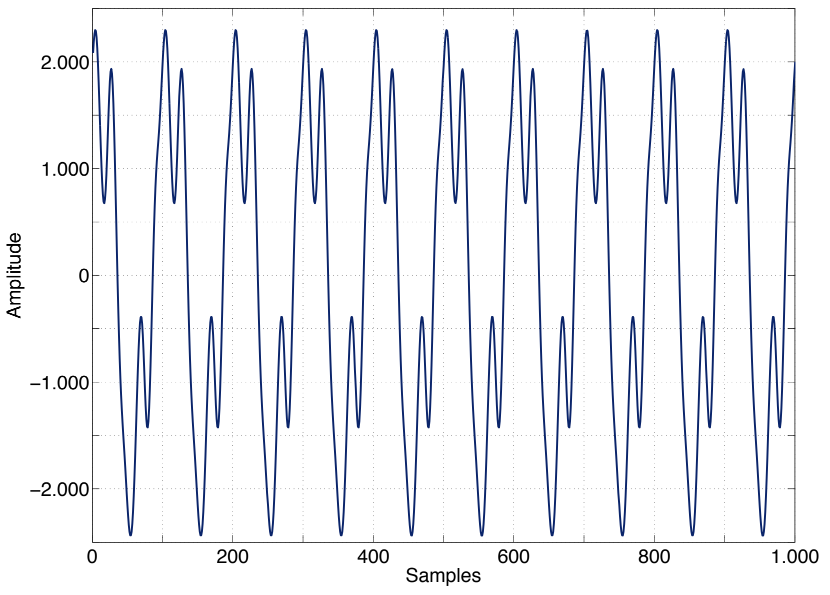


(a) Harmonic distortion of loudspeaker APW300.

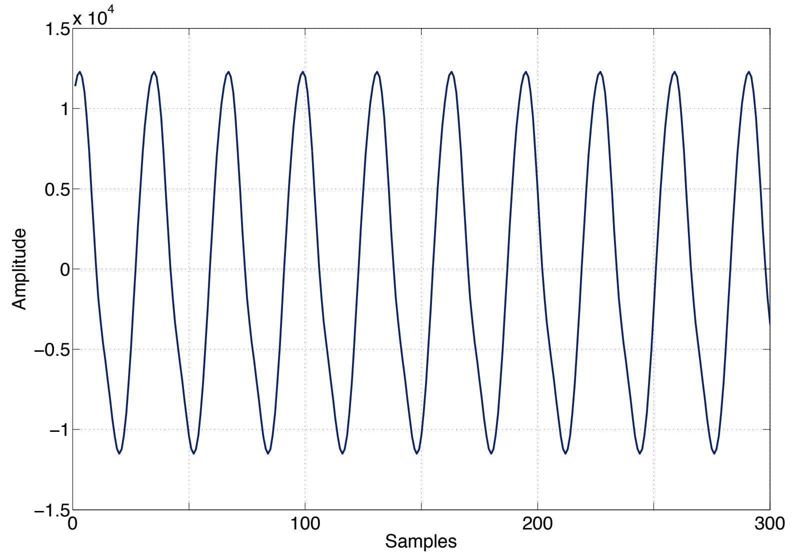


(b) Harmonic distortion of neural model.

Fig. 7.3: *Comparison between the harmonic distortion of the loudspeaker APW300 (a) and the neural loudspeaker model (b). The input signal is a sweep.*



(a) Sine at 80 Hz.



(b) Sine at 250 Hz.

Fig. 7.4: *Distortion effect of the neural loudspeaker model (a) on a sine at 80 Hz and (b) on a sine at 250 Hz. Being a professional loudspeaker the distortion is more evident at low frequencies.*

neurons [55, 113] with 28 points and fixed step $\Delta x = 0.5$. A *backpropagation algorithm* is used as learning rule; however, the learning rate is normalized with a quantity proportional to the input signal energy. In order to evaluate the distortion produced by this loudspeaker and the identification capability of the adopted ANN it is possible to compare Fig. 7.3 (a) and Fig. 7.3 (b).

The distortion effect of the neural loudspeaker model is depicted in Fig. 7.4 where it is clear that, being the APW300 a professional loudspeaker, nonlinearities affect a signal at low frequencies, so that a sine at 80 Hz results more distorted than a sine at 250 Hz. However, this is sufficient to produce a worsening in the performance of an AEC.

7.2.3 Performance worsening in an AEC process

In order to evaluate the loss of quality caused by loudspeaker distortions in an AEC process, we compare AEC performance using both an ideal purely linear model and the neural loudspeaker model previously described. We use a common hands-free scenario of a typical office room with a reverberation time of $T_{60} \approx 130$ ms, thus resulting the AIR depicted in Fig. 6.1 (b). We evaluate performance in terms of ERLE in three cases: when the far-end signal is a white Gaussian noise with zero mean and unitary variance, when the far-end signal is a coloured noise obtained through an autoregressive process of the white Gaussian noise signal, and eventually when the far-end input is a female speech signal. In all the cases an additive white Gaussian noise is added providing 20 dB of SNR in order to simulate some near-end background noise.

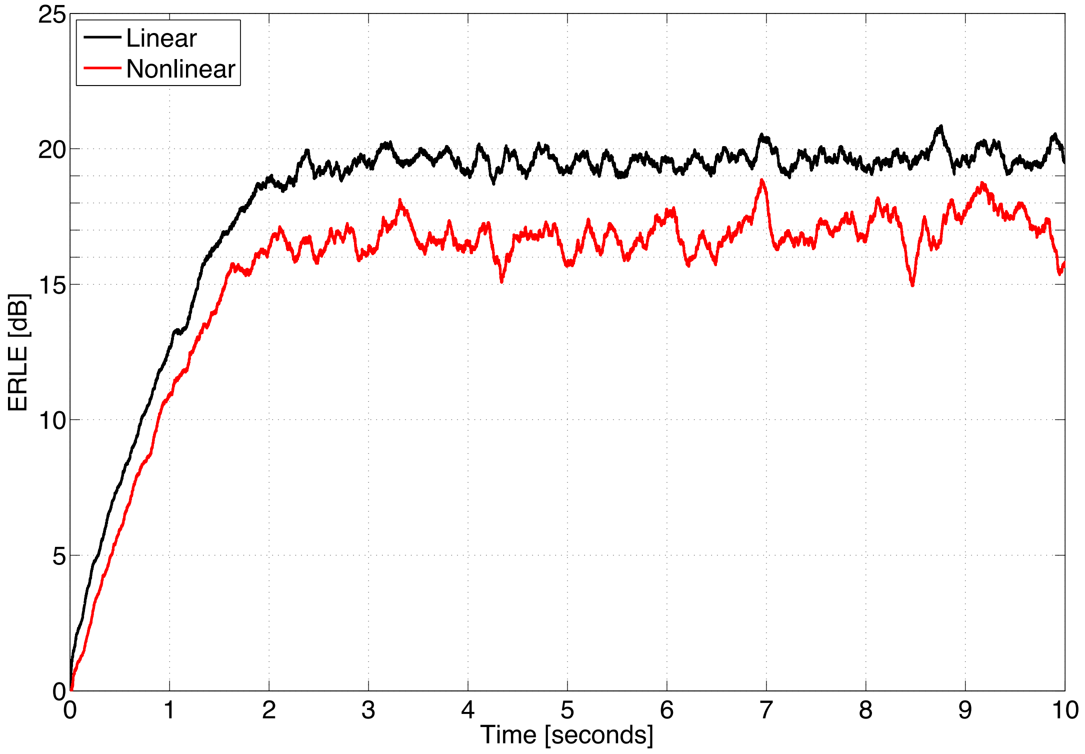


Fig. 7.5: *Loss of quality in terms of ERLE caused by loudspeaker distortions when the far-end input is a white Gaussian noise.*

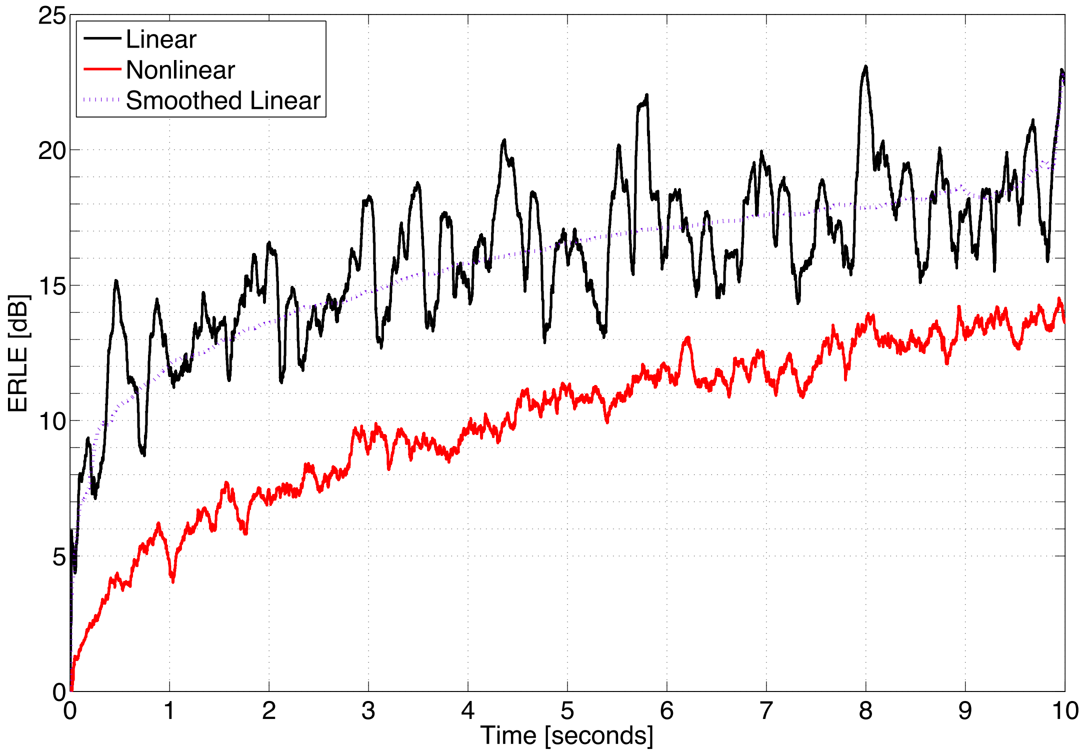


Fig. 7.6: *Loss of quality in terms of ERLE caused by loudspeaker distortions when the far-end input is a coloured noise. The dotted line represents the average performance in the linear case which clarifies the difference from the nonlinear performance.*

In Fig. 7.5, AEC performance in terms of ERLE is represented when the far-end signal is white Gaussian noise. The black line represents the ERLE performance in absence of distortions while the red line denotes the ERLE performance in presence of loudspeaker distortions. It is quite evident from this graph that the presence of distortions in the echo signal causes a loss

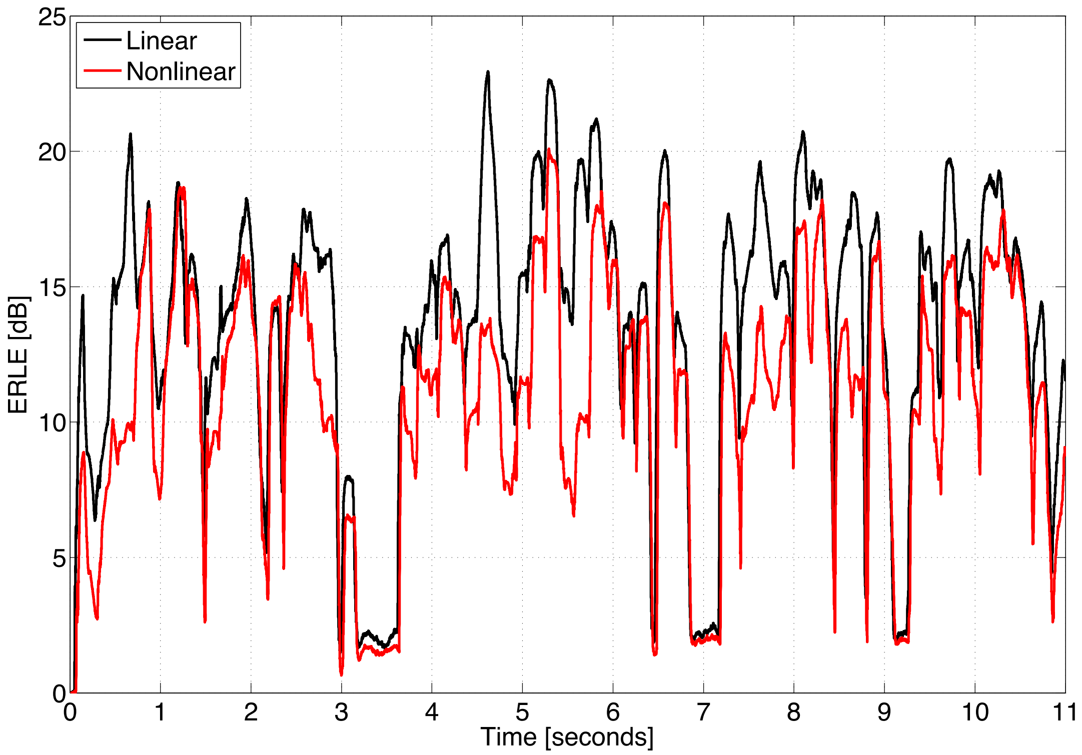


Fig. 7.7: *Loss of quality in terms of ERLE caused by loudspeaker distortions when the far-end input is a female speech signal.*

of quality of about 3 dB. The gap between performance in absence and in presence of distortions is more evident when the far-end signal is a coloured signal, i.e. a speech-like signal, as it is possible to see in Fig. 7.6, when the loss of quality is comprised on average within the range from about 3 to 7 dB. A confirmation of this trend is achieved when the far-end signal is a speech signal, as depicted in Fig. 7.7, where the loss of quality is larger than 7 dB in some peak of the signal. These results show that in hands-free acoustic applications, even with a professional loudspeaker, an important loss of quality can be obtained in presence of nonlinearities. Let us note that the performance in the linear ideal case represents the maximum achievable quality. Therefore an NAEC may improve the performance of an AEC in presence of distortions and may reach at most the achievable performance, thus we may expect to plug the performance gap as much as possible.

8 Functional Link Adaptive Filters: A New Class of Nonlinear Filters

Contents

THIS chapter introduces a new class of nonlinear filters, whose structure is based on Hammerstein model. The *functional link adaptive filters* (FLAF) are defined by a nonlinear input expansion, which enhances the representation of the input signal through a projection in a higher dimensional space, and a subsequent linear filtering. The most important element of a functional link adaptive filter is the nonlinear expansion, in which the a set of *functional links* processes the input signal allowing an enhanced modelling of nonlinearities. The functional expansion block allows to design a suitable filter according to scopes and field of application. This flexibility enables the filter to find the optimal trade-off between performance and computational complexity, according to the specifications of the problem.

8.1 INTRODUCTION

The problem of modelling linear systems has been widely tackled in last decades [116, 69] and, nowadays, it may be considered definitely solved. A linear system can be considered as a *white box*, since all information necessary to describe the system is available. Therefore, an effective estimate of the impulse response of a linear system may be achieved by using linear adaptive filtering algorithms [120, 59]. However, real-world systems often involve some degree of nonlinearity. In particular, if a system introduces a weak degree of nonlinearity it can be considered as a *grey box*, since, although information concerning the system is not entirely known, a linear approximation may be adopted. However, if a system shows a strong degree of nonlinearity it can be considered as a *black box*, since no information concerning the system is *a priori* available, thus a nonlinear system identification technique must be taken into account [157].

A popular approach to the problem of nonlinear system identification is the use of a cascade of a linear dynamic system and a memoryless nonlinear function. This kind of model is known in literature as *Wiener model* [97, 157]. On the other side, a cascade of a memoryless nonlinear function and a linear dynamic system is a very useful system in many practical applications and it is known as *Hammerstein model* [97]. Among the several other solutions to nonlinear filtering problem, one of the most popular technique proposed in literature is based on the so-called *polynomial filters* [86], which is a quite general model for nonlinear filtering. In this kind of filters, the adaptive nonlinearity consists in a polynomial-type nonlinearity: the filter output can be evaluated from its input through a polynomial model, truncated to a suitable order.

A particular case of polynomial filters is represented by Volterra filters [150]. The Volterra model can be very effectiveness in many practical applications, however, as said in Section 7.1, its computational cost may be very huge due the enormous number of coefficients required for higher-order kernels.

A more general framework for nonlinear filtering is provided by *artificial neural networks* (ANNs) [60], which represent an easily and flexible way to implement a such nonlinear filtering. The nonlinear transformations, applied by each neuron of an ANN, realize the searched nonlinearity. ANNs are capable of generating complex mapping between input and output space, therefore, arbitrarily complex nonlinear decision boundaries can be approximated by these networks. A drawback of this approach is the high computational cost of such a network. A particular type of ANN with reduced computational cost is characterized by activation functions implemented as *flexible spline nonlinear functions*, which are piecewise polynomials [148, 55, 129]. The term *spline*, in fact, comes from the flexible spline devices used by drafters to draw smooth shapes. Such networks, due to the adaptability of their activation functions, can solve hard problems with a low number of neurons [114].

In this chapter, we propose a novel nonlinear adaptive filtering model based on *functional links*. The functional link is a functional operator which allows to represent an input pattern in a feature space where its processing turns out to be enhanced. The functional links have been initially proposed by Pao [103] with the aim of developing a class of single-layer feedforward neural networks, known as *functional link artificial neural networks* (FLANNs). Pao has shown that FLANN may be conveniently used for function approximation and pattern recognition with faster convergence rate and lesser computational load than a *multi-layer perceptron* (MLP) ANN [103]. The FLANN is basically a flat net and the removal of the hidden layer allows a very simple use of the *backpropagation* learning algorithm [103, 60]. Functional links have been used for many applications, ranging from pattern recognition [104] to process control [128].

In this research study we develop a novel nonlinear model based on functional links that is not built on an ANN but on an adaptive filter structure. Such model, named *functional link adaptive filter* (FLAF), exploits the nonlinear modelling capabilities of functional links and the filtering properties of linear adaptive algorithms, which are definitely less computationally expensive than ANNs, thus resulting an effective tool to model nonlinearities (especially) in acoustic applications.

8.2 NONLINEAR SYSTEM IDENTIFICATION PROBLEM

Before describing the proposed nonlinear model, we briefly introduce a problem formulation concerning the nonlinear system identification. It needs to notice that the correspondent acoustic application of nonlinear system identification is the nonlinear acoustic echo cancellation, that we address in the next chapter.

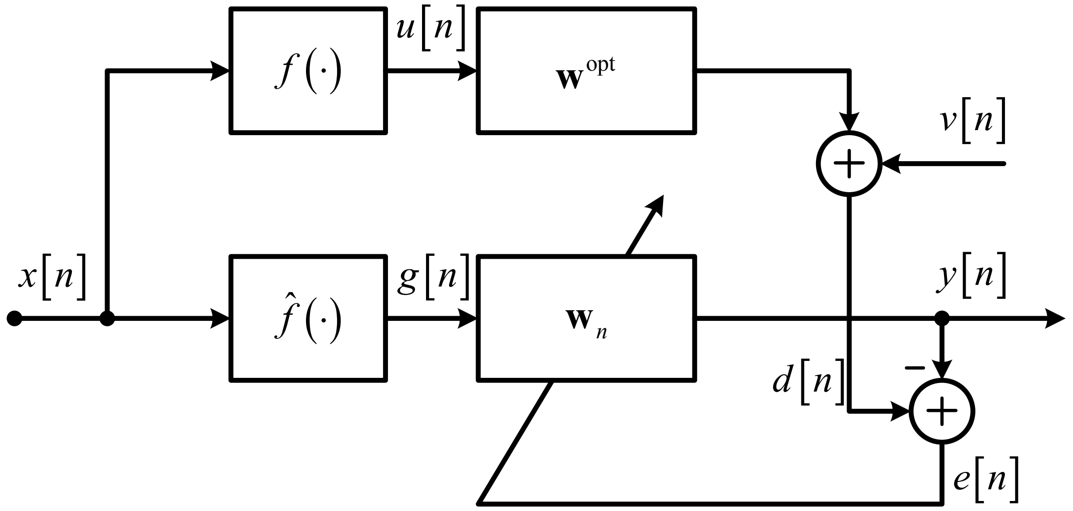


Fig. 8.1: *Hammerstein-based nonlinear system identification scheme.*

A nonlinear system identification problem based on a Hammerstein model is depicted in Fig. 8.1, in which it is possible to notice that the desired signal $d[n]$ results from the convolution between the input signal $x[n]$ and the unknown system to identify, denoted as:

$$\mathbf{w}^{\text{opt}} = \left(\mathbf{x}_n^T \mathbf{x}_n\right)^{-1} \mathbf{x}_n d[n] \tag{8.1}$$

as it is the optimal solution that solves the least-mean squares problem:

$$\min_{\mathbf{w}} \mathrm{E}\left\{\left|d[n] - \mathbf{x}_n^T \mathbf{x}_{n-1}\right|^2\right\}. \tag{8.2}$$

In a Hammerstein model the system to identify is preceded by a nonlinearity which is *a priori* unknown and may only be approximated. Therefore the identification of a Hammerstein model strictly depends on the nonlinearity upstream the filter.

In Fig. 8.1 it is possible to notice that the signal $x[n]$ is fed into a nonlinear system, thus the input signal to the unknown system gets to be $u[n] = f(u[n])$. Therefore the desired signal is:

$$d[n] = \mathbf{u}_n^T \mathbf{w}^{\text{opt}} + v[n] \tag{8.3}$$

where $v[n]$ is an additive noise, usually a white Gaussian noise with zero mean and unitary variance, thus resulting independent and identically distributed (i.i.d.). Consequently, the adaptive nonlinear filter, that aims at identifying the unknown system, is composed of a linear adaptive algorithm preceded by an artificial nonlinearity $\hat{f}(\cdot)$, which aims at approximating the nonlinearity of the unknown system. Therefore, the nonlinear input to the linear adaptive filter is denoted as $g[n] = \hat{f}(x[n])$.

The scheme depicted in Fig. 8.1 is generic for a system identification problem based on a Hammerstein model; with some specific changes, it allows to analyze a wide class of adaptive nonlinear filters based on Hammerstein model and described by the following adaptation rule:

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \mu \mathbf{g}_n \gamma(e[n]) \tag{8.4}$$

where $\gamma(\cdot)$ represents some function of the *a priori* output error signal:

$$e[n] = d[n] - \mathbf{g}_n^T \mathbf{w}_{n-1}. \tag{8.5}$$

Therefore, the scope is to define a suitable nonlinear function $\hat{f}(\cdot)$, which allows, through the update of an adaptive filter $\mathbf{w}_n$, to minimize the mean square error.

8.3 FUNCTIONAL LINK ADAPTIVE FILTERS

8.3.1 Functional link approach

The main idea which underpins our FLAF approach is that of asking whether it might be possible to enhance the original representation right from the start in a linearly independent manner. A way of enhancing the original input signal is to represent it in a space of higher dimension [103]. This process derives directly from the machine learning theory, and more exactly from Cover's Theorem on the separability of patterns [60]. Size and nature of the enhanced space are described by the functional links chosen to perform the nonlinear filtering. The functional link adaptive filtering is carried out in two stages: a nonlinear functional expansion of the input and a subsequent linear filtering, as it is possible to see in Fig. 8.2.

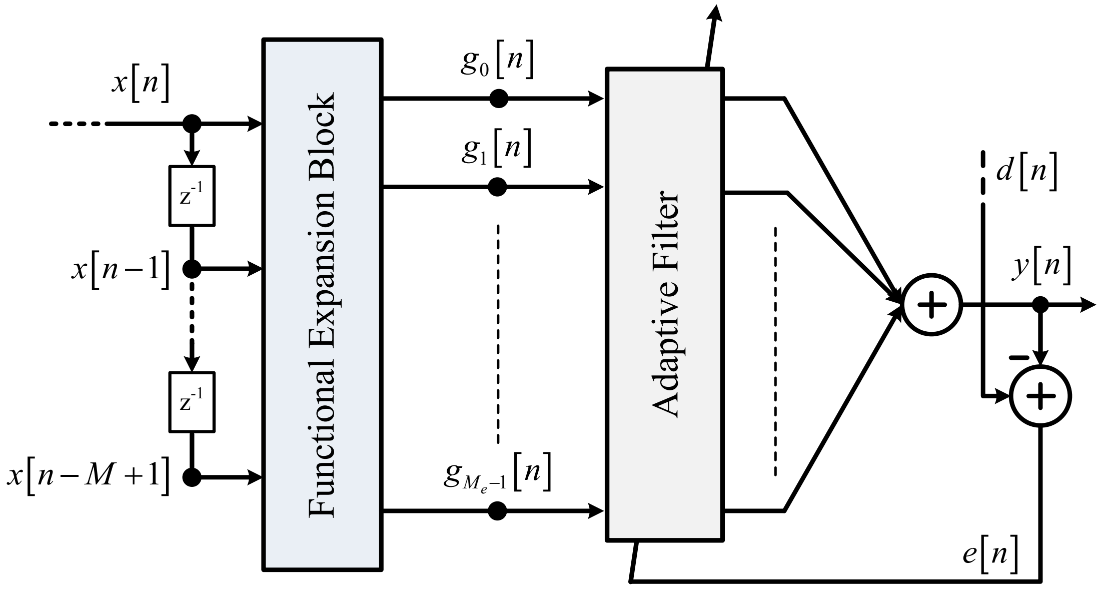


Fig. 8.2: *The functional link adaptive filter.*

At n-th time instant FLAF receives an input buffer $\mathbf{x}_n \in \mathbb{R}^M = \begin{bmatrix} x[n] & x[n-1] & \ldots & x[n-M+1] \end{bmatrix}^T$, where M is the input buffer length; differently from the linear weighting carried out by a linear filter, FLAF processes the input buffer by means of a *functional expansion block* (FEB). The FEB generates a series of linearly independent functions, which might be a subset of a complete set of orthonormal basis functions, satisfying universal approximation constraints [34]. The term functional links actually refers to this series of functions. The FEB processes the input buffer by passing each element of the buffer as argument for the chosen functions. The described process results in an *expanded buffer* $\mathbf{g}_n$, whose length is $M_e \geq M$. A deeper description of the expansion process will be drawn in Subsection 8.3.2.

In one sense, no new *ad hoc* information has been inserted into the process; however, the representation of the original buffer has been definitely expanded, and nonlinear modeling becomes possible in the expanded space. Once achieved the expanded buffer, the functional link adaptive filtering process is completed simply linearly filtering the expanded buffer. This aspect is an important theoretical novelty, with respect to the original formulation of functional links [103] and their recent use [162, 125], due to the significant advantages that it provides to FLAF, as described in Subsection 8.3.4.

8.3.2 Nonlinear input expansion

The most important element of the FLAF is the FEB, whose processing plays a leading role in the nonlinear modelling. The expansion process carried out by the FEB is depicted in Fig. 8.3, where it is possible to see how the input buffer $\mathbf{x}_n$ is projected in a higher dimensional space yielding the expanded buffer.

At n-th time instant, the i-th sample of the input buffer $x[n-i]$, being $i = 0, 1, \ldots, M-1$, is expanded by means of a chosen set of functional links $\mathbf{\Phi} = \{\varphi_0(\cdot), \varphi_1(\cdot), \ldots, \varphi_{Q-1}(\cdot)\}$, where Q is the number of functional links of the chosen set $\mathbf{\Phi}$.

The effectiveness of the FEB relies on two main feature of the chosen set of functional links $\mathbf{\Phi}$. The first feature will be detailed in Section 8.4 and concerns the nature of the expansion and, therefore, the choice of the functional links. The second feature is the correspondence between the input and the output samples of the FEB which can be characterized by the choice of taking into account some memory of the input buffer. This feature will be described in Section 8.5. The former feature depends on the kind of scenario of application and on the nature of involved signals; on the other hand, the latter feature depends on the nature of the input signal and, more specifically, on the kind of distortion which affects the desired signal.

8.3.3 FLAF learning algorithm

Once chosen the set of basis functions, the problem focuses on finding out the coefficients of the FLAF weight vector $\mathbf{w}_n \in \mathbb{R}^{M_e}$, defined as:

$$\mathbf{w}_n = \left[\begin{array}{cccc} w_0[n] & w_1[n] & \ldots & w_{M_e-1}[n] \end{array} \right]^T, \tag{8.6}$$

in order to yield the best possible approximation of the nonlinear model within a small error value ε. Therefore, the explicit representation of the FLAF error signal $e[n]$ is:

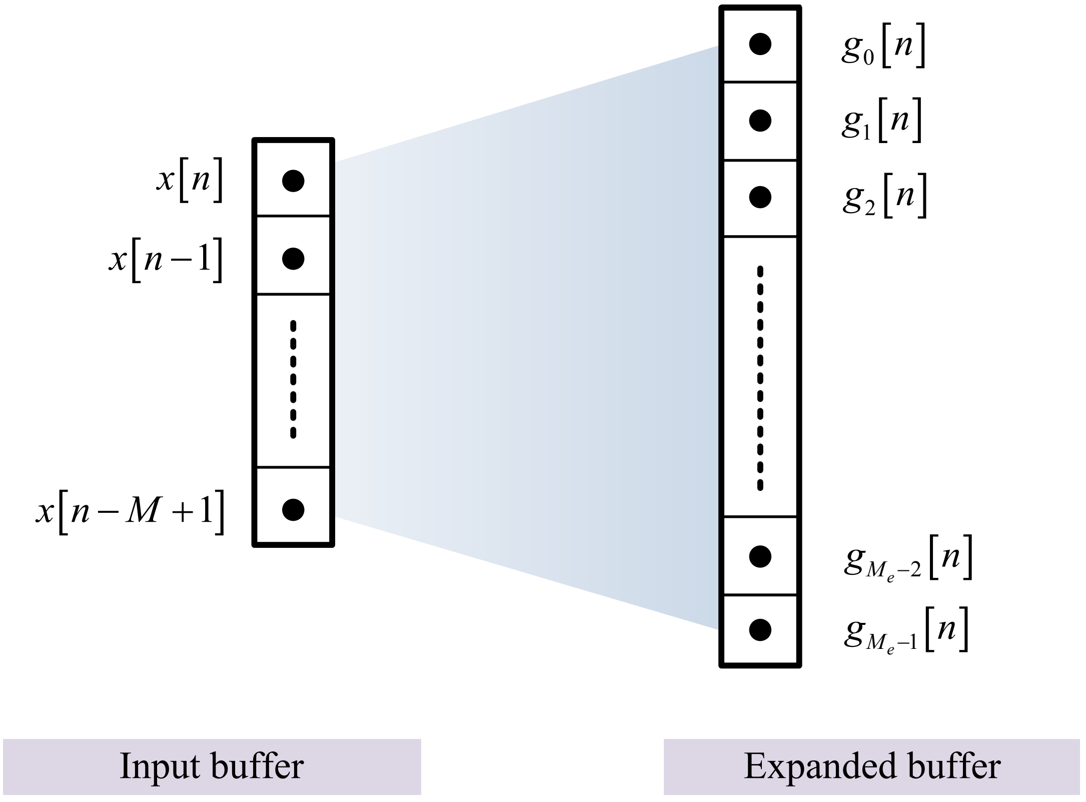


Fig. 8.3: *Functional link expansion.*

$$\begin{aligned} e[n] &= d[n] - y[n] \\ &= d[n] - \mathbf{g}_n^T \mathbf{w}_{n-1} \end{aligned} \tag{8.7}$$

whose minimization depends on a proper estimate of the weights of the filter $\mathbf{w}_n$. In order to find the coefficients of $\mathbf{w}_n$ it is possible to use any adaptive algorithm based on gradient descent rule [120]. In this work we use linear adaptive algorithms based on stochastic gradient rule (see Chapter 4) to adapt the filter coefficients.

8.3.4 Advantages and drawbacks of FLAF

The use of FLAF entails several attractive advantages. Firstly, FLAF has a hugely flexible architecture due to its scalable nonlinear expansion and to its scalable structural complexity. The former property allows to choose *a priori* a suitable series of functional links according to the application of interest. On the other hand, the latter property allows to deal with high dimension input signals, modelling the FEB structure in order to find the right trade-off between performance and computational complexity, according to application requirements and disposable computational resources. Moreover, the flexibility of FLAF architecture allows an easy integration of any *a priori* knowledge of a certain nonlinear system.

Furthermore, it is well known that the introduction of high-order functions in FLAF structure entails an increase of the learning rate [72] and a robust generalizing ability [90]. This property becomes more solid in FLAF, compared to FLANN [109, 72], due to the abilities to exploits the theory of linear adaptive filters [120] by using fast learning algorithms. In addition, the use of a linear filter provides FLAF with significant tracking capabilities that makes it suitable for DSP applications.

However, FLAF might also show some drawbacks, mainly caused by certain applications. A substantial difficulty might be definitely caused by the extreme flexibility of the architecture and in particular by the lack of a well-defined choice of an optimum nonlinear expansion and by a possible need of an *a priori* knowledge of the nonlinear system to design the expansion. Actually FLAF performance is strictly sensitive to the choice of nonlinear functions. Another drawback is that FLAF might incur in a biased convergence resulting in a non-optimum estimation [83, 135]. The described drawbacks will be certainly matter of future researches.

8.4 CHOICE OF FUNCTIONAL EXPANSION TYPE

The functional expansion process can be designed according to models and signals involved in the application. An important choice in the FEB design concerns the expansion type, i.e. the basis functions, or a subset of it, to assign for each functional link. This choice mostly depends on the application and in particular on the signals involved in the processing.

8.4.1 Choosing a proper set of functional links

The FLAF structure is a cascade of a nonlinear expansion and a linear filter; therefore the learning of a FLAF aims at approximating a continuous multivariate function $f(\mathbf{x}_n)$. In FLAF, the approximating function $\hat{f}(\mathbf{x}_n)$ is represented by a set of basis functions and by the coefficients of the adaptive filter $\mathbf{w}_n$. Inside the functional expansion process, a critical point is enacted by the choice of the complete set of orthonormal basis functions and its subset, which represents the functional links actually used. We start to analyze this problem by using a mathematical derivation.

Let I be a compact simply connected subset of $\mathbb{R}^n$ and $\mathcal{L}^m(I)$ be the subset of Lebesgue measurable functions $\hat{f}: I \subset \mathbb{R}^n \to \mathbb{R}^m$ such that the supremum norm of $\hat{f}$, denoted as $\left\|\hat{f}\right\|_I$ is bounded, i.e. $\left\|\hat{f}\right\|_I = \sup_{\mathbf{x}_n \in I} \left|\hat{f}(\mathbf{x}_n)\right| < \infty$. The space of all continuous functions $\hat{f}: I \to \mathbb{R}^m$ is a subset of $\mathcal{L}^m(I)$ and it is denoted as $\mathcal{C}^m(I)$. Let $\mathfrak{B}_Q = \{\varphi_j\}_{j=0}^{Q}$ be a subset of basis functions of a linearly independent set $\mathfrak{B}_Q \in \mathcal{L}^m(I)$. Being $\hat{f}(\mathbf{x}_n)$ a continuous function over a compact set, according to the Stone-Weierstrass theorem [139], there exist several subsets of $\mathfrak{B}$ that can uniformly approximate $\hat{f}(\mathbf{x}_n)$ by a discriminant:

$$\begin{aligned}\hat{f}(\mathbf{x}_n) &= \sum_{i=0}^{M-1}\sum_{j=0}^{Q-1}\varphi_j(x[n-i])\,w[n-iQ-j-1] \\ &= \mathbf{g}_n^T\mathbf{w}_{n-1}\end{aligned} \tag{8.8}$$

such that:

$$\max_{\mathbf{x}_n\in I}\left|f(\mathbf{x}_n)-\hat{f}(\mathbf{x}_n)\right| < \varepsilon \tag{8.9}$$

where ε is a small threshold, $\mathbf{x}_n \in I \subset \mathbb{R}^n$ is the FLAF input and $\hat{f}(\mathbf{x}_n)$ represents the FLAF output signal, also denoted as $y[n]$.

8.4.2 Most popular functional link sets

The solution of equation (8.8) depends on the existence of the inverse of the correlation matrix of the enhanced buffer. This can be assured by choosing a proper set of basis functions, which have to be linearly independent. Basis functions satisfying this property may be a subset of orthogonal polynomials, like Chebyshev [91], Legendre [107] and trigonometric polynomials [103], or just approximating functions, such as sigmoid [92] and Gaussian functions [22]. In the following we deal with the most employed functional link bases.

Trigonometric basis functions

It has been pointed out that when trigonometric polynomials are used in upstream, i.e. before the adaptive filtering, the weight estimate will approximate the desired impulse response in terms of multidimensional Fourier series decomposition [154]. In particular, compared with other orthogonal basis functions, trigonometric polynomials provide the best compact representation of

any nonlinear function in the mean square sense, even for nonlinear dynamic systems as proved in [109]. Moreover, trigonometric functions are computationally cheaper than power series-based polynomials. Due to its properties, trigonometric polynomial functions are very popular in functional link expansion, ranging from from function approximation applications [103, 72] and channel equalization [162] to active noise control applications [125]. Functional links with trigonometric functions are also used for dynamic system identification [109].

It is possible to generalized the set of functional links using trigonometric basis expansion in the following way:

$$g_j[n] = \begin{cases} x[n-i], & j=0 \\ \sin(p\pi x[n-i]), & j=2p+1 \\ \cos(p\pi x[n-i]), & j=2p+2 \end{cases} \tag{8.10}$$

where $j = 0,\ldots,Q-1$ is the functional link index, and $p = 0,\ldots,P-1$ is the expansion index, being P the *expansion order*. In (8.10) it is possible to notice that the first element of the set of functional links, $\varphi_0(x[n-i])$, is the replica of the current i-th input sample. In this way, the expanded buffer contains both linear and nonlinear elements.

Chebyshev polynomial functions

It is well known that Chebyshev polynomial functions are endowed with powerful nonlinear approximation capability [76]. This is the reason why their use is widespread in different fields of application. In particular, Chebyshev polynomials have been widely used both in pattern classification [91] and in functional approximation [76] problems. These works pointed out that an ANN with Chebyshev polynomial expansion has universal approximation capability and faster convergence than a MLP network. Moreover, Chebyshev polynomials were also used in FLANN structure [108] for the problem of identification of nonlinear dynamic systems in presence of input plant noise, showing a strong effectiveness. Furthermore, FLANN using Chebyshev expansion has been used in channel equalization [151, 161].

The effectiveness of Chebyshev polynomials is mainly due to the fact that the Chebyshev expansion of an input entry includes functions of the previous functions. Moreover, Chebyshev expansion is based on power series expansion, which may approximate a nonlinear function with a very small error near the point of expansion. However, far from the point of expansion, the error increases rapidly [35]. With reference to other power series of the same degree, Chebyshev polynomials are quite computationally cheap and more efficient [76], and this is the reason why they are frequently used for function approximation. However, when the power series converges slowly the computational cost dramatically increases.

Chebyshev functions are easier to compute with respect to trigonometric polynomial functions. Taking into account the i-th input sample $x\left[n-i\right]$, the Chebyshev polynomial expansion can be written as:

$$g_j\left[n\right]=\begin{cases}1, & j=0\\ x\left[n-i\right], & j=1\\ 2x^2\left[n-i\right]-1, & j=2\\ 2x\left[n-i\right]g_{j-1}\left[n\right]-g_{j-2}\left[n\right], & j=3,\ldots,Q-1\end{cases}\tag{8.11}$$

in which both linear and nonlinear terms are included, similar to the trigonometric case (8.10).

Legendre polynomial functions

Similar to Chebyshev polynomials, the Legendre functional links provides computational advantage while promising better performance [107]. Legendre polynomial functions have been widely used for function approximation by means of orthonormal ANN [159] and also functional link based ANN [111, 107]. Legendre-based *quadrature amplitude modulation* (QAM) equalizer [107] performs better than Radial Basis Function (RBF)-based and linear FIR-based equalizers; however, its performance is similar to that of Chebyshev-based equalizer [110].

Considering the i-th input sample $x\left[n-i\right]$, the Legendre polynomials are given by:

$$g_j\left[n\right]=\begin{cases}1, & j=0\\ x\left[n-i\right], & j=1\\ \left(3x^2\left[n-i\right]-1\right)/2, & j=2\\ \left\{\left(2j-1\right)x\left[n-i\right]g_{j-1}\left[n\right]-\left(j-1\right)g_{j-2}\left[n\right]\right\}/j, & j=3,\ldots,Q-1\end{cases}\tag{8.12}$$

where, as the previous two cases, both linear and nonlinear elements are involved.

8.5 MEMORY AND MEMORYLESS FLAF

In addition to the choice of considering the type of functional link set, another important choice in the FLAF design concerns the memory of the input buffer, which bears on the correspondence between samples of the input buffer and those of the expanded buffer. The choice of taking into account some memory is strictly related to the nature of the input signal. In particular, it depends a lot on the type of nonlinearity which deteriorates the input signal, in particular on whether the nonlinearity is *instantaneous*, i.e. it is independent from the time instant, or *dynamic*, i.e. the nonlinearity depends even on the time instant.

8.5.1 Memoryless functional links

The simplest and most commonly implemented type of nonlinearity is the *memoryless* (or instantaneous) one. Given an input signal $x\left[n\right]$, the generic output of any memoryless nonlinearity can be written as:

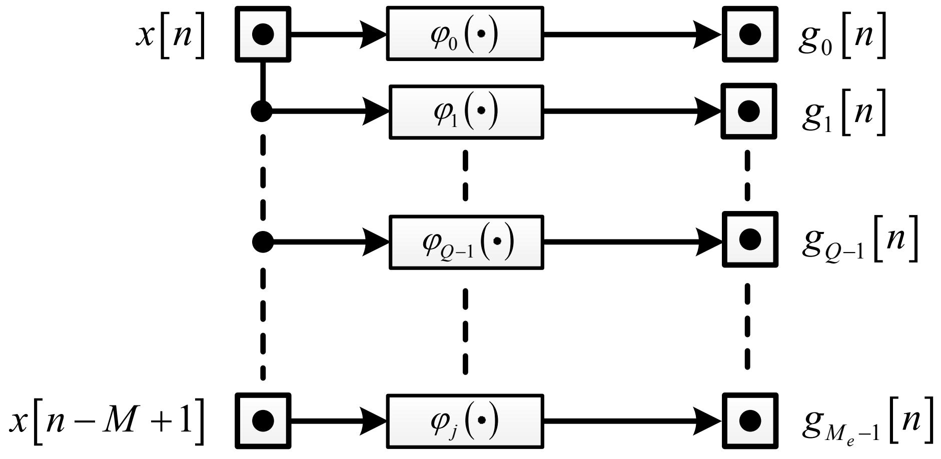


Fig. 8.4: *Functional expansion in memoryless FLAF.*

$$y[n] = f(x[n]) \tag{8.13}$$

where $f(\cdot)$ is some function which maps each input value to a unique output value [127]. Memoryless nonlinearities are very popular since many complex nonlinear systems can be broken down into a linear system containing a memoryless nonlinearity. Memoryless nonlinearities require *memoryless FLAF* which generates an unambiguous relation between the input buffer and the expanded buffer, as depicted in Fig. 8.4.

In a memoryless FLAF, it is possible to define a set $\mathbf{\Phi}_{\mathrm{ml}}$ of memoryless functional links, each of which takes one input sample as argument, yielding the corresponding sample of the expanded buffer. Since the memoryless set is defined as in Subsection 8.3.2, we omit any subscript and refer to it simply as $\mathbf{\Phi}$. For the first $M-1$ input samples we apply the full set of memoryless functional links $\mathbf{\Phi} = \{\varphi_0(\cdot), \varphi_1(\cdot), \ldots, \varphi_{Q-1}(\cdot)\}$; however, for the M-th input sample, we may choose to stop at j-th functional link, with $j = 0, \ldots, Q-1$, or to apply the full set $\mathbf{\Phi}$, depending on whether we want to control the expanded buffer length M_e or not.

8.5.2 Functional links with memory

The set of memoryless functional links described above provides a satisfying approximation of a continuous multivariate function, whether the nonlinearity is instantaneous or dynamic. However, in case of nonlinear dynamic systems, better results may be achieved exploiting the flexibility of the

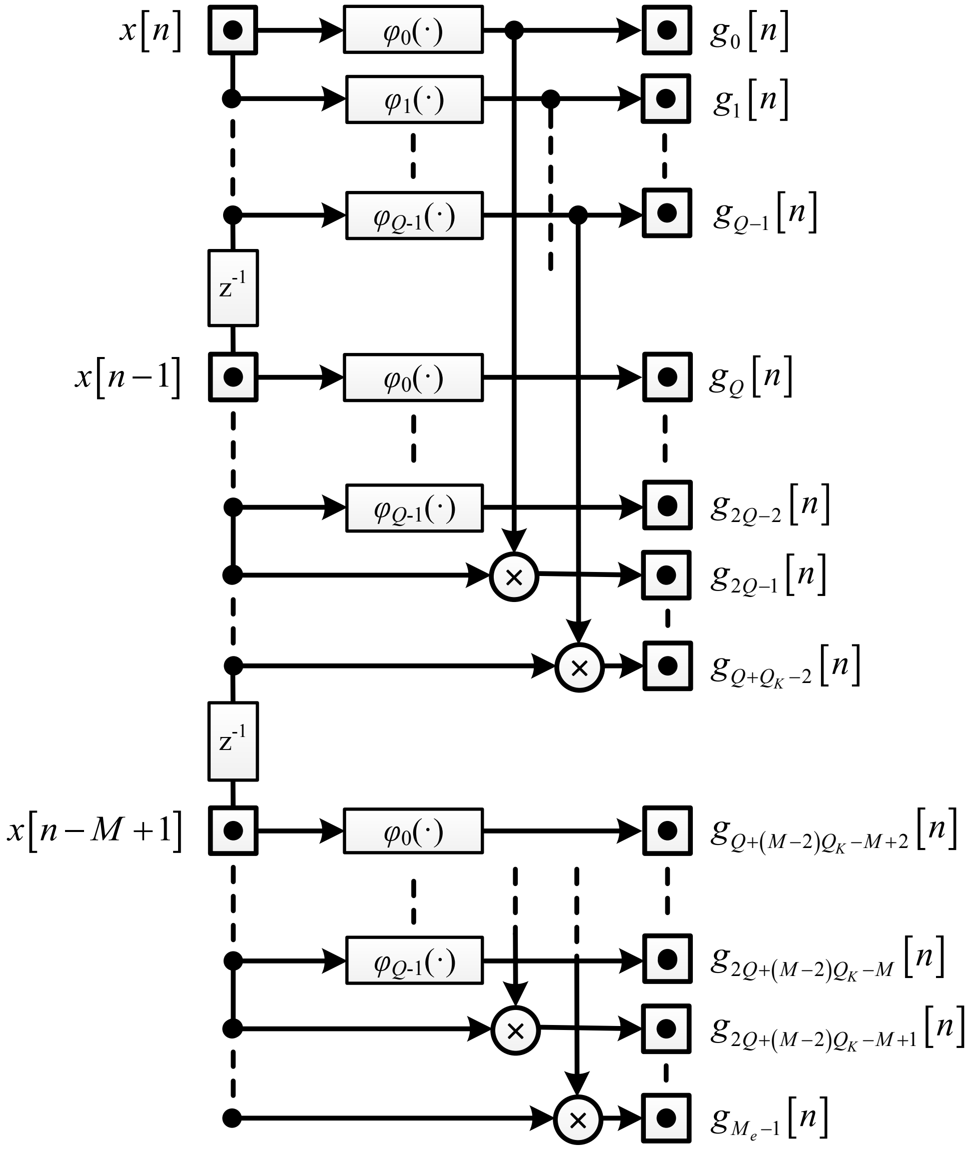


Fig. 8.5: *Functional expansion in FLAF with memory.*

FEB; in particular, it is possible to add to memoryless ones further functional links which take into account the memory of a certain dynamic nonlinearity. We refer to the new set $\mathbf{\Phi}_{\mathrm{m}} = \left\{\varphi_0(\cdot), \ldots, \varphi_{Q-1}(\cdot), \varphi_Q(\cdot), \ldots, \varphi_{Q_K-1}(\cdot)\right\}$ as a set of *functional links with memory*, where $Q_K > Q$ is the number of functional links with memory. A way of considering the memory of a nonlinearity is that of taking into account the outer products of the *i*-th input sample with the functional links of the previous input samples, as depicted in Fig. 8.5.

In designing the FLAF with memory, it is possible to define a memory order K which determines the length of the additional functional links, i.e. the depth of the outer products between the *i*-th input sample and the functional links related to the previous input samples. Fig. 8.5 shows an expansion with memory order $K = 1$.

8.6 MEAN-SQUARE PERFORMANCE ANALYSIS

8.6.1 Energetic approach to performance analysis

Transient and steady-state performance analyses of adaptive algorithms may be derived considering the expectation and the mean-square of the solution of its stochastic difference equation, which can be described by the expression (8.4). In particular, such analyses are conducted considering the asymptotic solution of the stochastic difference equation, defined as the limit, for $n \to \infty$, of $\mathbf{w}_n$. However, the presence of nonlinearities makes this approach impracticable. An alternative approach for the study of transient and steady-state performance analyses of adaptive algorithms is based on an *energy conservation relation* [120].

We start the derivation considering an important consequence of the data analysis model depicted in Fig. 8.1. Indeed, due to the independence property of the additive noise signal [120], it is possible to neglect $v[n]$, thus equation (8.3) turns into:

$$d[n] = \mathbf{u}_n^T \mathbf{w}^{\mathrm{opt}} \tag{8.14}$$

Therefore, similarly to equation (8.15), it is possible to define the *a priori* estimation error as:

$$e_a[n] = \mathbf{u}_n^T \mathbf{w}^{\mathrm{opt}} - \mathbf{g}_n^T \mathbf{w}_{n-1}. \tag{8.15}$$

which measures how close the nonlinear estimator $\mathbf{g}_n^T \mathbf{w}_{n-1}$ is to the desired response $d[n]$. Similarly, it is possible to the define the *a posteriori* estimation error as:

$$e_p[n] = \mathbf{u}_n^T \mathbf{w}^{\mathrm{opt}} - \mathbf{g}_n^T \mathbf{w}_n. \tag{8.16}$$

We consider the generic form (8.4) of the Hammerstein nonlinear adaptive filter; multiplying both sides of (8.4) by $\mathbf{g}_n^T$ from the left we obtain:

$$\mathbf{g}_n^T \mathbf{w}_n = \mathbf{g}_n^T \mathbf{w}_{n-1} - \mu \left\|\mathbf{g}_n\right\|^2 \gamma\left(e[n]\right) \tag{8.17}$$

Then, subtracting (8.17) from the desired response defined in (8.14), we achieve a relation between the *a priori* and *a posteriori* error signals:

$$e_p[n] = e_a[n] - \mu \left\|\mathbf{g}_n\right\|^2 \gamma\left(e[n]\right) \tag{8.18}$$

Equation (8.18) provides an alternative description of the stochastic equation (8.4). Generally, it is possible to analyse the behaviour of an adaptive filter in terms of estimation errors, $e_a[n]$ and $e_p[n]$, and in terms of misalignment vector $\widetilde{\mathbf{w}}_n = \mathbf{w}_n - \mathbf{w}^{\mathrm{opt}}$. However, in case of Hammerstein nonlinear filter it is not possible to take into account the information about the misalignment vector, thus the estimation errors are the only useful quantities in order to determine the behaviour of the filter. This is the reason why equation (8.18) assumes a significant relevance, since it turns out to be the only relation from which it is possible to accomplish a performance analysis. In particular, it is possible to derive the following behaviours:

- *Steady-state behaviour*, by means of the expectations $\mathrm{E}\left\{|e_a[n]|^2\right\}$ and $\mathrm{E}\left\{|e[n]|^2\right\}$.
- *Stability*, by determining the range of values of the step-size μ over which $\mathrm{E}\left\{|e_a[n]|^2\right\}$ remains bounded.
- *Transient behaviour*, by studying the evolution of the curve $\mathrm{E}\left\{|e_a[n]|^2\right\}$.

Therefore, in order to address these behaviours we may deal with an energy equality that relates the squared norms of the estimation errors.

8.6.2 Derivation of the energy conservation principle

The energy conservation relation does not depend on the error nonlinearity $\gamma(\cdot)$ [120], thus, in order to generalize this approach, it is possible to use equations (8.18) and (8.4) to solve for $\gamma(\cdot)$, distinguishing between three different cases.

1. $\mathbf{x}_n = \mathbf{0}$.
   The *degenerate case* is common for any linear adaptive filter and both Wiener and Hammerstein-based nonlinear filter. $\mathbf{x}_n = \mathbf{0}$ implies that $\mathbf{u}_n = \mathbf{g}_n = \mathbf{0}$, therefore it is obvious from (8.4) and (8.18) that $\mathbf{w}_n = \mathbf{w}_{n-1}$ and $e_p[n] = e_a[n]$, thus resulting:

$$\|\mathbf{w}_n\|^2 = \|\mathbf{w}_{n-1}\|^2 \quad \text{and} \quad |e_p[n]|^2 = |e_a[n]|^2 \tag{8.19}$$

2. $\mathbf{x}_n \neq \mathbf{0}$, $\mathbf{g}_n = \mathbf{u}_n$.
   As the previous case, this condition is still common for any linear and nonlinear adaptive filter. We solve for $\gamma(\cdot)$ from (8.18), using the constraint $\mathbf{g}_n = \mathbf{u}_n$, and substitute it into (8.4), obtaining:

$$\mathbf{w}_n = \mathbf{w}_{n-1} - \frac{\mathbf{u}_n}{\|\mathbf{u}_n\|^2}\left(e_a[n] - e_p[n]\right) \tag{8.20}$$

It is possible to notice that in equation (8.20) even the step-size μ is cancelled out. Moreover, in equation (8.20) the two estimation errors appear. In order to have an equality between the two errors, it is possible to rearrange equation (8.20):

$$\mathbf{w}_n + \frac{\mathbf{u}_n}{\|\mathbf{u}_n\|^2} e_a[n] = \mathbf{w}_{n-1} + \frac{\mathbf{u}_n}{\|\mathbf{u}_n\|^2} e_p[n]\,. \tag{8.21}$$

If we evaluate the energy of both sides of (8.21), we find out the following energy equality:

$$\|\mathbf{w}_n\|^2 + \frac{1}{\|\mathbf{u}_n\|^2}|e_a[n]|^2 = \|\mathbf{w}_{n-1}\|^2 + \frac{1}{\|\mathbf{u}_n\|^2}|e_p[n]|^2\,. \tag{8.22}$$

in which we do not take into account irrelevant cross-terms in order to have a fair energy relation.

3. $\mathbf{x}_n \neq \mathbf{0}$, $\mathbf{g}_n \neq \mathbf{u}_n$.
   The third case is not common for any adaptive filter, but it is specific to a Hammerstein nonlinear adaptive filter. Similarly to case 2 but without using any constraint, we solve for $\gamma(\cdot)$ from (8.18):

$$\gamma(e[n]) = \frac{1}{\mu\|\mathbf{g}_n\|^2}\left(e_a[n] - e_p[n]\right) \tag{8.23}$$

and then we substitute $\gamma(e[n])$ into (8.4), obtaining:

$$\mathbf{w}_n = \mathbf{w}_{n-1} - \frac{\mathbf{g}_n}{\|\mathbf{g}_n\|^2}\left(e_a[n] - e_p[n]\right) \tag{8.24}$$

and the correspondent energy relation:

$$\|\mathbf{w}_n\|^2 + \frac{1}{\|\mathbf{g}_n\|^2} |e_a[n]|^2 = \|\mathbf{w}_{n-1}\|^2 + \frac{1}{\|\mathbf{g}_n\|^2} |e_p[n]|^2 . \tag{8.25}$$

The results achieved in the three different cases can be combined together by defining a common term $\overline{\mu}[n]$:

$$\overline{\mu}[n] = \begin{cases} 0, & \mathbf{x}_n = \mathbf{0} \\ 1/\|\mathbf{u}_n\|^2, & \mathbf{x}_n \neq \mathbf{0}, \quad \mathbf{g}_n = \mathbf{u}_n \\ 1/\|\mathbf{g}_n\|^2, & \mathbf{x}_n \neq \mathbf{0}, \quad \mathbf{g}_n \neq \mathbf{u}_n \end{cases} \tag{8.26}$$

Using (8.26), we can combine (8.19), (8.22) and (8.25) into a single identity:

$$\|\mathbf{w}_n\|^2 + \overline{\mu}[n] |e_a[n]|^2 = \|\mathbf{w}_{n-1}\|^2 + \overline{\mu}[n] |e_p[n]|^2 \tag{8.27}$$

which generalizes the energy conservation relation and provides a unifying framework for the performance analysis of any linear and nonlinear adaptive filters.

Theorem 1 ***Energy conservation relation.*** *For any linear adaptive filter and for both Wiener and Hammerstein model-based nonlinear filter, it always holds that:*

$$\|\mathbf{w}_n\|^2 + \overline{\mu}[n] |e_a[n]|^2 = \|\mathbf{w}_{n-1}\|^2 + \overline{\mu}[n] |e_p[n]|^2$$

where $e_a[n] = \mathbf{u}_n^T \mathbf{w}^{\mathrm{opt}} - \mathbf{g}_n^T \mathbf{w}_{n-1}$, $e_p[n] = \mathbf{u}_n^T \mathbf{w}^{\mathrm{opt}} - \mathbf{g}_n^T \mathbf{w}_n$, *and* $\overline{\mu}[n]$ *is defined as in* (8.26).

9 Functional Link Adaptive Filters for NAEC

Contents

In this chapter we apply the nonlinear model of FLAF, described in the previous chapter, to nonlinear acoustic echo cancellation (NAEC), which is the correspondent acoustic application of the nonlinear system identification. As said in Sections 3.2 and 7.1, when nonlinearities occur in the echo path it is necessary to employ a nonlinear echo canceller in order to reduce the quality loss and preserve the intelligibility of a speech communication. In particular, in this chapter we introduce a novel type of FLAF designed *ad hoc* to tackle nonlinearities in an NAEC application. Some experimental results show that FLAFs are an effective alternative to VFs in NAEC applications[1].

9.1 FLAFs FOR ACOUSTIC APPLICATIONS

Flexibility is one of the stronger points of FLAF. However, the FLAF structure may turn out to be not always optimal depending on applications and on the nonlinearity degree engendered by an unknown system. This issue is mainly due to the fact that the FEB expands the whole input buffer. Thereby the expanded buffer may contain more nonlinear elements than necessary and an overfitting case may occur, thus resulting in non-optimal filtering performance. Moreover, a control over the expanded buffer seems to be problematic, as the choice of the input buffer length is bound to an accurate estimate of the acoustic impulse response. Additionally, the choice of optimal parameters of the adaptive filter, such as the step size, is the same for both linear and nonlinear elements of the expanded buffer. This is the reason why also this choice results critical in many situations, in particular when the nonlinearity degree in the echo path varies in time, as it is often the rule in acoustic echo cancellation. Due to these changes of the nonlinearity degree, it could be desirable to have a control over the nonlinearity degree in order to achieve always the best possible fitting. In that sense, improvements can be achieved modifying the FLAF structure up to yield the robust filtering architectures, some of which will be described in this chapter.

9.2 THE SPLIT FUNCTIONAL LINK ADAPTIVE FILTER

A significant improvement can be achieved separating the adaptation of linear and nonlinear elements of the expanded buffer. In particular, it is possible to consider two different adaptive filters in parallel, one completely linear and the other purely nonlinear. The linear filter receives the whole input buffer and aims only at estimating the echo path. On the other hand, the nonlinear filter is an FLAF in which the set of functional link does not include the replica of the linear element, as described in Chapter 8, thus the expanded buffer is only composed of nonlinear elements.

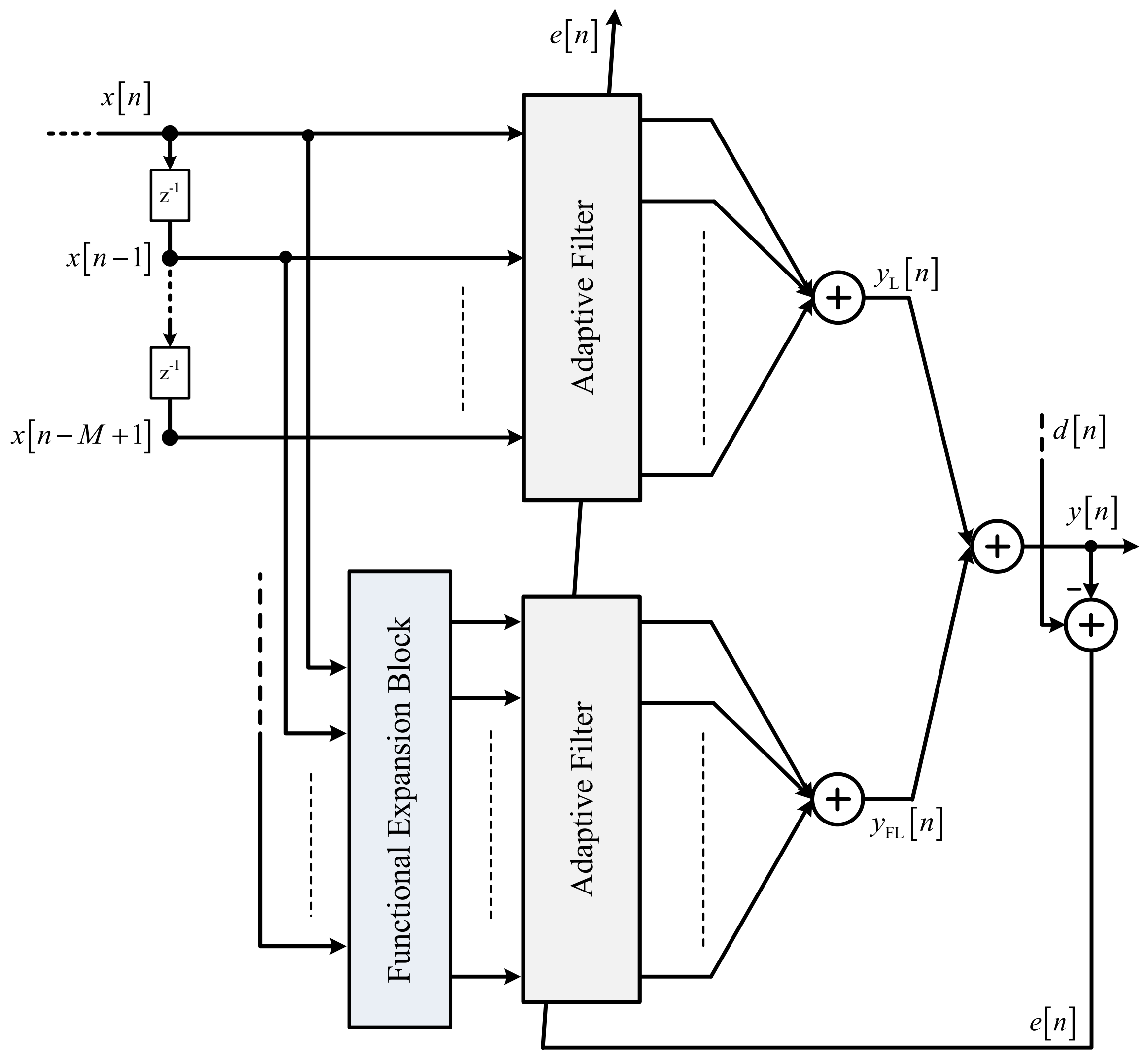


Fig. 9.1: *The split functional link adaptive filter.*

Therefore, the nonlinear FLAF only aims at modelling the nonlinearity affecting the echo signal. In this way it is possible to distinguish two different

[1] The work in this chapter has been partly performed while the author was a visiting Ph.D. student at the Department of "Teoría de la Señal y Comunicaciones", at "Universidad Carlos III de Madrid".

filterings with two different settings of the parameters, such that each filter can accomplish its task at best. Moreover, using this structure, the FEB can receive the whole input buffer or just a portion of it. This yields a further degree of freedom compared to the FLAF described in Chapter 8.

We call this filtering architecture *Split Functional Link Adaptive Filter* (SFLAF), thus remarking the separation between linear and nonlinear elements of the expanded buffer compared to the FLAF. The SFLAF structure is depicted in Fig. 9.1, where it is possible to notice that the SFLAF output signal results from the sum of the output of the linear filter and the output of the nonlinear FLAF:

$$y\left[n\right] = y_{\mathrm{L}}\left[n\right] + y_{\mathrm{FL}}\left[n\right] \tag{9.1}$$

in which $y_{\mathrm{L}}\left[n\right] = \mathbf{x}_n^T \mathbf{w}_{\mathrm{L},n}$, and where $\mathbf{w}_{\mathrm{L},n} \in \mathbb{R}^M = \left[\begin{array}{cccc} w_0\left[n\right] & w_1\left[n\right] & \ldots & w_{M-1}\left[n\right] \end{array}\right]^T$ is the coefficient vector of the linear filter, and $y_{\mathrm{FL}}\left[n\right] = \mathbf{g}_n^T \mathbf{w}_{\mathrm{FL},n}$, where $\mathbf{w}_{\mathrm{FL},n} \in \mathbb{R}^{M_e} = \left[\begin{array}{cccc} w_0\left[n\right] & w_1\left[n\right] & \ldots & w_{M_e-1}\left[n\right] \end{array}\right]^T$ is coefficient vector of the nonlinear FLAF.

From Fig. 9.1 it is possible to gather that both linear and nonlinear filters are adapted using the overall error signal $e\left[n\right] = d\left[n\right] - y\left[n\right]$. However, each filter can be adapted using a different adaptation rule and different parameter settings. This “splitting” feature of FLAFs opens new interesting scenarios in acoustic applications since it is even more possible to exploit at best the capabilities of linear adaptive algorithms and the effectiveness of functional links for acoustic applications. In fact, the flexibility of the FEB and the possibility to choice the proper adaptive filter, for both the nonlinear and the linear paths of SFLAF, make the SFLAF a versatile and effective tool for the modelling of the AIR affected by nonlinearities.

9.3 EXPERIMENTAL RESULTS

In this section we investigate the performance of the proposed SFLAF architecture in an echo cancellation scenarios. The scenario is a simulated teleconferencing environment in which the AIR is the one depicted in Fig. 6.1 (b), corresponding to a reverberation time of $T_{60} \approx 130$ ms, and truncated after $M = 512$ samples. In order to introduce a nonlinearity in the echo path which can simulate a loudspeaker distortion, we apply a *symmetrical soft-clipping* to the echo signal before that it activates the echo path according to the scheme in Fig. 9.2. The soft-clipping distortion is described by the following expression [165]:

$$f\left(x\left[n\right]\right) = \begin{cases} 2x\left[n\right] & \text{for} \quad 0 \leq x\left[n\right] \leq \zeta \\ \mathrm{sign}\left(x\left[n\right]\right) \frac{3-\left(2-3x\left[n\right]\right)^2}{3} & \text{for} \quad \zeta \leq x\left[n\right] \leq 2\zeta \\ 1 & \text{for} \quad 2\zeta \leq x\left[n\right] \leq 1 \end{cases} \tag{9.2}$$

where ζ is a threshold chosen in the range $(0\ ,\ 0.5]$. We obviously suppose that the input signal is normalized at 1.

Two kinds of input signal are used for this scenarios: a white Gaussian noise input with zero mean and unitary variance and a female speech input. Additive Gaussian noise is added at the output of the echo path in order to provide 20 dB of signal to noise ratio (SNR). The length of the experiments is $t = 10$ seconds. In Fig. 9.3 it is possible to see the effect of the nonlinear

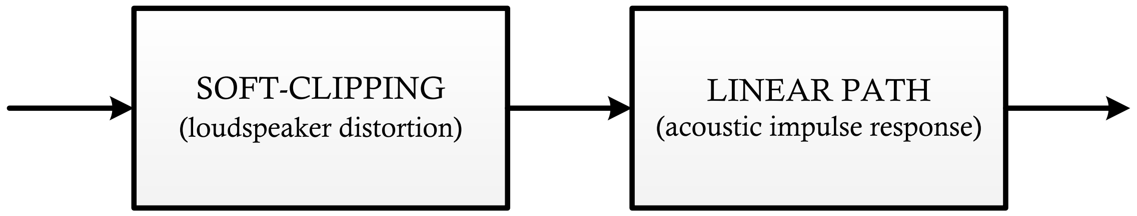


Fig. 9.2: *Scheme of the nonlinearity introduced in the input signal.*

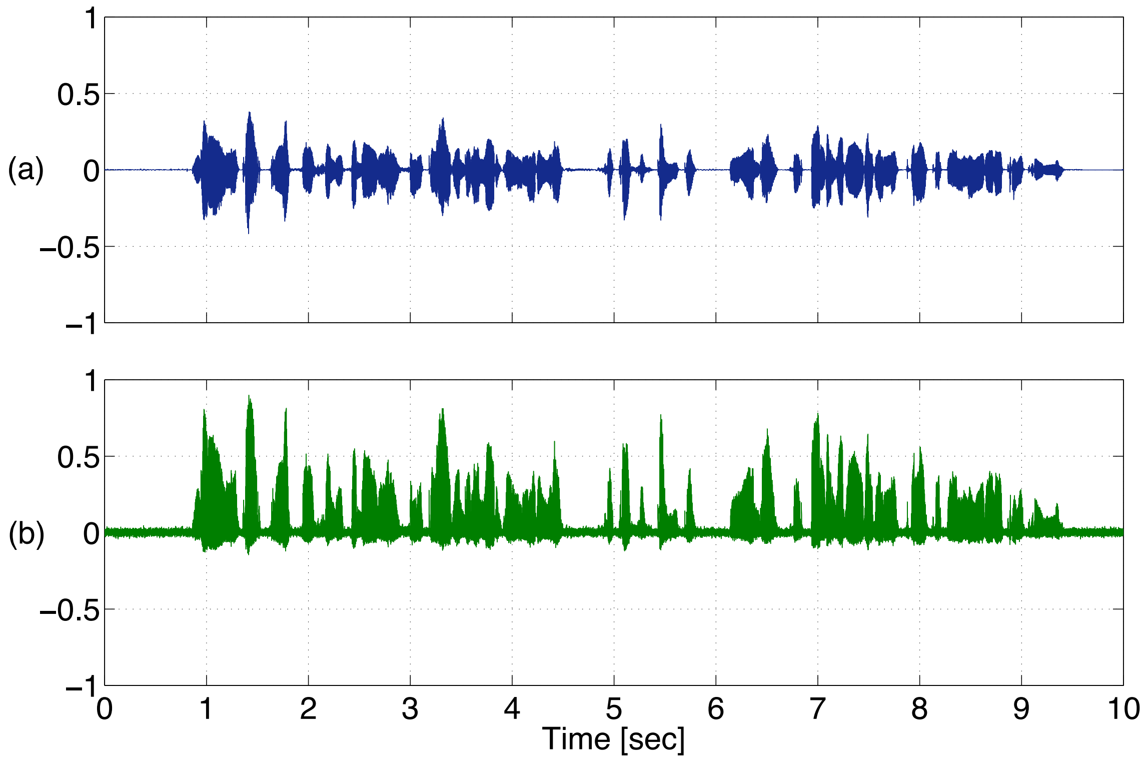


Fig. 9.3: *(a) Far-end female speech input signal. (b) Signal acquired by the microphone after being distorted and reverberated.*

distortion on the speech input signal using a clipping threshold of $\zeta = 1/8$.

9.3.1 Performance improvement of SFLAFs

First of all it is important to show the performance improvement brought by SFLAF compared to FLAF. We use the same parameter setting for both the FLAF and the SFLAF. The input buffer length is set to $M_e = M$, i.e. for the FLAF we use a filter length which is the same of the AIR length and for the SFAF we use the same length for both the linear and nonlinear filters. We choose an expansion order of $P = 5$ and a step size parameter of $\mu = 0.2$ for the FLAF and for both the filters of the SFLAF. Both FLAF and SFLAF are memoryless. All the adaptive filters are updated using an NLMS algorithm. We compare the performance of FLAF and SFLAF in terms of ERLE, both for the white Gaussian noise input and for the female speech input. Results

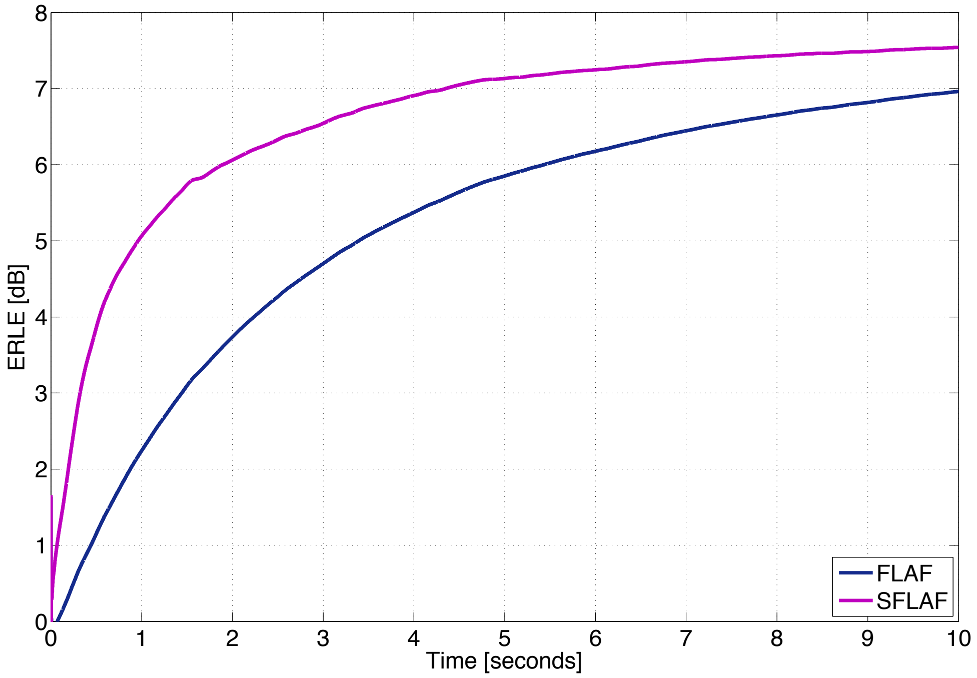


Fig. 9.4: *Performance comparison in terms of ERLE between an FLAF and an SFLAF in case of white Gaussian input.*

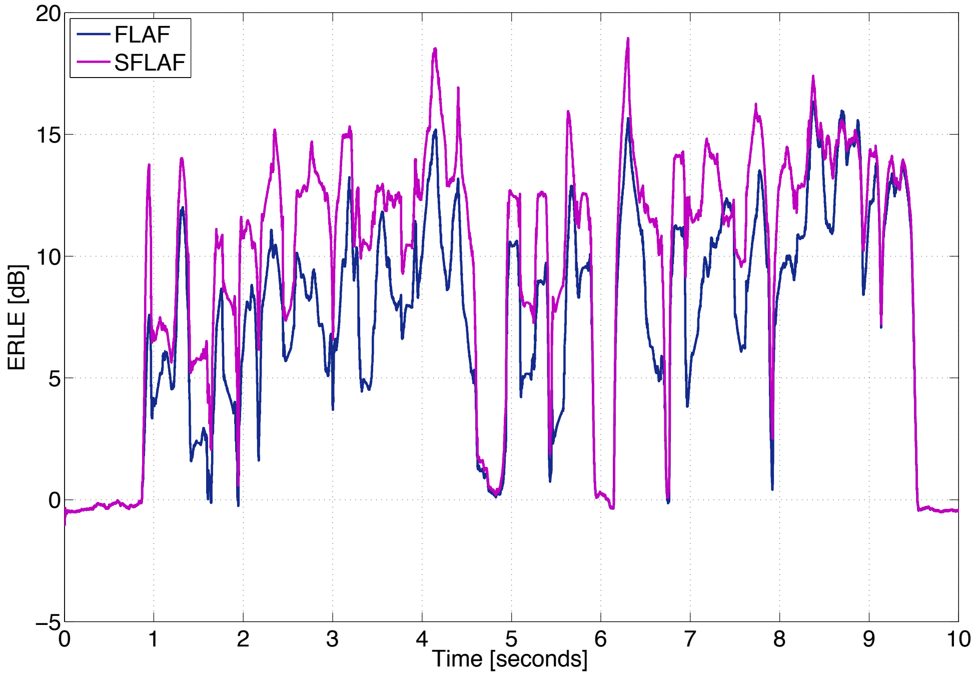


Fig. 9.5: *Performance comparison in terms of ERLE between an FLAF and an SFLAF in case of female speech input.*

are respectively depicted in Fig. 9.4 and Fig. 9.5 in which is evident the performance improvement brought by the SFLAF due to the separation of linear and nonlinear elements. Moreover, it has to be considered that it is also possible to change some parameters values of the SFLAF, such as the step size parameter and the input buffer length of the nonlinear path. A proper choice of such parameters may bring a further improvement of the performance of the SFLAF in terms of ERLE.

9.3.2 An effective alternative to Volterra filters

In the following set of experiments we compare the overall performance of an SFLAF with that of a VF, which, as previously said, remains the most popular NAEC in literature (see Section 7.1). We adopt the parameter setting

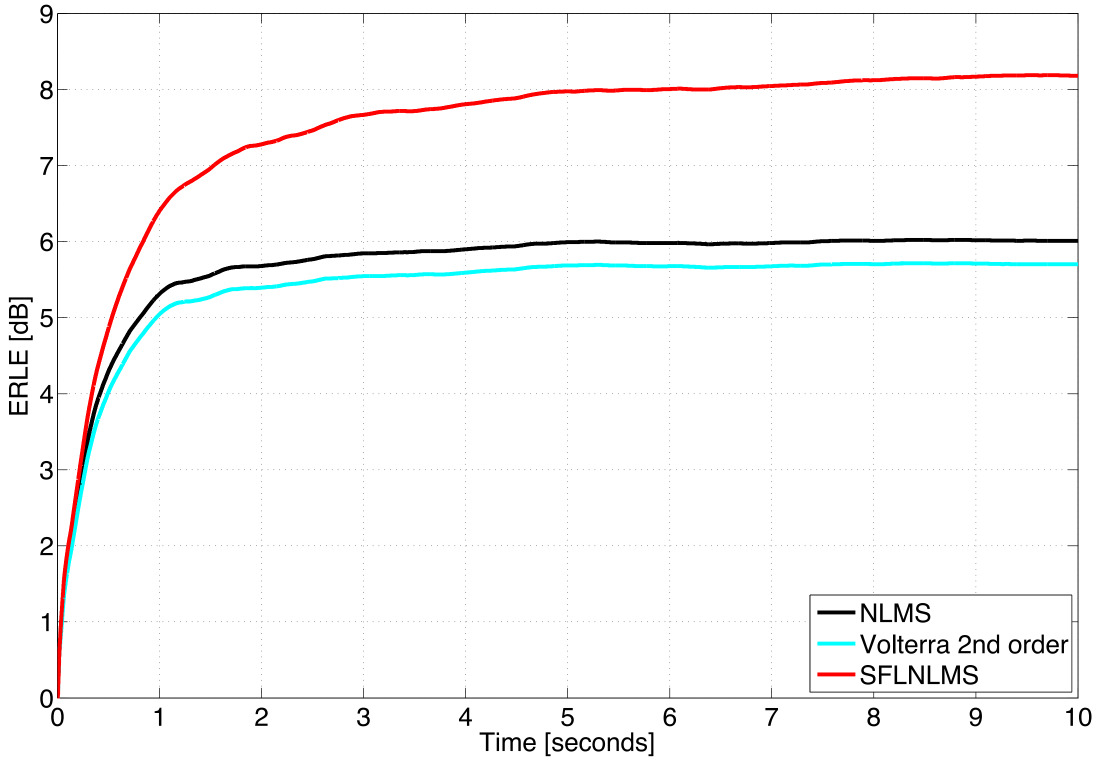


Fig. 9.6: *Performance comparison in terms of ERLE between the SFLAF and a 2nd order VF in case of white Gaussian input.*

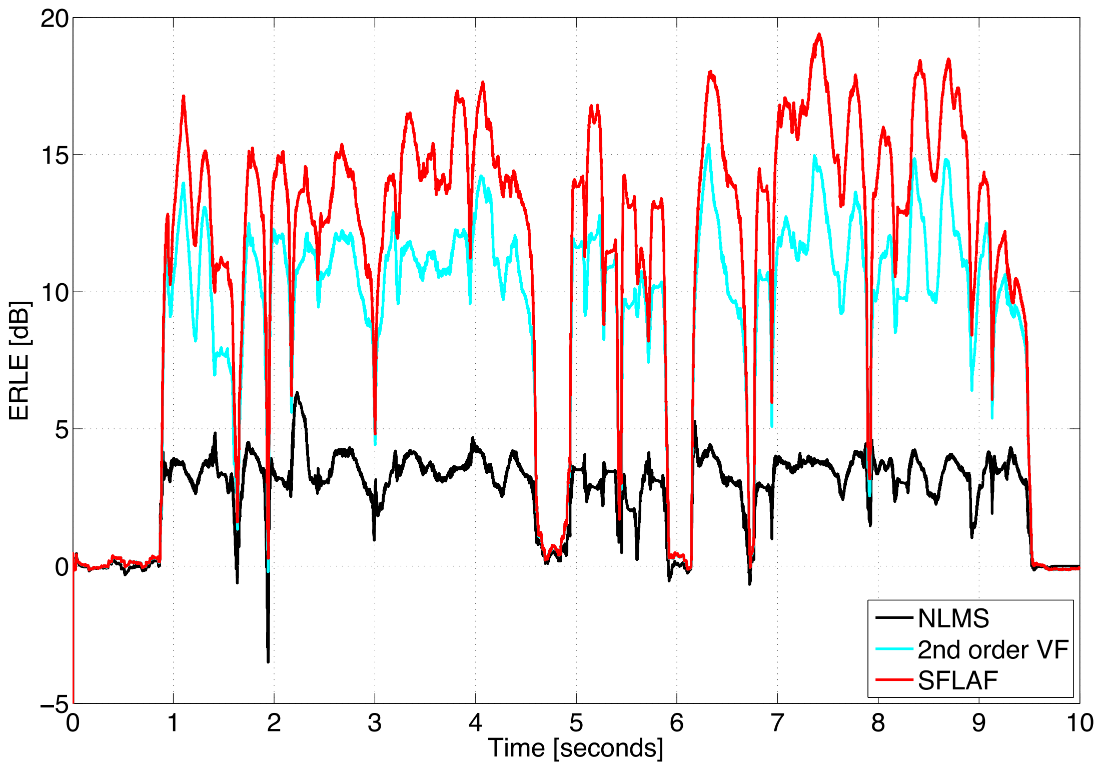


Fig. 9.7: *Performance comparison in terms of ERLE between the SFLAF and a 2nd order VF in case of female speech input.*

of the previous subsection for the SFLAF, but an expansion order of $P = 3$. The same buffer length, step size value and updating algorithm are also used for the adaptive Volterra filter, which is of the second order. Performance are evaluated in terms of ERLE for both the white Gaussian input and for the speech input, and results are respectively depicted in Fig. 9.6 and Fig. 9.7, where also the performance of an NLMS is taken into account as standard reference. Results show that SFLAF overcomes VF in terms of ERLE performance, thus resulting an effective alternative to VF for NAEC.

In terms of computational load, an SFLAF results more advantageous compared with a VF, especially when the SFLAF is memoryless. In a case like the one investigated, it is more convenient to use a memoryless SFLAF since the difference with an SFLAF with memory is quite small as it is possible to see from the comparison in Fig. 9.8. However, when the system may

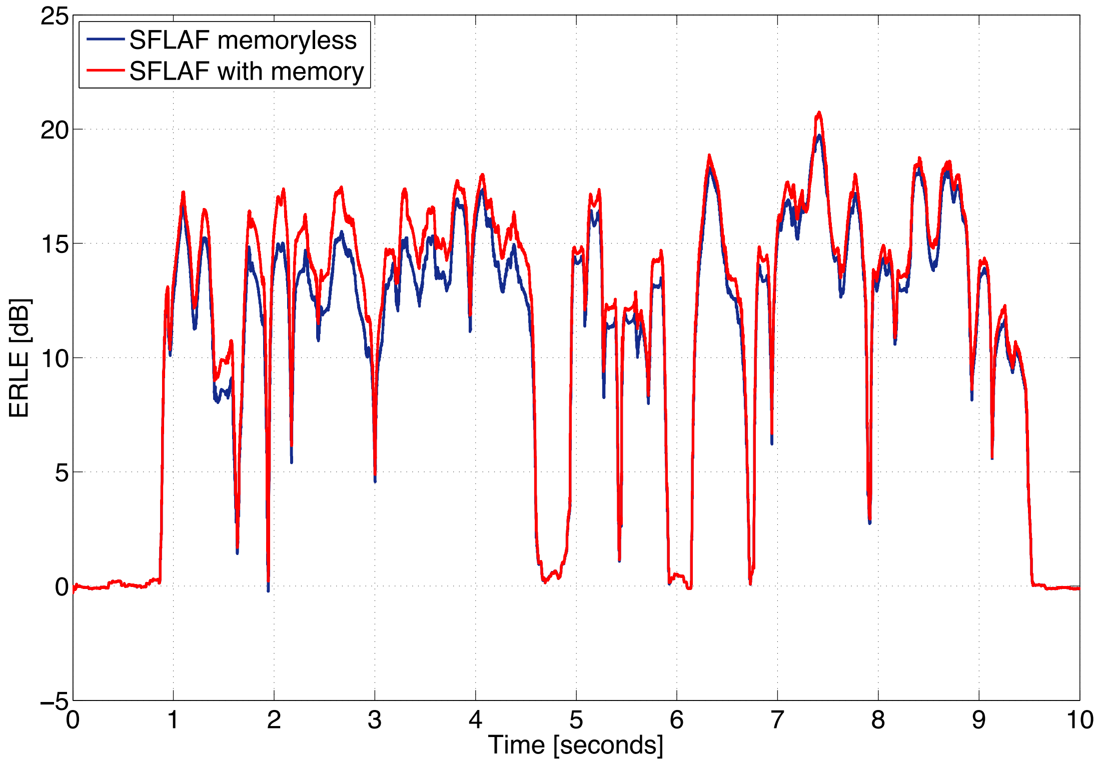


Fig. 9.8: *Performance comparison in terms of ERLE between an SFLAF with memory and a memoryless SFLAF.*

introduce a more dynamic nonlinearity than the used soft-clipping an SFLAF with memory may result definitely the best choice. It has to be taken into account that for speech input the use of an APA instead of the NLMS for the linear path of the SFLAF may bring further improvements in terms of ERLE. Further experiments on FLAFs for NAEC can be found in [29, 27].

9.4 CONCLUSIONS

In this chapter we have introduced a new class of nonlinear adaptive algorithms based on the FLAF model, described in Chapter 8, for acoustic applications. In particular, we have investigated the performance of the *splitting functional link adaptive filters* for NAEC, thus resulting an effective alternative to adaptive Volterra filters. The nonlinear model of SFLAFs has a great worthiness in this research project since it opens new research scenarios in the nonlinear acoustic echo cancellation due to the fact that SFLAFs allow future developments. In fact, using proper adaptive algorithms and nonlinear expansion it is possible to achieve further improvements in terms of ERLE. Moreover, it is possible to apply the proportionate techniques, introduced in the chapters of Part II, thus achieving a better modelling of nonlinearities, especially when they are highly time-variant.

PART IV

ROBUST ADAPTIVE FILTERING ARCHITECTURES

—Wherever we are, what we hear is mostly noise.
When we ignore it, it disturbs us.
When we listen to it, we find it fascinating.
John Cage

10

Filtering Architectures Based on Adaptive Combination of Filters

Contents

In the previous two parts of this work we have seen interesting adaptive algorithms for linear and nonlinear modelling of the acoustic impulse response. Even if such algorithms have shown remarkable results not always they provide optimal performance. In fact, they might suffer the initial choice of parameter settings when conditions of the environment, or in general of a system to identify, change during the adaptation, such that the initial setting becomes unsatisfying. In the linear case, such a situation may occur due to a nonstationary or a change in the environment which leads to a different choice of the step size parameter rather than the filter length or the regularization factor. Similarly, in the nonlinear case, a kind of nonlinearity highly varying, in amplitude or in time, may require to change the filter design during the adaptation. Moreover, another important troubling situation occurs when the desired signal is not known *a priori*, thus it is difficult to choose whether adopting a linear filter or a nonlinear model.

In order to tackle these problems we introduce robust adaptive filtering architectures based on the *adaptive combination of filters*. The idea of filters combination is very interesting because it is possible to model a wide range of applications [81, 67]. Using such technique it is possible to develop *combined filtering architectures* able to change their parameter setting automatically during the adaptation. An experimental example of combined filtering architectures for acoustic application can be found in Chapter 11.

Moreover, the adaptive combination of filters may be used also to develop *collaborative filtering architectures* able to model an impulse response apart from its nature, whether it is linear or nonlinear. This results very useful in acoustic applications, such as AEC, when it is not possible to know *a priori* if the AIR conveys any nonlinearity, thus biasing the design choices about an acoustic echo canceller. An experimental example of collaborative filtering architecture for AEC can be found in Chapter 12.

However, first of all in this chapter it is necessary to introduce the adaptive combination of filters.

10.1 ADAPTIVE COMBINATION OF FILTERS

Real-world processes comprise both linear and nonlinear components, together with deterministic (that can be precisely described by a set of equations) and stochastic ones. In this way, models used to describe these real-world processes can be classified with a certain degree of nonlinearity and uncertainty, and described in a diagram (see Fig. 10.1). In literature only few cases

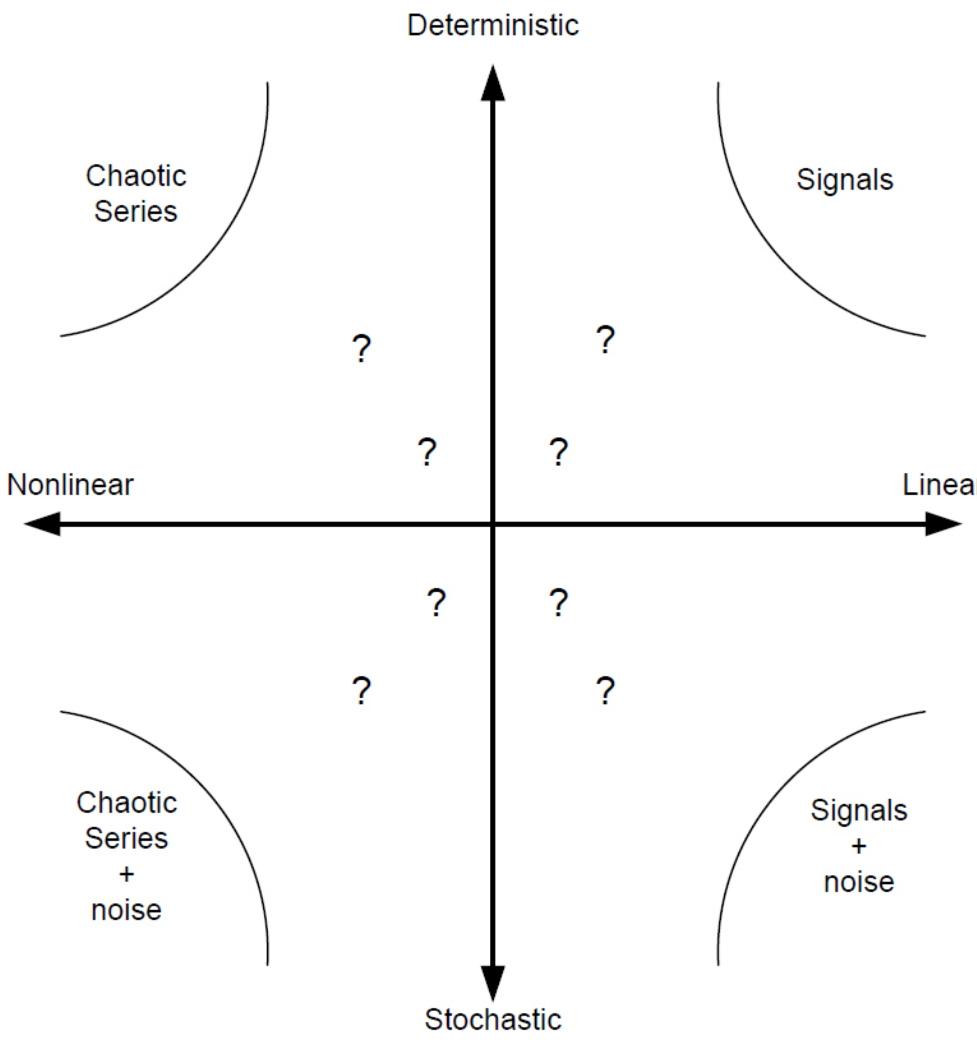


Fig. 10.1: *Possible variety of signals spanned by a certain degree of nonlinearity and uncertainty.*

as the linear stochastic ARMA and chaotic models are well understood, while real-world processes are often a combination of the previous four possibilities. In order to automatically take into account all the previous possibilities, a possible solution is to think to a system that automatically selects the right subsystem working on the relative quadrant.

It is possible to generalize Fig. 10.1 to the adaptive filtering, such that each subsystem corresponds to an adaptive filter. Using the fusion of the outputs of adaptive filters it is possible to produce a single hybrid filtering architecture

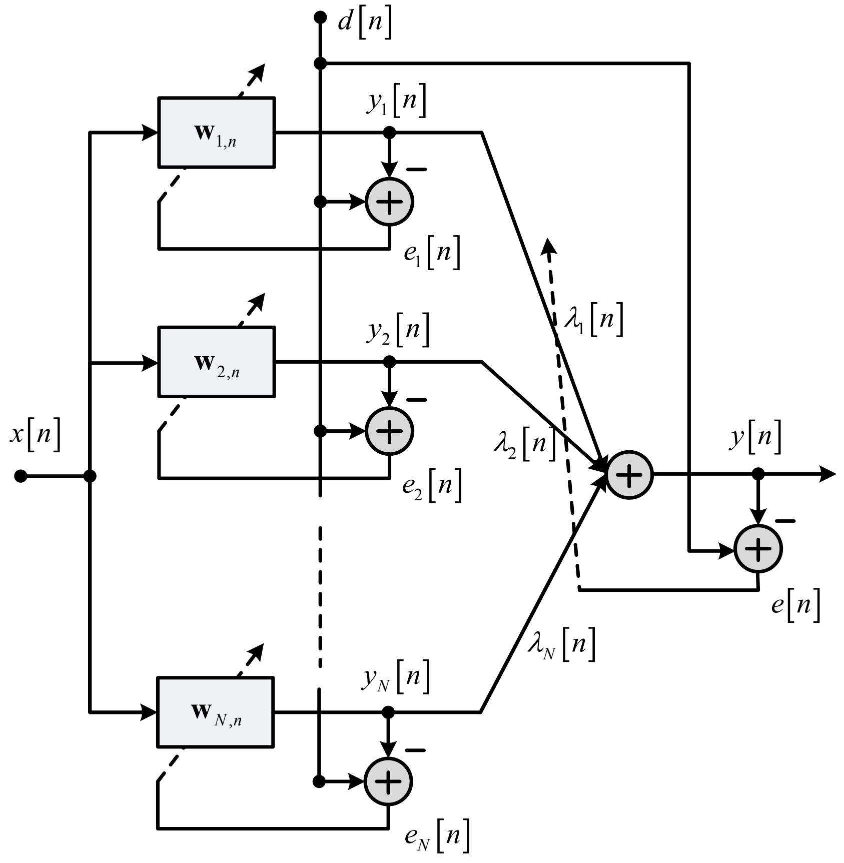


Fig. 10.2: *Adaptive combination of transversal adaptive filters.*

which provides at each time-instant the best performance among those of individual adaptive filters [67].

Adaptive combination of filters, as depicted in Fig. 10.2, consists of multiple individual adaptive subfilters operating in parallel and all feeding into a mixing algorithm which produces the single output of the filter [5, 73]:

$$\begin{aligned} y[n] &= \sum_{i=1}^{N} \lambda_i[n]\, y_i[n] \\ &= \sum_{i=1}^{N} \lambda_i \mathbf{x}_{i,n}^T \mathbf{w}_{i,n-1} \end{aligned} \tag{10.1}$$

where N is the number of filters in parallel, $y_i[n]$, are the outputs of the individual filters, with $i = 1, \ldots, N$, and $\lambda_i[n]$ are the mixing parameters, which are nothing but the coefficients of the filter on the output stage. Such mixing parameters can be updated using an adaptive algorithm. Therefore, the mixing coefficients are also adaptive and combine the outputs of each subfilter based on the estimate of their current performance on the input signal from their instantaneous output error. The mixing parameters are updated in such a way to minimize the global MSE in output. This minimization may be subjected to a constraint. The most used optimization constraints in the adaptive combination of filters are the affine and the convex constraints.

The *affine combination of adaptive filter* is characterized by an affine constraint, according to which:

$$\sum_{i=1}^{N} \lambda_i[n] = 1. \tag{10.2}$$

On the other side, the *convex combination of filters*, in addition to satisfy the affine constraint, is characterized by the fact that all the mixing parameters are not negative, i.e.:

$$\sum_{i=1}^{N} \lambda_i[n] = 1 \quad \text{with} \quad 0 \le \lambda_i[n] \le 1, \;\; i = 1, \ldots, N \tag{10.3}$$

In the next section we deepen the convex combination which is quite used in acoustic applications.

10.2 CONVEX COMBINATION OF ADAPTIVE FILTERS

A simple form of mixing algorithm for two adaptive filters is a convex combination. Convexity can be described as [26]:

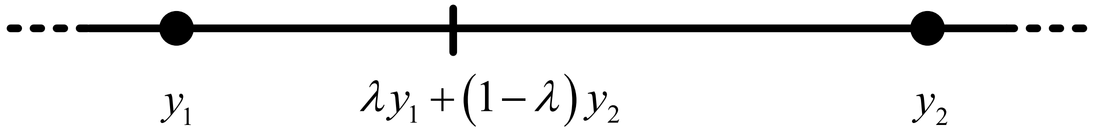


Fig. 10.3: *Convexity.*

$$\lambda y_1 + (1-\lambda) y_2 \tag{10.4}$$

where $\lambda \in [0, 1]$. For y_1 and y_2 being two points on a line, as shown in Fig. 10.3, their convex mixture (10.4) will lie on the same line between y_1 and y_2. The convex combination between two adaptive filters is represented in Fig. 10.4, in which, due to the convex constraint, the mixing parameters can be written as $\lambda_1 = \lambda$ and $\lambda_2 = 1 - \lambda$.

Therefore, in this case the output of the combined structure can be written as:

$$y[n] = \lambda y_1[n] + (1-\lambda) y_2[n] \tag{10.5}$$

It has been showed, in [5, 6], that the convex combination method is universal with respect to the component filters, i.e., in steady-state, it performs at least as well as the best component filter. Furthermore, when the correlation between the *a priori* errors of the components is low enough, their combination is able to outperform both of them [6]. This is the reason why the convex combination of filters is very attractive in adaptive filtering. In fact, it is known that on-line adaptation of certain filter parameters or even cost functions has been attempted to influence filter performance, such as adjusting the forgetting factor of recursive least squares (RLS) algorithms [164] or minimizing adjustable cost functions [25, 105]. However, a widespread use of adaptive combination of filters is to optimally set the step size parameter. Variable step size adaptive filters (see also Section 5.5) allow the filters to dynamically adjust their performance in response to conditions in the input data and error signals [58, 75, 124]. For example, it is possible to choose a convex combination of two adaptive filters [84, 8], one fast, i.e. with a large step size value, and one slow, i.e. with a small step size value. These filters are combined in such a manner that the advantages of both component filters are kept: the rapid convergence from the fast filter, and the reduced steady-state error from the slow filter. This scheme, that has also proven to outperform previous variable step approaches, is an analogy of a well-known neurological fact: human brains combine fast and coarse reactions against abrupt changes in the environment, with an early processing at the amygdala, and more elaborated but slower responses taken in the neocortex at a conscious level [7].

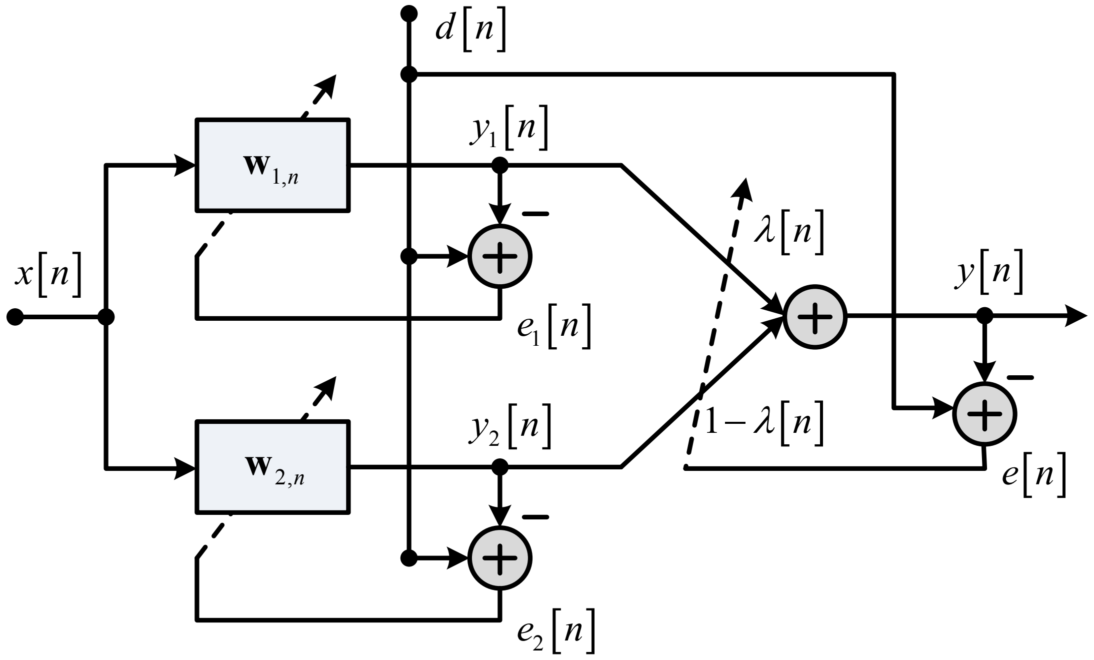


Fig. 10.4: *Convex combination of two adaptive filters.*

10.3 ADAPTATION OF MIXING PARAMETERS

The adaptation of the mixing parameters follows the updating rule of stochastic gradient adaptive algorithms (see Section 4.4). As it is possible to see also from Fig. 10.2 and Fig. 10.4, the individual filters are independently adapted using their own error signals, while the combination, both affine and convex, is adapted by means of a stochastic gradient algorithm in order to minimize the error of the overall structure. In this section we introduce the LMS and the NLMS adaptation for the mixing parameters, however other stochastic gradient algorithms might be adopted.

10.3.1 LMS adaptation of a convex combination of two filters

Let us consider the convex combination of two adaptive filters, as depicted in Fig. 10.4, described by equation (10.5). Let M the length of both the adaptive filter and let the input signal buffer $\mathbf{x}_n \in \mathbb{R}^M$. The *least mean square* updating equations for the two filters result:

$$\mathbf{w}_{i,n} = \mathbf{w}_{i,n-1} + \mu_i \mathbf{x}_n^T e_i\left[n\right], \quad \text{with } \; i = 1,2 \tag{10.6}$$

where:

$$e_i\left[n\right] = d\left[n\right] - y_i\left[n\right] \tag{10.7}$$

is the instantaneous error relative to individual filters.

Concerning the mixing parameter $\lambda\left[n\right]$, the adaptation may be carried out in convex mode imposing that $0 \leq \lambda\left[n\right] \leq 1$ by means of a sigmoidal activation function defined as:

$$\begin{aligned}\lambda\left[n\right] &= \mathrm{sgm}\left(a\left[n\right]\right) \\ &= \frac{1}{1+e^{-a[n]}}\end{aligned} \tag{10.8}$$

i.e., such that $\lambda\left[n\right]$ derive from the adaptation of an auxiliary parameter, $a\left[n\right]$, which is updated by means of a gradient descent rule, such as $a\left[n+1\right] = a\left[n\right] + \Delta a\left[n\right]$. Therefore, $\Delta a\left[n\right]$ may be computed applying a *least mean square* adaptation rule:

$$\begin{aligned}\Delta a\left[n\right] &= -\frac{1}{2}\mu_a \frac{\partial e^2\left[n\right]}{\partial a\left[n\right]} \\ &= -\mu_a e\left[n\right] \frac{\partial\left(d\left[n\right] - \lambda\left[n\right] y_1\left[n\right] - \left(1-\lambda\left[n\right]\right) y_2\left[n\right]\right)}{\partial \lambda\left[n\right]} \frac{\partial \lambda\left[n\right]}{\partial a\left[n\right]} \\ &= \mu_a e\left[n\right]\left(y_1\left[n\right] - y_2\left[n\right]\right)\lambda\left[n\right]\left(1-\lambda\left[n\right]\right).\end{aligned} \tag{10.9}$$

where μ_a is a step size parameter.

The benefits of employing the sigmoidal activation function are twofold. First, it serves to keep $\lambda\left[n\right]$ within the desired range $\left[0,1\right]$. Second, as seen from (10.9), the adaptation rule of $a\left[n\right]$ reduces both the stochastic gradient noise and the adaptation speed near $\lambda\left[n\right] = 1$ and $\lambda\left[n\right] = 0$ when the combination is expected to perform close to one of its component filters without degradation. Still, note that the update of $a\left[n\right]$ in (10.9) stops whenever $\lambda\left[n\right]$ is too close to the limit values of 0 or 1. To circumvent this problem, we shall restrict the values of $a\left[n\right]$ to lie inside a symmetric interval $\left[-a^+, a^+\right]$, which limits the permissible range of $\lambda\left[n\right]$ to $\left[1-\lambda^+, \lambda^+\right]$, where $\lambda^+ = \mathrm{sgm}\left(a^+\right)$ is a constant close to 1. In this way, a minimum level of adaption is always guaranteed.

10.3.2 A normalized adaptation

In [9] a normalized adaptation scheme has been introduced in order to be more robust to changes in the filtering scenario. Considering equation (10.7), it is possible to rewrite (10.5) as:

$$y\left[n\right] = y_2\left[n\right] + \lambda\left[n\right]\left(e_2\left[n\right] - e_1\left[n\right]\right) \tag{10.10}$$

so that we can think of the overall combination scheme as a two-stage adaptive filter. In the first stage, the two component filters operate independently of each other and according to their own rules, while the second layer consists of a filter with input signal $e_2[n] - e_1[n]$ that minimizes the overall error.

This interpretation of the combination scheme suggests that further advantages could be obtained if we used a normalized LMS rule for adapting the mixing parameter rather than standard LMS. Since $e_2[n] - e_1[n]$ plays the role of the input signal at this level, it makes sense to use the following adaptation scheme:

$$a[n+1] = a[n] + \mu_a \frac{\lambda[n](1-\lambda[n])}{(e_2[n]-e_1[n])^2} e[n](e_2[n]-e_1[n]) . \tag{10.11}$$

In practice, however, the performance of this scheme is quite unsatisfactory given that the instantaneous value $(e_2[n] - e_1[n])^2$ is a very poor estimate of the power of the “second stage” input signal. Similar to the normalized LMS (NLMS) algorithm with power normalization [120], better behaviour is obtained from:

$$a[n+1] = a[n] + \frac{\mu_a}{r[n]}\lambda[n](1-\lambda[n])e[n](e_2[n]-e_1[n]) \tag{10.12}$$

where:

$$r[n] = \beta r[n-1] + (1-\beta)(e_2[n]-e_1[n])^2 \tag{10.13}$$

is a rough (low-pass filtered) estimate of the power of the signal of interest. Selection of the forgetting factor β is rather easy. For instance, using $\beta = 0.9$ gives a good enough approximation, and typically ensures that $r[n]$ is adapted faster than any component filter.

10.4 CONCLUSIONS

In this chapter adaptive combination of filters has been introduced. In the following two chapters we use this technique to develop robust combined filtering architectures, for the linear modelling, and collaborative filtering architectures, for the nonlinear modelling. Adaptive combination of filters still remains a fertile argument for future researches since, as we have seen, the adaptation of mixing parameters is conducted by means of stochastic adaptive algorithms; it can be thinkable to adopt more appropriate adaptation rules, especially for the modelling of an acoustic path.

11

Combined Architectures for Adaptive Beamforming

Contents

ADAPTIVE combination of filters is a very effective and flexible approach to balance the compromises inherent to the settings of adaptive filters. In this chapter we exploits the capabilities of adaptive combination of filters in order to introduces novel adaptive beamforming methods for speech enhancement applications, designed to be robust against adverse environment conditions. The proposed architectures derive from the *generalized sidelobe canceller* (GSC); the novelty relies on the use of hybrid adaptive sidelobe cancelling structures which allow the system to achieve robustness in nonstationary environments. The novel structures are based on the convex combination of two *multiple-input single-output* (MISO) adaptive systems with complementary capabilities. The whole beamformer benefits from the combination and results to be able to preserve the best properties of each system. Experiments show that the proposed beamforming systems are capable of enhancing the desired speech signal even in adverse environment conditions[1].

11.1 INTRODUCTION

In immersive speech communications, taking place in multisource environments, the presence of interfering signals and reverberation may cause the loss of spatial information, thus resulting in compromising the speech intelligibility. In order to tackle this problem, speech enhancement systems are widely employed in distant talking applications. Microphone array beamforming represents a class of such speech enhancement techniques which are highly effective in acquiring a desired source signal while reducing the interfering components, thus resulting in recovering the binaural perception. Beamforming systems exploit the properties of microphone interfaces which facilitate binaural hearing.

The *generalized sidelobe canceller* (GSC) [54] is one of the most popular beamforming techniques for speech enhancement. The potency of a GSC system strictly relies on the adaptive algorithm chosen to perform the sidelobe canceller in the adaptive path. Generally the adaptation of filters in time-domain may be performed by *gradient*-based adaptive algorithms (see Section 4.4), such as the LMS-type algorithms. Although this family of algorithms is computationally quite cheaper, when the filter length is quite large a rather slow convergence occurs [120], thus the adaptation of the filter weights becomes unpractical in hands-free applications. Another time-domain standard approach is *Hessian*-based adaptive filtering, which is typical of algorithms such as the RLS. The latter approach displays a faster convergence rate compared with gradient-based algorithms [120]. However, RLS adaptive filtering entails a high computational complexity; therefore, adaptation may become prohibitively expensive, thus compromising real-time implementations. Moreover, the RLS may perform worse than LMS algorithm in nonstationary environment, depending on the statistics of acquired source signals [41]. A good compromise between performance and computational load may be obtained by using the family of APA [98], which is quite used in adaptive beamforming [163, 30], since it shows better convergence rates and manageable computational complexity compared with other time-domain algorithms. Moreover, APA is the best suitable algorithm to process speech signals compared with other classic time-domain adaptive algorithms. However, despite its good capabilities, APA suffers adverse environment conditions, especially in presence of multiple nonstationary sources which make the adaptation process unstable and reduce speech enhancement performance.

In order to address this problem we propose robust microphone beamforming architectures based on the adaptive combination of MISO systems, that are nothing but filter banks. Combined adaptive schemes are usually adopted with filters of the same family and complementary properties, e.g. using different step sizes, different filter lengths; however, they are used even with filters of different families using different updating rules or different

[1] The work in this chapter has been performed while the author was a Ph.D. student collaborating with the Fondazione Ugo Bordoni.

cost functions [82, 160, 17, 126, 74]. A combined architecture is capable of adaptively switching between filters according to the best performing filter, thus always providing the best possible filtering (see Chapter 10).

In this chapter we propose two different beamforming architectures based on the combination of MISO systems using different updating approaches. In particular, we propose a *system-by-system* combined architecture, in which the overall output of the first MISO system is convexly combined with the overall output of the second MISO system, and a *filter-by-filter* combined architecture, in which each adaptive filter of the first MISO system is convexly combined with the correspondent filter of the second MISO system. Moreover, in order to use the best parameter setting for each filter and further improve the tracking performance we use both the combination of filters with different updating approaches and the combination of filters with different step size values in a *multi-stage* combined architecture in which the filtering process is carried out in two steps [73].

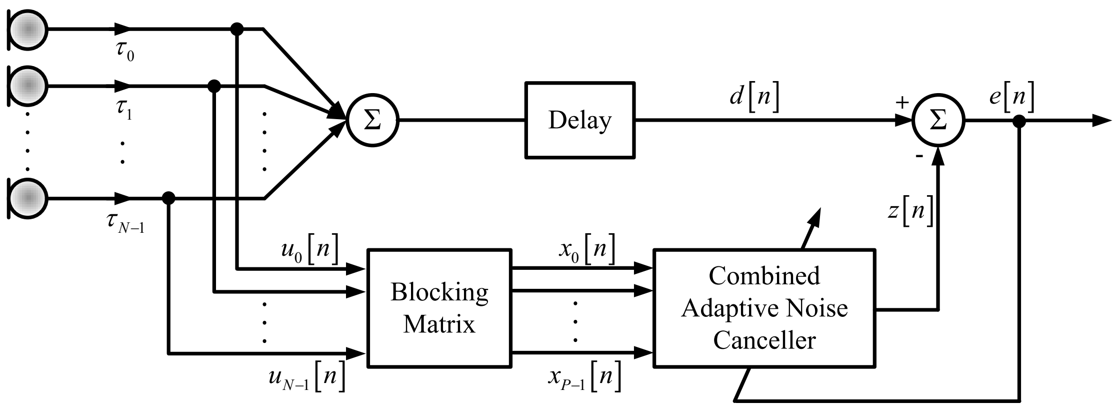


Fig. 11.1: *Microphone array beamforming architecture.*

11.2 COMBINED MICROPHONE ARRAY BEAMFORMING ARCHITECTURE

The beamforming architecture adopted in this paper is a typical GSC configuration [54] composed of a microphone array interface, a fixed *delay-and-sum beamformer* (DSB), and an *adaptive noise cancelling* (ANC) path, as depicted in Fig. 11.1. Let us consider a microphone array interface composed of N sensors. The signal $u_i\left[n\right]$ acquired by the i-th microphone, with $i = 0, \ldots, N-1$, is a delayed replica of the target signal $s\left[n\right]$ convolved with the (AIR) $\mathbf{a}_i$ between the i-th microphone and the desired source with the addition of background noise $v_i\left[n\right]$. The DSB spatially aligns the microphone signals with reference to the desired source direction, yielding the speech reference signal $d\left[n\right]$:

$$\begin{aligned} d\left[n\right] &= \sum_{i=0}^{N-1} u_i\left[n\right] \\ &= \sum_{i=0}^{N-1} \sum_{m=0}^{M-1} a_i\left[m\right] s\left[n-m-\tau_i\right] + v_i\left[n\right] \end{aligned} \tag{11.1}$$

where we suppose that each AIR between the desired source and the i-th microphone has the same length denoted with M. τ_i represents the delay relative to the i-th microphone.

In the adaptive path of the beamformer, the *blocking matrix* (BM) generates the noise references $x_p\left[n\right]$, with $p = 0, \ldots, P-1$, being $P = N-1$. The BM is implemented by pairwise differences between microphone signals [20], i.e. the sum of the elements of each column, except the first one, is null.

The noise reference signals are then processed by means of the *combined adaptive noise canceller* (CANC), whose structure will be detailed in the next section. The goal of the CANC is to remove any residual noise components in the speech reference signal, minimizing the output power and yielding the beamformer output signal $e\left[n\right]$.

11.3 ADAPTIVE COMBINATION OF MISO SYSTEMS

11.3.1 Convex combination of adaptive filters using APA

The trademark of the proposed beamforming approach is represented by the structure of the CANC. Generally, a conventional ANC is composed of an adaptive filter bank forming an MISO system. However, the adopted architecture results from combinations of adaptive filters. In particular, the structure is composed of two or more different MISO systems, each bringing different filtering capabilities to the whole beamformer. Each MISO system receives the same input signals, which are the noise reference signals resulting from the BM. Taking into account a number J of MISO systems, the p-th filter of the j-th MISO system, with $j = 0, \ldots, J-1$, receives as input a noise reference matrix $\mathbf{X}_{n,p}^{(j)}$, defined similarly to (5.2), but using a projection order K_j relative to all the filters of the j-th MISO system. We denote the coefficient vector of the p-th filter belonging to the j-th MISO system at n-th time instant as $\mathbf{w}_{n,p}^{(j)} \in \mathbb{R}^M$, which contains the same number of coefficients, M, and is adapted according to the *affine projection algorithm* (APA) [98], whose updating rule is derived similarly to (4.42):

$$\mathbf{w}_{n,p}^{(j)} = \mathbf{w}_{n-1,p}^{(j)} + \mu_j \mathbf{X}_{n,p}^{(j),T} \left(\delta_j \mathbf{I} + \mathbf{X}_{n,p}^{(j)} \mathbf{X}_{n,p}^{(j),T} \right)^{-1} \mathbf{e}_n^{(j)} \qquad (11.2)$$

where $\mathbf{e}_n^{(j)} \in \mathbb{R}^{K_j}$ is the error vector of the j-th MISO system containing the last K_j samples of the j-th error signal, which results from:

$$\mathbf{e}_n^{(j)} = \mathbf{d}_n^{(j)} - \sum_{p=0}^{P-1} \mathbf{y}_{n,p}^{(j)} \qquad (11.3)$$

where $\mathbf{d}_n^{(j)} \in \mathbb{R}^{K_j}$ is the vector containing the last K_j samples of the desired signal and $\mathbf{y}_{n,p}^{(j)} \in \mathbb{R}^{K_j} = \mathbf{X}_{n,p}^{(j)} \mathbf{w}_{n-1,p}^{(j)}$ is the vector containing the K_j projections of the output signal relative to the p-th filter of the j-th MISO system. Moreover, in equation (11.2), the parameters μ_j and δ_j are respectively the *step size* and the *regularization factor* common for all the filters of the j-th MISO system.

Using the updating rule described by (11.2) it is possible to differentiate the considered MISO systems simply changing the values of the step sizes or of the projection orders. However, aside from the chosen distinguishing parameters, there are two ways to combine the MISO system. The first way is to convexly combine the outputs of the two MISO systems and the second is to combine each filter of the first MISO system with the correspondent filter of the second MISO system under a convex constraint. We denote the former way as *system-by-system combined architecture* and the latter as *filter-by-filter combined architecture*, which are both described in the following two subsections.

11.3.2 System-by-system combined architecture

The first proposed scheme is the *system-by-system* CANC, depicted in Fig. 11.2 (a). The output of each MISO system, that we denote as $y^{(j)}[n] = \sum_{p=0}^{P-1} y_p^{(j)}[n]$, yields two system outputs that are then convexly combined generating the overall CANC output:

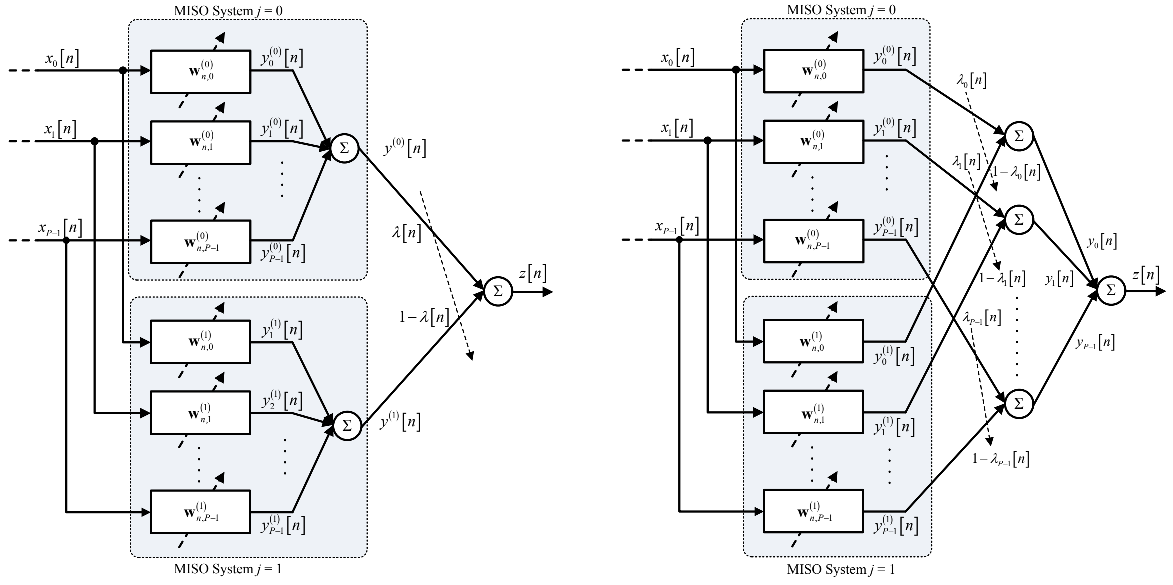
$$z[n] = \lambda[n]\, y^{(0)}[n] + (1 - \lambda[n])\, y^{(1)}[n] \qquad (11.4)$$

where $\lambda[n]$ is the *mixing parameter* (see Chapter 10). Therefore, the beamformer output signal $e[n]$, using the system-by-system combination, is achieved as $e[n] = d[n] - z[n]$.

The mixing parameter in (11.4) is usually updated using a gradient descent rule through the adaptation of an auxiliary parameter, $a[n]$, related to $\lambda[n]$ by a sigmoidal activation function, similarly to (10.12).

11.3.3 Filter-by-filter combined architecture

The second proposed scheme is the *filter-by-filter* CANC in which the output signal $z[n]$ is built in a different way. As it is possible to see in Fig.



(a) (b)

Fig. 11.2: *Combined adaptive noise canceller architectures: (a) system-by-system and (b) filter-by-filter combination schemes.*

11.2 (b), the p-th filter output of the first MISO system is convexly combined with the correspondent p-th filter output of the second MISO system, thus generating $P-1$ outputs, each relative to a noise reference:

$$y_p[n] = \lambda_p[n]\, y_p^{(0)}[n] + (1-\lambda_p[n])\, y_p^{(1)}[n] \tag{11.5}$$

where $\lambda_p[n]$ is the p-th *mixing parameter*, adapted using the p-th auxiliary parameter, $a_p[n]$, similarly to (10.12). Once computing the convex combinations, it is possible to achieve the CANC output signal $z[n]$ by summing the individual output contributions deriving from the combinations, as it is possible to see in Fig. 11.2 (b):

$$z[n] = \sum_{p=0}^{P-1} y_p[n] \tag{11.6}$$

from which we derive the overall beamformer output signal $e[n] = d[n] - z[n]$,

relative to the filter-by-filter combination scheme.

Both the combined architectures presented above improve the tracking capabilities of CANC giving robustness to the overall beamforming system in nonstationary environments.

11.4 MULTI-STAGE MICROPHONE ARRAY BEAMFORMING

The microphone beamforming schemes described in Section 11.3 are effective in presence of multiple nonstationary sources both choosing different step size values (μ_0 small and μ_1 large) and different projection orders ($K_0 = 1$ and $K_1 > 1$). However, further improvements may be achieved if we consider the joined capabilities deriving from choosing both different step size values and projection orders. To this end we propose a *multi-stage combined architecture* in which the filtering process may involve more convex combinations of MISO systems.

In particular, in order to yield an adaptive beamforming architecture robust against adverse conditions, we may consider a CANC composed of a number $J = 4$ of MISO systems, as depicted in Fig. 11.3, each bringing different capabilities to the whole architecture. We differentiate by twos the four systems according to the step size values and the projection orders. In particular, we choose a small step size $\mu_j = \mu_{\mathrm{A}}$ for $j = 0, 2$ and a large step size value $\mu_j = \mu_{\mathrm{B}}$ for $j = 1, 3$. Moreover, we update the first two MISO systems using a gradient-based algorithm and the second two systems with a Hessian-based algorithm. This is obtained by setting a unitary projection order $K_j = 1$ for $j = 0, 1$ and a superior projection order $K_j > 1$ for $j = 2, 3$.

The choice of different step size values affects the convex combinations on the first stage, in which the first MISO system is combined with the second and the third with the four. In this stage the convex combination may follow the system-by-system scheme or the filter-by-filter scheme. In Fig. 11.3 a multi-

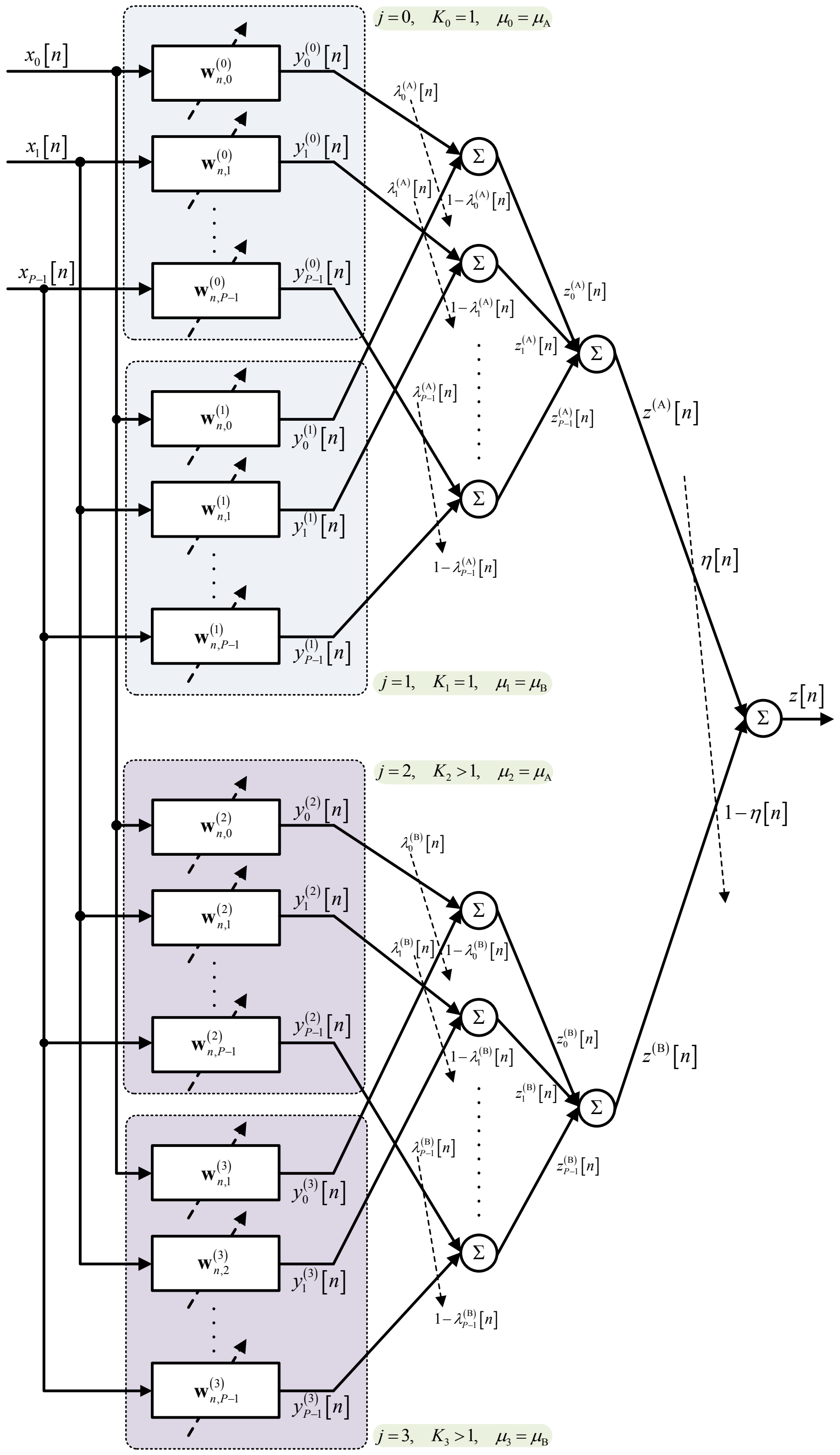


Fig. 11.3: *Multi-stage combined adaptive noise canceller.*

stage beamformer with a filter-by-filter scheme on the first stage is depicted. On the other hand, the choice of different projection order affects the convex combination on the second stage, in which the output signal resulting from the combination of the first and the second MISO systems is in turn combined with the output signal resulting from the combination of the third and the four MISO systems. The convex combination on the second stage follows the system-by-system combination scheme.

Output signals of the convex combinations on the first stage, denoted as $z^{(\mathrm{A})}[n]$ and $z^{(\mathrm{B})}[n]$, may be achieved similarly to (11.4), according to a system-by-system combination scheme, or similarly to (11.6), according to a filter-by-filter combination scheme as depicted in Fig. 11.3. In turn, the convex combination on the second stage may be achieved according to a system-by-system scheme, thus resulting the following output signal from the multi-stage CANC:

$$z[n] = \eta[n]\, z^{(\mathrm{A})}[n] + (1 - \eta[n])\, z^{(\mathrm{B})}[n] \tag{11.7}$$

where $\eta[n]$ is the mixing parameter of the second stage, even adapted using an auxiliary parameter.

Once computing the second stage convex combination, it is possible to derive the overall multi-stage beamformer output signal $e[n] = d[n] - z[n]$, as done for the single-stage combination schemes in Section 11.3.

The multi-stage beamforming architecture introduced above exploits the capabilities of each MISO system, thus improving speech enhancement performance compared to both conventional beamformers (using a single MISO ANC) and single-stage combined beamformers in presence of nonstationary interfering signals.

11.5 EXPERIMENTAL RESULTS

In the this section we carry out two different sets of experiments: the first set, in Subsection 11.5.1, aims at assessing the effectiveness of the described combined filtering schemes adopted in the proposed beamforming method; the second set of experiments, detailed in Subsection 11.5.2, is performed to evaluate the proposed combined beamforming architectures for speech enhancement application in a multisource scenario.

11.5.1 Convergence performance of combined architectures

In the first set of experiments we prove the filtering effectiveness of proposed CANC schemes through a tracking analysis which describes the convergence performance. To this end we use conventional ANC MISO systems and the proposed combined architectures to identify an unknown nonstationary system and to compare their performance.

The initial optimal solution is formed with $M = 7$ independent random values between -1 and 1. In the following examples the initial system is: $\mathbf{w}_1^{\mathrm{opt}} = \left[\begin{array}{ccccccc} 0.4125 & 0.7632 & -0.5484 & -0.6099 & -0.4622 & -0.4826 & -0.5296 \end{array}\right]^T$. The input signal is generated by means of a first-order autoregressive model, whose transfer function is $\sqrt{1-\alpha^2}/\left(1-\alpha z^{-1}\right)$, with $\alpha = 0.8$, fed with an i.i.d. Gaussian random process. The length of the input signal is of $L = 10000$ samples. However, in order to study the ability of combined schemes to react to nonstationary environments, at time instant $n = L/2$ the system changes into $\mathbf{w}_2^{\mathrm{opt}} = \left[\begin{array}{ccccccc} -0.4223 & 0.0848 & -0.1228 & 0.3876 & 0.9950 & 0.9806 & -0.2700 \end{array}\right]^T$. Furthermore, an additive i.i.d. noise signal $e_0[n]$ with variance $\sigma_0^2 = 0.01$ is added to form the desired signal.

In order to identify the unknown solutions $\mathbf{w}_1^{\mathrm{opt}}$ and $\mathbf{w}_2^{\mathrm{opt}}$ we use both conventional MISO systems and the adaptive combined filtering schemes described in Sections 11.3 and 11.4 and we compare their performance in terms of *excess mean square error* (EMSE), defined as $\mathrm{EMSE}[n] = \mathrm{E}\left\{\left(e[n] - e_0[n]\right)^2\right\}$,

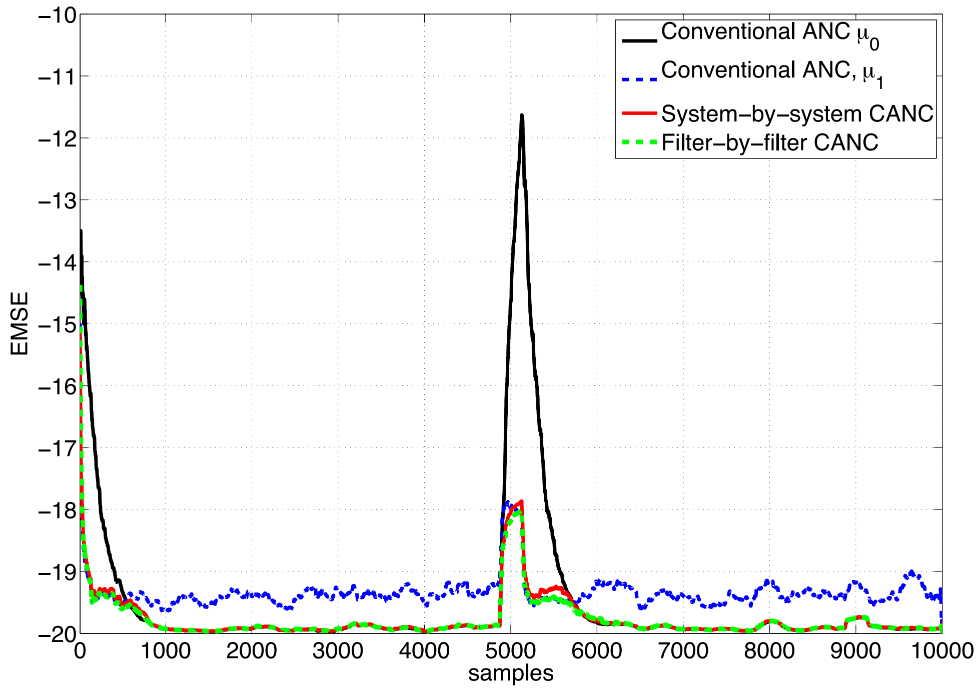


Fig. 11.4: *EMSE comparison between single-stage combined filtering architectures and conventional ones using the same projection order and different step size values.*

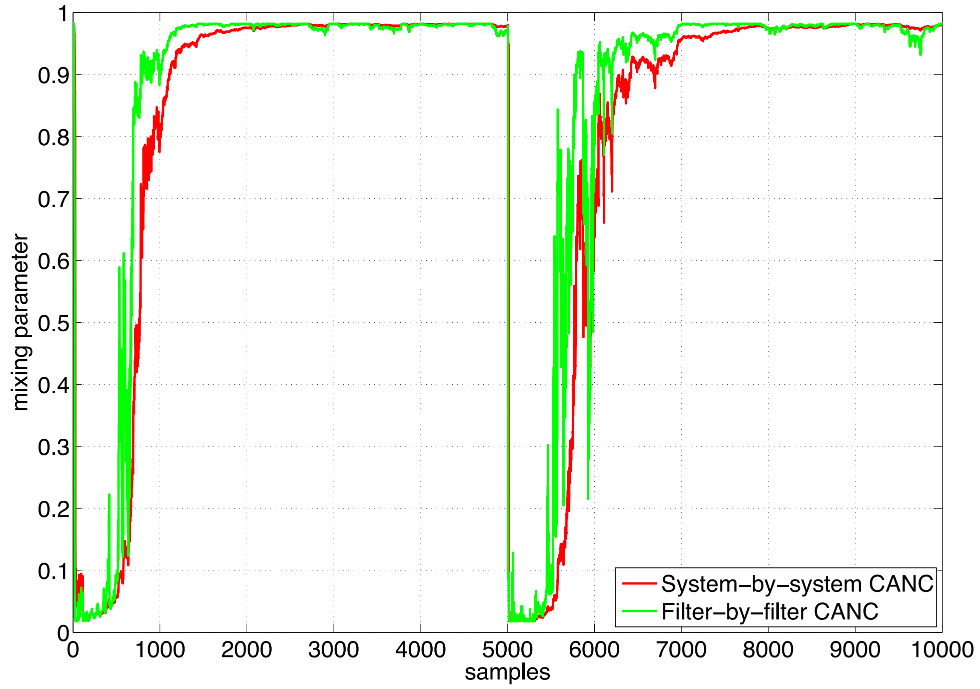


Fig. 11.5: *Behaviour comparison between the mixing parameter of the system-by-system CANC and the mixing parameter relative to the first channel of the filter-by-filter CANC.*

where $e[n]$ is the error signal of the filtering architecture, $e_0[n]$ is the additive noise signal (which is the same for all the filtering architectures) and the operator $\mathrm{E}\{\cdot\}$ is the mathematical expectation. The EMSE of each filtering structure is evaluated over 1000 independent runs. Moreover, in order to facilitate the visualization, the EMSE curves are filtered by a moving-average filter. All the filtering architectures, included the conventional ones, use MISO systems with $P = 4$ channels.

In a first experiment, we compare a conventional MISO architecture and both single-stage combined architectures described in Section 11.3, i.e. the system-by-system CANC and the filter-by-filter CANC. Both the system-by-system and the filter-by-filter schemes are composed of two MISO systems, as depicted in Fig. 11.2. All the MISO systems use APA filters. For the adaptation of the mixing parameter of the system-by-system filtering architecture we use a step size value of $\mu_{\mathrm{s}} = 10^2$, while a step size value of $\mu_{\mathrm{f}} = 10^3$ is adopted for the adaptation of all the mixing parameters of the filter-by-filter scheme. Both the step size values provide the best performance in each case. We evaluate the filtering architectures choosing the same projection order $K = 2$ for all the MISO systems and different step size values for the MISO systems of the combined schemes: a slower one $\mu_0 = 0.01$ and a faster one $\mu_1 = 0.1$. In Fig. 11.4 we have compared the performance of combined architecture with those of conventional ANC using both μ_0 and μ_1. As it is possible to see, both system-by-system and filter-by-filter schemes take advantage from using the combined filtering with respect to conventional filtering. In fact, combined schemes always show the behaviour of the best performing system and in transient state they behave even better than the best conventional filtering. Both the combined schemes provide good convergence performance, however the filter-by-filter scheme is slightly better than the system-by-system one due to the fact that the adaptation of the mixing parameters in the filter-by-filter scheme is faster than the system-by-system one, as can be seen in Fig. 11.5. This results in a quality improvement of the processed signal that can be

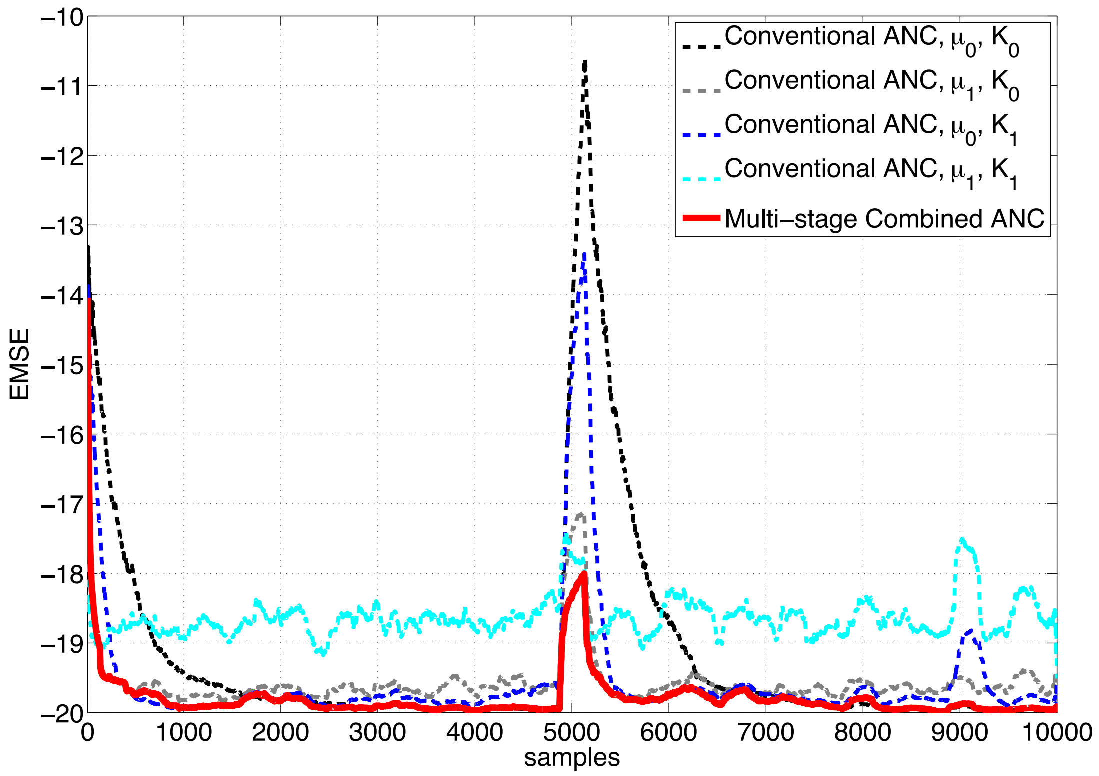


Fig. 11.6: *EMSE comparison between multi-stage combined filtering architectures and conventional ones.*

decisive in speech applications. A similar result was achieved choosing the same step size value and different projection orders.

In a second experiment, keeping the same scenario, we study now the convergence performance of the multi-stage combined architecture. As stated in Section 11.4, in a multi-stage combined scheme the combinations on the first stage may be performed in both system-by-system or filter-by-filter way. However, in light of previous result we take into account the performance of a multi-stage scheme whose combinations on the first stage are performed according to a filter-by-filter scheme, as depicted in Fig. 11.3. Therefore, we consider a two-stage combined scheme composed of four different MISO systems and we choose two different step size values, μ_0 = 0 .01 and μ_1 = 0 .1, and two different projection orders K$_0$ = 1 and K$_1$ = 4 . In Fig. 11.6 the comparison between the multi-stage combined filtering architecture and the

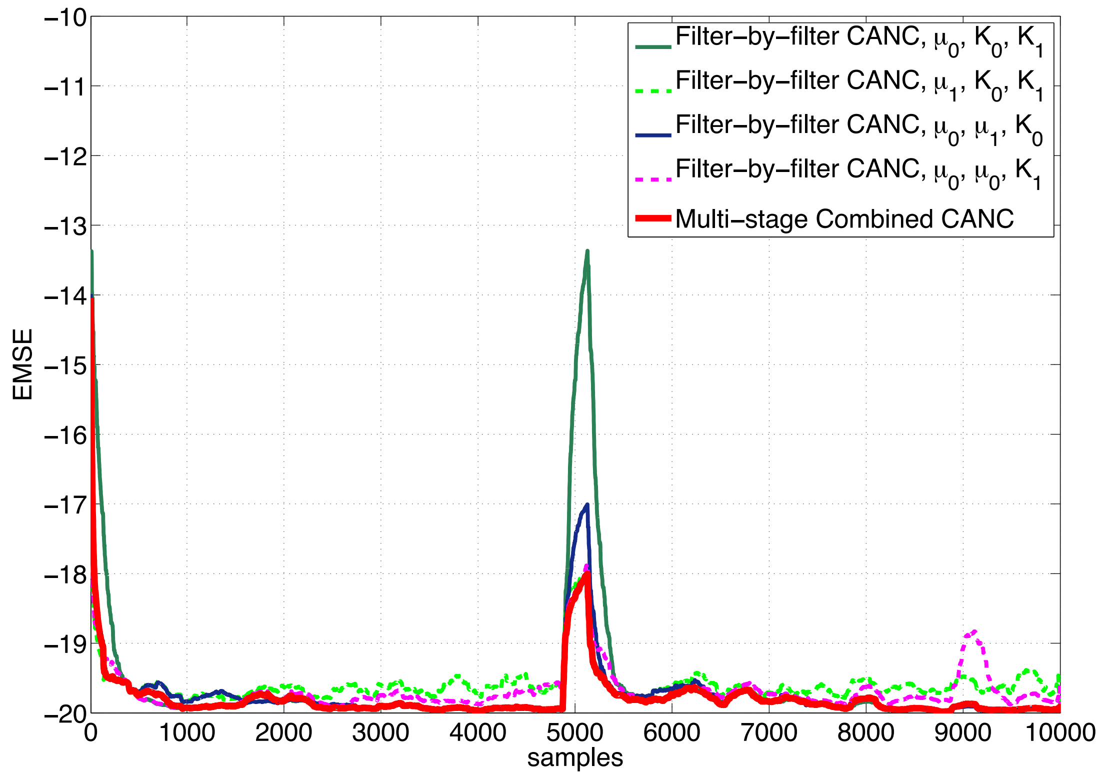


Fig. 11.7: *EMSE comparison between multi-stage combined filtering architectures and single-stage ones. Multi-stage combined architecture always provide the best overall performance.*

individual conventional filterings shows that the multi-stage filtering results the best performing architecture. Moreover, the performance improvement of the multi-stage architecture results even from the comparison with the single-stage filter-by-filter architectures, as depicted in Fig. 11.7.

Results achieved in this subsection the filtering ability of proposed combined schemes compared to conventional filtering. Moreover, a slightly preference is given to the filter-by-filter schemes which show a better reaction to abrupt changes in the environment due to the fact that the adaptive combination is performed for each channel. Furthermore, filter-by-filter schemes may exploit spatial diversity and thus different step size values for the adaptation of the mixing parameters may be chosen according to the scenario requirements. Finally, it has been shown that the multi-stage combined filtering always achieves the best convergence performance.

11.5.2 Speech enhancement evaluation of combined beamformers

In the second set of experiments we assess the effectiveness of the proposed combined beamforming architectures in terms of speech enhancement in multisource nonstationary environments. Experiments take place in a $6 \times 5 \times 3,3$ m room with a reverberation time of $T_{60} \approx 120$ ms. The source of interest is a female speaker located 70 cm from the center of the microphone array, as depicted in Fig. 11.8. Two interfering sources are initially located respectively $1,9$ m and $2,8$ m about from the center of the acoustic interface: the first source is a female speaker located on the left of the array, while on the right is located the second source which is a male speaker. White Gaussian noise is added at

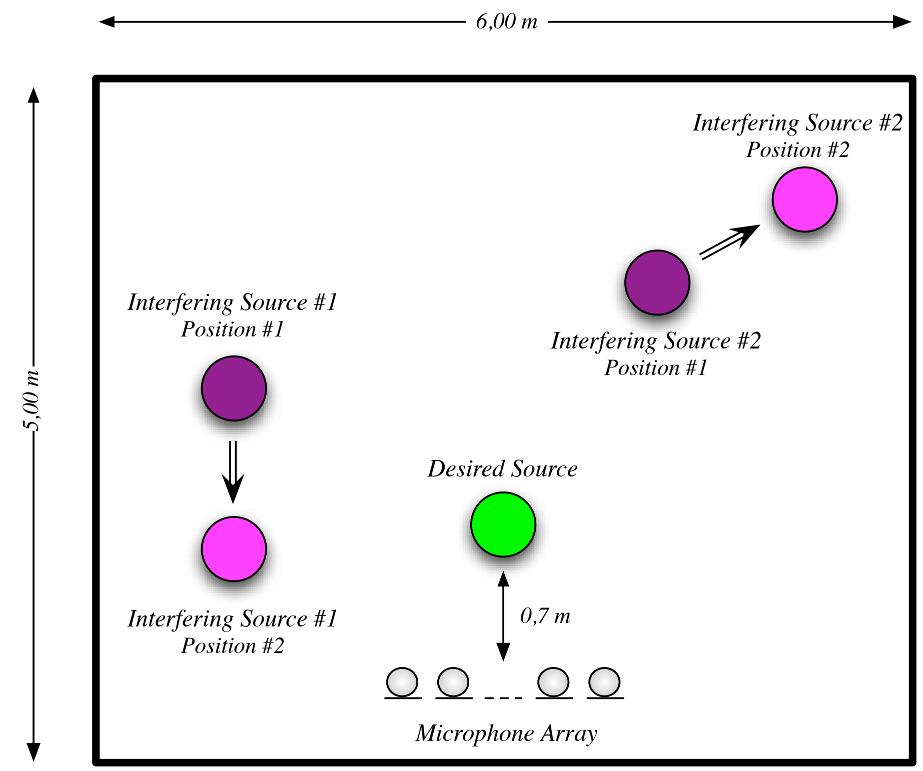


Fig. 11.8: *Speech enhancement nonstationary scenario. The source of interest is a female speaker located in front of the microphone array and two interfering speakers are located respectively on the left and on the right of the desired source. After 5 seconds from the start of the experiment the first interfering source moves to position 2 and at second 10 also the second interfering source changes its position.*

microphone signals as diffuse background noise, thus providing 20 dB of SNR (*signal-to-noise ratio*) with respect to the desired source. The overall input SNR level, measured for each microphone signal, is of about 3 dB. After 5 seconds from the start of the experiment the first source changes its position and at second 10 also the second source changes its position. The overall length of the experiment is of 15 seconds.

The AIRs between sources and microphones are simulated by means of *Roomsim*, which is a Matlab tool [24]. Each AIR is measured by using an 8 kHz sampling rate and it is truncated after $M = 340$ samples, which is also the length of each filter. The microphone interface is a classic *uniform linear array* (ULA) composed of 8 omnidirectional sensors equally spaced with a distance of 5 cm, thus having a good spatial resolution even at mid-low frequencies.

The enhancement of the speech, provided by the beamformer, and the resulting noise reduction, are usually associated with an SNR improvement, defined as [20]:

$$\mathrm{SNR} = 10\log\left[\frac{\mathrm{E}\left\{s_{in}^2\left[n\right]\right\}}{\mathrm{E}\left\{s_{in}^2\left[n\right]\right\} - \mathrm{E}\left\{s_{out}^2\left[n\right]\right\}}\right] \tag{11.8}$$

where $s_{in}\left[n\right]$ is the generic input clean signal and $s_{out}\left[n\right]$ is the processed signal. We compute the SNR level over the total length of the experiment $(0-15$ seconds) and also in 3 different sub-intervals of time: the initial state, from $0-5$ seconds, when the two interfering sources are located in their initial position; the first change, from $5-10$ seconds, which includes the position change of the first interfering source and the consequent readaptation of the filtering system; the second change, from $10-15$ seconds, when also the second interfering source changes its position. We compare GSC beamformers having different ANCs: conventional ANCs with with different parameter settings, the single-stage filter-by-filter combined ANC with different parameter settings, and the two-stage combined ANC. Filter parameters μ_0, μ_1, K_0, K_1, μ_s and μ_f are the same used in the first set of experiments. Results are collected in Table 11.1, in which it is possible to notice the behaviour of the different beamformers taken into account and their contribution to the noise reduction in terms of SNR improvement. We could have shown performance of both system-by-system and filter-by-filter combination schemes and both varying the step size values and the projection order. However, for a better ease of reading results, we only show the performance relative to filter-by-filter combination schemes, which achieve the more relevant results, and we only vary the step size value for the single-stage combined ANCs. From Table 11.1 can infer that all the conventional ANCs show difficulties when a source change its position, thus decreasing speech enhancement performance. The more stable conventional ANC is the one having a large step size value and a large projection order, however, its performance is the poorest in terms of SNR. A significant enhancement is achieved by means of the filter-by-filter combined ANC (FF CANC) and a further improvement is provided by the multi-stage filter-by-filter combined ANC (MFFC ANC) which achieves the best performance in each time interval in terms of SNR.

GSC	**0-5 s**	**5-10 s**	**10-15 s**	**0-15 s**
Conventional ANC, μ_0, K_0	17.2	14.2	14.9	15.6
Conventional ANC, μ_0, K_1	17.8	16.7	16.8	16.9
Conventional ANC, μ_1, K_0	18.1	16.3	16.5	16.8
Conventional ANC, μ_1, K_1	13.4	13.2	13.4	13.4
FF CANC, μ_0, K_0, K_1	18.4	17.0	18.1	18.0
FF CANC, μ_1, K_0, K_1	18.2	17.5	18.0	17.9
MFF CANC, μ_0, K_0, K_1	18.8	18.1	19.1	18.7

Table 11.1: *SNR comparison in dB. We evaluate the beamformers over three sub-intervals of time, 0-5, 5-10 and 10-15 seconds, and over the total length of the experiment, 0-15 seconds. Multi-stage combined beamformer always performs the best reduction of interfering signals.*

SNR values obtained from this experiment are not definitely the best achievable values, since better results may be obtained using more sophisticated GSC beamformers, i.e. involving any *voice activity detectors* (VADs) and post-filters. However, the obtained results are sufficient to show the effectiveness of the proposed combined beamformers compared to conventional methods. Further results can be found in [28].

11.6 CONCLUSIONS

In this chapter we have introduced novel beamforming methods whose goal is to improve the performance, in terms of speech enhancement, in presence of a multisource nonstationary environment . The trademark of proposed methods relies on the use of combined filtering schemes in the ANC block. These combined schemes are based on the adaptive combination of MISO systems with different parameter settings thus involving complementary capabilities. The whole beamforming system benefits from the different capabilities of each MISO systems, yielding improved performance. We introduced two different way of combining the MISO systems which are the system-by-system scheme and the filter-by-filter one. Both the combined architectures provides better performance compared to conventional beamformers, however filter-by-filter schemes are slightly preferable due to the fact that the adaptive combination is performed for each channel. This allows filter-by-filter beamformers to better react to abrupt changes in the environment and to exploit spatial diversity by choosing different step size values for the adaptation of the mixing parameter of each channel. Finally, a multi-stage combined beamformer has been introduced in which the adaptive combination of MISO systems can be performed in subsequent stages. In particular, we have taken into account a two-stage combined beamformer which outperforms the single-stage schemes, thus always providing the best performance when nonstationary sources interfere with the enhancement of a desired speech signal.

12

Collaborative Architectures for NAEC

Contents

Adaptive combination of filters may result very useful in setting the critical parameters of a filter during the adaptation, as shown in the previous chapter. However, the adaptive combination might result non-optimal when the goal is to exploit the capabilities of different models, or adaptive filters having different modelling tasks. In fact, in such situations in order to reach a desired performance the contribution of each filter might be necessary to reach a goal. This is the reason why affine and convex constraints might be not appropriate, since the sum of the mixing coefficients could be larger than one. In this chapter we use the adaptive combination of filters in a different way, in order to develop not combined but collaborative filtering architectures through the introduction of a virtual filter. We apply such collaborative architectures for the nonlinear acoustic echo cancellation. Experimental results show that proposed architectures show a more robust behaviour compared with other nonlinear echo cancellers aside from the nonlinearity level in the echo path[1].

12.1 A SERIUOS PROBLEM IN NAEC

In immersive speech communications, the necessity of using NAECs is increasingly pressing due to the growing spread of low-cost loudspeakers for commercial hands-free communication systems, that cause significant nonlinearities in the echo path and lead to communication quality degradation [18], [147]. However, when the echo path is roughly linear or contains negligible nonlinearities an NAEC could perform worse than a conventional AEC due to the gradient noise introduced by the nonlinear filter. Moreover, the ratio between linear and nonlinear echo signal power is unknown *a priori* and it is time-varying for nonstationary signals like speech. Thereby, it is not possible *a priori* to know if an NAEC will improve or deteriorate the cancellation. This trouble, along with the expensive computational cost of a NAEC, affects the strategies of many companies that provide teleconferencing services, which often choose to drop the use of nonlinear echo cancellers even at the expense of communication quality.

A possible solution to this problem is the use of collaborative filtering architectures. Collaborative filtering architectures are based on the convex combination of an adaptive filter with an *all-zero kernel* (AZK), i.e. a *virtual* kernel whose coefficients are set to zero and do not need adaptation [10]. Such convex combination is depicted in Fig. 12.1, where it is possible to see that, while the adaptive filter is updated according to its own error signal $e_1[n]$, the AZK is not adapted since it is a vector with null coefficients. As a consequence, the output signal of the AZK $y_2[n]$ is a null contribution. This scheme is nothing but an *intelligent switch circuit*. In fact, according to the cost function chosen for the adaptation of the mixing parameter $\lambda[n]$, the circuit can automatically activate or deactivate the adaptive filter. Such switching is performed by the convex combination: according to equation (10.5), when the mixing parameter $\lambda[n]$ is close to 1 the circuit output $y[n]$ will bear the adaptive filter contribution $y_1[n]$, while when $\lambda[n] \to 0$ the circuit selects the AZK output, thus resulting in a null output signal for the overall circuit.

[1] The work in this chapter has been partly performed while the author was a visiting Ph.D. student at the Department of "Teoría de la Señal y Comunicaciones", at "Universidad Carlos III de Madrid".

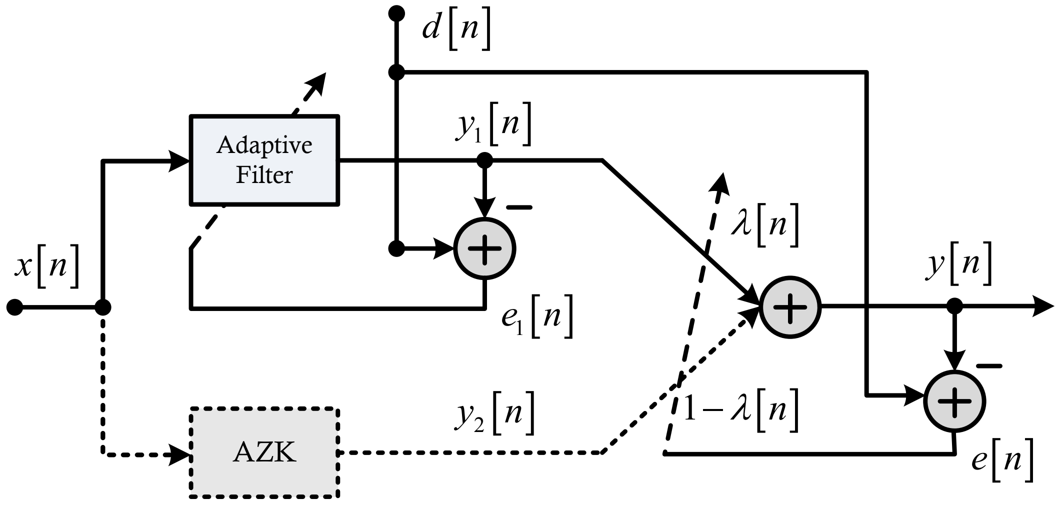


Fig. 12.1: *Intelligent switching circuit. The structure is composed of a convex combination between an adaptive filter and an all-zero kernel.*

Adaptive schemes using such intelligent switching circuit are introduced in [10] for NAEC employing Volterra filters and kernels, which are frequently employed as nonlinear solutions [138]. These collaborative schemes offer improved performance over the use of a single linear or nonlinear filter when the nonlinearity level is unknown or time-varying. However, the computational cost remains expensive due to the employment of Volterra kernels.

An effective collaborative architecture for NAEC is introduced in this chapter using the intelligent switching circuit in combination with a functional link adaptive filter (FLAF) (see Chapters 8 and 9). The resulting collaborative NAEC exploits the capabilities of FLAF-based NAECs introduced in Chapter 9 and, in addition, shows robustness against the variations of nonlinearity degree in an acoustic path.

12.2 COLLABORATIVE FLAF

Changes proposed in SFLAF (see Section 9.2), compared to the standard FLAF in Chapter 8, gives robustness to the flexibility of an NAEC based on functional links, due to the possibility to make the right choice for the critical parameters of the filter. However, some drawbacks may linger on when the nonlinearity degree varies in time. In particular, a non-optimal filtering may occur when the nonlinearity degree changes from a medium/high level to a very low one, such that the nonlinearity can be considered as irrelevant. It is well known [10, 30], indeed, that NAEC performance may result inferior than that of a conventional linear AEC when the desired signal is not affected by any nonlinearity, or when the nonlinearity degree is negligible. In that case, the nonlinear filter only brings some gradient noise in the filtering process, thus NAEC performance is subjected to a decrease. This is also the reason why conventional AEC devices are more commercially available than NAECs.

In order to design an NAEC robust to the changes of nonlinearity degree, we propose a collaborative architecture based on the convex combination of adaptive filters (see Section 10.2). Using the convex combination it is possible to exploit the capabilities of the individual filters, thus performing at least as well as the best contributing filter. Convex combination may result very useful in setting the critical parameters of a filter, as it is shown in [6, 126, 4]. However, convex combination might result non-optimal when the goal is to exploit the capabilities of different models, or adaptive filters having different modelling tasks, as in our case. As a matter of fact, the convex combination of a nonlinear FLAF with a linear filter might not fully exploit the linear filter capability to model the acoustic echo path when the desired signal is affected by any nonlinearity.

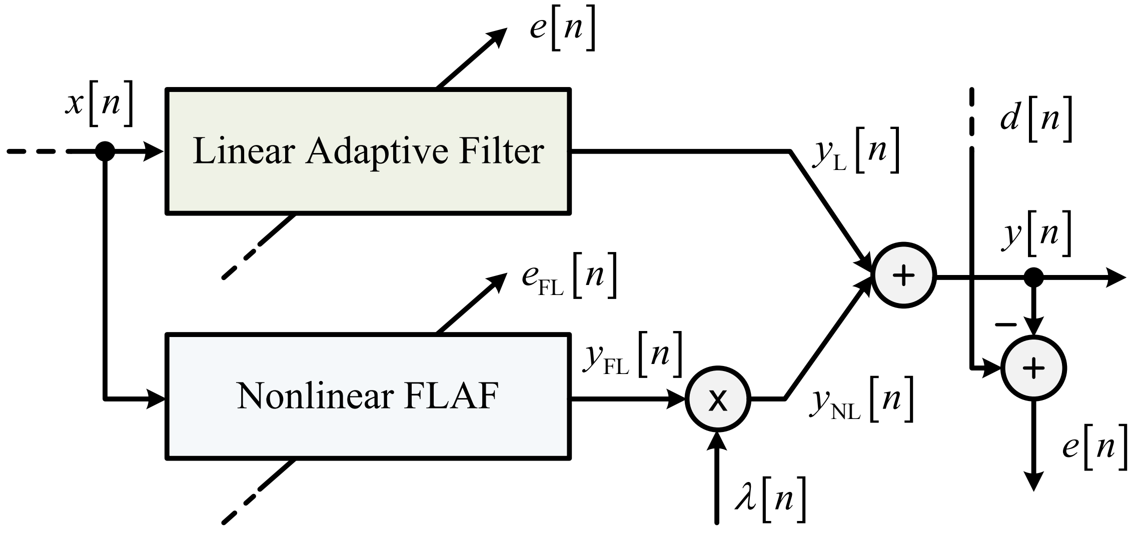


Fig. 12.2: *Collaborative functional link adaptive filter.*

Contrariwise, in designing an NAEC, it is desirable to enable the nonlinear modelling only if necessary. This is the reason why the proposed collaborative architecture exhibits a linear filtering always active and a nonlinear filtering which can be adaptively enabled and deactivated by means of an *intelligent switching circuit*, as depicted in Fig. 12.2. Such a collaborative architecture avoids the nonlinear contribution, and consequently the introduction of any gradient noise, when the echo path is almost linear, and the nonlinear FLAF is unnecessary.

The *collaborative FLAF*-based NAEC, that we denote as CFLAF, is depicted in Fig. 12.2, in which it is possible to notice that the overall output signal results as:

$$y[n] = y_{\mathrm{L}}[n] + \lambda[n]\, y_{\mathrm{FL}}[n] \tag{12.1}$$

where the *mixing parameter* $\lambda[n]$ allows to either keep or remove the output of the nonlinear FLAF as required by the filtering scenario. In equation (12.1) we omit the term weighted with $(1 - \lambda[n])$ and related to the AZK, as its contribution is null.

Due to the fact that linear and nonlinear filterings have different tasks, each filter is updated using different error signals in order completely to exploit the collaborative structure. In particular, the linear filter $\mathbf{w}_{\mathrm{L},n}$ pursues the minimization of the overall error signal $e[n] = d[n] - y[n]$, as the output contribution of the linear filter is always present. Differently, the nonlinear FLAF $\mathbf{w}_{\mathrm{FL},n}$ is updated using the local error $e_{\mathrm{FL}}[n]$ from which the linear output $y_{\mathrm{L}}[n]$ is subtracted, as it is always taken into account by the linear filtering:

$$e_{\mathrm{FL}}[n] = d[n] - \left(y_{\mathrm{L}}[n] + y_{\mathrm{FL}}[n]\right). \tag{12.2}$$

The mixing parameter $\lambda[n]$ can be adapted in a convex way assuming that $0 \leq \lambda[n] \leq 1$ through the adaptation of an auxiliary parameter, $a[n]$, related to $\lambda[n]$ by means of a sigmoidal function defined as (10.8). Therefore, $\lambda[n]$ is computed adapting $a[n]$ through a gradient descent rule as $a[n+1] = a[n] + \Delta a[n]$, where $\Delta a[n]$ results from a *normalized least mean squares* (NLMS) adaptation (see Paragraph 10.3.2):

$$\begin{aligned}\Delta a[n] &= -\frac{1}{2}\mu_a \frac{\partial e^2[n]}{\partial a[n]} \\ &= -\frac{\mu_a}{r[n]} e[n] \frac{\partial\left(d[n] - y_{\mathrm{L}}[n] - \lambda[n] y_{\mathrm{FL}}[n]\right)}{\partial \lambda[n]} \frac{\partial \lambda[n]}{\partial a[n]} \\ &= \frac{\mu_a}{r[n]} e[n] y_{\mathrm{FL}}[n] \lambda[n] \left(1 - \lambda[n]\right)\end{aligned} \tag{12.3}$$

where

$$r[n] = \beta r[n-1] + (1-\beta) y_{\mathrm{FL}}^2[n] \tag{12.4}$$

is a rough low-pass filtered estimate of the power of the signal of interest [9]. The parameter β is a smoothing factor which ensures that $r[n]$ is adapted

faster than any filter component. The value of $a[n]$ is kept within $[4, -4]$ for practical reasons [6] (see Section 10.3).

The proposed CFLAF architecture is robust against any nonlinearity level, since when the echo path is merely linear $\lambda[n]$ converges towards 0 and the whole scheme behaves like a purely linear filter, thus avoiding any gradient noise from the nonlinear FLAF. On the other hand, when the echo path conveys nonlinearities the mixing parameter approaches 1 according to the nonlinearity level in the echo path. Note that when $\lambda[n] = 1$ the CFLAF architecture performs like the SFLAF.

12.3 BLOCK-BASED COLLABORATIVE FLAF

A further weak spot of an FLAF-based NAEC may be a failed control over the expanded buffer. In fact, a control in that sense can be useful when the nonlinearity degree is unknown. In the previous subsection, we saw how a CFLAF is able to be robust when the nonlinearity degree varies from a negligible value to a detectable one and *vice-versa*. However, significant differences may occur when the nonlinearity degree varies between detectable levels with different intensity. As a matter of fact, a high expansion order may be necessary in order to model a high nonlinearity degree, so that the length of the expanded buffer is sufficiently large to ensure a high number of nonlinear elements. On the other hand, in case of detectable nonlinearity with a low/medium-intensity a large number of coefficients may cause an overfitting plight and, therefore, introduce some gradient noise, thus degrading filtering performance.

In order to overcome this drawback, we propose an improved CFLAF architecture featuring a block-based convex combination [4], that we name as *block-based collaborative FLAF* (BCFLAF). As we saw in the previous section, the adaptive combination in CFLAF allows to adaptively deactivate the whole nonlinear filtering whether not necessary. Similarly, the main idea which underpins BCFLAF approach is that of dividing the expanded buffer into blocks and adapting each block with its own mixing parameter, so that it is

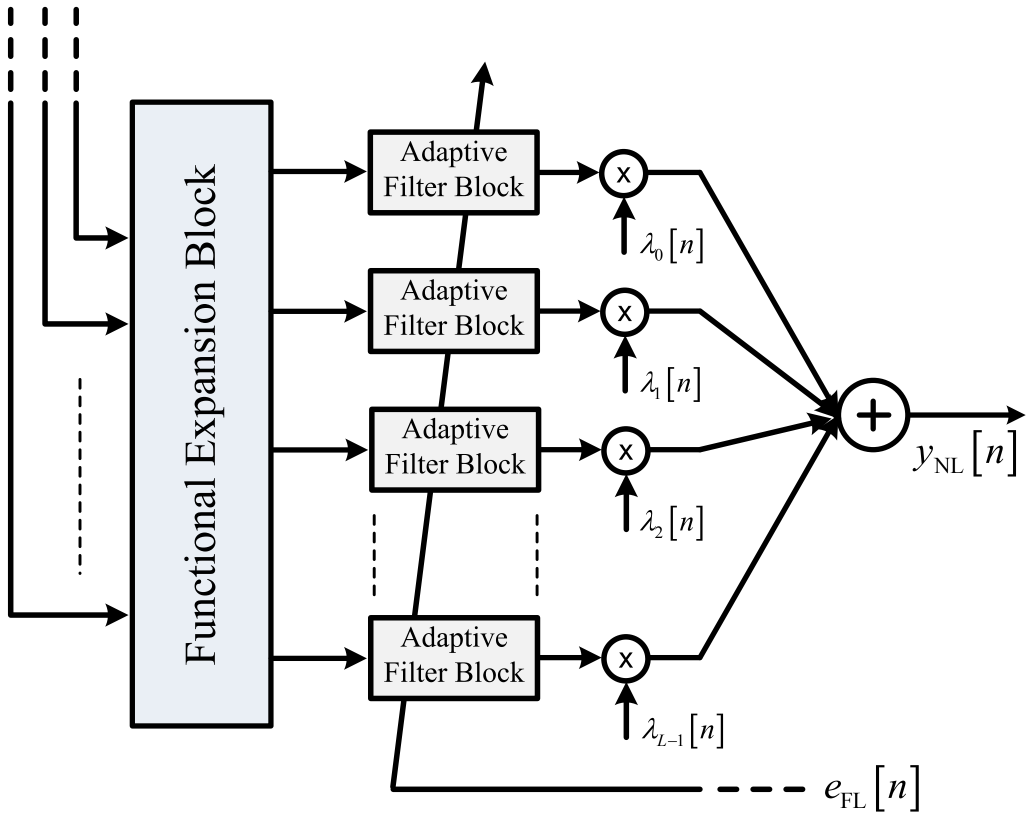


Fig. 12.3: *Nonlinear adaptive path in a block-based collaborative FLAF.*

possible to adaptively deactivate those blocks which are not useful to model nonlinearities. Due to the fact that the nonlinear filtering strictly depends on the length of the expanded buffer, and therefore on the number of nonlinear elements, it is possible to divide the expanded buffer in blocks just according to the desired accuracy. A sufficiently large number [6] of blocks may result in a high accuracy but also in an increase of the computational cost. Therefore, the block-based combination actually reduces the number of nonlinear elements selected for the nonlinear filtering and therefore avoids the introduction of any gradient noise.

The convex combination introduced in CFLAF, and described by equation (12.1), adopts the same mixing parameter for all weights. On the other

hand, considering a number of L blocks, each one consisting of $M_b = M_e/L$ coefficients, it is possible to express the output of the BCFLAF as:

$$y[n] = y_{\rm L}[n] + \sum_{l=0}^{L-1} \sum_{k\in\mathcal{I}_l} \lambda_l[n]\, g_k[n]\, w_{{\rm FL},k}[n-1] \tag{12.5}$$

where $\lambda_l[n]$ is the mixing parameter related to the l-th block, $w_{{\rm FL},k}[n-1]$ refers to the m-th coefficients of each block, and $\mathcal{I}_l = [l \cdot M_b, \ldots, (l+1)\, M_b - 1]$ is the range of indices related to the coefficients of the l-th block.

The block-based combination also affects the adaptation of the nonlinear filter $\mathbf{w}_{{\rm FL},n}$, which becomes:

$$\mathbf{w}_{{\rm FL},n} = \mathbf{w}_{{\rm FL},n-1} + \mu_{\rm FL} \frac{e_{\rm FL}[n] \sum_{l=0}^{L-1} \sum_{k\in\mathcal{I}_l} \lambda_l[n]\, g_k[n]}{\delta_{\rm FL} + \sum_{l=0}^{L-1} \sum_{k\in\mathcal{I}_l} |\lambda_l[n]\, g_k[n]|^2} \tag{12.6}$$

where $\mu_{\rm FL}$ and $\delta_{\rm FL}$ are respectively the step size and the regularization parameter for the all the blocks of the nonlinear filter.

The L mixing parameters can be adapted similarly to the equation (12.3) of the CFLAF case. Therefore, defining $\lambda_l[n] = \mathrm{sgm}(a_l[n])$, with $l = 0, \ldots, L-1$, the updating rule for each auxiliary parameter is given by:

$$\begin{aligned} a_l[n+1] = & a_l[n] + \frac{\mu_a}{r[n]} e[n]\, \lambda_l[n]\, (1 - \lambda_l[n]) \\ & \times \sum_{l=0}^{L-1} \sum_{k\in\mathcal{I}_l} g_k[n]\, w_{{\rm FL},k}[n]\,. \end{aligned} \tag{12.7}$$

A graphical representation of the nonlinear filtering carried out by BCFLAF architecture is depicted in Fig. 12.3.

12.4 EXPERIMENTAL RESULTS

In this section we evaluate the performance of the proposed CFLAF in an acoustic echo cancellation scenario. We use the same experimental setup of Section 9.3 and also the same input signals having a length of 10 seconds. However, if the acoustic channel is nonlinear and the degree of nonlinearity remains constant, an NAEC using the CFLAF yields the same performance of the SFLAF, according to what said in Section 12.2. Therefore, in order to show the advantages of the convex combination, we consider a change of the nonlinearity level in the echo path. In fact, we start the process in linear conditions, i.e. the nonlinearities in the AIR are neglegible so that the acoustic path can be assumed as linear. After 5 seconds from the start of the process we introduce a clipping nonlinearity, the same as in Section 9.3.

In these scenario conditions, we compare the performance of three acoustic echo canceller in terms of ERLE: a conventional linear AEC, an NAEC based on the SFLAF and an NAEC based on the CFLAF. In a first experiment we consider the white Gaussian input and we use an NLMS algorithm to update the filters for all the three echo cancellers. The result is depicted in Fig. 12.4 in which it is possible to see that in the first half of the process, the best performing filter is the conventional NLMS, due to the fact that the AIR is purely linear. In this case the SFLAF shows a worse behaviour due to the gradient noise introduced by the nonlinear elements of the filter. However, it is possible to notice that, for the first 5 seconds, the CFLAF displays the same behaviour of the NLMS, and this is due to the fact that the intelligent switching circuit inside the CFLAF detects the absence of nonlinearities and selects the output contribution of the AZK; in this way the whole CFLAF reduces to be a linear filter.

However, in the second half of the process the nature of the AIR turns to be nonlinear and an immediate consequence is the performance decrease of the NLMS in Fig. 12.4. On the other side both the SFLAF and the CFLAF exploit the capabilities of the functional link based filtering and display better performance than the linear AEC. However, due to the different initial conditions (at second 5) the CFLAF performs better than the SFLAF. Therefore, it is possible to state that, comparing to the NLMS and the SFLAF, the CFLAF is always the best performing acoustic echo canceller notwithstanding the nonlinearity degree in the echo path.

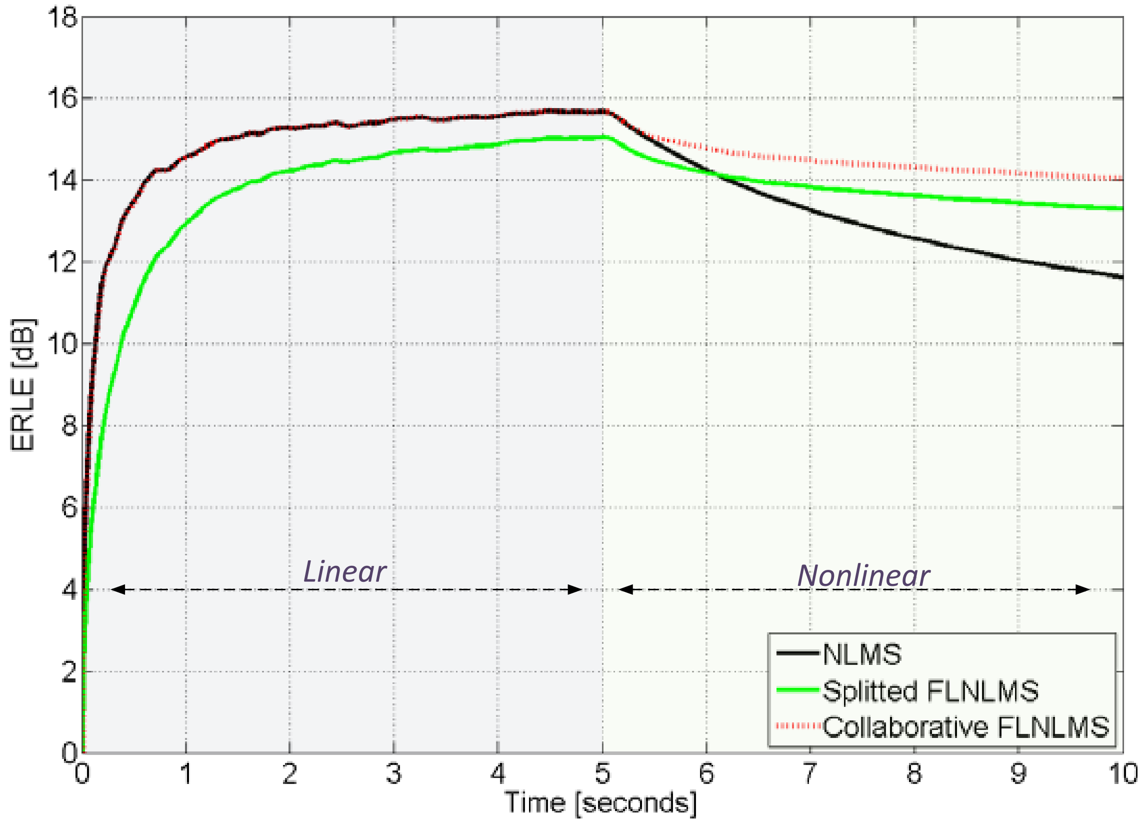


Fig. 12.4: *Performance comparison in terms of ERLE between a linear, an SFLAF-based and a CFLAF-based echo cancellers in case of white Gaussian input. All the filters are updated using an NLMS algorithm.*

Same conclusions, even if with less evident differences, result from a second experiment using the female speech signal as input, as it is possible to see from Fig. 12.5. In this second experiment all the filters are updated using an APA with a projection order of $K = 3$.

Let us note that in this case it is difficult to comprehend the real benefits deriving from the collaborative architectures due to the fact that the ERLE does

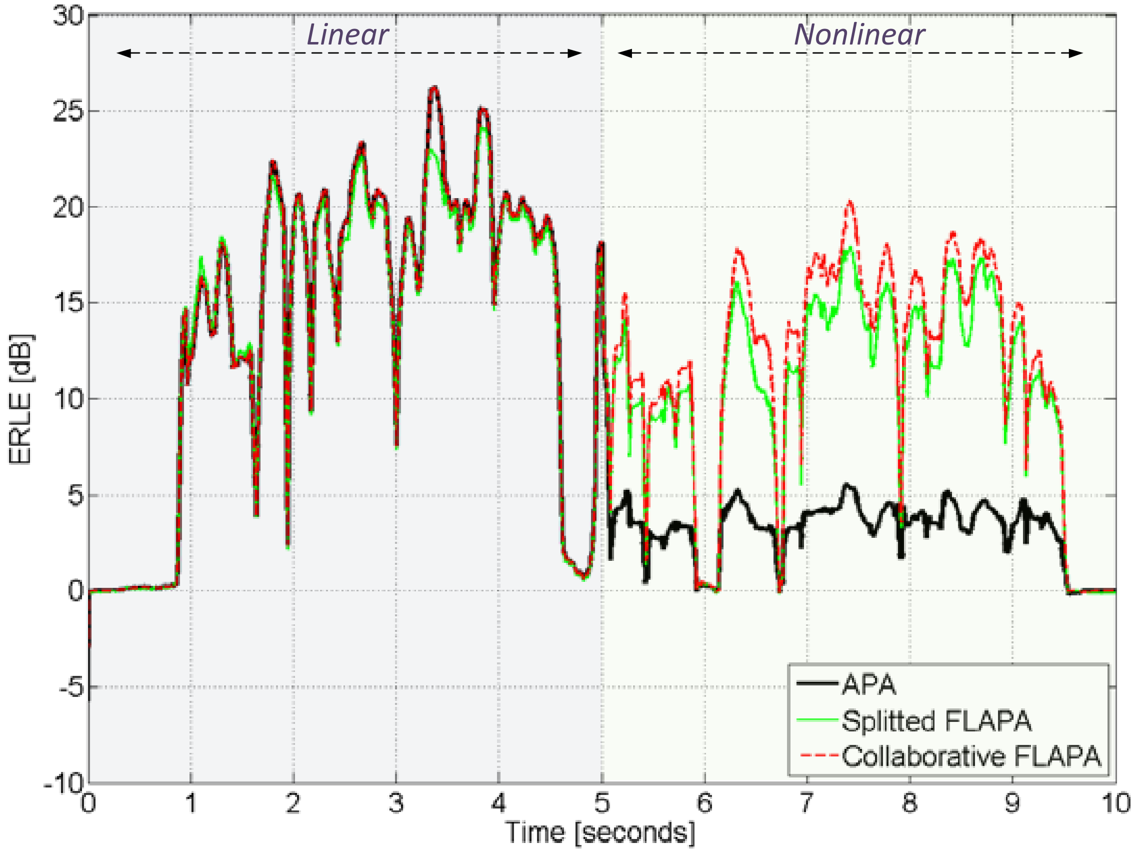


Fig. 12.5: *Performance comparison in terms of ERLE between a linear, an SFLAF-based and a CFLAF-based echo cancellers in case of female speech input. All the filters are updated using an APA.*

not reflect the perceived quality improvement of the speech, which is more evident than the ERLE improvement. In the linear case this lack is plugged by the *normalized misalignment* (see Section 3.4), however in the nonlinear case it is not possible to dispose of a similar performance measure, and it is often difficult to achieve a complete evaluation of an NAEC only using the ERLE, even if it is the most used measure in literature for the evaluation of a nonlinear echo canceller.

12.5 CONCLUSIONS

In this chapter we have introduced robust acoustic echo cancellers based on the adaptive combination of filters. In particular, we exploits the capabilities of the convex combination to develop an intelligent switching circuit which allows the combination of adaptive filters from different models. In this case, we have combined a linear adaptive filter and a nonlinear adaptive filter, thus obtaining collaborative filtering architecture that can be used for nonlinear echo cancellation. Such collaborative architectures have shown a more robust behaviour compared with other nonlinear echo cancellers notwithstanding the nonlinearity level in the echo path. This result paves the way for the development of more sophisticated architectures able to solve similar problems both for acoustic applications and also for other kinds of application.

PART V

CONCLUSIONS

—There are two possible outcomes:
if the result confirms the hypothesis, then you've made a measurement.
If the result is contrary to the hypothesis, then you've made a discovery.
Enrico Fermi

13

Conclusions and Outlooks

The main motivations which underlie this dissertation work spring from the new directions towards which speech telecommunications are going to. *Immersive speech communications* are becoming a reality and soon enough will become part of our daily life. However, immersive communications entail the use of displaced microphones and moreover take place in multisource environments where interfering signals may degrade quality and intelligibility of the desired speech source. Therefore, acquisition of desired signals with high quality is far more difficult and challenging for immersive communications than in the classical telephony environment where the microphone is close to the user.

Thereof the necessity to develop *intelligent acoustic interfaces* is increasingly pressing. An intelligent acoustic interface aims at extracting, from audio signals, desired informations for an acoustic environment, and, at the same time, has to reproduce remote desired acoustic information taking into account the perceptive requirements of a speech communication. To this end an intelligent acoustic interface has to model the acoustic channel, and the more "intelligent" way to do that, looking on the user requirements, is to employ adaptive filtering algorithms.

In this dissertation work we have investigated adaptive algorithms expressly designed for intelligent acoustic interfaces. For this purpose, the work is structured in three main parts.

In the first part we dealt with linear adaptive algorithms in order to tackle acoustic limitations deriving from the modelling of the acoustic impulse response, the presence of nonstationary sources, the presence of interfering phenomena, such as the "double talk". In this part, starting from the study of new class of adaptive algorithms, such as the *proportionate algorithms*, we have formulated an alternative framework for the derivation of both classic stochastic algorithms and proportionate ones. Moreover, we proposed efficient proportionate algorithms based on the *affine projection* and on the *variable step size*, able to model an acoustic path even in adverse environment conditions.

In the second part we took into account nonlinear limitation, caused by the introduction of loudspeaker distortions in the acoustic path. This is a quite tricky problem, since nonlinearities strongly decrease the quality of a speech communication and due to the fact that commercial nonlinear filtering algorithms are not able yet to satisfy the quality requirements of a speech communication. In order to address this problem we proposed a novel nonlinear filtering model, called *functional link adaptive filter*, that we have used to develop *ad hoc* nonlinear adaptive algorithms for the modelling of nonlinear acoustic paths.

In the last part of this thesis, exploiting the adaptive algorithms proposed in the previous two parts, we developed more sophisticated adaptive filtering architectures which are more robust against adverse conditions of real scenarios. Such architectures have been developed exploiting the capabilities of *adaptive combinations of filters*. The main motivation, which underlies this study, is based on a common problem in the modelling of a nonlinear acoustic path. In fact, in this case, a kind of nonlinearity highly varying, in amplitude or in time, may require to change the filter design during the adaptation. Moreover, another important troubling situation occurs when the desired signal is not known *a priori*, thus it is difficult to choose whether adopting a linear filter or a nonlinear model. This trouble, along with the expensive computational cost of commercial nonlinear adaptive filters, affects the strategies of many companies that provide teleconferencing services, which often choose to employ only linear filters even at the expense of communication quality. In order to tackle this problem we proposed *collaborative filtering architectures* which are able to model an acoustic impulse response apart from its nature, whether it is linear or nonlinear.

The results achieved in this work pave the way for future research. A main relevance could be reserved to the modelling of nonlinear acoustic channel. In fact, the introduction of a new nonlinear model leads to novel interesting scenarios that can be deepened.

First of all, it could be possible to work on FLAF model in order to reduce the drawbacks making it a more consistent model. Moreover, it could be possible to exploits the capabilities of adaptive algorithms to develop more robust nonlinear adaptive filters. For example it is thinkable to apply the sparsity constraints to the modelling of the nonlinearities. This could lead to a further performance improvement.

Another important point is the fact that all the filtering techniques introduced in this work can be extended in the multichannel domain, due to the fact that immersive speech communications are based on the use of MIMO systems.

Moreover, as we have seen, a weak point of such techniques is their not appropriate evaluation. Immersive communications are based on perceived quality of a speech signal, thus the use of performance measures that includes also a perceptive evaluation of the filtering could be more proper.

Furthermore, being these proposed techniques very flexible, their use is not limited only to acoustic application, thus it could be possible to exploit their capabilities to develop *ad hoc* adaptive filtering algorithms.

References

[1] F. Albu and H. K. Kwan, "A new block exact affine projection algorithm," in *Proc. of the IEEE International Symposium on Circuits and Systems (ISCAS '05)*, vol. 5, Kobe, Japan, May 23-26 2005, pp. 4337–4340.

[2] S. Amari, "Natural gradient works efficiently in learning," *Neural Computation*, vol. 10, pp. 251–276, Feb. 1998.

[3] S. Amari and S. C. Douglas, "Why natural gradient?" in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '98)*, vol. 2, Seattle, WA, May 12-15 1998, pp. 1213–1216.

[4] J. Arenas-García and A. R. Figueiras-Vidal, "Adaptive combination of proportionate filters for sparse echo cancellation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 6, pp. 1087–1098, Aug. 2009.

[5] J. Arenas-García, A. R. Figueiras-Vidal, and A. H. Sayed, "Steady state performance of convex combinations of adaptive filters," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '05)*, vol. 4, Philadelphia, PA, Mar. 18-23 2005, pp. 33–36.

[6] ——, "Mean-square performance of a convex combination of two adaptive filters," *IEEE Transactions on Signal Processing*, vol. 54, no. 3, pp. 1078–1090, 2006.

[7] J. Arenas-García, V. Gómez-Verdejo, and A. R. Figueiras-Vidal, "New algorithms for improved adaptive convex combination of LMS transversal filters," *IEEE Transactions on Instrumentation and Measurement*, vol. 54, no. 6, pp. 2239–2249, Dec. 2005.

[8] J. Arenas-García, V. Gómez-Verdejo, M. Martínez-Ramón, and A. R. Figueiras-Vidal, “Separate-variable adaptive combination of LMS adaptive filters for plant identification,” in *Proc. of the IEEE International Workshop on Neural Networks for Signal Processing (NNSP ’03)*, Toulouse, France, Sep. 17-19 2003, pp. 239–248.

[9] L. A. Azpicueta-Ruiz, A. R. Figueiras-Vidal, and J. Arenas-García, “A normalized adaptation scheme for the convex combination of two adaptive filters,” in *Proc. of the IEEE 13th International Conference on Acoustics, Speech and Signal Processing (ICASSP ’08)*, Las Vegas, NV, Mar. 30 - Apr. 4 2008, pp. 3301–3304.

[10] L. A. Azpicueta-Ruiz, M. Zeller, A. R. Figueiras-Vidal, J. Arenas-García, and W. Kellermann, “Adaptive combination of Volterra kernels and its application to nonlinear acoustic echo cancellation,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 1, pp. 97–110, Jan. 2011.

[11] J. Benesty and P. Duhamel, “A fast exact least mean square adaptive algorithm,” *IEEE Transactions on Signal Processing*, vol. 40, no. 12, pp. 2904–2920, Dec. 1992.

[12] J. Benesty, T. Gänsler, D. R. Morgan, M. M. Sondhi, and S. L. Gay, *Advances in Network and Acoustic Echo Cancellation*. Berlin, Heidelberg, New York: Springer-Verlag, 2001.

[13] J. Benesty and S. L. Gay, “An improved PNLMS algorithm,” in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP ’02)*, vol. 2, Orlando, FL, May 13-17 2002, pp. 1881–1884.

[14] J. Benesty and Y. Huang, “The LMS, PNLMS, and exponentiated gradient algorithms,” in *Proc. of the European Signal Processing Conference (EUSIPCO ’04)*, Vienna, Austria, Sep. 6-10 2004, pp. 721–724.

[15] E. Benetos and S. Dixon, “Joint multi-pitch detection using harmonic envelope estimation for polyphonic music transcription,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 6, pp. 1111–1123, Oct. 2011.

[16] D. A. Berkley and O. M. M. Mitchell, “Seeking the ideal in “hands-free” telephony,” *Bell Labs Record*, vol. 52, no. 10, pp. 318–325, Nov. 1974.

[17] N. J. Bershad, J. C. M. Bermudez, and J. Tourneret, “An affine combination of two LMS adaptive filters - Transient mean-square analysis,” *IEEE Transactions on Signal Processing*, vol. 56, no. 5, pp. 853–1864, May 2008.

[18] N. Birkett and R. A. Goubran, “Limitations of handsfree acoustic echo cancellers due to nonlinear loudspeaker distortion and enclosure vibration effects,” in *Proc. of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA ’95)*, New Paltz, NY, Oct. 15-18 1995, pp. 103–106.

[19] N. A. Birkett and R. A. Goubran, “Acoustic echo cancellation using NLMS-neural network structures,” in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP ’95)*, vol. 5, Detroit, MI, May 9-12 1995, pp. 3035–3038.

[20] M. Brandstein and D. Ward, Eds., *Microphone Arrays: Signal Processing Techniques and Applications*. New York, NY: Springer, 2001.

[21] R. Brooks, “Elephants don’t play chess,” *Robotics and Autonomous Systems*, vol. 6, no. 1-2, pp. 3–15, Jun. 1990.

[22] D. S. Broomhead and D. H. Lowe, “Multivariable functional interpolation and adaptive networks,” *Complex Systems*, vol. 2, pp. 321–355, 1988.

[23] T. G. Burton, R. A. Goubran, and F. Beaucoup, “Nonlinear system identification using a subband adaptive Volterra filter,” *IEEE Transactions on Instrumentation and Measurement*, vol. 58, no. 5, pp. 1389–1397, May 2009.

[24] D. R. Campbell, K. J. Palomaki, and G. J. Brown, “Roomsim, a MATLAB simulation of “shoebox” room acoustics for use in teaching and research,” *Computing and Information Systems*, vol. 9, no. 3, pp. 48–51, 2005.

[25] J. A. Chambers, O. Tanrikulu, and A. Constantinides, “Least mean mixed-norm adaptive filtering,” *Electronics Letters*, vol. 30, no. 19, pp. 1574–1575, 1994.

[26] A. Cichocki and R. Unbehauen, *Neural Networks for Optimisation and Signal Processing*. Chichester, UK: John Wiley & Sons, Ltd., 1993.

[27] D. Comminiello, L. A. Azpicueta-Ruiz, M. Scarpiniti, A. Uncini, and J. Arenas-Garcia, “Functional link based architectures for nonlinear acoustic echo cancellation,” in *Proc. of the IEEE Joint Workshop on Hands-free Speech Communication and Microphone Arrays (HSCMA ’11)*, Edinburgh, UK, May 30 - Jun. 1 2011, pp. 180–184.

[28] D. Comminiello, M. Scarpiniti, R. Parisi, A. Cirillo, M. Falcone, and A. Uncini, “Multi-stage collaborative microphone array beamforming in presence of nonstationary interfering signals,” in *Proc. of the International Workshop on Machine Listening in Multisource Environments (CHiME ’11)*, Florence, Italy, Sep. 1 2011, pp. 64–67.

[29] D. Comminiello, M. Scarpiniti, R. Parisi, and A. Uncini, “A functional link based nonlinear echo canceller exploiting sparsity,” in *Proc. of the International Workshop on Acoustic Echo and Noise Control (IWAENC ’10)*, Tel Aviv, Israel, Aug. 30- Sep. 2 2010.

[30] ——, “A novel affine projection algorithm for superdirective microphone array beamforming,” in *Proc. of the IEEE International Symposium on Circuits and Systems (ISCAS ’10)*, Paris, France, May 30 - Jun. 2 2010, pp. 2127–2130.

[31] J. R. Cooperstock, “Multimodal telepresence systems,” *IEEE Signal Processing Magazine*, vol. 28, no. 1, pp. 77–86, Jan. 2011.

[32] J.-P. Costa, A. Lagrange, and A. Arliaud, “Acoustic echo cancellation using nonlinear cascade filters,” in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP ’03)*, vol. 5, Hong Kong, Apr. 6-10 2003, pp. 389–392.

[33] M. Cristani, M. Bicego, and V. Murino, “Audio-visual event recognition in surveillance video sequences,” *IEEE Transactions on Multimedia*, vol. 9, no. 2, pp. 257–267, Feb. 2007.

[34] G. Cybenko, “Approximation by superpositions of a sigmoidal function,” *Mathematics of Control, Signals, and Systems (MCSS)*, vol. 2, no. 4, pp. 303–314, Dec. 1989.

[35] S. Dehuri and S. B. Cho, “A comprehensive survey on functional link neural networks and an adaptive PSO-BP learning for CFLNN,” *Neural Computing & Applications*, vol. 19, no. 2, pp. 187–205, 2010.

[36] P. S. R. Diniz, *Adaptive Filtering: Algorithms and Practical Implementations*, 3rd ed. New York, NY: Springer, 2008.

[37] S. C. Douglas and S. Amari, “Natural gradient adaptation,” in *Unsupervised Adaptive Filtering, Vol. 1: Blind source Separation*, S. Haykin, Ed. New York, NY: John Wiley & Sons, Ltd., 2000, ch. 2, pp. 13–61.

[38] H. L. Dreyfus, *What Computers Still Can’t Do - A Critique of Artificial Reason*. Cambridge, MA: MIT Press, 1992.

[39] D. L. Duttweiler, “A twelve-channel digital echo canceler,” *IEEE Transactions on Communications*, vol. 26, no. 5, pp. 647–653, May 1978.

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

[41] E. Eweda, “Comparison of RLS, LMS and sign algorithms for tracking randomly time-varying channels,” *IEEE Transactions on Signal Processing*, vol. 42, no. 11, pp. 2937–2944, Nov. 1994.

[42] B. Farhang-Boroujeny, *Adaptive Filters Theory and Applications*. Chichester, UK: John Wiley /& Sons, 1999.

[43] A. Fermo, A. Carini, and G. L. Sicuranza, “Analysis of different low complexity nonlinear filters for acoustic echo cancellation,” in *Proc. of the 1st International Workshop on Image and Signal Processing and Analysis (IWISPA ’00)*, Pula, Croatia, Jun. 14-15 2000, pp. 261–266.

[44] ——, “Simplified Volterra filters for acoustic echo cancellation in GSM receivers,” in *Proc. of the European Signal Processing Conference (EUSIPCO ’00)*, Tampere, Finland, Sep. 4-8 2000.

[45] J. L. Flanagan, R. Johnson, J. D. andZahn, and G. W. Elko, “Computer-steered microphone arrays for sound transduction in large rooms,” *Journal of Acoustical Society of America*, vol. 78, no. 5, pp. 1508–1518, Nov. 1985.

[46] J. Fu and W.-P. Zhu, “A nonlinear acoustic echo canceller using sigmoid transform in conjunction with RLS algorithm,” *IEEE Transactions on Circuits and Systems II, Express Briefs*, vol. 55, no. 10, pp. 1056–1060, Oct. 2008.

[47] ——, “A simplified structure of second-order Volterra filters for nonlinear acoustic echo cancellation,” in *Proc. of the IEEE International Symposium on Circuits and Systems (ISCAS ’10)*, Paris, France, May 30 - Jun. 2 2010, pp. 2366–2369.

[48] T. Gänsler, J. Benesty, S. L. Gay, and M. M. Sondhi, “A robust proportionate affine projection algorithm for network echo cancellation,” in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP ’00)*, vol. 2, Istanbul, Turkey, Jun. 5-9 2000, pp. 796–796.

[49] F. X. Y. Gao and W. M. Snelgrove, “Adaptive linearization of a loudspeaker,” in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP ’91)*, vol. 5, Toronto, Canada, Apr. 14-17 1991, pp. 3589–3592.

[50] S. L. Gay, “An efficient, fast converging adaptive filter for network echo cancellation,” in *Proc. of the IEEE 3rd Asilomar Conference on Signals, Systems & Computers (ACSSC ’98)*, vol. 1, Pacific Grove, CA, Nov. 1-4 1998, pp. 394–398.

[51] S. L. Gay and S. C. Douglas, “Normalized natural gradient adaptive filtering for sparse and non-sparse systems,” in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP ’02)*, vol. 2, Orlando, FL, May 13-17 2002, pp. 1405–1408.

[52] S. L. Gay and S. Tavathia, “The fast affine projection algorithm,” in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP ’95)*, vol. 5, Detroit, MI, May 9-12 1995, pp. 3023–3026.

[53] G. H. Golub and C. F. Van Loan, *Matrix Computation*. Baltimore, MD and London: John Hopkins University Press, 1989.

[54] L. Griffiths and C. Jim, “An alternative approach to linearly constrained adaptive beamforming,” *IEEE Transactions on Antennas and Propagation*, vol. 30, no. 1, pp. 27–34, Jan. 1982.

[55] S. Guarnieri, F. Piazza, and A. Uncini, “Multilayer feedfoward networks with adaptive spline activation function,” *IEEE Transactions on Neural Networks*, vol. 10, no. 3, pp. 672–683, May 1999.

[56] A. Guerin, G. Faucon, and R. Le Bouquin-Jeannes, “Nonlinear acoustic echo cancellation based on Volterra filters,” *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 6, pp. 672–683, Nov. 2003.

[57] E. Hänsler and G. Schmidt, *Acoustic Echo and Noise Control. A Practical Approach*. Hoboken, NJ: John Wiley & Sons, Inc., 2004.

[58] R. W. Harris, D. M. Chabries, and F. A. Bishop, “A variable step (VS) adaptive filter algorithm,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 34, no. 2, pp. 309–316, 1986.

[59] S. Haykin, *Adaptive Filter Theory*, 4th ed. Upper Saddle River, NJ: Prentice Hall, Sep. 2001.

[60] ——, *Neural Networks and Learning Machines*, 3rd ed. Upper Saddle River, NJ: Prentice Hall, Nov. 2008.

[61] W. E. Hefley and D. Murray, “Intelligent user interfaces,” in *Proc. of the 1st International Conference on Intelligent User interfaces (IUI ’93)*. Orlando, FL: ACM, Jan. 4-7 1993, pp. 3–10.

[62] T. T. Hewett, R. Baecker, S. Card, T. Carey, J. Gasen, M. Mantei, G. Perlman, S. G., and W. Verplank, *ACM SIGCHI Curricula for Human-Computer Interaction*, B. Hefley, Ed. New York, NY: The Association for Computing Machinery, Inc., 1992.

[63] D. Hongyun and W.-P. Zhu, “Compensation of loudspeaker nonlinearity in acoustic echo cancellation using raised-cosine function,” *IEEE Transactions on Circuits and Systems II, Express Briefs*, vol. 53, no. 11, pp. 1190–1194, Nov. 2006.

[64] O. Hoshuyama, R. A. Goubran, and A. Sugiyama, “A generalized proportionate variable step-size algorithm for fast changing acoustic environments,” in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP ’04)*, vol. 4, Montreal, Canada, May 17-21 2004, pp. 161–164.

[65] Y. Huang, J. Benesty, and J. Chen, *Acoustic MIMO Signal Processing*. Berlin: Springer-Verlag, 2006.

[66] Y. Huang, J. Chen, and J. Benesty, “Immersive audio schemes,” *IEEE Signal Processing Magazine*, vol. 28, no. 1, pp. 20–32, Jan. 2011.

[67] B. Jelfs, P. Vayanos, S. Javidi, V. Su Lee Goh, and D. P. Mandic, *Signal Processing Techniques for Knowledge Extraction and Information Fusion*. New York, NY: Springer Science+Business Media, LLC, 2008, ch. Collaborative Adaptive Filters for Online Knowledgs Extraction and Information Fusion, pp. 3–21.

[68] J.-M. Jot, “Real-time spatial processing of sounds for music, multimedia and interactive human-computer interfaces,” *ACM Multimedia Systems Journal*, vol. 7, no. 1, pp. 55–69, Jan. 1999.

[69] T. Kailath, *Linear Systems*. Englewood Cliffs, NJ: Prentice-Hall, Jan. 1980.

[70] A. J. M. Kaizer, “Modeling of the nonlinear response of an electrodynamic loudspeaker by a Volterra series expansion,” *Journal of the Audio Engineering Society*, vol. 35, no. 6, pp. 421–433, Jun. 1987.

[71] J. Kivinen and M. K. Warmuth, “Exponentiated gradient versus gradient descent for linear predictors,” *Information and Computation*, vol. 132, no. 1, pp. 1–64, Jan. 1997.

[72] M. S. Klassen and Y. H. Pao, “Characteristics of the functional link net: A higher order delta rule net,” in *Proc. of the IEEE 2nd Annual International Conference on Neural Networks (ICNN ’88)*, vol. 1, San Diego, CA, Jul. 24 1988, pp. 507–513.

[73] S. Kozat and A. Singer, “Multi-stage adaptive signal processing algorithms,” in *Proc. of the IEEE Workshop on Sensor Array and Multichannel Signal Processing (SAM ’00)*, Cambridge, MA, Mar. 16-17 2000, pp. 380–384.

[74] S. S. Kozat, A. T. Erdogan, A. C. Singer, and A. H. Sayed, “Steady-state MSE performance analysis of mixture approaches to adaptive filtering,” *IEEE Transactions on Signal Processing*, vol. 58, no. 8, pp. 4050–4063, Aug. 2010.

[75] R. H. Kwong and E. W. Johnston, “A variable step size LMS algorithm,” *IEEE Transactions on Signal Processing*, vol. 40, no. 7, pp. 1663–1642, Jul. 1992.

[76] T. T. Lee and J. T. Jeng, “The Chebyshev polynomial-based unified model neural networks for functional approximation,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 28, no. 6, pp. 925–935, Dec. 1998.

[77] N. Levinson, “The Wiener RMS (root-mean-square) error criterion in filter design and prediction,” *Journal of Mathematics and Physics*, vol. 25, no. 4, pp. 261–278, Jan. 1947.

[78] L. Liu, M. Fukumoto, S. Saiki, and S. Zhang, “A variable step-size proportionate affine projection algorithm for identification of sparse impulse response,” *EURASIP Journal on Advances in Signal Processing*, vol. 2009, p. 10, 2009.

[79] R. W. Lucky and H. R. Rudin, “Generalized automatic equalization for communication channels,” *Proc. of the IEEE*, vol. 54, no. 3, pp. 439–440, Mar. 1966.

[80] A. Mader, H. Puder, and G. U. Schmidt, “Step-size control for acoustic echo cancellation filters - an overview,” *Signal Processing*, vol. 80, no. 9, pp. 1697–1719, Sep. 2000.

[81] D. P. Mandic, M. Chen, T. Gautama, M. M. Van Hulle, and A. Constantinides, "On the characterization of the deterministic/stochastic and linear/nonlinear nature of time series," *Proc. of the Royal Society*, vol. 464, no. 2093, pp. 1141–1160, Feb. 2008.

[82] D. P. Mandic, P. Vayanos, C. Boukis, B. Jelfs, S. Goh, T. Gautama, and T. Rutkowski, "Collaborative adaptive learning using hybrid filters," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '07)*, vol. 3, Honolulu, HI, Apr. 15-20 2007, pp. 921–924.

[83] D. G. Manolakis, V. K. Ingle, and S. M. Kogon, *Statistical and Adaptive Signal Processing: Spectral Estimation, Signal Modeling, Adaptive Filtering and Array Processing*. Norwood, MA: Artech House, Inc., Apr. 2005.

[84] M. Martínez-Ramón, J. Arenas-García, A. Navia-Vázquez, and A. R. Figueiras-Vidal, "An adaptive combination of adaptive filters for plant identification," in *Proc. of the International Conference on Digital Signal Processing (DSP '02)*, vol. 2, Santorini, Greece, Jul. 1-3 2002, pp. 1195–1198.

[85] V. J. Mathews and S. C. Douglas, *Adaptive Filters*. Upper Saddle River, NJ: Prentice Hall, 2003.

[86] V. J. Mathews and G. L. Sicuranza, *Polynomial Signal Processing*. New York, NY: John Wiley & Sons, 2000.

[87] M. Maybury, "Intelligent user interfaces: An introduction," in *Proc. of the 4th International Conference Intelligent User Interfaces (IUI '99)*. Los Angeles,CA: ACM, 1999, pp. 3–4.

[88] J. McCarthy, M. Minsky, N. Rochester, and C. Shannon, "A proposal for dartmouth summer research project on artificial intelligence," Dartmouth College, Tech. Rep., 1955.

[89] E. Milios, B. Kapralos, A. Kopinska, and S. Stergiopoulos, "Sonification of range information for 3-D space perception," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 11, no. 4, pp. 416–421, Dec. 2003.

[90] A. Namatame and Y. Kimata, "Improving the generalizing capabilities of a back-propagation network," *International Journal of Neural Networks*, vol. 1, no. 2, pp. 86–94, 1989.

[91] A. Namatame and N. Ueda, "Pattern classification with Chebyshev neural networks," *International Journal of Neural Networks*, vol. 3, pp. 23–31, Mar. 1992.

[92] K. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Transactions on Neural Networks*, vol. 1, no. 1, pp. 4–27, Mar. 1990.

[93] P. A. Naylor, J. Cui, and M. Brookes, "Adaptive algorithms for sparse echo cancellation," *Signal Processing*, vol. 86, no. 6, pp. 1182–1192, Jun. 2005.

[94] S. T. Neely and J. B. Allen, "Invertibility of a room impulse response," *Journal of Acoustical Society of America*, vol. 68, pp. 165–169, Jul. 1979.

[95] A. Newell and H. A. Simon, "Computer science as empirical inquiry: Symbols and search," *Communications of the ACM*, vol. 19, no. 3, pp. 113–126, Mar. 1976. [Online]. Available: http://doi.acm.org/10.1145/360018.360022

[96] B. S. Nollet and D. L. Jones, "Nonlinear echo cancellation for hands-free speakerphones," in *Proc. of the IEEE-EURASIP Workshop on Nonlinear Signal Image Processing (NSIP '97)*, Mackinac Island, MI, Sep. 8-10 1997.

[97] T. Ogunfunmi, *Adaptive Nonlinear System Identification: The Volterra and Wiener Model Approaches*. Berlin: Springer-Verlag, 2007.

[98] K. Ozeki and T. Umeda, "An adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties," *Electronics and Communications in Japan*, vol. 67-A, no. 5, pp. 19–27, 1984.

[99] C. Paleologu, J. Benesty, and S. Ciochină, "A variable step-size affine projection algorithm designed for acoustic echo cancellation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 8, pp. 1466–1478, Nov. 2008.

[100] ——, *Sparse Adaptive Filters for Echo Cancellation*. Morgan & Claypool Publishers, 2010.

[101] C. Paleologu, S. Ciochină, and J. Benesty, "Variable step-size NLMS algorithm for under-modeling acoustic echo cancellation," *IEEE Signal Processing Letters*, vol. 15, pp. 5–8, 2008.

[102] ——, "An efficient proportionate affine projection algorithm for echo cancellation," *IEEE Signal Processing Letters*, vol. 17, no. 2, pp. 165–168, Feb. 2010.

[103] Y.-H. Pao, *Adaptive Pattern Recognition and Neural Networks*. Reading, MA: Addison-Wesley, 1989.

[104] Y.-H. Pao and R. D. Beer, "The functional link net: A unifying network architecture incorporating higher order effects," in *Proc. of the First Annual Meeting of the International Neural Network Society (INNS '88)*, Boston, MA, Sep. 6 1988.

[105] E. V. Papoulis and T. Stathaki, "A normalized robust mixed-norm adaptive algorithm for system identification," *IEEE Signal Processing Letters*, vol. 11, pp. 56–59, 2004.

[106] R. Parisi, R. Russo, M. Scarpiniti, and A. Uncini, "Performance of acoustic nonlinear echo cancellation in the presence of reverberation," in *Proc. of the International Symposium on Frontiers of Research in Speech and Music (FRSM '09)*, Gwalior, India, Dec. 15-16 2009, pp. 106–111.

[107] J. C. Patra, W. C. Chin, P. K. Meher, and G. Chakraborty, “Legendre-FLANN-based nonlinear channel equalization in wireless communication system,” in *Proc. of the IEEE International Conference on Systems, Man and Cybernetics (SMC ’08)*, Singapore, Oct. 12-15 2008, pp. 1826–1831.

[108] J. C. Patra and A. C. Kot, “Nonlinear dynamic system identification using Chebyshev functional link artificial neural networks,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 32, no. 4, pp. 505–511, Aug. 2002.

[109] J. C. Patra, R. N. Pal, B. N. Chatterji, and G. Panda, “Identification of nonlinear dynamic systems using functional link artificial neural networks,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 29, no. 2, pp. 254–262, Apr. 1999.

[110] J. C. Patra, W. B. Poh, N. S. Chaudhari, and A. Das, “Nonlinear channel equalization with QAM signal using Chebyshev artificial neural network,” in *Proc. of the IEEE International Joint Conference on Neural Networks (IJCNN ’05)*, vol. 5, Montreal, Canada, Jul. 31 - Aug. 4 2005, pp. 3214–3219.

[111] J. C. Patra and A. Van Den Bos, “Modeling of an intelligent pressure sensor using functional link artificial neural networks,” *ISA Transactions*, vol. 39, no. 1, pp. 15–27, Feb. 2000.

[112] V. Petrini, “Teoria ed applicazioni del metodo perturbativo nellÕapprendimento di reti neurali,” Master’s thesis, UniversitĹ degli Studi di Ancona, 1996.

[113] F. Piazza, A. Uncini, and M. Zenobi, “Artificial neural networks with adaptive polynomial activation function,” in *Proc. of the IEEE International Joint Conference on Neural Networks (IJCNN ’92)*, vol. 2, Bejing, China, Nov. 3-6 1992, pp. 343–349.

[114] ——, “Neural networks with digital LUT activation function,” in *Proc. of the IEEE International Joint Conference on Neural Networks (IJCNN ’93)*, vol. 2, Nagoya, Japan, Oct. 25-29 1993, pp. 1401–1404.

[115] G. Rombouts and M. Moonen, “A sparse block exact affine projection algorithm,” *IEEE Transactions on Speech and Audio Processing*, vol. 12, no. 2, pp. 100–108, Feb. 2002.

[116] W. J. Rugh, *The Linear System Theory*, 2nd ed. Englewood Cliffs: NJ: Prentice-Hall, 1996.

[117] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd ed. Upper Saddle River, NJ: Prentice Hall, 2003.

[118] W. C. Sabine, *Collected Papers on Acoustics*. Cambridge, MA: Harvard University Press, 1992.

[119] K. Sakhnov, “An improved proportionate affine projection algorithm for network echo cancellation,” in *Proc. of the IEEE International Conference on Systems, Signals and Image Processing (IWSSIP ’08)*, Bratislava, Slovakia, Jun. 25-28 2008, pp. 125–128.

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

[121] M. Scarpiniti, D. Comminiello, R. Parisi, and A. Uncini, “Comparison of Hammerstein and Wiener systems for nonlinear acoustic echo cancelers in reverberant environments,” in *Proc. of the IEEE International Conference on Digital Signal Processing (DSP ’11)*, Corfù, Greece, Jul. 6-8 2011, pp. 1–6.

[122] R. Serizel, M. Moonen, J. Wouters, and S. H. Jensen, “Integrated active noise control and noise reduction in hearing aids,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 11, no. 6, pp. 1137–1146, Aug. 2010.

[123] K. Shi, X. Ma, and G. T. Zhou, “Adaptive acoustic echo cancellation in presence of multiple nonlinearities,” in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP ’08)*, Las Vegas, NV, Mar. 30 - Apr. 4 2008, pp. 3601–3604.

[124] H.-C. Shin, A. H. Sayed, and W.-J. Song, “Variable step-size NLMS and affine projection algorithms,” *IEEE Signal Processing Letters*, vol. 11, no. 2, pp. 132–135, Feb. 2004.

[125] G. L. Sicuranza and A. Carini, “A generalized FLANN filter for nonlinear active noise control,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 8, pp. 2412–2417, Nov. 2011.

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

[127] J. O. Smith III, *Physical Audio Signal Processing: for Virtual Musical Instruments and Digital Audio Effects*. W3K Publishing, Dec. 2010, vol. 3.

[128] D. J. Sobajic, Y.-H. Pao, and D. T. Lee, “Robust control of nonlinear systems using pattern recognition,” in *Proc. of the IEEE International Conference on Systems, Man and Cybernetics (SMC ’89)*, vol. 1, Cambridge, MA, Nov. 14-17 1989, pp. 315–320.

[129] M. Solazzi and A. Uncini, “Regularizing neural networks using flexible multivariate activation function,” *Neural Networks*, vol. 17, no. 2, pp. 247–260, 2004.

[130] M. M. Sondhi, “An adaptive echo canceller,” *Bell System Technical Journal*, vol. 46, no. 3, pp. 497–511, Mar. 1967.

[131] ——, *Springer Handbook of Speech Processing*. Berlin: Springer-Verlag, 2008, ch. Adaptive echo cancellation for voice signals, pp. 903–927, ch. 45, pt. H.

[132] M. M. Sondhi and A. J. Presti, "A self-adaptive echo canceler," *Bell System Technical Journal*, vol. 45, no. 10, pp. 1851–1854, Dec. 1966.

[133] T. P. Spexard, M. Hanheide, and G. Sagerer, "Human-oriented interaction with an anthropomorphic robot," *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 852–862, Oct. 2007.

[134] C. Stanciu, C. Anghel, C. Paleologu, J. Benesty, F. Albu, and S. Ciochină, "A proportionate affine projection algorithm using dichotomous coordinate descent iterations," in *Proc. of the IEEE International Symposium on Signals, Circuits and Systems (ISSCS '11)*, Iasi, Romania, Jun. 30 - Jul. 1 2011, pp. 1–4.

[135] L. Stanković, "Performance analysis of the adaptive algorithm for bias-to-variance tradeoff," *IEEE Transactions on Signal Processing*, vol. 52, no. 5, pp. 1228–1234, May 2004.

[136] A. Stenger and R. Rabenstein, "An acoustic echo canceller with compensation of nonlinearities," in *Proc. of the European Signal Processing Conference (EUSIPCO '98)*, Isle of Rhodes, Greece, Sep. 8-11 1998, pp. 969–972.

[137] ——, "Adaptive Volterra filters for acoustic echo cancellation," in *Proc. of the IEEE-EURASIP Workshop on Nonlinear Signal Image Processing (NSIP '99)*, vol. 2, Antalya, Turkey, Jun. 20-23 1999, pp. 679–683.

[138] A. Stenger, L. Trautmann, and R. Rabenstein, "Nonlinear acoustic echo cancellation with 2nd order adaptive Volterra filters," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '99)*, vol. 2, Phoenix, AZ, Mar. 15-19 1999, pp. 877–880.

[139] M. H. Stone, "The generalized Weierstrass approximation theorem," *Mathematics Magazine*, vol. 21, no. 4, pp. 167–184, 1948.

[140] M. Tanaka, Y. Kaneda, S. Makino, and J. Kojima, "A fast projection algorithm for adaptive filtering," *IEICE Transactions on Fundamentals*, vol. E78-A, no. 10, p. 1355Ð1361, Oct. 1995.

[141] M. Tanaka, S. Makino, and J. Kojima, "A block exact fast affine projection algorithm," *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 1, pp. 79–86, Jan. 1999.

[142] A. Tate, Y.-H. Chen-Burger, J. Dalton, S. Potter, D. Richardson, J. Stader, G. Wickler, I. Bankier, C. Walton, and P. Williams, "I-Room: A virtual space for intelligent interaction," *IEEE Intelligent Systems*, vol. 25, no. 4, pp. 62–71, Aug. 2010.

[143] E. J. Thomas, "Some considerations on the application of the Volterra representation of nonlinear networks to adaptive echo canceller," *Bell System Technical Journal*, vol. 50, no. 8, pp. 2797–2905, Oct. 1971.

[144] A. M. Turing, "Computing machinery and intelligence," *Mind*, vol. 59, pp. 433–460, 1950.

[145] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, Jul. 2002.

[146] A. Uncini, *Elaborazione Adattativa dei Segnali*. Rome: Aracne Editrice S.R.L., 2010.

[147] A. Uncini, A. Nalin, and R. Parisi, "Acoustic echo cancellation in the presence of distorting loudspeakers," in *Proc. of the European Signal Processing Conference (EUSIPCO '02)*, vol. 1, Tolouse, France, Sep. 3-6 2002, pp. 535–538.

[148] L. Vecci, F. Piazza, and A. Uncini, "Learning and approximation capabilities of adaptive spline activation function neural networks," *Neural Networks*, vol. 11, no. 2, pp. 259–270, Mar. 1998.

[149] L. R. Vega, H. Rey, J. Benesty, and S. Tressens, "A family of robust algorithms exploiting sparsity in adaptive filters," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 4, pp. 572–581, May 2009.

[150] V. Volterra, *Theory of Functionals and of Integral, and Integral-Differential Equations*. London, UK: Blackie & Son, Ltd., 1930.

[151] W. D. Weng, C. S. Yang, and R. C. Lin, "A channel equalizer using reduced decision feedback Chebyshev functional link artificial neural networks," *Information Sciences*, vol. 177, no. 13, pp. 2642–2654, Jul. 2007.

[152] S. Werner, J. A. Apolinário Jr., and P. S. R. Diniz, "Set-membership proportionate affine projection algorithms," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2007, no. 1, pp. 1–10, 2007.

[153] B. Widrow and M. E. Hoff Jr., "Adaptive switching circuits," in *IRE WESCON Convention Record*, vol. IV, Los Angeles, CA, Aug. 23-26 1960, pp. 96–104.

[154] B. Widrow and M. A. Lehr, "30 years of adaptive neural networks: Perceptron, Madaline, and backpropagation," *Proc. of the IEEE*, vol. 78, no. 9, pp. 1415–1442, 1990.

[155] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*, 1st ed. Englewood Cliffs, NJ: Prentice Hall, Inc., 1985.

[156] N. Wiener, *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*. New York, NY: John Wiley /& Sons, 1949.

[157] ——, *Nonlinear Problems in Random Theory*, 1st ed. New York, NY: John Wiley & Sons, Dec. 1958.

[158] N. Wiener and E. Hopf, "On a class of singular integral equations," in *Proc. of Prussian Academy, Math.-Phys. Series*, 1931, p. 696.

[159] S. S. Yang and C. S. Tseng, "An orthonormal neural network for function approximation," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 26, pp. 779–785, Oct. 1996.

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

[161] H. Zhao and J. Zhang, "Functional link neural network cascaded with Chebyshev orthogonal polynomial for nonlinear channel equalization," *Signal Processing*, vol. 88, no. 8, pp. 1946–1957, Aug. 2008.

[162] ——, "Adaptively combined FIR and functional link artificial neural network equalizer for nonlinear communication channel," *IEEE Transactions on Neural Networks*, vol. 20, no. 4, pp. 665–674, Apr. 2009.

[163] Y. R. Zheng and R. A. Goubran, "Adaptive beamforming using affine projection algorithms," in *Proc. of the 5th IEEE International Conference on Signal Processing (WCCC-ICSP '00)*, vol. 3, Beijing, China, Aug. 21-25 2000, pp. 1929–1932.

[164] W. Zhuang, "RLS algorithm with variable forgetting factor for decision feedback equalizer over time-variant fading channels," *Wireless Personal Commununications*, vol. 8, pp. 15–29, 1998.

[165] U. Zölzer and X. Amatriain, *DAFX: Digital Audio Effects*. John Wiley & Sons, Inc., 2002.

INDEX