UNIVERSITY OF NICE - SOPHIA ANTIPOLIS

DOCTORAL SCHOOL STIC

SCIENCES ET TECHNOLOGIES DE L'INFORMATION
ET DE LA COMMUNICATION

PHD THESIS

to obtain the title of

PhD of Science

of the University of Nice - Sophia Antipolis

Specialty : Automatics, Signal and Image Processing

Defended by

Moctar MOSSI IDRISSA

Non-linear AEC with loudspeaker modelling and pre-processing

Thesis Supervisor: Nicholas W.D. EVANS

prepared at EURECOM Sophia Antipolis, MultiMedia Department

defended on October 17, 2012

Jury :

Reviewers :	Pr. Régine LE BOUQUIN-JEANNES	-	Rennes 1 University
	Pr. Dominique PASTOR	-	Telecom Bretagne
President :	Pr. Dirk SLOCK	-	EURECOM
Advisors :	Dr. Christophe BEAUGEANT	-	Intel Mobile Communications
	Dr. Nicholas D.W. EVANS	-	EURECOM

Abstract

This thesis presents new solutions to non-linear echo cancellation using loudspeaker pre-processing. A theoretical and experimental analysis of linear echo cancellation behaviour in non-linear environments is first introduced and shows that performance is typically degraded in the presence of non-linearities. This supports the need for dedicated non-linear solutions.

Two new approaches to non-linear acoustic echo cancellation are proposed. They involve a common approach to loudspeaker modelling which is based on measurements from a real mobile phone and simulations. Results are used to characterise and model the loudspeaker which is proven to be the dominant cause of non-linearities. The loudspeaker model is used in one of two different pre-processing structures both with the aim of improving acoustic echo cancellation performance in non-linear environments. The pre-processor is placed either before the linear acoustic echo cancellation module or before the loudspeaker in an otherwise conventional approach to acoustic echo cancellation.

The first arrangement aims to emulate loudspeaker behaviour so that non-linearities are taken into account by the linear acoustic echo cancellation module. Performance remains affected by clipping and subject to increased computational burden. An improved approach, combining clipping compensation in the pre-processor and decorrelation filtering in the linear acoustic echo cancellation module is subsequently introduced and demonstrates improved convergence and tracking capability compared to the existing state of the art.

When placed before the loudspeaker the pre-processor aims to linearise the loudspeaker output in a form of pre-compensation. This approach naturally improves the performance of otherwise standard approaches to linear acoustic echo cancellation. Compared to current state-of-the-art solutions, where the pre-processor is static, the new algorithm can dynamically adapt to the changes in loudspeaker characteristics over time. However, the pre-processor adaptation can be paused without significant losses in performance so that re-initialisation of parameters is not required for each new call.

Finally, we report a comparative analysis of the different non-linear acoustic echo cancellers which shows that the classical approach using loudspeaker emulation has a good reactivity to echo path changes, however convergence can be slow in highly non-linear conditions. Hence, by incorporating clipping compensation and decorrelation filtering, the system is more robust to clipping distortion, has better convergence and echo reduction performance. When the pre-processor is used to pre-compensate the loudspeaker, the robustness of linear acoustic echo cancellation to echo path changes and echo reduction performance are both improved. The analysis demonstrate that the combination of clipping compensation and decorrelation filtering represent a good practical solution to non-linear acoustic echo cancellation for mobile communication systems. The new algorithms are shown to outperform existing, well-known solutions with real signals.

Acknowledgements

I would like to thank my supervisors, Dr. Nick Evans and Dr. Christophe Beaugeant for their help, support and advice during my study. I would also like to express my gratitude to the members of the Jury, Pr. Regine Le-Bouquin Jeannes and Pr. Dominique Pastor for accepting to review my thesis and also Pr. Dirk Slock for being the president of the jury.

I would also like to thank the administration at eurecom specially the secretariat for their support on different visa issues I had. I am also thankful to my colleagues from eurecom especially my co-worker Melle Christelle Yemdji. My great thanks go to my colleague, room-mate and friend Dr. Simon Bozonnet for his help during the period of my study. I would also like to thank Infineon technologies (currently Intel Mobile Communications) for funding my thesis and the entire Intel Mobile Communications DSP group at Sophia-Antipolis for their contribution during my work.

My warm thanks are due to all my friends and my family especially my parents for their support, my dear wife Balkissa for being patient and comprehensive, Kader and Nasser for their all-time support.

Finally, I would like to thank all my teachers for their devotion to our school success and their tireless encouragements.

To all my teachers,
$\cdots$ Specially my dear mother.

For her commitment

Symbols

n : Time indice.
$x(n)$: Far end speech signal.
$s(n)$: Near end speech signal.
$d(n)$: Echo signal.
$\hat{d}(n)$: Estimate of the echo signal.
$y(n)$: Microphone signal.
$h(n)$: Impulse response of the Loudspeaker Enclosure Microphone System (LEMS) system (target impulse response).
$\hat{h}(n)(n)$: Impulse response of the Acoustic Echo Cancellation (AEC) filter.
$e(n)$: Estimation error.
$n(n)$: Ambient noise at the microphone.
λ : Eigenvalue.
M : Matrix or vector dimension.
$\mathbf{x}(n) = [x(n), x(n-1), ..., x(n-M+1)]^T$: Input vector of the filter.
$\mathbf{h}(n) = [h_0(n), h_1(n), ..., h_{M-1}(n)]$: Filter taps vector.
$h^0(n)$: Optimal filter in MMSE sense.
$h_p(n)$: Sub-filter of a non-linear filter system.
$h_Q(n)$: Second order Volterra kernel.
$\mathbf{R}$: Auto-correlation matrix of the input vector.
$\mathbf{P}$: Cross-correlation matrix of the input vector and the reference signal.
$\mathbf{Q}$: Eigenvector matrix. v_s : sound velocity.
f_s : sampling frequency.

Abbreviations

ADC Analog-to-Digital Converter

AEC Acoustic Echo Cancellation

APA Adaptive Projection Algorithm

AR Auto Regressive

ASPM Adaptive Sub-gradient Projected Method

AIR Aachen Impulse Responses

BLMS Block LMS

CC Clipping Compensation

CD Cepstral Distance

CS Cascaded Structure

CS1 CS 1

CS + CC Cascaded Structure with Clipping Compensation

CS + CC + DF Cascaded Structure with Clipping Compensation and Decorrelation Filtering

CS + DF Cascaded Structure with Decorrelation Filtering

DAC Digital-to-Analog Converter

DL Down-Link

DCL Dynamic Compression and Limitation

DCT Discrete Cosine Transform

DF Decorrelation Filtering

DFT Discrete Fourier Transform

DCTLMS Discrete Cosine Transform-LMS

DFTLMS Discrete Fourier transform-LMS

DSP Digital Signal Processor

DT Double Talk

DTD Double Talk Detector

E-RLS Extended RLS

email Electronic mail

EP Echo Path

EPC Echo Path Change

ERLE Echo Return Loss Enhancement

FAPA Fast Adaptive Projection Algorithm

FBLMS Frequency Block LMS

FFT Fast Fourier Transform

FIR Finite Impulse Response

FRLS Fast RLS

FTF Fast Transversal Filter

ICASSP International Conference on Acoustics, Speech, and Signal Processing

ICSP International Conference on Signal Processing

IDFT Inverse Discrete Fourier Transform

IFFT Inverse Fast Fourier Transform

i.i.d independent and identically distributed

IIR Infinite Impulse Response

IMC Intel Mobile Communications

IPNLMS Improved PNLMS

ITU International Telecommunication Union

ITU-T ITU Telecommunication Standardization Sector

IWAENC International Workshop on Acoustic Echo and Noise Control

LEMS Loudspeaker Enclosure Microphone System

LTI Linear Time Invariant

LTV Linear Time Variant

LMS Least Mean Square

LP Loudspeaker Pre-processing

LP1 Loudspeaker Pre-processing 1

LS Least Square

LRLS Lattice Recursive Least Square

MMD Multi Memory Decomposition

MISO Multiple Inputs Single Output

MIMO Multiple Inputs Multiple Output

MMSE Minimum Mean Square Error

MSE Mean Square Error

NLMS Normalized-LMS

PAPA Proportionate APA

PC Personal Computer

PFBLMS Partitioned Frequency Block LMS

PFBVLMS Partitioned Frequency Block Volterra LMS

PNLMS Proportionate NLMS

POCS Projection Onto Convex Set

PS Parallel Structure

QR-RLS QR Recursive Least Square

Re-NLMS Re-estimated NLMS

RLS Recursive Least Square

SD System Distance

LsD Log-spectral Distance

SER Signal-to-Estimate Ratio

SIMO Single Input Multiple Output

SISO Single Input Single Output

SMS Short Message Service

SNeR linear echo to non-linear echo ratio

SNR Signal-to-Noise Ratio

ST Single Talk

STFT Short-Term Fourier Transform

TDLMS Transform Domain LMS

THD Total Harmonic Distortion

VAD Voice Activity Detection

VoIP Voice-over-IP

w.r.t with respect to

UL Up-Link

WASPAA IEEE Workshop on Applications of Signal Processing to Audio and Acoustics

Contents

Chapter 1

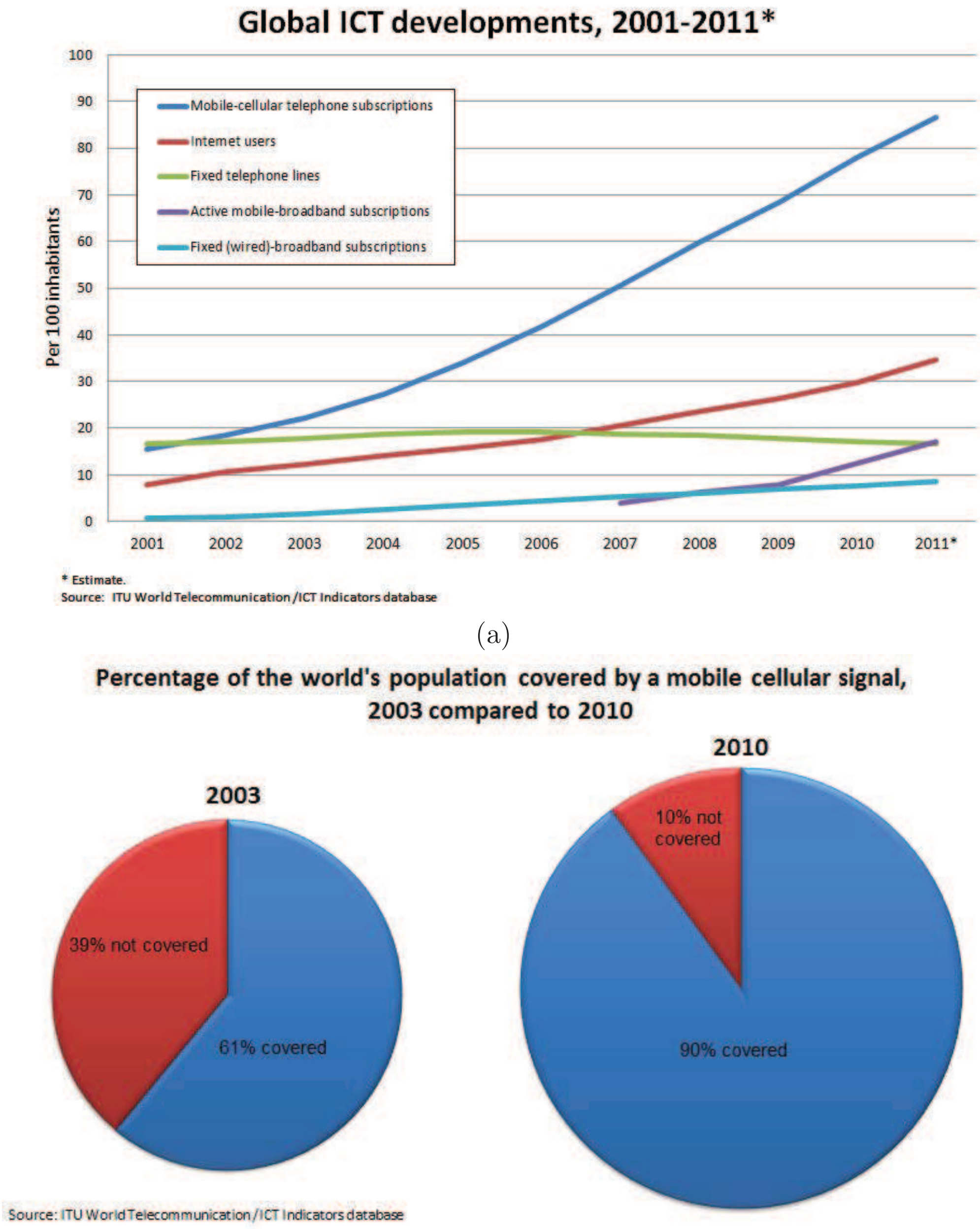
Introduction

Communications has become very important in our daily life and since the development of mobile phones the communications systems market has grown rapidly. Most recently the demand relates to smart-phones which have the capability to support mobile communications and Internet applications. These smart-phones provide the possibility of voice communication via switched-circuits but also applications such as Voice-over-IP (VoIP). The latter provides low cost options for some long distance communications.

The growth of business-related sectors implies people from different countries working together on the same project. This has lead to an increasing demand for teleconferencing applications. Even if teleconferencing requires the use of image and speech components the latter is the most important. This shows that even with the growth of text messaging and email other communication mediums, speech still remains the most important. The advantage of speech communication is mainly due to the fact that it is a traditional medium of communication and presents the additional advantage to provide the mood sentiment and other non-linguistic information which is difficult to transcribe through text messages.

Figure 1.1 presents statistics data extract from ITU information and communications technologies (ICT) database. It shows the development of different communications systems from to 2001 to 2011 and the growth of the world population that have access to the mobile phone network between 2003 and 2010. We observe on Figure 1.1 (a) the increase in demand for all communication systems, except for the fixed phone which is more-less constant. We observe that demand for broadband fixed phones is increasing and, in particular, that demand for mobile phones is increasing rapidly. The increase in mobile broadband subscriptions will increase the use of mobile VoIP. In Figure 1.1 (b) we observe that mobile phone deployment covers about 90% of the world population as the channel is not dedicated to the user, which is not the case for fixed phones. This shows the growth of the mobile market and the interest of the operator to cover more and more people. This also requires the provision of accessible mobile terminals meaning low cost devices.

All these progresses rely on some improvements in different research domains which aims to provide a better quality of service. In communications systems such as mobile the speech quality is very important. The enhancement of speech quality has lead to the development of many research areas. One of the most important is that related to this thesis, namely that of acoustic echo cancellation.



(a)

(b)

Figure 1.1: ITU statistic on information and communications technologies. (a) Global ICT developments, (b) Percentage of world population covered of mobile (*Source ITU world Telecommunication/ICT indicators database*).

1.1 Acoustic echo cancellation

Speech quality is important for acceptable communications and a large amount of the processing capacity of a typical mobile telephone is dedicated to general speech enhancement. A significant contribution to degradation in speech quality can be attributed to echo, i.e. when we hear a delayed version of our own voice. In mobile communications there are two sources of echo: the line echo due to mismatched impedances and that attributed to the acoustical coupling between the loudspeaker and the microphone. Even if there are many similarities in the way in which they are treated, the work described in this thesis relates to the latter, namely Acoustic Echo Cancellation (AEC).

The requirement for long distance calls with the possibility of full-duplex communication has mainly introduced the problem of acoustic echo. Acoustic echo arises when the signal of the loudspeaker is coupled to the microphone and sent to the far-end user who will hear his/her own voice. However, the delay is an important characteristics of the echo problem. When the delay is small the signal is perceived by the far-end listener as a reverberation whereas when it exceeds $30 - 50$ ms it is an echo signal and becomes disturbing [Burnett *et al.* 1988]. Nowadays communications delay is greater than 100 ms, and sometimes up to 700 ms. There is thus a need to reduce echo in communications and there is accordingly a large amount of research in the literature which is dedicated to the topic of echo cancellation [Hänsler & Schmidt 2004, Vary & Martin 2006]. Switching systems were originally used to prevent such problems but these systems do not allow full-duplex communication. The principal solution for full-duplex communication which reduces acoustic echo is based on the assumption of linearity of some components such as loudspeakers and microphones.

AEC is based on a system identification approach. It generally uses an adaptive filter to estimate the echo signal which is then subtracted from the microphone signal. AEC is a challenging problem which has been investigated first with the linearity assumption before being investigated in the non-linear domain. Linear AEC approaches often provide acceptable performance in linear condition, however, in presence of non-linearity such as loudspeaker distortion or amplifier saturation their performance degrades. Non-linear AEC is the topic of this thesis and our goal is to propose new solutions to improve the performance of non-linear AEC approaches.

1.2 Non-linear acoustic echo cancellation

With the growth of the mobile market the demand for cheaper and small terminals can lead to increased speech distortion which can be non-linear in nature. Non-linearity can degrade speech quality by introducing some other components in the original signal. It also reduces the performance of algorithms which are based on assumption of linearity. One of the most affected algorithms is the echo canceller.

Non-linearity sources

One of the factors which increases this non-linearity in mobile communications is the use of hands-free mode. Hands-free mode entails amplification using a small battery to provide a loud signal. As the battery is limited in size and power the amplifier is not always sufficient to reach certain amounts of amplification which lead to clipping distortion. The loudspeaker also generates some non-linearities. When the loudness of the signal increases these non-linearities become perceptible and disturbing. Distortions generated by the amplifier and loudspeaker are the most studied in the literature, even if they are not the only source of non-linearity. There are also those introduced by the casing vibration, the microphone and the different Analog-to-Digital Converter (ADC) or Digital-to-Analog Converter (DAC). The casing vibration non-linearities are less investigated due to the complexity and also the fact that they have shown to be independent from the original signal [Birkett & Goubran 1995b]. The converter non-linearities are generally considered as additive noise and are mostly ignored in non-linear AEC.

Non-linearity effects

In general non-linear distortions affect speech quality during communication. A collateral effect arises when non-linearities disturb algorithms which rely on linearity, such as linear AEC. Whereas linear AEC has proved to enhance speech quality in communication systems, the presence of non-linearities degrades linear AEC performance. This degradation leads to a more audible echo signal at the far-end, and thus perturbs communication. To solve this problem a solution proposed is the use of non-linear echo cancellation. This solution generally relies on linear AEC approaches but takes into account the non-linearities generated by the devices to improve performance. Many solutions have been proposed to solve this problem which are exposed further in this thesis.

The work presented in this thesis is dedicated to the problem of acoustic echo cancellation in non-linear environments. This work principally focuses on different strategies to increase linear AEC performance in non-linear environments based on linear AEC or loudspeaker analysis and pre-processing. Two different approaches are used here: first an approach based on loudspeaker emulation and the second on the linearisation of the loudspeaker.

1.3 Context of the thesis

This work was supported by Intel Mobile Communications (IMC) group. IMC is a leader in the mobile communications field. The work was overseen by the DSP group of Infineon Technologies at Sophia-Antipolis which become part of IMC in 2011.

The challenge is to provide solutions to the non-linear acoustic echo problem. Indeed some solutions have already been proposed in this area. They are mainly

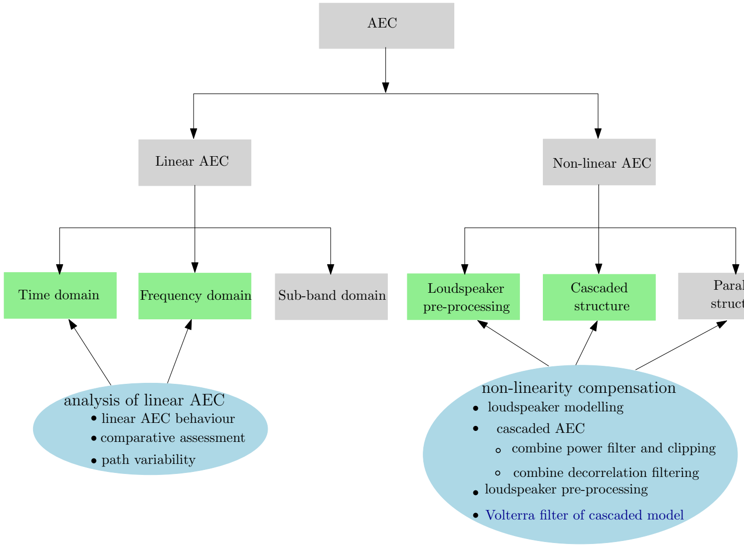


Figure 1.2: AEC applications and our contributions in the blue boxes

based on the Volterra approach, which is generally complex. Other solutions with different structures have been proposed, i.e. cascaded structures or non-linear post-processing which uses the noise suppression approach to non-linear residual echo suppression.

In this thesis our approach started with an analysis of linear AEC solutions in non-linear environments to well understand the effects of non-linearity on linear AEC and mainly focuses on their robustness to non-linearities. We have identified the non-linear sources in the Loudspeaker Enclosure Microphone System (LEMS) and propose a model for these non-linearities. Then we propose the use of this model to the compensation of non-linearity in different approaches. We first use an improved cascaded approach then a new solution based on on-line loudspeaker linearisation.

1.4 Contributions

The main contributions of this work are three-fold. They are (i) an investigation of non-linear distortion and noise effects on linear AEC performance, (ii) two different, novel approaches to loudspeaker modelling, and (iii) new solutions to non-linear AEC with loudspeaker non-linearity pre-processing. The three contributions are described in more detail below.

- **Analysis of non-linear distortion and noise effects on linear AEC performance**
  Most current approaches to non-linear AEC are based upon, or have their roots in standard linear algorithms. Initial work aims to highlight the nature of non-linear artefacts and how they degrade AEC performance. First, the

contribution relates to a thorough comparative performance analysis of various linear AEC algorithms in the presence of non-linear distortion. Since the performance of linear AEC in the presence of acoustic noise has received a great deal of attention, and thus many diverse noise compensation algorithms have been developed, the contribution also relates to a comparison of system behaviour in the face of acoustic noise and non-linear echo. This latter work aims to determine whether or not approaches to noise compensation have potential utility in attenuating the effects of non-linear distortion. Second, the contribution relates to a new theoretical analysis of linear AEC in non-linear environments. The analysis is based on the derivation of the Wiener solution under the assumption that the linear and non-linear components are correlated.

- *Comparative performance analysis of linear AEC with non-linear distortion*: in general, most approaches to linear AEC assume that the input signal is independent and identically distributed (i.i.d). This assumption is unrealistic in the face of non-linear distortion which can be dependent on signal characteristics. The thesis reports a new comparative assessment of different linear AEC algorithms and their performance in non-linear environments. Reported are experiments which measure the difference in Echo Return Loss Enhancement (ERLE) between linear and non-linear environments, convergence time and system distance (linear component only). Frequency block-filtering approaches are shown to be the most disturbed in non-linear environments. An Adaptive Projection Algorithm (APA) approach is furthermore shown not to perform any better than a standard Normalized-LMS (NLMS) algorithm. A comparative assessment of non-linear echo and acoustic noise effects is also presented according to the same experimental approach. Results highlight better robustness to non-linear distortion than to noise and clearly show that non-linear distortion cannot be considered as additive thus necessitating specific approaches to AEC in non-linear environments.
- *New theoretical analysis of linear AEC in non-linear environments*: in order to better explain behaviours and results observed in the comparative study a new theoretical analysis of non-linear effects is presented. According to the proposed analysis non-linear echo is divided into correlated and uncorrelated components, where correlation relates to the far-end signal. Using this decomposition we show that non-linear environments can be characterised according to a pseudo-variable echo path which depends on the far-end signal characteristics. The new theoretical analysis better accounts for observed experimental results than any existing theory and shows why NLMS algorithms often perform better than APA algorithms in the presence of non-linear distortion; their use of less memory affords increased robustness to non-linear distortion. The analysis furthermore shows that post-processing to attenuate non-linear

artefacts is likely to be more complex than that for noise since, under such conditions, the linear AEC filter is not guaranteed to converge to the linear Wiener solution.

The assessment of linear AEC in non-linear environments was presented at the International Conference on Acoustics, Speech, and Signal Processing (ICASSP) in 2010 [Mossi *et al.* 2010a]. The comparison of non-linear and noise effects was presented at the International Conference on Signal Processing (ICSP) also in 2010 [Mossi *et al.* 2010b]. The same is presented in a technical report [Mossi *et al.* 2010c] which extends the work to include the new theoretical analysis.

- **Novel approaches to loudspeaker modelling**
  The analysis of linear AEC shows that the performance of linear algorithms can degrade significantly in the presence of non-linear distortion. The second contribution thus relates to an analysis of non-linear distortion typically introduced by system components and their modelling as a precursor to the design of suitable compensation algorithms. The objective here is to model non-linearities introduced by loudspeakers, which have been identified as the main source of non-linearity in the literature and as confirmed in our own experimental tests.

  – *Loudspeaker modelling based on harmonic summation*: this approach consists in harmonic estimation according to the frequency and the amplitude of each signal component in the discrete-frequency domain. Harmonic components arising from normalised test signals are measured and stored in a two-dimensional matrix according to the base and harmonic frequencies. The matrix thus represents a model of non-linear distortions introduced by the loudspeaker and hence the non-linear distortion stemming from any discrete-frequency signal component can be estimated based on the matrix of harmonics. The approach can be used to generate effective estimates of the loudspeaker output and does not assume a predefined model of the loudspeaker but relies instead on empirical measurements of the loudspeaker response to a certain frequency and amplitude. Being based on simple harmonic estimation however, this approach does not take into account inter-modulation effects.

  Loudspeaker modelling based on polynomial expansion: with this alternative approach harmonics are generated according to a cosine power expansion and appropriately attenuated to form the output signal. This approach is less complex than harmonic summation and takes intermodulation effects into account. It is difficult to control, however, since the loudspeaker model is difficult to properly parameterise in the presence of inter-modulation. Nevertheless the approach is shown to provide a reliable estimate of the loudspeaker output and is less complex than existing approaches based on Volterra models.

This work was presented at the International Workshop on Acoustic Echo and Noise Control (IWAENC) in 2010 [Mossi *et al.* 2010d].

- **Loudspeaker non-linearity pre-processing**
  The third contribution relates to the use of loudspeaker models to implement non-linearity pre-processing algorithms, and hence to improve AEC performance in the presence of non-linear distortions. Due to their lower complexity, time domain approaches are preferred to frequency domain implementations. Two new algorithms have been developed.

  - *Cascaded structure*: The first approach is based on an adaptive pre-processing of the linear AEC input. The pre-processor aims to mimic the behaviour of the loudspeaker so that the pre-processor output is linear compared to that of the loudspeaker, thus the linear AEC module will reliably estimate the echo signal. While parallel implementations are possible, a cascaded structure is preferred since it requires fewer parameters to optimise and is more efficient in terms of tracking. Two extensions to the original approach have also been investigated:

    - *Combined hard-clipping compensation*: it was observed that variations in amplification can affect pre-processing performance and thus a combined loudspeaker pre-processing and hard-clipping compensation algorithm was also investigated. Given the added computational burden, a computationally efficient approach is proposed to reduce complexity.
    - *Reduced-complexity implementation*: this work aims to reduce the complexity of cascaded structures to loudspeaker pre-processing. Since pre-processing generally increases signal correlation, the work also considered the application of decorrelation filtering applied at the input of the linear AEC. While being based on well-known, existing algorithms, improved convergence requires efficient control of the different algorithms such that they function coherently in a cascaded structure.

  - *Loudspeaker pre-processing*: the second approach involves a combination of loudspeaker pre-processing (linearisation) and linear AEC. Using suitable loudspeaker models, linearisation pre-processing is applied at the input of the loudspeaker to reduce non-linear distortion at the output. This approach places no constraints on the use of any particular AEC algorithm and avoids the introduction of distortion in the error signal which can occur with alternative approaches to non-linear AEC. The proposed approach can thus give better near-end speech quality than existing solutions.

This work was presented at the ICASSP in 2011 and 2012 [Mossi *et al.* 2011a, Mossi *et al.* 2012]. The loudspeaker pre-processing was presented at the

IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA) in 2011 [Mossi *et al.* 2011b].

1.5 Organization

In Chapter 2 we describe the general approach to AEC based on adaptive filtering. We first introduce the Least Mean Square (LMS) algorithm which forms the basis of most Minimum Mean Square Error (MMSE) approaches and describe the different constraints involved in the use of adaptive filtering in AEC applications. They relate to characteristics of the input speech signal, the Echo Path (EP) and on the possible presence of noise or near-end speech. We then present several existing solutions to AEC. The weaknesses of each approach are described along with proposed solutions. In general they focus on the characteristics of the system such as speech signal eigenvalues spread or EP sparsity to improve the adaptive filter or processing in other domain such as frequency or sub-band.

The emphasis in Chapter 3 switches to non-linear environments which are now typical on account of device miniaturization and imperfections of low-cost devices. We describe existing solutions to non-linear AEC and include those relating to non-linear adaptive filtering and non-linear post-processing. The non-linear adaptive filtering approach, with which this thesis is concerned, generally extends linear adaptive filtering solutions to a non-linear LEMS model. Two main structures are presented: the parallel structure, where the LEMS is globally model by a non-linear system, and the cascaded structure where the LEMS is assumed to be a cascade of two different systems. In the cascaded structure two solutions are presented: a non-linear pre-processor followed by a linear AEC, and the loudspeaker linearisation approach. Finally we present the non-linear post-processing approach which uses similar procedure developed in residual echo reduction. We present the solutions proposed in this domain which are mainly based on frequency domain echo suppression.

Chapter 4 presents an analysis of the effects of non-linearities on the performance of linear AEC. This analysis is based on two experimental works based on a widely used non-linear model. In the first part we assess linear AEC in non-linear environments and then we compare the effects of non-linearities to the effects of noise. In the second part we present a new mathematical analysis of linear AEC behaviour in the presence of non-linearities. It is based on the assumption that non-linearities can be considered as correlated noise (correlated with the far-end signal). Based on this assumption we derive a Wiener solution of the echo path estimate and show that the presence of non-linearities degrades estimation of the linear echo path. We show that, due to the instability of the correlation between the non-linear echo component and the far-end speech signal, the estimate echo path effectively fluctuates around the optimal solution. We then show that the non-linear environment can be considered as a noise environment with a time variant EP by dividing the non-linear component into a correlated (far-end) component, which introduces fluctuation and

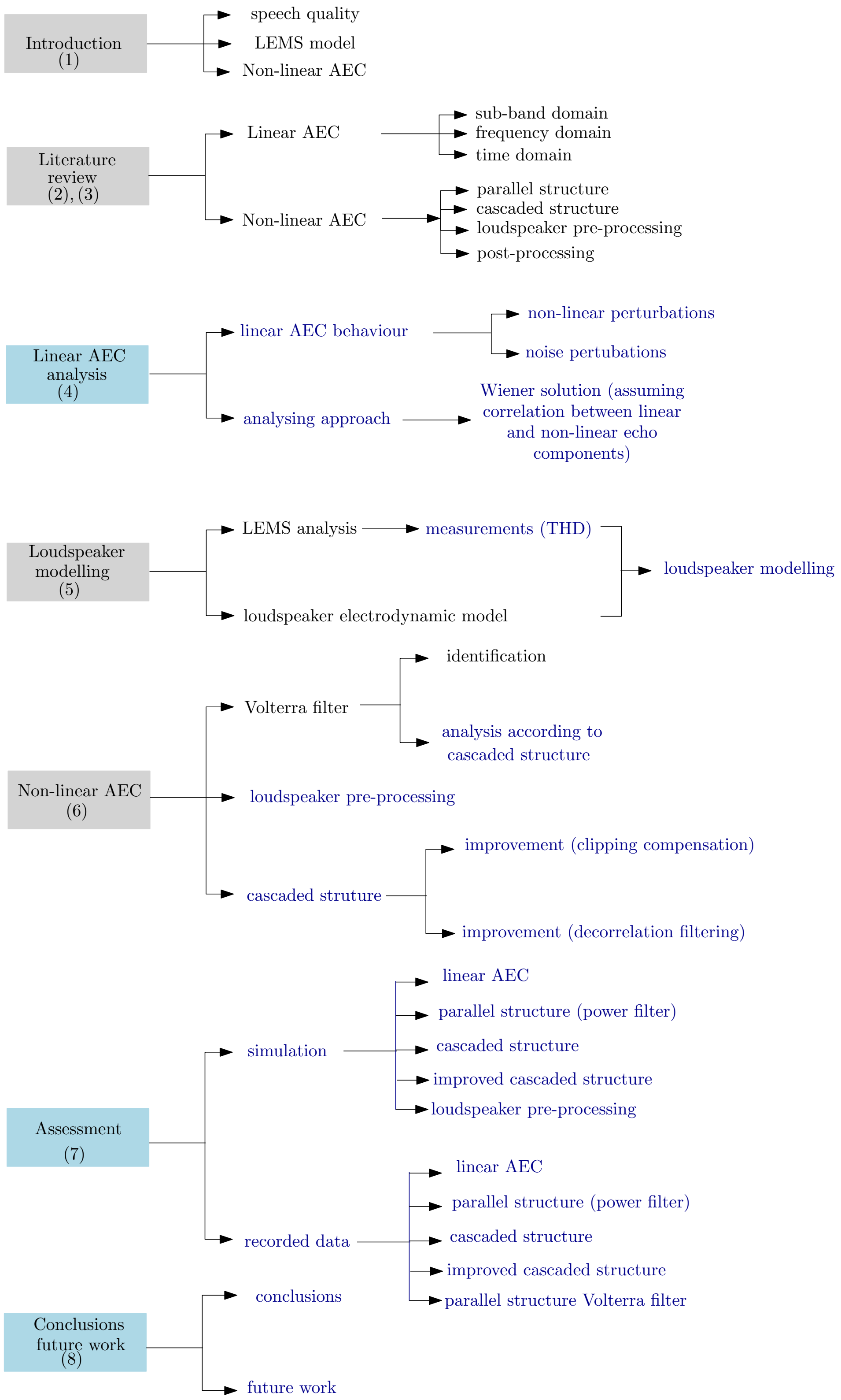


Figure 1.3: Thesis organization

time variability, and a non-correlated (far-end) component, which is considered as noise.

Since our analysis shows that linear AEC performances are degraded in non-linear environments, we propose in Chapter 5 new approaches to model non-linearity. We first present an analysis of the distortion introduced by the non-linear component using real device measurements. Since the loudspeaker is the main source of non-linearity, we present an electro-acoustic model of the loudspeaker. Based on a literature review of loudspeaker modelling and non-linear AEC we then introduce two new loudspeaker models: a time domain model based on cosine power expansion, and a frequency domain model based on harmonic estimation.

Chapter 6 reports non-linear adaptive filtering based on Volterra solution, cascaded structure and loudspeaker pre-processing where the two later solutions propose the use of a pre-processor based on the new loudspeaker model. The Volterra solution is presented here as the most widely used approach in non-linear AEC application and forms a baseline for reported experiments. Here we propose an analysis of the Volterra solution based on our conclusions on the LEMS in Chapter 5. This means that we assume a non-linear model of the loudspeaker and a linear model for the rest of the LEMS. We show that the Volterra quadratic kernel of the equivalent LEMS has a memory equal in length to that of the three paths (down-link path, acoustic channel and up-link path) but that the non-linearity memory does not change from that of the loudspeaker. This shows that the kernel contains a number of negligible taps which increase unnecessarily the complexity of the standard approach.

We propose in the second section a cascaded solution to non-linear AEC based on the time domain model of the loudspeaker developed in Chapter 5. The loudspeaker model is used as a pre-processor to emulate non-linearities introduced by the loudspeaker so that the following AEC is entirely linear. In this section we also discuss about local minima that affect cascaded structure.

In the third section we propose to improve the cascaded structure into two directions; an extension of the pre-processor model and the use of a decorrelation filter. The pre-processor model is extended to global loudspeaker and amplifier non-linearity compensation by incorporating a clipping compensator in the previous pre-processor. This allows the system to efficiently model clipping distortion that may arise in loudspeaker amplifier. We then use a decorrelation filter to reduce correlation in the speech signal in order to improve the convergence of the linear AEC.

The model developed in Chapter 5 is also used to linearise the loudspeaker in section four. This approach combines an on-line loudspeaker pre-processing and a linear AEC based on NLMS. It avoids introducing distortions in the microphone signal compared to parallel and cascaded non-linear AEC approaches and permits the use of conventional linear AEC.

In Chapter 7 we present an assessment of a linear AEC, a parallel structure, a cascaded structure and an improved cascaded structure. We first present an analysis based on a synthetized environment results then an analysis based on real recorded signals. In the synthetized environment analysis, all the algorithms parameters are chosen to fit with the model which is already known. A linear AEC, a parallel structure and a cascaded structure are used for a first assessment. The objective is mainly to show the behaviour of the different systems and their performance in terms of echo reduction and robustness to echo path changes. In the next step of the synthetized analysis, the decorrelation filtering procedure and clipping compensation combined to the cascaded structure are assessed and compared to the basic cascaded structure and parallel structure.

The loudspeaker pre-processing is then assessed with the linear AEC. Here the analysis of the system is based on echo reduction and linearisation performance. The objectives are to show that, with the loudspeaker pre-processing, a better echo reduction is achieved by conventional linear AEC and the output of the loudspeaker can also be efficiently linearised by the pre-processor.

In the second section, a smart-phone is used to record the data signals. The objective is to assess the tracking performance of the algorithms by changing the position of the mobile and generate non-linearities by applying a loud signal to the loudspeaker. These data are then used to assess the algorithms presented in the synthetized environment except the loudspeaker pre-processing which uses an online procedure. However, in this assessment as no a priori was made on the loudspeaker model we additionally analyse the behaviour of the Volterra filter.

Finally in Chapter 8 we present the conclusions and the perspectives. We explain the different steps of this work and provide some recommendations on the choice of a non-linear AEC structure regarding the environment characteristics. We have then make some propositions to improve non-linear acoustic echo cancellation.

Chapter 2

Linear AEC

This chapter presents different approaches developed in linear Acoustic Echo Cancellation (AEC) research field. We first present the general approach to acoustic echo cancellation in linear environment. We then introduce the Least Mean Square (LMS) algorithm which serves as basis for many adaptive filters used in AEC. Adaptive filtering algorithms which are developed to improve the LMS algorithm against the communication environment constraints are presented for the linear AEC applications. We decided to present the linear AEC in this work since they still widely use for AEC application due to stability and complexity reasons and above all many non-linear AEC approaches rely on algorithms developed for linear systems.

2.1 General approach

In this section we introduce the general approach to AEC. We first explain how the loudspeaker and the microphone environment (this environment is referred to as the Loudspeaker Enclosure Microphone System (LEMS)) can be approximated as a linear, time variant filter. We then introduce linear system identification approach used in AEC.

2.1.1 Linear modelling approach

In the linear approach the acoustical coupling between the loudspeaker and the microphone is assumed to constitute many acoustic reflections. The echo signal is simply the summation over all reflected paths. With this simplified approach we ignore any non-linearities that may be introduced by the amplifiers, the loudspeaker and the mobile terminal casing which corresponds to the perfect linear system. This model is illustrated in Figure 2.1 where the LEMS is assumed to be linear and represented by a linear system S^e with an impulse response $h(n)$. Hence, each reflected path is characterized by its delay τ and its attenuation $h(\tau)$. This can be modelled mathematically as:

$$d(t) = \int_0^\infty h(\tau)x(t-\tau)d\tau \tag{2.1}$$

where t indicates continuous time. Given that highly delayed paths incur high attenuation, and thus contribute relatively little in terms of echo, we may obtain a reasonably accurate model by performing the summation over a small, finite number

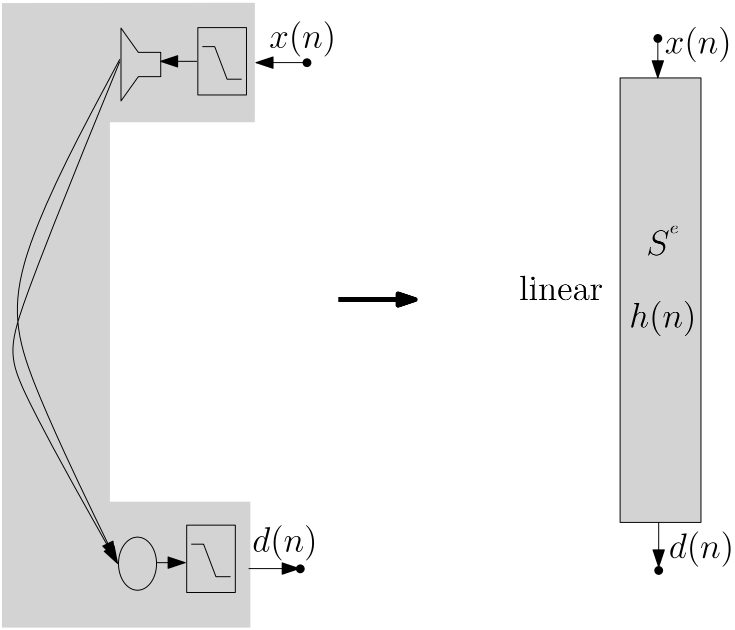


Figure 2.1: Linear LEMS model. The summed reflections (left) are modelled by the system S^e which has a linear impulse response, $h(n)$, (right) and the echo, $d(n)$, is equivalently the result of the convolution between the far end signal, $x(n)$, and the filter, $h(n)$.

of paths. As we work with discrete signals we can also discretize Equation 2.1 and, supposing only M ($h_i \approx 0$ for $i \geq M$) echo paths, we can write:

$$d(n) = \sum_{i=0}^{M-1} h_i x(n-i) \tag{2.2}$$

where i is a path index according to the delay which is a time discrete representation of τ in Equation 2.1. Hence $i = 0$ represents the first tap of h and h_0 the respective attenuation. In reality, though, when the speaker moves or a change arises in the LEMS (e.g. when a door in the room is opened) the coefficients h_i become time varying, so Equation 2.2 can be rewritten as [Hänsler & Schmidt 2004]:

$$d(n) = \sum_{i=0}^{M} h_i(n) x(n-i) \tag{2.3}$$

The LEMS is now modelled as a time varying filter, $h(n)$, so it becomes more important to have an idea of its characteristics which depend on many aspects of the environment, e.g. the materials coefficient of absorption. One important characteristic is the filter impulse response length which depends on the system sampling rate and the amount of time that the sound persists in the LEMS especially the acoustic channel. This is referred to as the reverberation time, which is defined as the

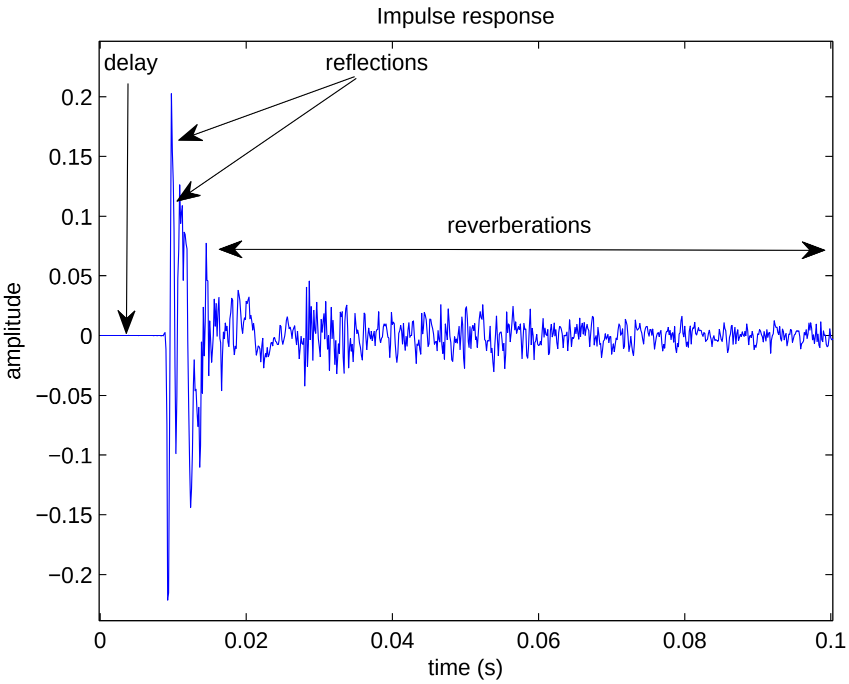


Figure 2.2: An example of LEMS impulse response. Illustrated are: the initial direct path delay, the first two dominant reflections and subsequent reverberation over a period of 0.1 s.

amount of time it takes for a sound to decay by 60 dB [Addington & Schodek 2005].

Figure 2.2 illustrates an example of LEMS impulse response, which can be divided into three parts as illustrated. The first part, where the level is close to zero, represents the direct-path delay between the loudspeaker and the microphone and is here in the order of 0.01 s. The second part is the most dominant and is composed of a high level coefficient that represents the first reflection at approximately 0.01 s and other smaller components for the second and the third reflections etc. The last part, with the smallest level, represents the most delayed reflections which are collectively referred to as reverberation. With a suitable LEMS model, the solution can be well formulated. This approach consists in identifying the filter impulse response, and is discussed in the identification section.

2.1.2 System identification

To mitigate the problem of echo, Acoustic Echo Cancellation (AEC) is often used. There is a wealth of relevant material in the literature and the general approach is illustrated in Figure 2.3. The AEC problem is viewed as one of system identification. The goal is to estimate the echo path $h(n)$ via an adaptive filter $\hat{h}(n)$ in order to synthesize an estimate of the echo signal, $\hat{d}(n)$. The estimate may then be subtracted

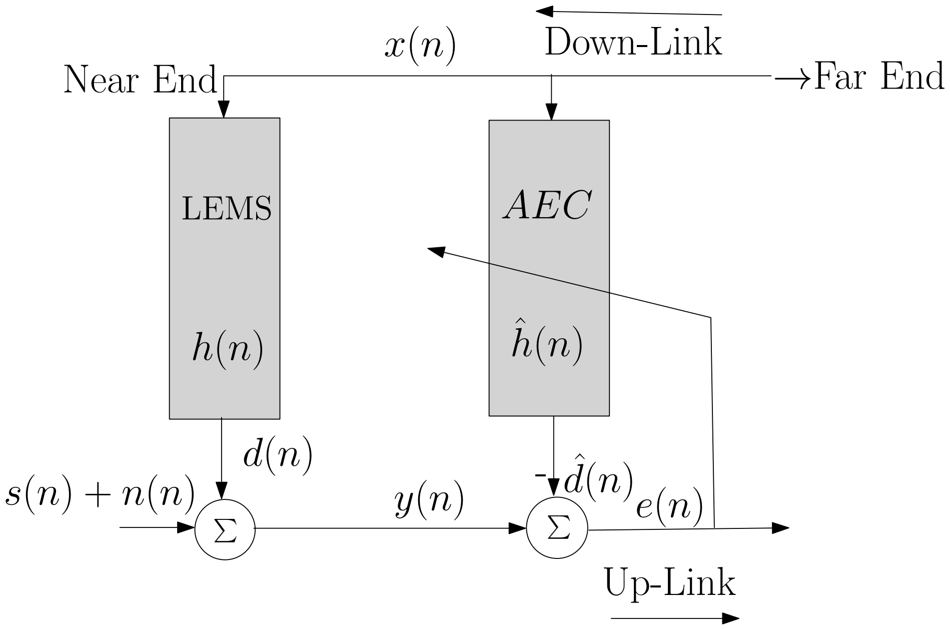


Figure 2.3: Concept of system identification in linear case

from the transmitted signal $y(n)$ which is the addition of the near-end speech signal $s(n)$, the echo component $d(n)$, and the noise $n(n)$. In so doing the echo in the Up-Link (UL) path is suppressed.

In the approach of system identification, the acoustic echo canceller tracks the time varying LEMS impulse response with the aim of creating a replica of the echo. In the ideal case the acoustic echo canceller maintains the same filter coefficients as the LEMS impulse response (if they were to have the same number of taps). Since the input of the AEC is the same as the output of the loudspeaker, the output of the acoustic echo canceller will thus be a perfect replica of the echo. Hence by subtracting the AEC output from $y(n)$, the echo component can be removed. To track the LEMS impulse response $h(n)$, system identification procedure generally relies on adaptive filtering approaches.

Adaptive filtering is an extremely important field of signal processing and there is a wealth of relevant material in the open literature. Figure 2.3 shows the procedure of the AEC using an adaptive filter. As new data $x(n)$ arrives the adaptive filter computes the error $e(n)$ between a reference signal $d(n)$ (echo in this case) and the output of the AEC $\hat{d}(n)$. This error is used to update the filter parameters $\hat{h}(n)$ according to certain criteria. In the next section the basic adaptive filtering algorithm known as LMS is presented then constraints in AEC application are provided. In general the echo signal $d(n)$ is corrupted by background noise $(n(n))$ and near-end signal $(s(n))$ but in the following calculations we assume a free noise environment $(n(n) = 0)$ and echo-only period $(s(n) = 0)$ for simplifications.

2.2 Least mean square algorithm

The least mean square (LMS) algorithm was proposed by Widrow and Hoff in 1960 [Haykin 2002] and has served as basis for many other adaptive filtering algorithms. In general it is derived using some approximation on the steepest descent algorithm which is an iterative method to reach the estimate of the Wiener solution. Here the steepest descent method is not presented but we give some necessary explanations necessary of the basis of the LMS algorithm. To derive the LMS algorithm we will choose an initial estimate $\hat{\mathbf{h}}(n)$ of $\mathbf{h}$ (assumed to be time invariant) at a certain time index n. This initial estimate is generally chosen to be a filter with all the taps equal to 0 and the initial index is generally 0 but, as the procedure is iterative we use the time index n. With this initial estimate $\hat{\mathbf{h}}(n)$, the echo estimate $\hat{d}(n)$ is computed as:

$$\hat{d}(n) = \hat{\mathbf{h}}^T(n)\mathbf{x}(n) \tag{2.4}$$

where $\mathbf{x}(n) = [x(n), x(n-1), \cdots, x(n-N+1)]$ is the input signal vector of length N. An error $e(n)$ is then computed as the difference between the echo signal $d(n)$ and its estimate $\hat{d}(n)$:

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

Equations (2.4) and (2.5) show that the square of the error is a quadratic function of the filter $\hat{\mathbf{h}}(n)$ and can be written as:

$$e^2(n) = d^2(n) - 2 \cdot d(n)\hat{\mathbf{h}}^T(n)\mathbf{x}(n) + \hat{\mathbf{h}}^T(n)\mathbf{x}(n)\mathbf{x}^T(n)\hat{\mathbf{h}}(n) \tag{2.6}$$

Figure 2.4 illustrates an example of a two-tap filter $\mathbf{h}(n)$, and shows the shape of the square error as a function of the two tap weights h_0 and h_1. The objective of the LMS is that $\hat{\mathbf{h}}(n)$ converges to the optimal filter $\mathbf{h}^0$ which is the filter that gives the minimum mean square error. Note that the paraboloid and the minimum square error $e^2_{min}(n)$ are time-dependent due to the variation of $\mathbf{x}^T(n)\mathbf{x}(n)$. Here the optimal or Wiener solution is the filter $\mathbf{h}^0$ which, in ideal case, satisfies $E\{e^2(n, h^0)\} = e^2_{min}$ (e^2_{min} is the minimum square error overall the process) for a signal $x(n)$ whereas the true filter $\mathbf{h}$ is the solution which gives zero error ($e(n, h) = 0$) whatever $x(n)$. The difference between $\mathbf{h}^0$ and $\mathbf{h}$ arises due to the fact that $\mathbf{h}$ is assumed to be a real filter whereas $\mathbf{h}^0$ is an estimate which generally depends on the excitation signal $x(n)$. Thus the optimal solution $\mathbf{h}^0$ is the best estimate which may be reached by $\hat{\mathbf{h}}(n)$ and is in general sufficiently accurate if the characteristics of the signal $x(n)$ do not change too much.

From Figure 2.4 we observe that the minimum square error $e^2_{min}(n)$ is reached when the derivative of the square error with respect to $\hat{\mathbf{h}}(n)$ is equal to zero. As the square error is quadratic in $\hat{\mathbf{h}}(n)$ all the derivatives of the square error with respect to $\hat{\mathbf{h}}(n)$ will point in the increasing direction of the error (under the assumption that $\mathbf{x}^T(n)\mathbf{x}(n)$ is positive-definite which is generally the case for speech signal) so that taking the opposite direction will lead towards the optimal filter. Hence the LMS algorithm consists of updating with each new sample, $x(n)$, the current filter

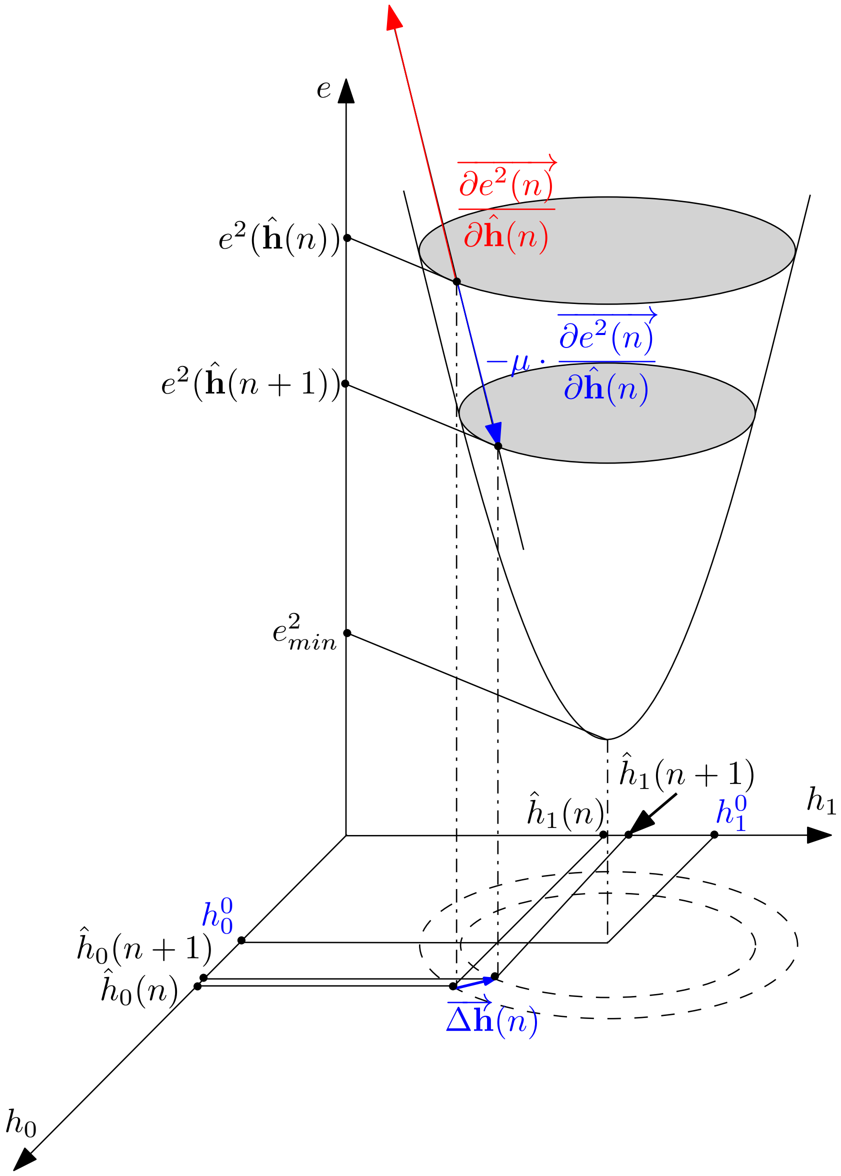

e
$\overrightarrow{\frac{\partial e^2(n)}{\partial \hat{\mathbf{h}}(n)}}$
$e^2(\hat{\mathbf{h}}(n))$
$-\mu \cdot \overrightarrow{\frac{\partial e^2(n)}{\partial \hat{\mathbf{h}}(n)}}$
$e^2(\hat{\mathbf{h}}(n+1))$
e^2_{min}
$\hat{h}_1(n+1)$
$\hat{h}_1(n)$
h_1^0
h_1
h_0^0
$\hat{h}_0(n+1)$
$\hat{h}_0(n)$
$\overrightarrow{\Delta \hat{\mathbf{h}}}(n)$
h_0

Figure 2.4: Illustration of the iterative square error minimization. Example of the square error shape with a two-tap optimal filter ($\mathbf{h}^{0} = [h_1^0, h_2^0]$). The red arrow points in the direction of the derivative of the square error w.r.t $\hat{\mathbf{h}}(n)$ whereas the blue one points the opposite direction which leads to the minimum square error e^2_{min}.

estimate $\hat{\mathbf{h}}(n)$ in the opposite direction of the derivative of the square error $e^2(n)$ w.r.t the filter $\hat{\mathbf{h}}(n)$. This can be formulated mathematically as:

$$\hat{\mathbf{h}}(n+1) = \hat{\mathbf{h}}(n) - \frac{\mu}{2}\frac{\partial e^2(n)}{\partial \hat{\mathbf{h}}(n)} \tag{2.7}$$

where μ is a step-size used to control the adaptation rate which is divided by two for simplification (see the final Equation 2.9). Equation 2.7 is illustrated in Figure 2.4 in which we observe that the derivative of the square error points in the increasing direction of the square error (red arrow) and the opposite direction, scaled by μ, points in the direction of the minimum square error $e^2_{min}(n)$ (blue arrow).

We therefore required to compute the derivative of the square error with respect to the filter $\hat{\mathbf{h}}(n)$ which is given by:

$$\begin{aligned}
\frac{\partial e^2(n)}{\partial \hat{\mathbf{h}}(n)} &= \frac{2e(n)\partial e(n)}{\partial \hat{\mathbf{h}}(n)} \\
&= \frac{2e(n)\partial(d(n) - \hat{\mathbf{h}}^T(n)\mathbf{x}(n))}{\partial \hat{\mathbf{h}}(n)} \\
&= 2e(n)\Big(\underbrace{\frac{\partial d(n)}{\partial \hat{\mathbf{h}}(n)}}_{=0} - \frac{\partial \hat{\mathbf{h}}^T(n)\mathbf{x}(n)}{\partial \hat{\mathbf{h}}(n)}\Big) \\
&= -2e(n)\mathbf{x}(n)
\end{aligned} \tag{2.8}$$

Combining Equations 2.7 and 2.8 we obtain the LMS algorithm given as in [Haykin 2002]:

$$\hat{\mathbf{h}}(n+1) = \hat{\mathbf{h}}(n) + \mu e(n)\mathbf{x}(n) \tag{2.9}$$

Equation 2.9 is the basic form of LMS and most of the Minimum Mean Square Error (MMSE)-based adaptive algorithms are derived from it.

From Figure 2.4 we see that if the step-size μ is too large the next estimate $\hat{\mathbf{h}}(n+1)$ may be farther from $\mathbf{h}^0$. This should be avoided and requires the study of stability regarding the step-size μ in Equation 2.9. Note that in general when used in terms of adaptive filtering stability refers to the conditions which permit the algorithm to converge to the optimal filter. In the same way, if the algorithm diverges from the optimal point, we refer to instability.

In the LMS algorithm the conditions for convergence relate to the step-size μ. In Figure 2.4 we observe that, if μ is high the optimal value $\mathbf{h}^0$ can be reached quickly, but if it is too high $\hat{\mathbf{h}}(n+1)$ can be far from the optimal point. It is thus necessary to strike a compromise between fast convergence and stability. Under the assumption of independent and identically distributed (i.i.d) inputs samples conditions for stability can be shown to relate to μ [Haykin 2002] such that:

$$0 < \mu < 2 \cdot \lambda_{max} \tag{2.10}$$

where λ_{max} is the maximum eigenvalue of the auto-correlation matrix of the input signal $x(n)$. In fact the LMS algorithm is unlikely to reach the exact optimum value and generally fluctuates around it instead. The error due to this fluctuation is proportional to the step-size μ so that, the bigger μ, the faster the adaptation but the bigger is the fluctuation error around the optimal solution. For this reason, adaptive μ values have been investigated in adaptive filtering. The main task is to find an adaptive μ which, at the beginning of the process is high and satisfies the stability condition then becomes smaller when $\hat{\mathbf{h}}(n)$ is close to $\mathbf{h}^0$. This task is easier in the general study of adaptive filtering but in AEC it is a challenge due to different constraints that are explained next.

2.3 Adaptive filtering constraints for AEC

Before the various approaches to adaptive filtering are presented in Section 2.4 some parameters that influence the adaptive filtering in AEC approaches are presented in this section. The constraints described below are the common guiding mechanisms involved with adaptive filtering for AEC.

2.3.1 Speech signal characteristics

When applied to speech signals adaptive filtering introduces two problems that are well described in the AEC literature [Hänsler & Schmidt 2004, Vary & Martin 2006]: the non-stationarity and eigenvalue spread of the speech signal. The non-stationarity of the speech signal arises from the fact that to generate the different phonemes the vocal track is time variant. This variation of the vocal track leads to a variation in the speech signal spectra. The speech signal can be approximated as the output of a time varying process to noise or impulse excitation input [Vary & Martin 2006] which induces its non-stationarity. Even with the slow variation of the vocal track, the speech signal may still be considered as short-term stationary. The non-stationarity has as effect in adaptive filtering to make the estimate fluctuate around a mean so that good tracking behaviour is required. In fact, under stationary condition, the term $\mathbf{x}^T(n)\mathbf{x}(n)$ will iteratively approximate $E\{\mathbf{x}^T(n)\mathbf{x}(n)\}$, which is time independent, but under non-stationary conditions it will be time dependent. According to Equation 2.6 this will cause the minimum square error $e^2_{min}(n)$ to fluctuate around a mean point so that no stable optimum can be reached. It has been shown that in these situations the LMS algorithm introduces an additional amount of error to the minimum reachable error obtained in the stationary environment [Widrow 1966].

The vocal track filter also introduces a correlation in the speech signal which reduces the convergence. The speech signal spectrum shows higher energy in lower frequencies and very low energy in higher frequencies [Hänsler & Schmidt 2004, Vary & Martin 2006] meaning that the speech signal auto-correlation has spread eigenvalues. It has been shown that the convergence rate is related to the ratio of the maximum and minimum eigenvalue [Haykin 2002] and the more spread the eigenvalues are the slower the adaptation. This is shown in Figure 2.4 where we observe that $\hat{h}_1(n)$ converges faster to h_1^0 than $\hat{h}_0(n)$ does to h_0^0. This is explained by the fact that the projection of the vector $\overrightarrow{\Delta h(n)}$ in Figure 2.4 onto the $h_1(n)$ axis is greater in magnitude than that on $h_0(n)$. If in one step the component of $\overrightarrow{\Delta h(n)}$ on h_1 is too big it may rise above h_1^0. The step may be small, however, for $\hat{h}_0$ to reach its optimal position h_0^0. This shows that, in this case, the stability condition is more dependent on h_1 (which in this case corresponds to the λ_{max} in Equation 2.10) than h_0. Hence whereas $\hat{h}_1$ can quickly converge to its optimal position h_1^0, $\hat{h}_0$ is constrained to slowly converge to its optimal position to avoid overshooting h_1^0. If the eigenvalue is not spread, as in the case for white noise, at each step the projection of $\overrightarrow{\Delta h(n)}$ onto the different axes will be similar so that all the parameters will converge toward their optimal positions at the same rate and none of them will impose a dominant influence. This is one of the reasons which motivate frequency domain adaptive filtering where each frequency bin of the filter $\hat{h}(n)$ can be parametrized independently from the others.

The speech signal is not the only factor which affects the LMS algorithm. To take full advantage of adaptive filtering it is necessary to appreciate the effect of echo path changes and the presence of noise and near-end speech which can be well handled with efficient parametrization.

2.3.2 Acoustic echo path variability

One problem inherent to AEC is the time variability of the Echo Path (EP). In fact the EP is dependent of the environment characteristics, e.g. room size. This environment is generally not static in mobile communication or teleconferencing due to speaker movement or changes in the environment such as opened doors or windows. A difficult situation arises when the position of the loudspeaker relative to the microphone changes (generally does not arise as the relative position between the loudspeaker and the microphone is fixed). This will introduce delay changes meaning that the position of the direct path also changes. As the direct path corresponds in general to the most significant filter tap of $\mathbf{h}(n)$ the adaptive filter will be perturbed and takes longer to re-converge. The effects of Echo Path Change (EPC) are also relative to the sampling frequency of the Digital Signal Processor (DSP). If the relative distance between the microphone and the loudspeaker changes by δd cm then the delay of the direct path will change by a number of samples equal to $N_d = \delta d \frac{f_s}{v_s}$ where f_s is the sampling frequency and v_s is the sound velocity. With a sampling frequency of 8 kHz and sound velocity of 341 m/s, a relative distance $\delta d = 4.2$ cm will result in a delay change of one sample [Breining *et al.* 1999]. This further complicates the tracking problem of the acoustic path to those problems already introduced by the speech signal characteristics. In [Van de Kerkhof & Kitzen 1992] it has been shown that with longer filters the LMS based adaptive filter is not able to track EPC introduced by a moving speaker.

The solutions proposed are in general faster adaptive filters which do not always solve the tracking problem. In fact the tracking problem introduced by EPC interacts with that introduced by speech signal non-stationarity. The echo path variability changes the parameters of the optimal filter $\mathbf{h}^0(n)$, which becomes time dependent, whereas the speech non-stationarity affects the estimator (by affecting the position of the minimum square error) which results in a more complex tracking problem. These effects introduce non-stationarity but to use adaptive filtering approach it is assumed that speech non-stationarity arises slowly so that the speech signal is considered as short-term stationary whereas EPC are expected to be smooth so that the difference is easily tracked by the adaptive filter. Abrupt echo path changes are also assumed to arise less often than for speech characteristics changes so that a fast adaptive filter may track the EP. Two solutions are used to solve the EPC problem and are either faster adaptive filters or better approaches to control the adaptation step. The latter increases the step-size when an EPC is detected and reduces the step-size when the adaptive filter has converged.

The length of the acoustic path is also critical in AEC as it is necessary to have enough taps in the adaptive filter to reach a good estimate of the echo. In general this length depends on the environment of the near-end speaker. The length of the acoustic path is relative to the reverberation time of the environment. Without an idea of the specific environment it is better to use a long filter and in general the acoustic path is between 100 taps to 2000 taps. Overly long filters increase the computational complexity and convergence time of the AEC, and may pose instability problems. These two factors are more difficult to manage in mobile applications where the amount of memory is limited and real-time performance is required.

2.3.3 Background noise and Double Talk

In general the echo signal used as a reference by the adaptive filter is corrupted by noise. The presence of noise at low levels is not problematic but at significant levels the performance of the AEC can be reduced and under such conditions a solution should be found. The second effect that may be considered as noise for AEC is the presence of a near-end speech signal. The presence of near-end speech is considered as high level noise for the AEC and, if no control is applied, it may diverge from its optimal point. The simultaneous presence of near-end and far-end signals is referred to as Double Talk (DT). The difficulty with DT is due to the fact that both speech signals have similar characteristics and lead to significant correlation in short-time processes between near-end and far-end signals so that the adaptive filter will be highly perturbed in such periods. Compared to the previous problems (speech signal and echo path characteristics) these do not in general require the development of new algorithms but are solved by controlling the AEC. In general it consists of using an estimate of the background noise to slow down the adaptation in higher noise. A Double Talk Detector (DTD) can be used to detect the presence of near-end speech so that adaptation is paused during periods of DT. This poses a real problem of step-size estimation as it becomes dependent on the noise estimation, near-end speech detection and EPC detection. For each of these effects a different decision must be taken. This explains the large amount of papers which focus on the estimation of the optimal step-size [Hänsler & Schmidt 2004, Vary & Martin 2006].

2.4 Linear AEC approaches

Many solutions have been proposed to improve the basic AEC approach which is based on the LMS algorithm. These efforts can be categorized into three general approaches: optimal parametrization of the LMS algorithm, new adaptive filters to improve performance and complexity reduction of the AEC.

The first approach aims to maximize the performance of the AEC algorithm according to the environment. This involves finding the optimal step-size to drive the algorithm combining DTD, EPC detection and noise level estimation. The second approach is based on new adaptive algorithms which may provide better performance compared to the baseline LMS algorithm. The third approach addresses the problem of complexity which is critical for mobile communications due to limited computational power and memory with the additional real time process requirement.

Under the constraints of AEC the LMS algorithm is not very efficient. Hence many solutions have been proposed to improve the convergence rate and complexity. In general a compromise should be done between the two. Here we present some popular solutions. Most are well developed in the AEC literature so more details can be found in [Hänsler & Schmidt 2004, Vary & Martin 2006] especially in AEC application and more generally [Haykin 2002, Farhang-Boroujeny 1998, Sayed 2008] for adaptive filtering. Here a brief description is given and we focus on the advantages and disadvantages of the different approaches. To avoid the repetition of certain equations we will use the basic formulation of the adaptive filtering algorithm:

$$\hat{\mathbf{h}}(n+1) = \hat{\mathbf{h}}(n) + \Delta\mathbf{h}(n) \tag{2.11}$$

where $\Delta\mathbf{h}(n)$ is the gradient which generally encapsulates the difference between one algorithm and another. Referring to Equation 2.9 the gradient $\Delta\mathbf{h}(n)$ is equal to $\mu e(n)\mathbf{x}(n)$ for the LMS algorithm.

2.4.1 Normalized-LMS algorithm

The stability of the LMS algorithm depends on μ which is bounded by the maximum eigenvalue. Since the input vector is time varying, it is difficult to choose a single, fixed value of μ that ensures consistent performance. One solution is to choose a small value of μ such that stability is assured for any input $\mathbf{x}(n)$. However, at the expense of increased stability comes slower convergence, and vice versa. A solution to the problem is a normalization of the set-size μ that adapts automatically according to the norm of the input vector. This normalization is obtained by bounding λ_{max} in Equation 2.10 by the trace of the matrix $\mathbf{x}^T(n)\mathbf{x}(n)$ which corresponds to the norm of the input vector. The new algorithm is referred to as the Normalized-LMS (NLMS) algorithm and its gradient is given by [Haykin 2002]:

$$\Delta\mathbf{h}(n) = \frac{\mu_n}{\|\mathbf{x}(n)\|^2 + \xi} \times \mathbf{x}(n) \times e(n), \tag{2.12}$$

where ξ is a small value used to avoid dividing by zero. The stability condition is given by [Haykin 2002]:

$$0 < \mu_n \leq 2. \tag{2.13}$$

The NLMS algorithm is thus equivalent to the LMS algorithm with a normalized step size. The advantage of the NLMS algorithm is the reduced sensitivity to the norm of the input vector and thus a general increase in stability and convergence. Due to its low complexity NLMS is one of the most popular LMS variants.

Even if the NLMS algorithm has better performance than the LMS, it does not bring a solution to the problem posed by the speech signal and echo path characteristics. But, due to its stability, it has been used as a basis for many other solutions. Three directions have been taken to address speech signal and echo path characteristics; the use of new algorithms such as Adaptive Projection Algorithm (APA) and Recursive Least Square (RLS), input signal pre-whitening or decorrelation, and sparse domain adaptive filters. The first category corresponds to adaptive filtering that is less affected by speech correlation. The second approach is based on a decorrelation filtering to reduce speech signal correlation before applying the NLMS algorithm. The third solution assumes that the acoustic path is sparse and uses a tap dependent step-size to increase the adaptation of the most significant taps in the acoustic path. These solutions are also implemented in different domains such as frequency or sub-band. The most widely used algorithm in these different domains is still the NLMS filter due to stability reason and computational efficiency.

2.4.2 Affine projection algorithm

The Affine Projection Algorithm (APA) is a generalization of the NLMS algorithm. It is based on a gradient estimate which takes into account the error in the previous input vector according to the filter. Hence, for each iteration, a vector of new estimates is computed using previous data convolved with the current filter estimate which can be written as [Haykin 2002]:

$$\hat{d}_n(n-k) = \hat{\mathbf{h}}^T(n)\mathbf{x}(n-k) \text{ for } k = 0, 1, ..., K-1 \tag{2.14}$$

where the subscript n of $\hat{d}_n(n-k)$ indicates that it is an estimate of $d(n-k)$ at time index n with the filter $\hat{\mathbf{h}}(n)$ and where K is the order of the APA algorithm. This leads to an error vector given by $e(n-k) = d(n-k) - \hat{d}_n(n-k)$. Minimizing the error vector leads to a more general version of the NLMS algorithm with a gradient given by:

$$\Delta\mathbf{h}(n) = \mu\mathbf{X}^T(n)\left(\mathbf{X}(n)\mathbf{X}^T(n) + \xi\mathbf{I}_N\right)^{-1}\mathbf{e}(n) \tag{2.15}$$

where the input matrix $\mathbf{X}(n)$ is equal to $[\mathbf{x}(n), \mathbf{x}(n-1), \cdots, \mathbf{x}(n-K)]$ and where $\mathbf{I}_K$ is the identity matrix of dimension $K \times K$. As in the NLMS algorithm ξ is a regularization factor.

Equation 2.15 represents the APA algorithm gradient and $\mathbf{X}(n)\mathbf{X}^T(n) + \xi\mathbf{I}_K$ serves as in the NLMS for normalization. We see that if $K = 1$ the APA algorithm is equivalent to NLMS.

With these additional constraints APA converges faster than NLMS. The higher K (order) the faster the convergence, but it has been shown that the more the order increases, the less significant is the improvement [Breining *et al.* 1999]. This gain of performance results in an increase in computational complexity and memory requirements. This is due to the matrix inversion with a dimension equal to the order (K) of the APA and also the computation of $K-1$ estimates of $\hat{d}_n(n-k)$ that requires to put in memory previous samples of $x(n)$ and $d(n)$. The computation of each estimate $\hat{d}_n(n-k)$ requires a convolution of the filter and the corresponding vector. To reduce the complexity of APA, fast versions have been introduced. The Fast Adaptive Projection Algorithm (FAPA) are based on fast convolution in Fourier domain or block convolution [Haykin 2002, Tanaka *et al.* 1999]. These fast versions are subject to instability, however. Hence, when these fast versions are applied the regularization of the APA is difficult and may lead to some instability [Haykin 2002]. Some solutions have been proposed in [Ding 2000, Chen *et al.* 2006, Challa *et al.* 2007] regarding the normalization of the FAPA. Another FAPA using an approximation of the present error vector $\mathbf{e}(n)$ according to the previous one $\mathbf{e}(n-1)$ ($e_n(n-k) \approx (1-\mu)e_{n-1}((n-1)-(k-1))$) is proposed in [Gay & Tavathia 1995]. This avoids the computation of the convolution between previous inputs and current filter taps but, at ultimately end, it can also result in instability.

Additionally, the APA has the drawback to introduce more perturbation compared to the NLMS algorithm during EPC periods. In fact, minimizing the previous error according to the current echo path estimate assumes that h^0 is static. Hence, if an EPC arises the APA will be perturbed during this period. This is due to the memory of the system which corresponds to the APA order K. A solution to reduce this effect is to use a lower order but on the other hand, this will increase the convergence time. Another drawback of the APA that has been shown in [Yamada *et al.* 2002] is its inefficiency in noisy environments. To demonstrate this inefficiency, in [Yamada *et al.* 2002] APA was formulated using Projection Onto Convex Set (POCS) theory and they propose an algorithm based on the Adaptive Sub-gradient Projected Method (ASPM) as a solution. In this solution, instead of assuming that $d(n-k) - \hat{d}_n(n-k)$ should be equal to zero, which is the target in APA, they take into account the presence of noise to bound the difference as $\left\| d(n-k) - \hat{d}_n(n-k) \right\| < \rho$ where ρ is related to the level of noise.

2.4.3 Recursive least square algorithm

The RLS algorithm is different from those described previously which are based on the MMSE criteria. The RLS algorithm is derived from the Least Square (LS) criteria where the cost function is given by:

$$J(n) = \sum_{l=0}^{L} \lambda^{n-l} e^2(n-l) \tag{2.16}$$

Here the cost function is given by the weighting of the square of the errors. The weighting parameter λ ensures that, when $\lambda < 1$, the more the error is in the past then the less important it is in the cost function. Based on this cost function the RLS filter can be derived by its minimization and leads to a gradient given by [Haykin 2002, Hänsler & Schmidt 2004]:

$$\Delta\mathbf{h}(n) = \mathbf{g}(n)e(n|n-1) \tag{2.17}$$

where $g(n)$ is the RLS gain factor given by:

$$g(n) = \frac{\lambda^{-1}\hat{\mathbf{R}}_{xx}^{-1}(n-1)\mathbf{x}(n)}{1+\lambda^{-1}\mathbf{x}^T(n)\hat{\mathbf{R}}_{xx}^{-1}(n-1)\mathbf{x}(n)} \tag{2.18}$$

with $\hat{\mathbf{R}}_{xx}^{-1}(n)$ computed as:

$$\hat{\mathbf{R}}_{xx}^{-1}(n) = \lambda^{-1}\hat{\mathbf{R}}_{xx}^{-1}(n-1) - \lambda^{-1}g(n)\mathbf{x}^T(n)\hat{\mathbf{R}}_{xx}^{-1}(n-1) \tag{2.19}$$

and where the error is given by:

$$e(n|n-1) = d(n) - \hat{\mathbf{h}}^T(n-1)\mathbf{x}(n) \tag{2.20}$$

This is the general approach used for the RLS algorithm even if the basic approach does not require the iterative computation of $\hat{\mathbf{R}}_{xx}^{-1}(n)$. The recursive estimation of $\hat{\mathbf{R}}_{xx}^{-1}(n)$ is used for complexity reduction.

Regarding the cost function in Equation 2.16, the closer λ is to 1 the more previous errors are taken into account. RLS therefore uses memory just like APA. The effect of this memory, which is longer when λ is close to one, helps improving convergence and stability but on the other hand reduces tracking performance [Hänsler & Schmidt 2004, Haykin 2002]. If λ is close to 1, during periods when the speech signal is not consistent the algorithm will not be much perturbed but during EPC it will require more time to re-converge. This explains the compromise between fast convergence, stability and tracking capability when using the RLS. To solve the tracking problem of RLS a variant has been proposed. Extended RLS (E-RLS) in [Haykin *et al.* 1997] takes advantage of the duality between RLS and Kalman filtering to improve the tracking capability.

Another drawback of this algorithm is the computational complexity. Hence, Fast RLS (FRLS) has been proposed to reduce complexity, these solutions are in general based on lattice algorithm and the computationally efficient Fast Transversal Filter (FTF) proposed in [Cioffi & Kailath 1984]. However, the latter solution suffers from instability with fixed-point processors. Other solutions have been developed to solve the problem of instability [Slock & Kailath 1988, Houacine 1991, Benesty & Gansler 2001, Callender & Cowan 1990, Arezki *et al.* 2006]. In [Slock & Kailath 1988] after analysing error propagation in the FTF based RLS, they propose to introduce some redundancy in the feedback to stabilize the algorithm.

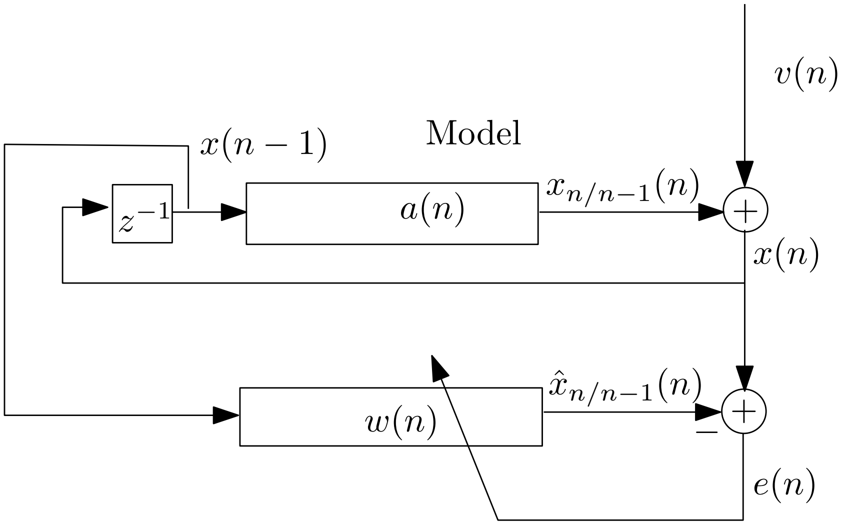


Figure 2.5: AR model of a speech signal and its prediction filter estimation using an adaptive filter. Note that, this is a normalized version fo AR model which is used to illustrate the decorrelation procedure since in general representation $v(n)$ is multiplied by the first coefficient of $a(n)$.

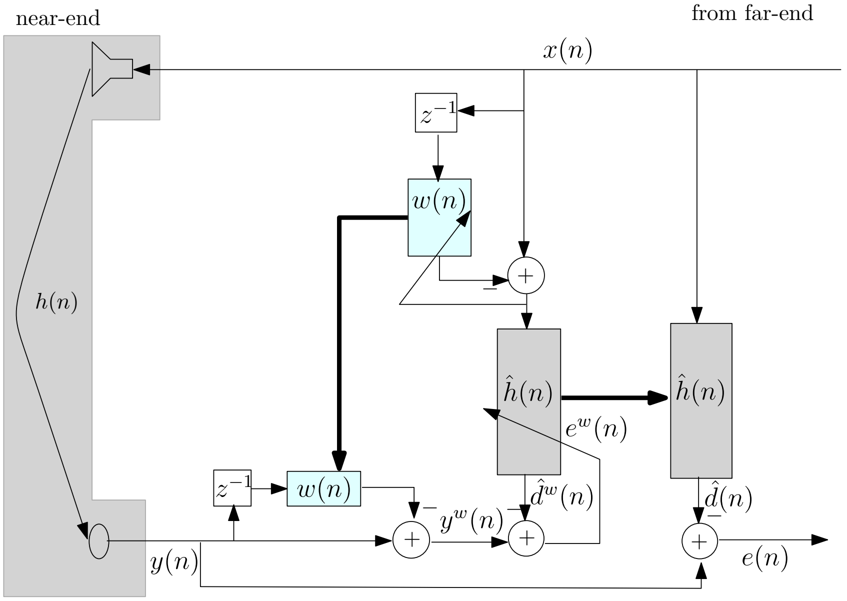


Figure 2.6: AEC using an adaptive predictor

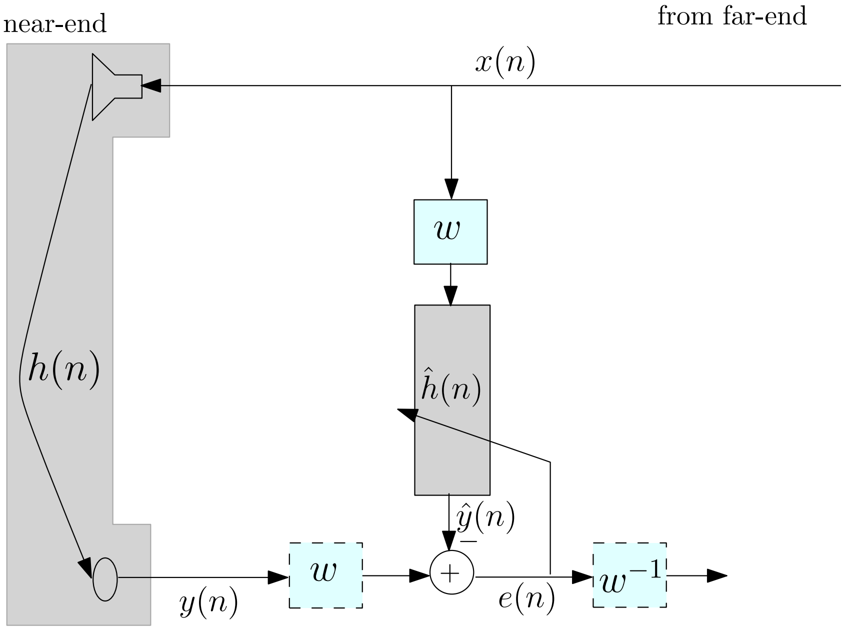


Figure 2.7: AEC using fixed predictor. The error $e(n)$ is reconstructed with the inverse filter of the predictor.

2.4.4 Normalized-LMS with decorrelation filtering

It is well known that the vocal track can be modelled using a filter. An Autoregressive (AR) model of the vocal track and its adaptive estimation is illustrated in Figure 2.5. The filter $\mathbf{a}(n)$ models the vocal track and is assumed to be slowly variable and fluctuates around a mean value. If $\mathbf{w}(n)$ is an efficient estimate of $\mathbf{a}(n)$ then the error $e(n)$ will be close to a white signal with less eigenvalue spread than the original signal $x(n)$. Hence, if $e(n)$ is used as input to the NLMS algorithm it results in faster convergence than with $x(n)$. If a decorrelation filter is applied only to the far-end signal then the estimate $\hat{\mathbf{h}}(n)$ will be perturbed and, accordingly the same filter is applied to the far-end signal as well [Breining *et al.* 1999, Hänsler & Schmidt 2004]. The two widely used approaches are illustrated in Figure 2.6 and 2.7. In Figure 2.6 the adaptive decorrelation filter procedure is used. This decorrelation filtering is generally based on the Levinson algorithm or with a simple NLMS filter [Frenzel & Hennecke 1992, Yasukana *et al.* 1988, Mboup *et al.* 1992, Mboup *et al.* 1994]. Figure 2.7 shows the fixed decorrelation filtering approach. In fact, due to speech non-stationarity, adaptive decorrelation filtering provides better performance than fixed filtering as it can follow the vocal track variations. On the other hand the fixed decorrelator is less complexity demanding by using the inverse filter $\mathbf{w}^{(-1)}$ $(w(n) * w^{(-1)}(n) = \delta(n))$. The inverse filter is not always easy to estimate but it is shown in [Haykin 2002] that $\mathbf{w}$ is a minimum phase filter so the estimation of $\mathbf{w}^{(-1)}$ is entirely feasible [Breining *et al.* 1999]. Another fixed decorrelation system has been proposed in the subband domain and uses the predictor filter on the near-end signal and the error signal. This simplifies the system but requires that the inverse filter to be estimated by the AEC filter. A comparison of complexity and performance of different approaches to decorrelation filtering is proposed in [Rupp 1993]. It has been shown that better improvement can be obtained with less taps and in general the maximum order of the decorrelation is about 20 taps [Breining *et al.* 1999].

2.4.5 Sparse adaptive filtering

Sparse adaptive filter is based on a fast adaptation of the most significant taps in the acoustic path. In general they are dedicated to electrical echo cancellation where the condition of sparsity is satisfied [Paleologu *et al.* 2010], but it has been shown in [Loganathan *et al.* 2011] that the first samples of the acoustic path can be assumed as sparse and might be well estimated using a sparse adaptive filter. The most well-known sparse adaptive filter is the Proportionate NLMS (PNLMS) algorithm introduced in [Duttweiler 2000] where the gradient is given by:

$$\Delta\mathbf{h}(n) = \mu\frac{\mathbf{G}(n-1)\mathbf{x}(n)e(n)}{\mathbf{x}^T(n)\mathbf{G}(n-1)\mathbf{x}(n)+\xi} \tag{2.21}$$

where $\mathbf{G}(n-1)$ is given as:

$$\mathbf{G}(n-1) = diag\{g_0(n-1), g_1(n-1), \cdots, g_{N-1}(n-1)\} \tag{2.22}$$

and where the diagonal elements $g_l(n)$ are given as:

$$g_l(n) = \frac{\gamma_l(n)}{\frac{1}{L}\sum_{l=0}^{L-1}\gamma_l(n)} \tag{2.23}$$

with:

$$\gamma_l = \{\rho max[\delta, |\hat{h}_0|, |\hat{h}_1|, \cdots, |\hat{h}_{N-1}|], |\hat{h}_l|\} \tag{2.24}$$

where ρ and δ are some small values that are typically 0.001. ρ is used to avoid stalling of small taps and δ is used for regularization [Paleologu *et al.* 2010]. We remark that, if $\mathbf{G}(n-1)$ is equal to the identity matrix, we obtain the gradient of the NLMS algorithm in Equation 2.21. This solution provides better performance but has the drawback of being more complex compared to the NLMS algorithm and may also result in lower performance when the LEMS is not sparse. This leads to the development of many other algorithms for sparse system identification, the most popular of which is the Improved PNLMS (IPNLMS) algorithm [Benesty & Gay 2002] which may have comparable performance to NLMS even for dispersive systems. The extension of the PNLMS to the APA has also been proposed as Proportionate APA (PAPA) [Gansler *et al.* 2000]. The disadvantage of these algorithms is that they require an a priori on the LEMS, sparsity otherwise they just increase complexity.

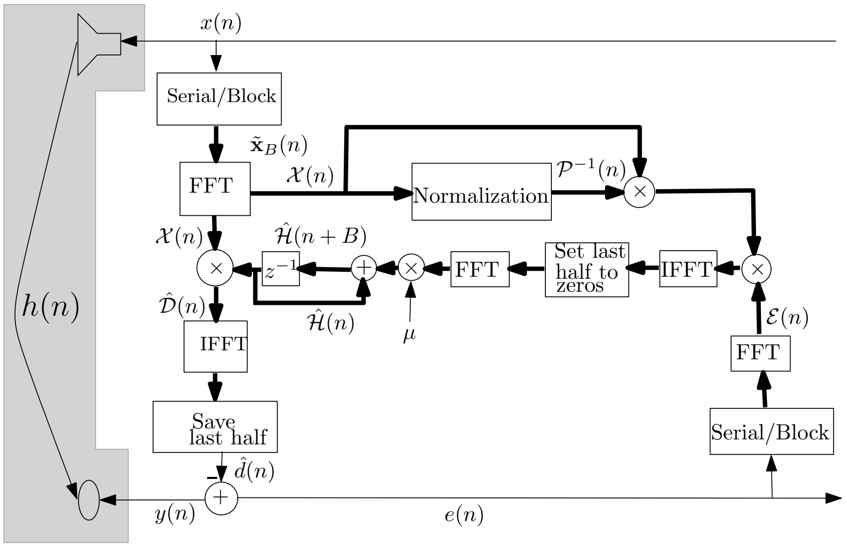


Figure 2.8: Frequency Block LMS (FBLMS) process.

2.4.6 Frequency domain approaches

The aim in block processing is to reduce the complexity of the LMS algorithm [Hänsler & Schmidt 2004]. Block processing uses a block of B input and output samples per iteration. The filter is updated every B samples and can significantly reduce the computational complexity. The disadvantage in block processing schemes is the control of parameters. All parameters are controlled block-by-block instead of sample-by-sample as with the NLMS algorithm which can be seen as reducing the time resolution [Hänsler & Schmidt 2004]. This resolution is important when the AEC algorithm needs to be controlled by a DTD. This is not efficient if the block size is too large. Block adaptive filtering is in general applied in the frequency domain to reduce complexity using fast convolution. The frequency domain gradient of an overlap/save fast convolution is given by [Haykin 2002, Hänsler & Schmidt 2004]:

$$\Delta\mathcal{H}(n) = \mu DFT[\text{first } N \text{ elements of IDFT}[\mathcal{P}^{-1}(n)\mathcal{X}(n)\mathcal{E}(n)], \mathbf{0}_{1\times(B-1)}] \tag{2.25}$$

where the vector $\mathcal{X}(n)$ corresponds to $DFT(\tilde{\mathbf{x}}_B(n))$ with $\tilde{\mathbf{x}}_B(n) = [x(n), x(n-1), \cdots, x(n-(M+B)-1)]$ and $\mathcal{E}(n) = DFT[e(n), e(n-1), \cdots, e(n-B-1)|\mathbf{0}_{1\times M}]$. Note that these Fourier transformations require zero padding to make the frequency domain multiplication equivalent to time domain convolution, which explains the presence of $\mathbf{0}_{1\times(B-1)}$ in Equation 2.25. The first N elements to which the Inverse Discrete Fourier Transform (IDFT) is applied in Equation 2.25 corresponds to the elements that are saved when using the overlap and save method. The vector $\mathcal{P}(n)$ is a normalization vector comparable to the NLMS algorithm but here it is applied per bin and is given by:

$$\mathcal{P}(n) = \lambda\mathcal{P}(n-1) + (1-\lambda)\mathcal{X}(n)\mathcal{X}^H(n) \tag{2.26}$$

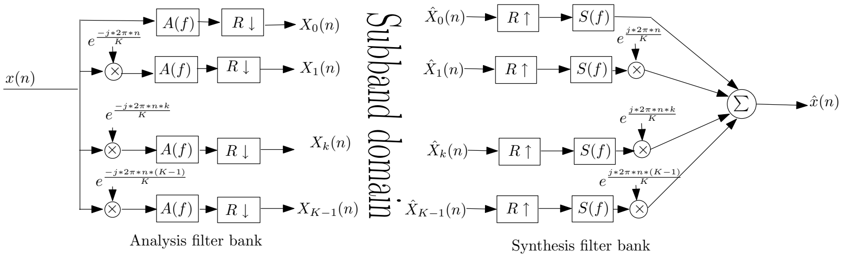


Figure 2.9: General filter bank approach

Each bin has its own normalization factor, it may result in better control of the adaptation rate than in the case of NLMS where the same adaptation step is used for all taps. The general block of the FBLMS approach is illustrated in Figure 2.8. In particular it shows the different Fast Fourier Transform (FFT) and Inverse Fast Fourier Transform (IFFT) routines which are used in the filtering and updating processes as presented in [Hänsler & Schmidt 2004].

A further complexity reduction is possible using block partitioning [Farhang-Boroujeny 1998]. This procedure consists of partitioning the filter length and applying independent adaptation processes to each partition. If we suppose that the filter length N is equal to $M \times L$ then the convolution:

$$\hat{d}(k) = \sum_{n=0}^{N-1} \hat{h}(n) x(k-n) \tag{2.27}$$

can be written as:

$$\hat{d}(k) = \sum_{m=0}^{M} \hat{d}_m(k) \tag{2.28}$$

where $\hat{d}_m(k)$ is given by:

$$\hat{d}_m(k) = \sum_{l=0}^{L} \hat{h}(L \times m + l)) x(k - (L \times m + l)) \tag{2.29}$$

Fast convolution in the frequency domain can be applied to compute Equation 2.29 and will reduce the complexity compared to standard FBLMS. This is generally achieved with a reduction of performance [Farhang-Boroujeny 1998].

The drawback of these techniques is the delay that will be introduced due to the block-by-block process and the lost of time resolution [Hänsler & Schmidt 2004]. To solve the resolution problem an efficient solution has been proposed which involves to a subband domain processing.

2.4.7 Subband domain approaches

Subband processing is another approach to reduce the complexity of a full-band process and is a mid-way between time domain processing and frequency domain processing. The general approach is to split the signal in different bands and then to apply an adaptive filter in each band as in conventional NLMS. After the process the output is reconstructed. The subband process itself requires two main blocks: an analysis filter bank to split signal into different bands, and a synthesis filter bank to reconstruct the subband signal.

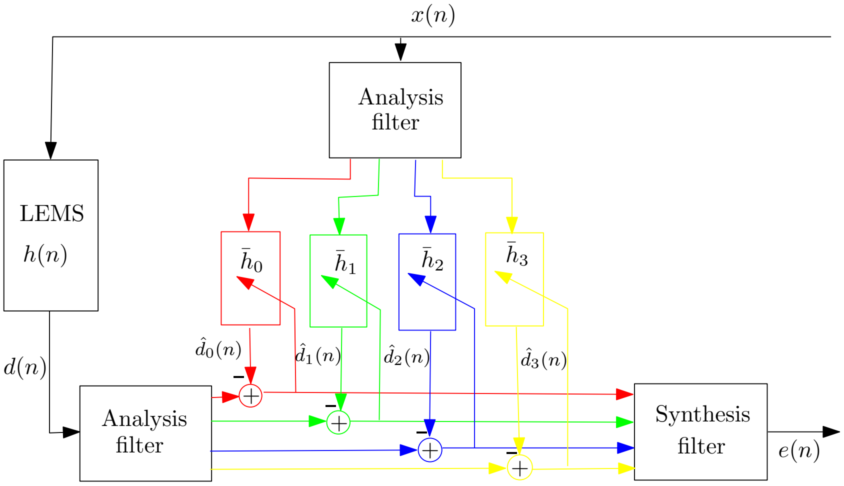


Figure 2.10: Synthesis-independent subband domain LMS

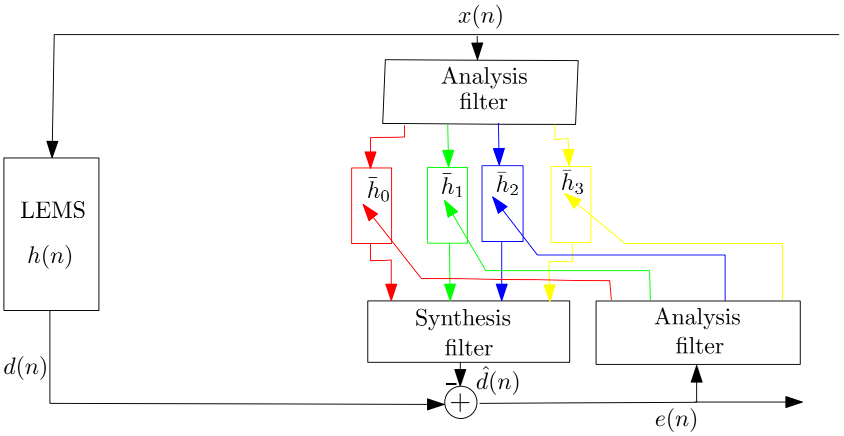


Figure 2.11: Synthesis-dependent subband domain LMS

The general subband process is illustrated in Figure 2.9. The analysis filter bank uses different non-overlapping filters to split the signal into different bands so, at the output of each filter, the signal is narrow band and is down-sampled. The down-sampling will reduce the complexity of the process as, in this case, the number of filter taps are reduced by a factor equal to the down-sampling factor. When the filter bank is well designed, the bands are symmetric so that this property can also be used to reduce the complexity. Another advantage is the possibility to use different filter lengths in each band relative to the strength of the speech in the subband in question. Lower bands in which speech energy is higher can therefore have more taps than higher bands where the level of speech is less significant.

In the case of adaptive filtering there are two possibilities to make the subband AEC. Figure 2.10 illustrates the most widely used. The input signal $x(n)$ and the echo $d(n)$ pass through an analysis filter and the error $e(n)$ is computed in the subband domain then reconstructed with a synthesis filter to obtain the time domain error. This structure is generally referred to as “synthesis-independent”, because the adaptation process is independent to the synthesis process.

The second structure is given in Figure 2.11. In this scheme the input signal $x(n)$ passes through an analysis filter then the estimated signal $\hat{d}(n)$ computed in subband domain is reconstructed and subtracted from the echo signal $d(n)$ in the time domain to obtain the error. The error passes through an analysis filter for the updating process of the subband filters. This structure is referred to as “synthesis-dependent” since we need the synthesis process to update the filter. The structure is comparable to the FBLMS structure illustrated in Figure 2.8 where the Discrete Fourier Transform (DFT) and IDFT are replaced respectively by analysis and synthesis filter banks.

The synthesis-independent structure requires a stop-band which is difficult to realize in practice and some distortions may appear in the error signal. The synthesis-dependent structure requires to use in parallel analysis and synthesis filtering which may have generally different delays that increase the complexity of the structure and stability [Farhang-Boroujeny 1998].

A less complex subband system is obtained with critical sampling, meaning that the down-sampling factor is equal to the number of subbands. The increased overlap between subbands may however reduce performance. The drawback of subband approaches is the delay introduced due to the use of analysis and synthesis filter banks. The subband system may also require some anti-causal taps in the estimated filter [Hänsler & Schmidt 2004]. The requirement of the anti-causal taps is due to the fact that the down-sampling process is not a time shift invariant process.

Chapter 3

Non-linear AEC

The focus in this chapter switches to non-linear Acoustic Echo Cancellation (AEC). Non-linear structures are presented without detailing the filtering process as they are mainly derived from linear AEC solutions. The non-linear echo cancellation arises with the problem of non-linearities introduced by the use of low-cost or miniaturized devices which exhibit some non-linearities. Here we present an overview of non-linear AEC approaches that have been proposed in the literature. The problem of non-linear echo has been tackled in two main directions: adaptive systems and post-filtering. The first is based on a non-linear model of the Loudspeaker Enclosure Microphone System (LEMS) and uses in general standard adaptive filtering algorithms to provide an estimate of the non-linear echo. This system can be viewed as an extension of linear AEC to a non-linear model of the LEMS. The second approach uses residual echo suppression techniques to estimate the useful near-end signal and is generally based on approaches similar to speech enhancement in noise. As with noise compensation this approach also relies on some a priori on the non-linearity model to estimate the non-linear component to be suppressed.

3.1 General approach

In this section we present the general approach of non-linear AEC system. We focus here on the different structures that can be derived according to the assumptions on the LEMS. As in the linear case the section is divided into two parts, a first part which focuses on the modelling of the LEMS and a second one which presents the identification approaches.

3.1.1 Non-linear modelling approaches

The non-linear model is more complex and requires knowledge of the characteristics of the non-linearities. In AEC and many other applications, even when a system is supposed to be non-linear it is nevertheless assumed to have a linear component. In general in AEC the linear component is assumed to dominate the non-linear component and is used as a priori in some identification processes. Figure 3.1 illustrates the LEMS in three mains blocks. This representation uses a simplification of the LEMS where, instead of a component decomposition, each block represents a subsystem comprising a group of components. Hence we have the down-link path, S_1, which involves all components between the AEC input and the loudspeaker. The acoustic channel, S_2, represents the coupling between the loudspeaker and the

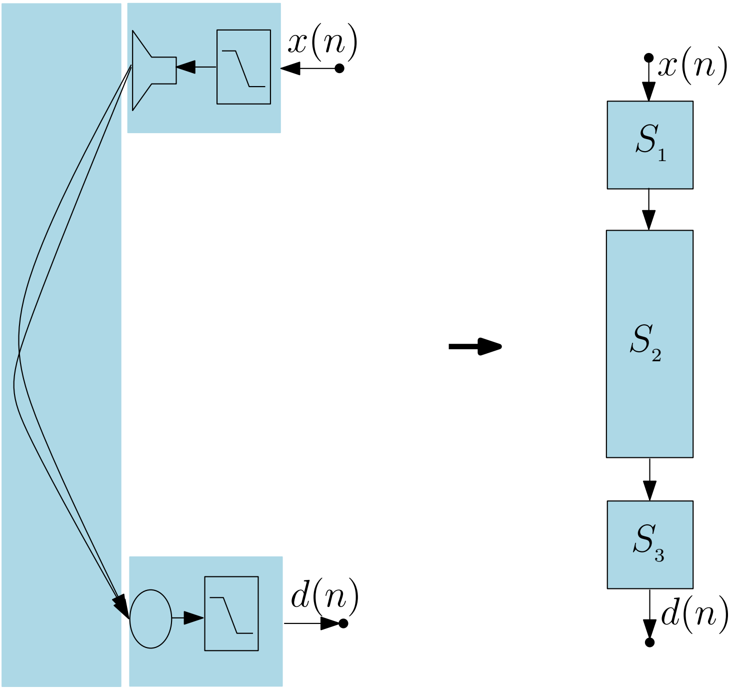


Figure 3.1: Non-linear LEMS model. Here the systems S_1, S_2 and S_3, correspond respectively to the down-link path, the acoustical channel and the up-link path. The LEMS is non-linear if at least one sub-system is non-linear.

microphone. The up-link path, S_3, involves all components between the microphone and the AEC reference point.

Hence the non-linear model differs from the linear model as soon as one of the blocks presents a non-linear characteristic. The block itself is considered as non-linear when a component in the model is not well defined as linear. The entire LEMS is then considered as non-linear and must be modelled according to the non-linearities that arise in the system.

Depending on the sub-system which introduces the non-linearities, different structures may be used to efficiently model the LEMS. The LEMS of Figure 3.1 may be differently modelled according to the position of the non-linear sub-system. The most widely used structures are illustrated in Figure 3.2. In this case S_1 is assumed to be non-linear and the rest of the system is assumed to be linear. This structure can be seen as the concatenation of a non-linear system and a linear system, since the concatenation of two linear systems is linear. Depending on the type of the non-linear model of S_1, the model can also be assumed to be globally non-linear by merging the non-linear system S_1 and the linear system $S^e_{2,3}$ to obtain $S^e_{1,2,3}$. Nevertheless, even when the global approach is used, we still need to define the blocks that are sources of non-linearity. A priori knowledge on the non-linearity sources helps to design a robust algorithm. Figure 3.2 (b) and (c) present two different

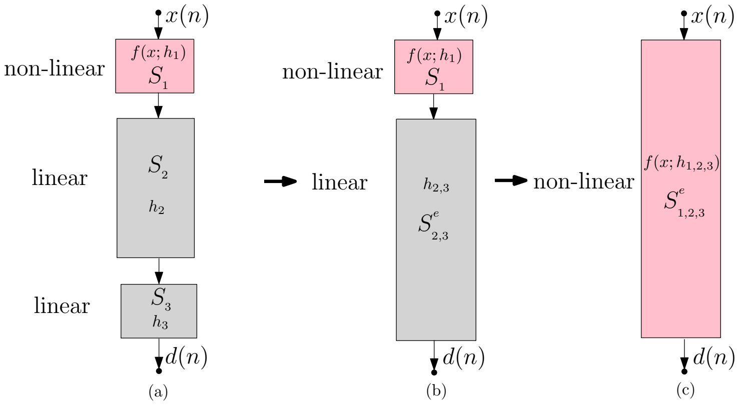


Figure 3.2: Example of non-linear LEMS. The down-link (S_1) is assumed to be a source of non-linearity whereas the others are assumed to be linear (S_2 and S_3). $f(x;h)$ represents a non-linear function parameterised by h and h_2,h_3 are linear impulse responses of system S_2 and S_3 respectively. In this example the impulse response of $S_{2,3}$, $h_{2,3}$ is equivalent to the convolution of h_2 with h_3.

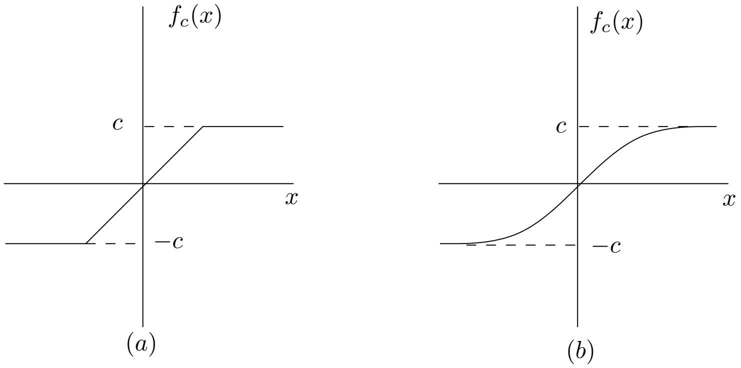


Figure 3.3: Example of non-linear function that are used to model clipping non-linearity. (a) represents a hard clipping model and (b) a smooth clipping model which can be represented by different types of functions.

structures to model the LEMS, the choice between these structures is important for the robustness of the system but a decision of efficiency cannot be made until the characteristics of the different system are well defined.

A well investigated non-linear model is the clipping (saturation) model which assumes that component above certain signal level or amplitude introduces some distortions. This is well known for amplifier which become saturated at significant signal level. The clipping model itself can be written in different ways, a simplified model is when one assumes that the limit is reached at a fixed point, which is called hard clipping as illustrated in Figure 3.3 (a). The more complex version is given in Figure 3.3 (b) where the amplification factor changes smoothly before the clipping

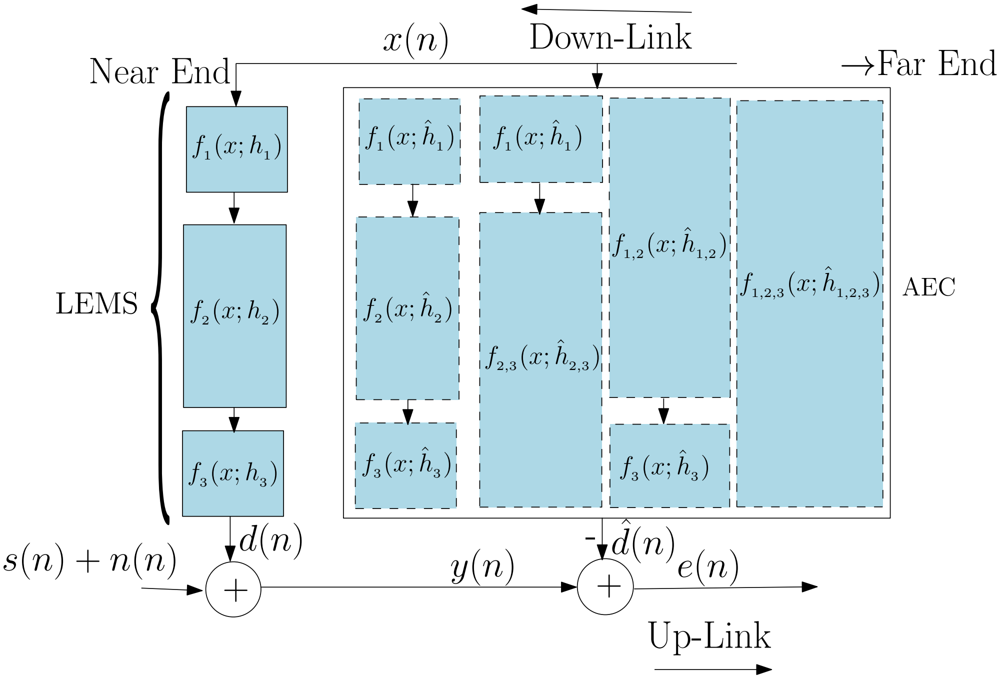


Figure 3.4: Concept of system identification in non-linear case, Here it is assumed that at least one of the functions $f_1(x;h_1)$,$f_2(x;h_2)$,$f_3(x;h_3)$ is non-linear and $h_{p,k}$ is not necessarily equivalent to $h_p * h_k$. Only one of the dashed structure is used as acoustic echo canceller.

level is reached. Due to different possible ways to model the non-linear LEMS details for a specific model are not provided and will be investigated latter in the core of the dissertation for the Volterra and clipping non-linearity models. After modelling of the LEMS the AEC algorithm uses an identification procedure to estimate the optimal model parameters.

3.1.2 System identification

In the non-linear case a similar approach as the linear case is generally adopted. According to the model and the structure that have been chosen the different parameters defined in the model are estimated. Hence, according to the number of systems adopted we may have one $(S^e_{1,2,3})$, two $(S_1, S^e_{2,3})$ or $(S^e_{1,2}, S_3)$ or three (S_1, S_2, S_3) to identify as illustrated in Figure 3.4. It shows the different structures that can be adopted regarding to the systems characteristics for a better identification procedure. In application generally the case (S_1, S_2, S_3) is used only when S_1 and S_3 are non-linear and S_2 linear and is still a difficult identification procedure.

Cascaded structure

The Cascaded Structure (CS) refers to identification procedures where at least two systems need to be identified such as $(S_1, S^e_{2,3})$, $(S^e_{1,2}, S_3)$ or (S_1, S_2, S_3). In the

LEMS model illustrated in Figure 3.2 we can write S_1 as a non-linear function of x and a vector of parameters h_1 to define the output $f(x; h_1)$. S_2 and S_3 are simply represented by linear filters h_2 and h_3 respectively. This example is chosen as it is used latter to developed some of the algorithms proposed in this thesis. In Figure 3.2 (a) which is the true LEMS model, the echo signal is given by:

$$d(n) = h_3 * h_2 * f(x; h_1)(n) \tag{3.1}$$

Three vectors of parameters are thus needed to identify the model but, since in AEC the identification of each individual sub-system is not an issue the LEMS can be simplified to Figure 3.2 (b) so that $f_{2,3}(x; h_{2,3}) = h_3(n) * h_2(n) * x(n) = h^e_{2,3} * x(n)$. Thus instead of identifying two filters we try to identify one filter representing the concatenation of S_2 and S_3. A reason to do so is the difficulty to identify two linear filters that are concatenated with current identification procedure that are based on error minimization. The use of $h^e_{2,3}$ is preferable as the concatenation of two linear systems is also linear. Hence the echo signal will be written as $h^e_{2,3} * f(x; h_1)$.

In Figure 3.2 (b), the objective will be to identify the functions $(f_1(x; h_1), h^e_{2,3})$. However, in the AEC application we can satisfy ourself with any couple $(h_r * f_1(x; h_1), h_s * h^e_{2,3})$ with the condition that h_r and h_s exist and $h_r * h_s = \delta(n)$ (they are invertible). In this case the output of the systems $(f_1(x; h_1), h^e_{2,3})$ and $(h_r * f_1(x; h_1), f_{2,3}(x; h_s * h_{2,3}))$ is identical but the systems are not identical.

Parallel structure

The Parallel Structure (PS) is the most widely used structure in AEC and corresponds to $(S^e_{1,2,3})$. The overall system is modelled by only one non-linear system as in Figure 3.2 (c). In this case we merge the non-linear and linear systems which results in a non-linear system which can be written as $f_{1,2,3}(x; h_{1,2,3})$, a non-linear function of x with a vector of parameters $h_{1,2,3}$ generally different to $h^e_{1,2,3}$ $(h_1(n) * h_2(n) * h_3(n))$. Depending on the model of non-linearity $h_{1,2,3}$ can be differently related to h_1 and $h^e_{2,3}$ and its derivation is not always guaranteed.

In Figure 3.2 (a), if $f(x; h_1)$ is a non-linear function which is linear in its parameters $(f(x; \alpha \cdot h_1) = \alpha \cdot f(x; h_1))$ hence we can write $h^e_{1,2,3} = h_1 \circledast h^e_{2,3}$[1] and results in $f_{1,2,3}(x; h_{1,2,3}) = f(x; h^e_{1,2,3})$. An example of a non-linear function which satisfies this condition is the polynomial function. The clipping models given in Figure 3.3 do not satisfy this condition but are generally approximated by a polynomial function which is linear in parameters using Taylor series. As it is easier to obtain $f_{1,2,3}(x; h_{1,2,3})$ using a polynomial approximation of $f(x; h_1)$ this explains the reason of the widely used Volterra filter. Taylor series avoid the implementation of complex functions such as exponentials functions that are used in some non-linear models and thus allow complexity reduction, however, generally leads to a Volterra model of the LEMS.

[1] The symbol "$\circledast$" is used here as the operation is slightly different from the linear convolution. An example is shown in Section 6.1.2 (Equation 6.9) with the concatenation of a second order Volterra kernel and a linear filter.

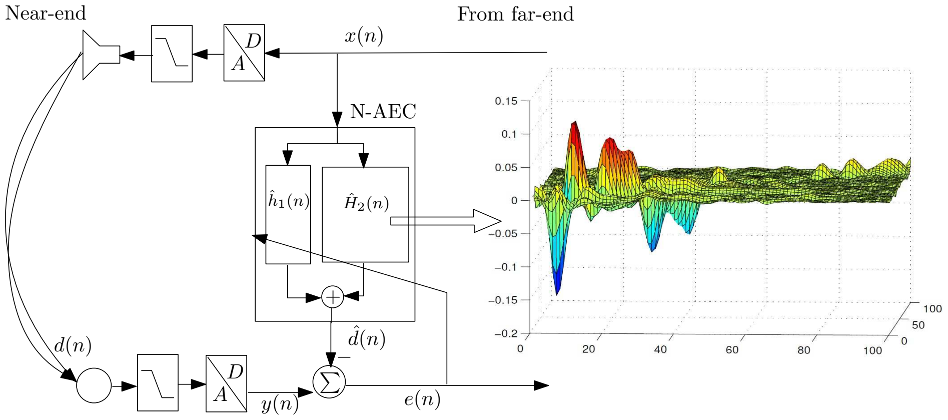


Figure 3.5: Parallel non-linear AEC based on quadratic Volterra adaptive filtering

3.2 Non-linear adaptive filtering

The constraints encountered for linear adaptive filtering remain also for non-linear solutions. Hence, non-linear AEC are known to be complex with slow convergence. All these problems have led to a significant effort in non-linear acoustic echo cancellation to devise solutions to non-linear adaptive filtering. These solutions can be divided into three main structures.

Parallel Structure PS approaches use a global model of the LEMS to estimate the echo signal. Cascaded Structure CS approaches use a concatenation of different types of systems to estimate the echo signal. Loudspeaker Pre-processing (LP) approaches where the linear AEC is combined with a pre-processor which aims to linearise the loudspeaker output.

3.2.1 Parallel structures

The parallel structure (PS) is a system where the linear echo component and the non-linear component are estimated and summed to obtain the echo component. The Volterra model of the LEMS is part of this structure as the linear echo component is estimated by the first order filter, whereas non-linear components are estimated by higher order kernels such as the second order non-linearity which are estimated by the quadratic kernel. Note that parallel solutions presented here are not specifically dedicated to acoustic echo cancellation, hence solutions from network echo cancellation and non-linear system identification are also cited. This is due to the fact that the widely used Volterra model is dedicated to a wide range of non-linear applications.

When the Volterra filter is used to model the LEMS the echo signal is given by:

$$d(n) = \sum_{p=1}^{P} d_p(n) \tag{3.2}$$

where $d_p(n)$ corresponds to the output of the $p-th$ kernel which is given by:

$$d_p(n) = \sum_{k_1=0}^{N_p-1} \cdots \sum_{k_p=k_{p-1}}^{N_p-1} h_{(n_{p,1},n_{p,2},\cdots,n_{p,p})} \prod_{q=1}^{p} x(n-k_q) \tag{3.3}$$

where N_p is the memory of the p^{th} kernel. Due to the linearity between the input power series and the Volterra kernel, linear adaptive filtering approaches are well adapted to Volterra filtering. Hence the quadratic kernel output can be written as:

$$d_2(n) = \mathbf{h}_Q^T(n)\mathbf{x}_Q(n) \tag{3.4}$$

where $\mathbf{h}_Q(n)$ is a vector that contains the taps of the quadratic kernel given by:

$$\mathbf{h}_Q(n) = [h_2(0,0), h_2(0,1)\cdots h_2(0,N_2-1),\cdots,h_2(N_2-1,N_2-1)]^T \tag{3.5}$$

Note that, for symmetry reasons i.e. $h_Q(l,p) = h_Q(p,l)$, only half of the taps are used. The input vector $\mathbf{x}_Q(n)$ is given by:

$$\mathbf{x}_Q(n) = [x^2(n), x(n)x(n-1),\cdots,x(n)x(n-N_2-1),\cdots,x(n-N_2-1)x(n-N_2-1)]^T \tag{3.6}$$

Using this structure most of the linear implementations are easily extended to the Volterra filter which is one of the reason why Volterra filters are widely used. In this case applying the Least Mean Square (LMS) algorithm will lead to:

$$\hat{\mathbf{h}}_1(n+1) = \hat{\mathbf{h}}_1(n) + \mu e(n)\mathbf{x}(n) \tag{3.7}$$

for the linear filter and,

$$\hat{\mathbf{h}}_Q(n+1) = \hat{\mathbf{h}}_Q(n) + \mu_Q e(n)\mathbf{x}_Q(n) \tag{3.8}$$

for the quadratic kernel adaptive estimation. We note that, in Equation 3.8, the same error is used for all the kernels meaning that they will affect each other. A second point is that the stability issue of the quadratic kernel is more challenging as the normalization is not as easy as for the linear case. Solutions to non-linear echo cancellation are presented in the following in which the majority use a quadratic Volterra filter for complexity reasons.

The Volterra solution, presented in [Thomas 1971], is one of the first solution for non-linear echo cancellation and in the conclusions the author explains that this solution suffers from complexity and that an efficient estimator for the Volterra filter parameters is required. Regarding the fact that the PS is affected by the shape of the acoustic channel between the loudspeaker and the microphone, in [Stenger & Rabenstein 1998, Stenger *et al.* 1999b] a truncated version of the Volterra quadratic kernel is proposed where the relatively null coefficients that represent the delay between the loudspeaker and the microphone are discarded in the process. Another simplification of the Volterra quadratic kernel has also been proposed in [Kuech & Kellermann 2002] where a cascaded structure is used to truncate the Volterra quadratic kernel to its diagonal which represents the most significant part of the kernel. This approach is a simplified form of the Multi Memory Decomposition (MMD) proposed in [Frank 1994, Frank 1995] for loudspeaker linearisation. Volterra filter orthogonalization has also been proposed by Mathews in [Mathews 1995a, Mathews 1995b] using Lattice Recursive Least Square (LRLS) or QR Recursive Least Square (QR-RLS) but the complexity is not taken into account as the system is dedicated to the identification process. Kuech [Kuech *et al.* 2005] proposed a power filter orthogonalization for small loudspeakers where a Gram-Schmidt orthogonalization is used followed by a bias correction which reduces the complexity compared to the Volterra filter. The power filter is an approximation of the Volterra kernels to their diagonal elements which is an equivalent model of a cascade of a memoryless polynomial expansion and linear filter. The input output relation is given by:

$$d(n) = \sum_{p=1}^{P} \mathbf{h}_p^T(n)\mathbf{x}_p(n) \tag{3.9}$$

where $\mathbf{x}_p(n)$ is equal to $[x^p(n), x^p(n-1), \cdots, x^p(n-N_p)]$ which corresponds to the diagonal elements of the $p-th$ kernel of the Volterra filter. Recently a multi-channel procedure has been proposed to improve the performance in [Malik & Enzner 2011]. This multi-channel procedure is based from a Markov model of the acoustic channel as in [Enzner & Vary 2006] which uses a frequency domain Kalman filtering to estimate the echo path.

Other solutions have been proposed to improve the Volterra filter using the Adaptive Projection Algorithm (APA) or Recursive Least Square (RLS) algorithms [Mathews & Lee 1988, Mathews 1991, Fermo *et al.* 2000, Lee & Mathews 1993]. Even if these algorithms improve performance compared to Normalized-LMS (NLMS)-based Volterra filters, they have the disadvantage of increasing computational complexity.

To improve the estimation procedure a Volterra filter combination is proposed in [Azpicueta-Ruiz *et al.* 2009, Azpicueta-Ruiz *et al.* 2011] which relies on the use of two Volterra filters, in a similar fashion to the combination of linear filters. With two quadratic kernels such solutions are too complex for real-time mobile applications. The sparsity of the quadratic kernel has lead to application of Proportionate NLMS (PNLMS) to Volterra filtering as proposed in [Kuech & Kellermann]. The use of a sparse adaptive filter for the quadratic kernel is well justified since the significant taps of the quadratic kernel are concentrated around the diagonal but it results in an increase in complexity. As the speech correlation has a great effect on NLMS algorithm a Volterra filtering based on NLMS algorithm with fixed decorrelation procedure is proposed in [Kuech *et al.* 2006].

The Volterra non-linear echo filter has also been extended to other domains like the frequency domain [Mansour & Gray 1981, Reed & Hawksford 2000]. These solutions are based on fast convolution, as used in the linear Frequency Block LMS (FBLMS) algorithm, to reduce complexity and it is also claimed that frequency domain adaptive filtering provides faster convergence. In [Kuech & Kellermann 2005] a block partitioning has been used. Based on this block partitioned, frequency domain Volterra filter structure, Zeller et al have proposed many improvements [Zeller & Kellermann 2010a], such as the iterated Partitioned Frequency Block Volterra LMS (PFBVLMS) [Zeller & Kellermann 2007, Zeller & Kellermann 2008] which is an extension of the work of [Eneman & Moonen 2003] on iterated Partitioned Frequency Block LMS (PFBLMS) to Volterra filtering. This iterated procedure can be written in the time domain as:

$$\begin{aligned} & \textit{for } r = 1 \textit{ to } R \textit{ do} \\ \mathbf{h}^{(r)}(n) &= \mathbf{h}^{(r-1)}(n) + \Delta\mathbf{h}^{(r)}(n) \\ & \textit{end} \\ \mathbf{h}^{(0)}(n+1) &= \mathbf{h}^{(R)}(n) \end{aligned} \tag{3.10}$$

where $\Delta\mathbf{h}^r(n)$ is computed using $x(n)$ and the error $e^r(n)$ which is given by:

$$e^{(r)}(n) = y(n) - \mathbf{x}^T(n)\mathbf{h}^{(r-1)}(n) \tag{3.11}$$

This data reusing method permits to use the same data R times to improve system performance but on the other hand, increases complexity and the process delay. An evolutionary Volterra filter is proposed in [Zeller *et al.* 2009, Zeller *et al.* 2010, Zeller *et al.* 2011] which aims to fit the quadratic kernel to its optimal size using the combination of two Volterra filters with different memory lengths. This solution permits to avoid under-modelling or over-modelling of the LEMS using a kernel size control. The control is based on the error resulting from the different filters and a technique that reduces the complexity of the adaptation. This method copies the taps of the filter with the best performance to the others with respect to their position and takes into account the difference in their number of taps.

The Volterra filter has also been used in the subband domain. Here the process is more complicated due to aliasing and cross-band effects which become more difficult in non-linear environments. However, subband approaches have been proposed by [Zhou *et al.* 2006, Burton *et al.* 2009, Furuhashi *et al.* 2006]. They are generally synthesis-dependent non-linear filters which are an extension of the linear structure illustrated in Figure 2.11. This approach aims to reduce the complexity of the system without losing too much time resolution and is claimed to improve system performance as well. This structure is more efficient than the second one (Figure 2.10) which introduces more constraints due to non-linear cross-terms between subbands. Nevertheless a solution has been proposed with the synthesis-independent structure in the Short-Term Fourier Transform (STFT) domain Volterra identification by [Avargel & Cohen 2009] which takes into account linear and non-linear cross terms. With this model it has been shown theoretically, with white Gaussian inputs

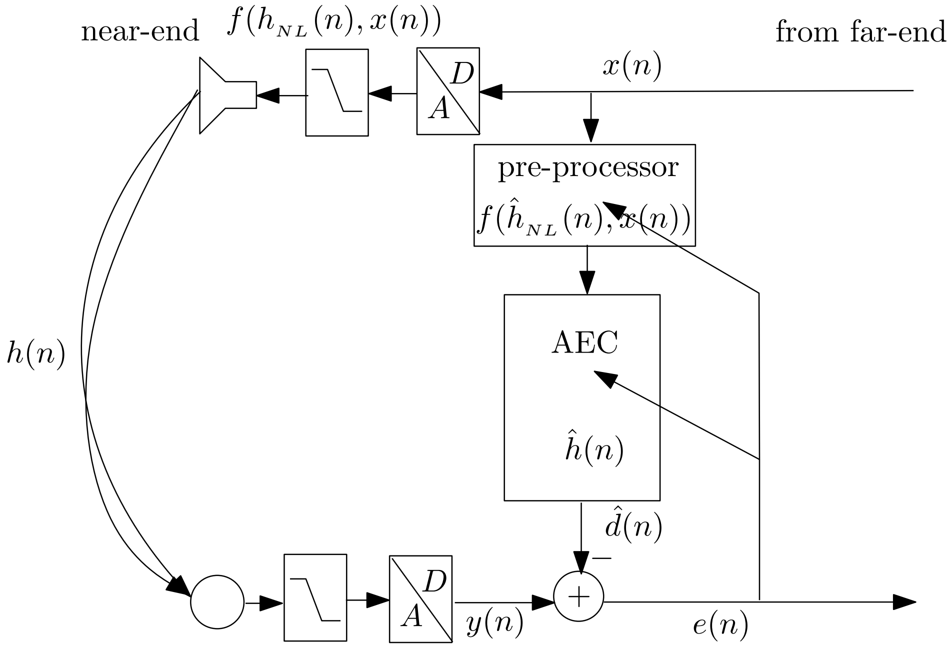


Figure 3.6: Example of cascaded non-linear AEC based on Wiener model. The pre-processor is used as loudspeaker model

that the quadratic model can improve system performance only for higher non-linear to linear component ratios. These results may not hold in the speech case as speech linear components are typically more correlated with the non-linear component due to speech signal harmonicity.

These solutions increase Volterra filter performance in terms of the complexity and convergence. Further improvements are still needed due to the higher number of parameters that are required to be updated which may easily be difficult for a real-time mobile application where memory is limited. The main drawback of the Volterra solution in non-linear AEC applications is that it is limited to static environments. This is due to the high number of parameters that need to be updated for each Echo Path Change (EPC). Many other solutions have been proposed where fewer parameters are required. These solutions are based on a cascaded structure of the LEMS model which is described next section.

3.2.2 Cascaded structure

The use of cascaded structures in AEC applications is based on the assumption that loudspeaker is the main source of non-linearities. This is confirmed by experiments which show that the loudspeaker is the main source of non-linearities due to hands-free mode and other imperfections related to the loudspeaker structure. In general non-linearities from enclosure vibration are uncorrelated with the far-end signal [Birkett & Goubran 1995b]. This explains why loudspeaker non-linearities have been studied in a more general way in addition to specific studies involving that for the echo cancellation application [Frank 1994, Quaegebeur 2007, Schurer 1997]. Models of the loudspeaker produced from these and related studies are typically used in solutions for non-linear AEC, even for parallel structures of the LEMS. Four cascaded structures have proved popular. They are the Wiener structure or NL-L (non-linear system (pre-processor) followed by a linear system), Hammerstein structure or L-NL a (linear system followed by a non-linear system), Wiener-Hammerstein or NL-L-NL and Hammerstein-Wiener or L-NL-L. In general in the Wiener and Hammerstein structure the NL system is a polynomial system without memory and is generally assumed to be static. To avoid being restrictive we consider as cascaded structure (CS) the concatenation of at least two systems where at least one of them is non-linear and the estimation of each system is performed separately. We use separate estimation procedures due to the fact that many cascaded structures can then be combined to form one PS (see Section 3.1.2). An advantage of the CS is that when it is well separated it requires less parameters and may give better convergence than parallel structures. On the other hand, however, it is more sensitive to local minima than parallel structures as the objective function is not always convex.

Example of a CS is illustrated in Figure 3.6. It uses a non-linear pre-processor represented by a non-linear function $f(\mathbf{h}_{NL}; x(n))$ depends on some parameters $\mathbf{h}_{NL}$ which need to be estimated and a linear AEC represented by $\mathbf{h}(n)$. The general estimation procedure with the NLMS adaptive filtering approach is given by:

$$\hat{\mathbf{h}}_{NL}(n+1) = \hat{\mathbf{h}}_{NL}(n) - \mu_{NL} \frac{\partial e^2(n)}{\partial \hat{\mathbf{h}}_{NL}(n)} \tag{3.12}$$

for the pre-processor filter and:

$$\hat{\mathbf{h}}(n+1) = \hat{\mathbf{h}}(n) + \mu_l \frac{\partial e^2(n)}{\partial \hat{\mathbf{h}}(n)} \tag{3.13}$$

for the linear filter estimation. The pre-processor function $f(\mathbf{h}_{NL}; x(n))$ is a non-linear function of $x(n)$. Many types of pre-processor have been proposed but the most popular use hard clipping, piecewise linear or sigmoid functions [Birkett & Goubran 1994, Nollett & Jones 1997, Stenger & Kellermann 2000, Fu & Zhu 2008] and polynomial or Volterra series [Stenger & Kellermann 2000, Guerin *et al.* 2003]. The parameters of which are represented by $\mathbf{h}_{NL}$ in Figure 3.6. From Equation 3.12 and 3.13 we see that the estimators are dependent as they use the same error.

A CS, where the loudspeaker is modelled using a polynomial function, is proposed in [Birkett & Goubran 1995a]. The proposed structure is based on three-layer time-delay neural network for the non-linear part, instead of the NLMS procedure in Equation 3.12, followed by a linear AEC based on the NLMS algorithm. This solution has shown that, in the cascaded approach, the higher the non-linearities are the better the performance of the cascaded approach compared to the linear AEC only. But, for small non-linearities, the linear system can perform better than the non-linear alternative. It has been extended to a Wiener-Hammerstein model using a non-linear clipping model [Nollett & Jones 1997]. This structure provides better performance than a linear AEC and, since all the parameters are estimated using the NLMS approach, it is less complex than the neural network approach in [Birkett & Goubran 1995a] though convergence to the global mean square minimum is not guaranteed. Some similar solutions have also been proposed in [Stenger *et al.* 1999a, Stenger & Kellermann 2000] which uses Wiener model, the clipping model of [Nollett & Jones 1997] and the polynomial model of [Birkett & Goubran 1994] for the loudspeaker model. In [Stenger & Kellermann 2000] a solution to improve the adaptation is proposed using an orthogonalization procedure based on speech statistics and an RLS algorithm in the place of the NLMS algorithm in Equation 3.12 for the pre-processor parameters estimation. The solution with the RLS algorithm is shown to improve performance but with increased of the system complexity, even if the number of pre-processor parameters is smaller compared to linear AEC.

Another solution proposed in [Shi *et al.* 2007] estimates the parameters of a polynomial expansion in a Wiener system based on the pseudo-coherence method. It is shown to be more accurate but more complex than the RLS algorithm [Shi *et al.* 2008a]. Based on the coherence method a Wiener-Hammerstein model of the LEMS is proposed in [Shi *et al.* 2008b] where the first, non-linear model is of the loudspeaker whereas the second model is the inverse of the microphone. The same authors propose in [Shi *et al.* 2009] a shortening filter for long impulse responses. A Wiener system is used in the acoustic link and a shortened filter is applied to the microphone signal. The objective of the shortened filter is to reduce the length of the Echo Path (EP). The convolution of the echo path with a shortened filter results to a filter which has its significant taps concentrated in the earlier part. This shortened approach which is more known in communication system has shown to give the possibility of reducing the length of the linear filter in the Wiener system. The drawback in AEC applications is that the shortened filter affects the near-end signal.

The more general CS proposed in [Guerin *et al.* 2003] uses a non-linear Volterra model of the loudspeaker and, to avoid local minima, the linear kernel of the Volterra filter is constrained to one. This approach introduces a general model of the loudspeaker and provides better results compared to the parallel approach in terms of complexity and convergence but increases AEC complexity compared to models with power expansions.

3.2.3 Loudspeaker pre-processing

The loudspeaker pre-processing (LP) approach uses the combination of two filters as in a CS. A non-linear filter is used before the loudspeaker to linearise its output and a linear filter is used to estimate the echo signal. In general this approach has a comparable complexity to the CS. The linearisation procedure is not well studied in the context of AEC. One reason may be the fact that, due to the speech harmonic-

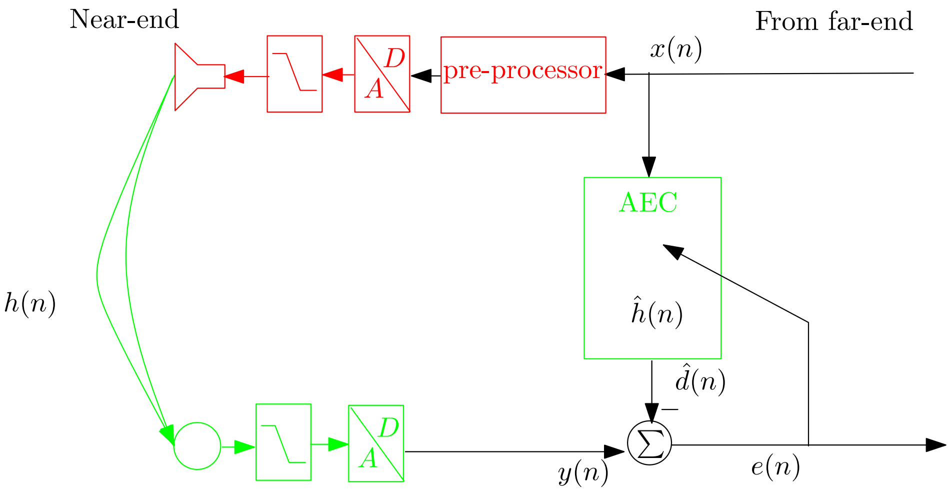


Figure 3.7: Non-linear AEC based on loudspeaker linearisation

ity, the effect of non-linearities under certain levels are not perceived. A solution presented in [Furuhashi *et al.* 2006] which combines linear AEC with an offline estimation of the non-linear system in the subband domain to linearise the loudspeaker output. It is shown that this solution may increase linear AEC performance. Another advantage of such structures is that the two systems are less coupled than with a CS. The drawback of the solution proposed in [Furuhashi *et al.* 2006] is the requirement to estimate the characteristics of each loudspeaker. Furthermore, even though the loudspeaker characteristics are largely static, they may vary over time. An advantage of the linearisation structure is that it does not require oversampling as proposed in [Frank 1996, Zeller & Kellermann 2010b, Mäkelä & Niemistö 2003] to reduce the effect of high-frequency non-linearities introduced in the residual echo when using parallel or cascaded structures. As non-linearities generated in the LEMS which are above the microphone sampling frequency are removed in the microphone signal so they are compensated in the AEC link by oversampling or low pass filtering. The disadvantage of loudspeaker pre-processing is the additional non-linearities generated at higher-frequencies and it also requires that the loudspeaker linear impulse response to be invertible.

3.3 Non-linear echo post-processing

Residual echo suppression has been proposed to suppress echo which cannot be estimated through conventional, linear AEC due to convergence issues or insufficient numbers of taps [Beaugeant *et al.* 1998]. It is also proposed as a solution to non-linear echo cancellation. One such approach is proposed in [Hoshuyama & Sugiyama 2006b]. This solution assumes that residual non-linear echo and the estimated echo are correlated so an offline estimation of the correlation coefficient in the frequency domain is developed. Since such a solution is somewhat

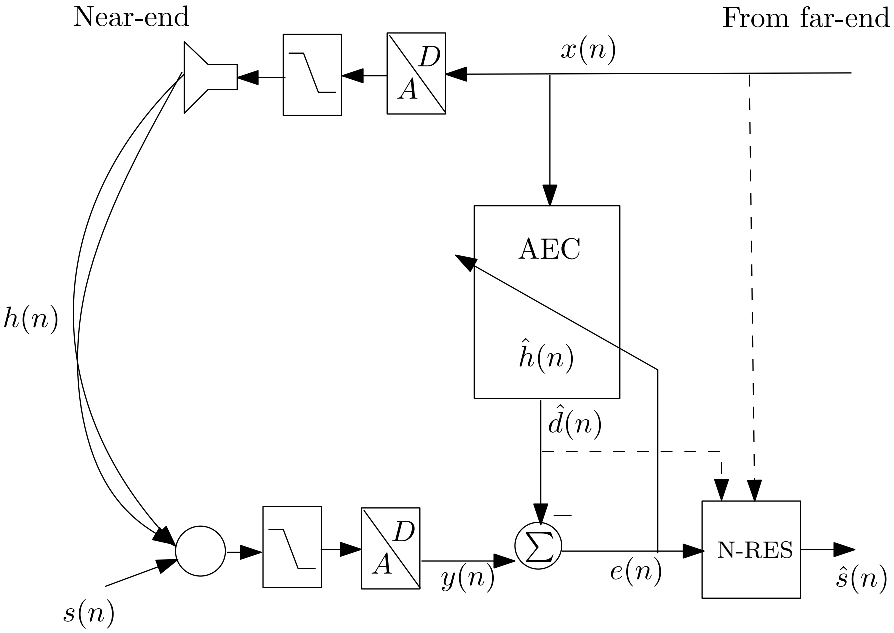


Figure 3.8: Acoustic echo suppression with non-linear echo post-processing. The adaptive filter estimates the linear component of the echo and residual echo is further suppressed by the non-linear residual echo suppressor.

device dependent an online coefficient estimation approach is proposed by the same authors in [Hoshuyama & Sugiyama 2006c].

The power filter used in [Kuech *et al.* 2005] has also been extended to residual echo suppression where non-linear echo is estimated based on an unconstrained adaptive frequency domain approach to reduce complexity. This approach can also be used to estimate the useful near-end signal [Kuech & Kellermann 2007]. Another solution involving the use of power expansion for non-linear model is proposed in [Shi *et al.* 2008c]. A subband domain solution is proposed in [Bendersky *et al.* 2008] to estimate the linear filter and also the non-linear residual echo where estimates of the non-linearities in the residual error are based on harmonic compensation.

Residual echo suppression is less complex than non-linear adaptive filtering approaches. The main disadvantage is distortion introduced in the useful signal. Another approach to non-linear echo post-processing is the solution proposed by [Wada & Juang 2012] where the objective is not residual echo suppression from the near-end signal but the enhancement of the feedback error applied to the linear adaptive filter. This structure tries to recover the true residual linear error so that it can be feedback to the linear AEC under the conditions where non-linearities are minimal. It has the advantage of making linear AEC robust to non-linearities without disturbing the near-end signal. The problem with this method is that the residual non-linear echo component will not be removed from the near-end signal.

Chapter 4

Linear AEC analysis

Since linear Acoustic Echo Cancellation (AEC) is still widely used even in the presence of non-linearity this chapter presents an analysis of different adaptive filtering algorithms dedicated to linear AEC and their robustness to non-linearity. The study is also directly relevant to non-linear AEC since the same adaptive filter algorithms are often applied in dedicated non-linear AEC solutions. For the analysis of the different approaches some metrics are required. Two metrics are used in this analysis. The System Distance (SD), which measures the distance between the real Echo Path (EP) and its estimate, and the Echo Return Loss Enhancement (ERLE) which measures the amount of echo suppressed. The assessment of the linear AEC algorithms in non-linear environments requires a model of non-linearities. Hence a polynomial model of non-linearities is used to simulate their effect.

The analysis of linear AEC in non-linear environments first aims to characterise the behaviour and robustness of the different linear adaptive filtering algorithms. This is helpful to identify the most reliable linear adaptive filter algorithms in non-linear environments. The analysis is also of use in choosing specific adaptive filters when we have an a priori on the environment i.e. linear, small non-linearities or highly non-linear. Also reported is a comparative analysis of AEC behaviour in the presence of non-linearity and noise. Many adaptive filtering solutions have been proposed for noisy environments and so this comparison shows if linear AEC algorithms exhibit similar behaviour in non-linear and noisy environments. This analysis is based on our works presented in [Mossi *et al.* 2010a, Mossi *et al.* 2010b].

According to the results provided by these tests a theoretical analysis of the non-linearity effect on the different AEC algorithms is presented. A time variant formulation of the Wiener AEC solution is proposed to incorporate the effect of non-linearities on linear AEC. We then use this time variable formulation to explain the behaviour of each linear AEC algorithm.

As all algorithms are already introduced in Chapter 2, we start by presenting the simulation set-up and the metrics use for the analysis.

4.1 Simulations set-up

Figure 4.1 illustrates the system model, where a linear AEC algorithm (Normalized-LMS (NLMS), Adaptive Projection Algorithm (APA) or Frequency Block LMS (FBLMS)) is used to reduce the echo signal in different linear or non-linear environments. The first environment is non-linear where non-linearities are generated artificially according to a non-linear model. All non-linearities are assumed to stem

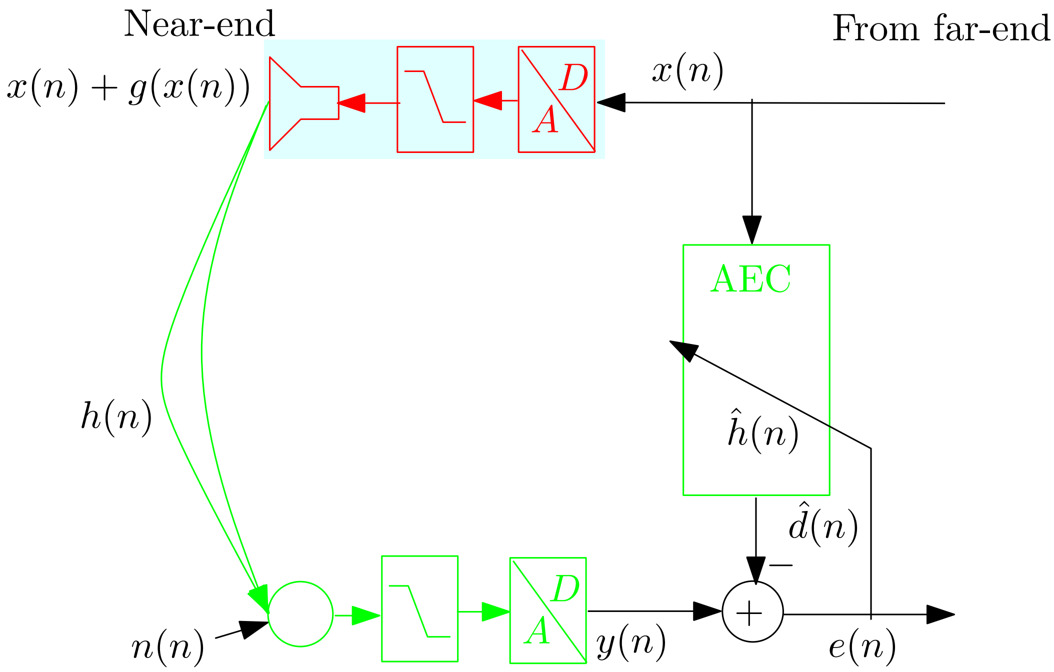


Figure 4.1: System model

from the loudspeaker and are represented by $g(x(n))$. The second environment contains background noise which is assumed to be generated in the near end environment. In general these situations arise in the same time in real applications but here, as our objective is to analyse separately the effects of these two perturbations and compare the behaviour of the AEC algorithms in each case they are thus investigated separately.

4.1.1 Non-linear model

Since the assessment presented here requires comparisons of performance both with and without non-linearities under otherwise identical conditions it is necessary that non-linear distortions be generated artificially so that they are well controlled. Here we briefly describe the sources of non-linearity which are already well defined in the literature and the model that has been chosen for the assessment.

In general non-linearities can be introduced by the Up-Link (UL) and Down-Link (DL) amplifiers, by the loudspeaker, the microphone, resonance from the mobile terminal housing and the acoustic EP. However, since the loudspeaker signal is usually of high level, especially in hands-free mode, it is commonly assumed that non-linearities from the DL amplifier and loudspeaker dominate and that, consequently, all other sources are negligible [Stenger & Kellermann 2000, Guerin *et al.* 2003, Kuech & Kellermann 2006]. It is also shown in [Birkett & Goubran 1995b] that non-linearities introduced by the housing can be considered as uncorrelated noise meaning that their assessment is comparable to that of ambient noise. Under this assumption the acoustic path may be considered as linear.

In [Guerin *et al.* 2003, Fermo *et al.* 2000] DL non-linearities may be adequately modelled using a Volterra model. As in the work of [Birkett & Goubran 1995a, Stenger & Kellermann 2000, Kuech & Kellermann 2006] a Volterra model of amplifier and loudspeaker non-linearities may be approximated by a cascade of memoryless saturation characteristics. Here we take into account only the second and third order non-linearities as they are generally assumed to be the most dominant components [Birkett & Goubran 1995a, Guerin *et al.* 2003]. For all experimental work reported in this chapter non-linearities are generated according to:

$$x_{_{ld}}(n) = x(n) + \alpha x^2(n) + \beta x^3(n), \tag{4.1}$$

where $x_{_{ld}}(n)$ is the non-linear output of the loudspeaker and the non-linear echo component $g(x(n))$ is equal to $\alpha x^2(n) + \beta x^3(n)$. α and β are the second and third order weighting components respectively and lie in the range of $\alpha, \beta \in [0, 1]$. It is worth mentioning that the couple $(\alpha, \beta) = (0, 0)$ corresponds to the linear case. This range of parameters was deemed to be representative of realistic non-linearities measured through laboratory tests of several popular, current mobile phones. It also agrees with those in the general literature, e.g. [Frank 1995]. The loudspeaker signal $x_{_{ld}}(n)$ is then convolved with an impulse response $h(n)$ to simulate the linear EP between the loudspeaker and the microphone.

4.1.2 Experimental set-up

We present the different test conditions for each adaptive filter and compare their performance in the presence of non-linearities or white noise. Echo reduction is assessed in terms of ERLE, convergence time and system distance SD. The duration of the far-end speech signal $x(n)$ is sufficient to ensure the convergence of each algorithm. In all cases ERLE measurements relate to intervals in which the algorithms are deemed to have converged. Non-linear artefacts are introduced into the down-link signal according to the model described in Section 4.1.1. The loudspeaker output, $x_{_{ld}}(n)$, is composed of the sum of the original speech signal $x(n)$ and a non-linear component $\alpha x^2(n)+\beta x^3(n)$ which are both convolved with the EP $h(n)$. This leads to a linear echo component $x(n) * h(n)$ and a non-linear echo component $[\alpha x^2(n)+\beta x^3(n)]*h(n)$. Then, a linear echo to non-linear echo ratio (SNeR) is computed as in [Vondrasek & Pollak 2005]:

$$SNeR = \frac{1}{K}\sum_{i=1}^{K} SNeR_{seg}(i), \tag{4.2}$$

where $SNeR_{seg}(i)$ is given by:

$$SNeR_{seg}(i) = 10log_{10}\frac{\sum_{m=0}^{M-1} x_i^2(m)}{\sum_{m=0}^{M-1} g_i^2(x(n))} \tag{4.3}$$

where $x_i(n)$ and $g_i(x(n))$ are the linear and non-linear echo components respectively in the i^{th} frame of analysed signals. The $SNeR_{seg}(i)$ is computed using windows of

32 ms ($M = 256$ samples for a sampling rate of 8 kHz) according to the short-term stationarity of speech. The SNeR level is used as a reference to generate a noisy signal with linear echo, where the mean Signal-to-Noise Ratio (SNR) is equal to the SNeR. In so doing we have two linear echo signals that are equally disturbed, one with non-linear echo, and another with additive noise. The weighting factors α and β are in the range of $[0, 1]$ as in [Mossi *et al.* 2010a]. This permits to artificially increase the level of the non-linear echo component (and noise) by increasing α and/or β. We compare the behaviour of each adaptive filter, with both non-linear echo and noise, when linear adaptive filters are configured with the same step size μ, and to obtain approximately the same level of ERLE by adjusting the regularization parameter. A second configuration is done by choosing the step-size μ so that each adaptive filter reaches its maximum ERLE.

4.2 Measurement metrics

To compare the performance of adaptive filtering algorithms two objective criteria are used. They are the SD and the ERLE which are described below.

4.2.1 System distance

The system distance (SD) criterion is based on the difference between the estimation by the AEC of the filter impulse response, $\hat{\mathbf{h}}(n)$ and the true impulse response of the Loudspeaker Enclosure Microphone System (LEMS), $\mathbf{h}(n)$ [Breining *et al.* 1999, Vary & Martin 2006]:

$$\bar{h}(n) = \|h(n) - \hat{h}(n)\| \tag{4.4}$$

In general the relative system distance is used to compare system performance under different conditions according to:

$$SD(n)_{dB} = 10 \cdot \log_{10} \left\{ \frac{\bar{h}^2(n)}{\|h(n)\|^2} \right\} \tag{4.5}$$

Figure 4.2 shows an example SD profile for an adaptive AEC filter, and is included here to help illustrate the concept. The algorithm begins at time $t = 0$ where, due to the initialization of the AEC taps to zero, the SD is equal to 0 dB. In this case the filter taps are updated once every sample (0.0625 ms) and the curve shows initially that $\hat{\mathbf{h}}(n)$ begins to converge toward the real filter $\mathbf{h}(n)$ as shown by the falling profile. A decreasing SD indicates the convergence of $\hat{\mathbf{h}}(n)$ toward $\mathbf{h}(n)$.

Some peaks are also observed in the SD profile of Figure 4.2. They correspond to disturbances, which may be due to periods of low SNR, deficient length or a bad parametrization. The general idea is that a quickly decreasing SD indicates faster adaptive filter converge.

The SD can only be used for simulation as in real conditions the acoustic path is completely unknown. The SD is furthermore dependent on the input signal frequency band. In fact, when the SD is used for the assessment of an adaptive filter,

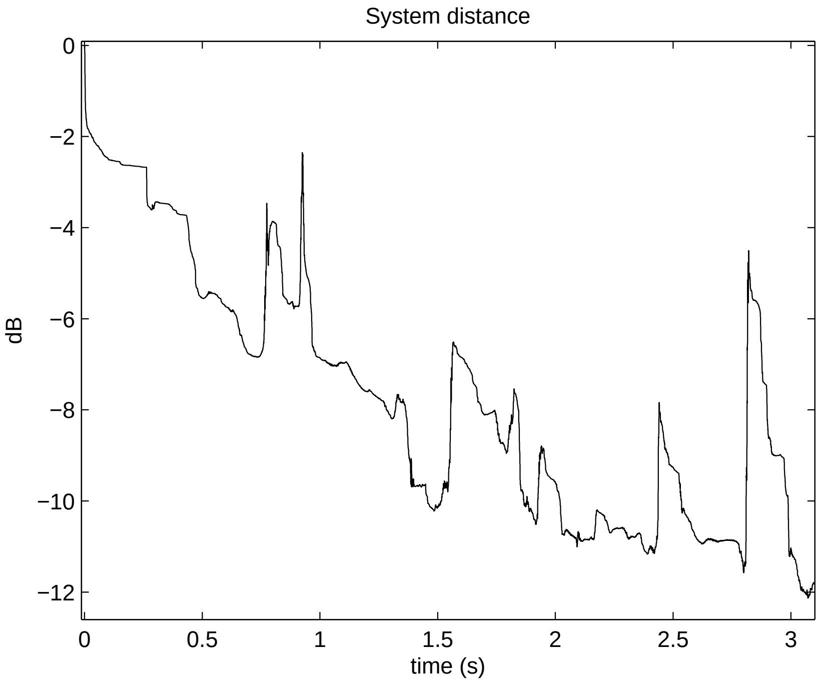


Figure 4.2: An example system distance profile.

where the input is a speech signal, not all frequencies of $h(n)$ are necessarily excited, meaning that the error $d(n) - \hat{d}(n)$ may go to zero even when the system distance profile is far from zero. A simple example of this case is when $x(n)$ is a sinusoid. In general, adaptive filters converge quickly in this case but on the other hand only one harmonic of $h(n)$ will be estimated by $\hat{h}(n)$. In the frequency domain $\hat{\mathcal{H}}(f) \approx 0$ whatever $\mathcal{H}(f)$ for all frequencies different from f_0 which is the frequency of $x(n)$ where $\hat{\mathcal{H}}(f) \approx \mathcal{H}(f_0)$. Hence $\hat{h}(n)$ can only estimate accurately the harmonics of $h(n)$ that are excited by $x(n)$. In other words, the band of $\hat{h}(n)$ is relative to that of $x(n)$, as explained in Section 2.3.1. This shows that the system distance is fully accurate only for identification processes where full band noise signals are used. Nevertheless the SD is still relevant for the comparison of different adaptive filters in linear environments. In practice, whilst the SD gives a quick and easy insight into AEC performance, it is not as useful in the case of non-linear environments. In fact, as will be seen later in non-linear environments the identification of the real system is quasi impossible so that other metrics are required. A circumstance in non-linear environments where the SD can be used is to assess the robustness of the adaptive filter in the estimation of the linear echo component when a linear $h(n)$ is available.

4.2.2 Echo return loss enhancement

The Echo Return Loss Enhancement (ERLE) is given by [Hänsler & Schmidt 2004, Vary & Martin 2006]:

$$ERLE(n)_{dB} = 10 \cdot \log_{10} \frac{E\left\{d^2(n)\right\}}{E\left\{(d(n) - \hat{d}(n))^2\right\}} \tag{4.6}$$

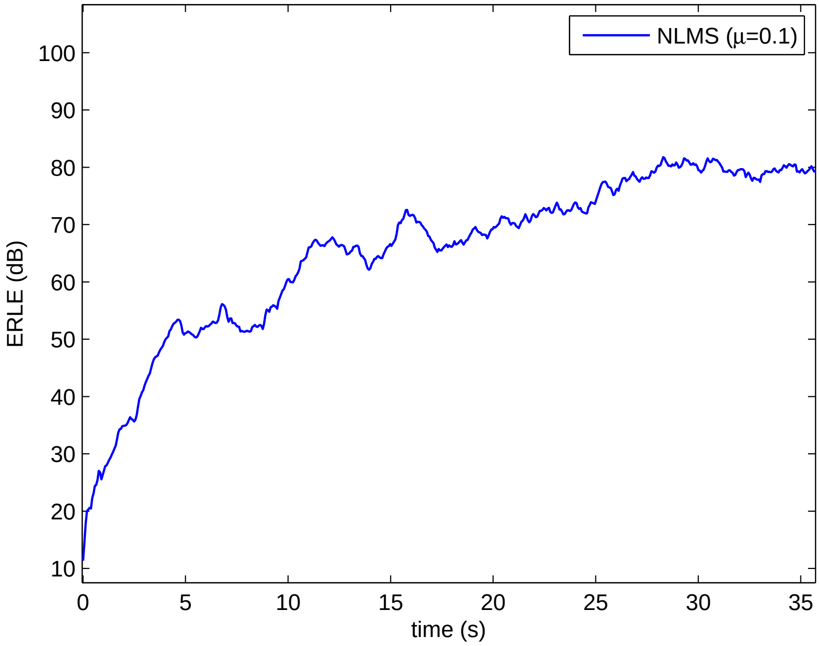


Figure 4.3: An example of ERLE for the NLMS.

The denominator corresponds to the error in the absence of near end speech, $s(n)$ and of noise, $n(n)$, so Equation 4.6 can also be written as:

$$ERLE(n)_{dB} = 10 \cdot \log_{10} \frac{E\left\{d^2(n)\right\}}{E\left\{e(n)^2\right\}} \tag{4.7}$$

The ERLE gives an indication of performance according to the level of the error signal, $e(n)$. As the error is in the denominator, the smaller its value, the greater the ERLE, and the better the convergence, i.e. the higher the ERLE the better the system. It gives an indication of system performance and can also be used in non-linear environments since it does not need any reference to a linear impulse response, i.e. $\hat{h}(n)$. However, the ERLE is also biased by the presence of noise or near-end speech in real applications. To show that let:

$$\begin{aligned}\sigma_d &= E\left\{d^2(n)\right\}\\ \sigma_e &= E\left\{(d(n)-\hat{d}(n))^2\right\}\\ \sigma_n &= E\left\{n^2(n)\right\}\end{aligned}$$

be the energies of the echo signal, the error signal (difference between the real echo and its estimate) and the noise. In the noise condition the ERLE is given by:

$$ERLE(n)_{dB} = 10\cdot\log_{10}\frac{E\left\{(d(n)+n(n))^2\right\}}{E\left\{(e(n)+n(n))^2\right\}} \tag{4.8}$$

where $e(n)$ represents here the difference between the echo signal $d(n)$ and its estimate $\hat{d}(n)$. With the decorrelation assumption between the noise and speech signal we can then write Equation 4.7 as:

$$\begin{aligned}ERLE(n)_{dB} &= 10\cdot\log_{10}\frac{\sigma_d+\sigma_n}{\sigma_e+\sigma_n} \qquad (4.9)\\ &= 10\cdot\log_{10}\{\frac{\sigma_d}{\sigma_e}\frac{\sigma_e(\sigma_d+\sigma_n)}{\sigma_d(\sigma_e+\sigma_n)}\}\\ &= 10\cdot\log_{10}\{\frac{\sigma_d}{\sigma_e}\}+10\cdot\log_{10}\{\frac{\sigma_e(\sigma_d+\sigma_n)}{\sigma_d(\sigma_e+\sigma_n)}\}\end{aligned}$$

If the convergence condition is satisfied then $\sigma_d \geq \sigma_e$ which implies that $\frac{\sigma_e(\sigma_d+\sigma_n)}{\sigma_d(\sigma_e+\sigma_n)}$ is always less than 1 and so $\log_{10}\{\frac{\sigma_e(\sigma_d+\sigma_n)}{\sigma_d(\sigma_e+\sigma_n)}\}$ is negative giving an impression of less echo reduction. Even if the ERLE is biased by the presence of noise for the same SNR the ERLE can indicate which algorithm has the lower residual error.

4.3 Assessment of linear AEC algorithms in adverse environments

In this section two different comparisons are conducted for each metric. The first assessment compares the robustness of the different algorithms in non-linear environments. The second is a comparative assessment in non-linear and noise environments.

4.3.1 Echo attenuation (ERLE)

Algorithm assessment

Figure 4.4 gives the ERLE against time for the APA, NLMS and FBLMS algorithms for different values of SNeR. Here the algorithms are parametrized to reach their maximum ERLE independently and we observe that APA converges more quickly than the others. These curves show clearly how non-linearities reduce the ERLE reached by each algorithm. As already mentioned the FBLMS is the most affected

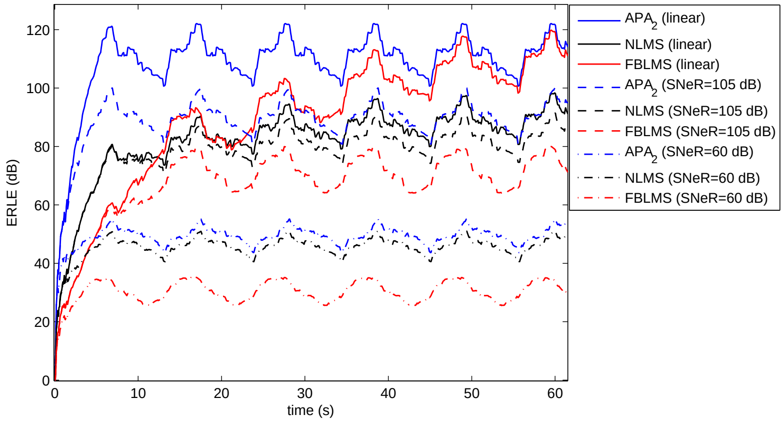


Figure 4.4: ERLE over time of NLMS, FBLMS and APA. Test results to compare the performance in linear and non-linear environments where each algorithm is parametrized to reach its maximum ERLE.

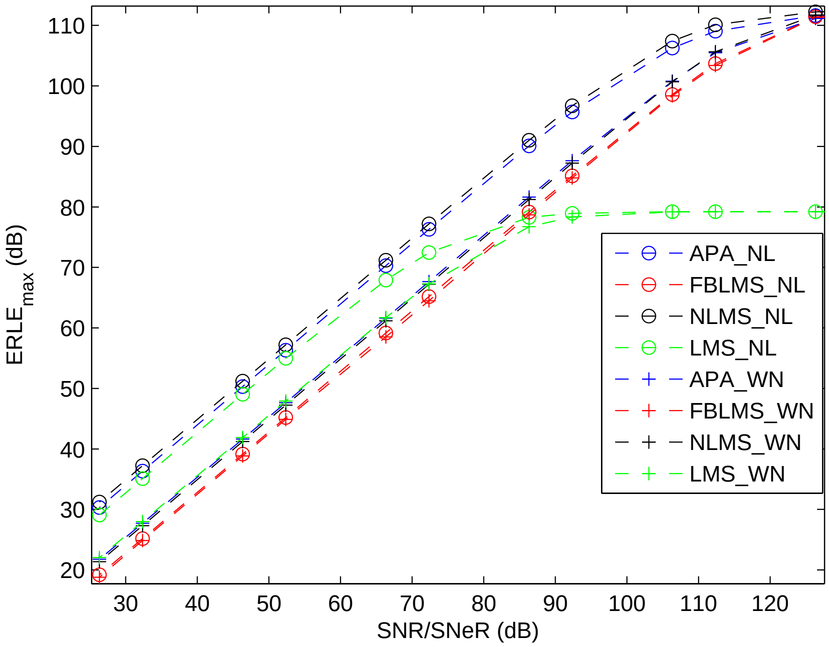


Figure 4.5: Maximum ERLE (in dB) achieved after convergence as a function of SNR/SNeR (also in dB). Here the SNR or SNeR corresponds to added noise (WN) or to non-linear echo (NL) as indicated. Profiles are illustrated for both perturbations and for each of the four approaches to AEC. APA, FBLMS and NLMS are all configured to give equivalent performance under linear echo conditions.

and its ERLE is lower than that of NLMS for SNeR$\leq$ 60 dB. Here we observe that, when the APA is parametrized to reach a higher amount of echo reduction with quick convergence, its performance decreases faster than that of NLMS. But, even if the APA algorithm is disturbed significantly by non-linearities, it still reaches a better ERLE than other algorithms after convergence. From these experiments, a first conclusion is that the faster an algorithm converges the more it is affected by non-linearities. The APA, for instance, is known to converge quickly compared to the NLMS but its performance drastically decreases when non-linearities increase.

FBLMS, however, is severely affected even though it does not converge quickly in linear environments. This behaviour is explained by the block-by-block processing nature of FBLMS. According to Equation 4.1, small input signals $x(n)$ lead to small non-linearities. As a result, even for high values of α and β, a sample-based algorithm will be, for certain periods of low $x(n)$, equivalent to a linear environment and thus, during such periods, it will be relatively less disturbed by non-linearities. Considering block-based processing such as FBLMS, a whole frame of low level $x(n)$ is needed to have the same effect. As a result, block-based algorithms are more disturbed by the same level of non-linear distortion. In fact improvements in convergence in the frequency domain are due to the decorrelation of the frequency bin. In the presence of non-linearity this decorrelation will be affected by the non-linear component in the error signal used for adaptation. This is explained by the fact that non-linearities introduce frequencies at multiples of the linear frequency component. A simple example is if we assume a signal containing two sinusoids, where the high frequency is the double of the low frequency. If the latter generates a second order harmonic it will directly perturb the high frequency components. This example may hold for speech if we consider its harmonicity. If this decorrelation is affected by the presence of non-linearities then convergence will be slower. Since updates are performed only every B samples, where B is the block length ($B = M = 256$ in our experiments), we may expect poor performance.

Figure 4.5 shows the ERLE after convergence against SNeR or SNR for each of the four different algorithms. We define the ERLE$_{max}$ value after convergence as the mean of the ERLE during the period of our test sequence between 50 to 60 s. The general trend of these curves shows that the maximum ERLE of all the algorithms decreases when the non-linearity increases (lower SNeR).

Generally, for high values of SNeR (small values of α and β) the ERLE difference is close to zero, indicating a low degradation in echo cancellation performance due to small non-linearities. For NLMS and APA when SNeR$\geq$ 105 dB, the ERLE is almost unaffected by the non-linearities. This is shown by the flatness of the curves in these ranges. The most affected algorithm is the FBLMS, where the difference in ERLE decreases even for low levels of non-linearity.

Comparative noise and non-linear assessment

Figure 4.5 also shows the maximum ERLE achieved by each algorithm in noisy environments. We observe that, as for non-linear echo, noise also decreases the performance of all adaptive filters. These properties are well known and expected [Vary & Martin 2006, Birkett & Goubran 1995b]. We see that the LMS algorithm is the most robust of all adaptive filters considered; it has the least degradation in performance as the SNR or SNeR decreases. This is due to its poor ERLE performance which is so low that the algorithm cannot even be configured to give equivalent performance to the other algorithms under linear echo conditions. This is expected as the stability over the process of the Least Mean Square (LMS) requires a very small step-size to ensure convergence. It is known that, the smaller the step-size, the lower the effects of the perturbations on the adaptation process and the better the resulting Minimum Mean Square Error (MMSE).

In noisy environments the ERLE of APA, NLMS and FBLMS algorithms decreases by approximately the same amount. For the APA and NLMS algorithms, when the SNR$<$ 100 dB, the difference between the ERLE in non-linear and noisy environments is about 10 dB with better performance in non-linear environments than noisy environments. The FBLMS algorithm seems to show the smallest differences between non-linear and noisy environments. This can again be explained by the averaging effect of block-by-block approaches. In the case of noise the perturbation is effectively averaged over the block and thus has a reduced impact on performance. This is not the case with non-linear echo, which is correlated with the input signal. The result is that noise perturbations have less effect than they do for the other approaches (compared to non-linearities) so that noise and non-linear echo have an equivalent effect on the performance of the FBLMS algorithm.

The difference between the effects of non-linearities and those of noise are explained by two hypotheses:

Noise spectrum: The filter frequency response depends on the differences in energy of the linear echo component and the perturbation (non-linear echo or noise). The spectrum of the non-linear echo component generally has a similar profile to that of the linear echo component whereas the spectrum of white noise is flat. This means that, during periods of voiced speech, the amplitude of the noise signal can be much lower than the amplitude of the speech signal at low frequencies, but much higher at high frequencies. At higher frequencies the linear echo component can thus be masked by the noise spectrum, leading to significant perturbation during periods of voiced speech.

Non-linearities are correlated with the far-end signal: Since non-linearities are correlated with the input signal, this can result in the adaptive filter underestimating the linear part but slightly attenuating the non-linearities. This is less the case in noisy environments as there is no correlation between the noise and the far-end speech signal.

Note that these results are comparable to results reported in [Birkett & Goubran 1995b] which show that non-linearities caused by the loudspeaker have less effect on linear AEC than those caused by enclosure vibrations. These results are explained by the fact that loudspeaker non-linearities are

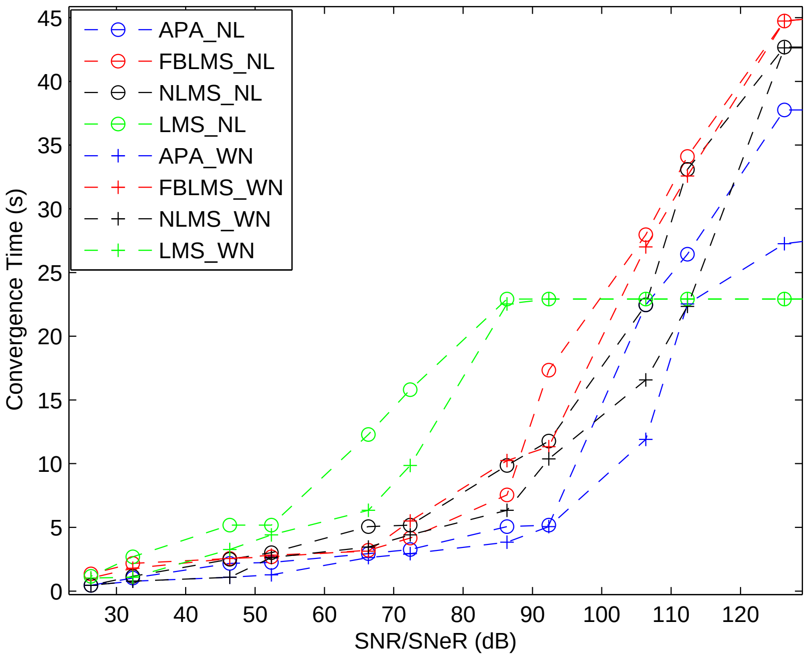


Figure 4.6: Convergence performance with non-linear (NL) and white noise (WN) perturbations for (a) APA, FBLMS and NLMS algorithms against SNeR and SNR.

more correlated to the far-end signal than enclosure vibrations which are noise-like.

4.3.2 Convergence Time

Algorithm assessment

We define the convergence time for each algorithm as the time needed for the adaptive filter to reach 95% of its maximum ERLE value. Convergence times are determined using the same speech signals as used previously and are estimated for all conditions: linear echo, non-linear echo, and linear echo with noise. Figure 4.6 shows the convergence time in seconds against SNR or SNeR for each of the four algorithms and both perturbations.

We see that, with the exception of the LMS algorithm, all profiles have a similar trend even though differences in convergence time are in the order of 25 s at 110 dB. In addition, for each algorithm, convergence times are greater for non-linear perturbations than they are for noise. The LMS algorithm is the slowest to converge where the SNeR or SNR is low but the fastest where they are high ($>$ 100 dB). This is explained by the fact that the ERLE obtained is lower: about 80 dB compared to 110 dB for all other algorithms in linear echo conditions (right side of Figure 4.5).

Figures 4.6 also shows that the convergence time decreases when non-linearities increase. Such unexpected results are explained by the fact that the algorithms

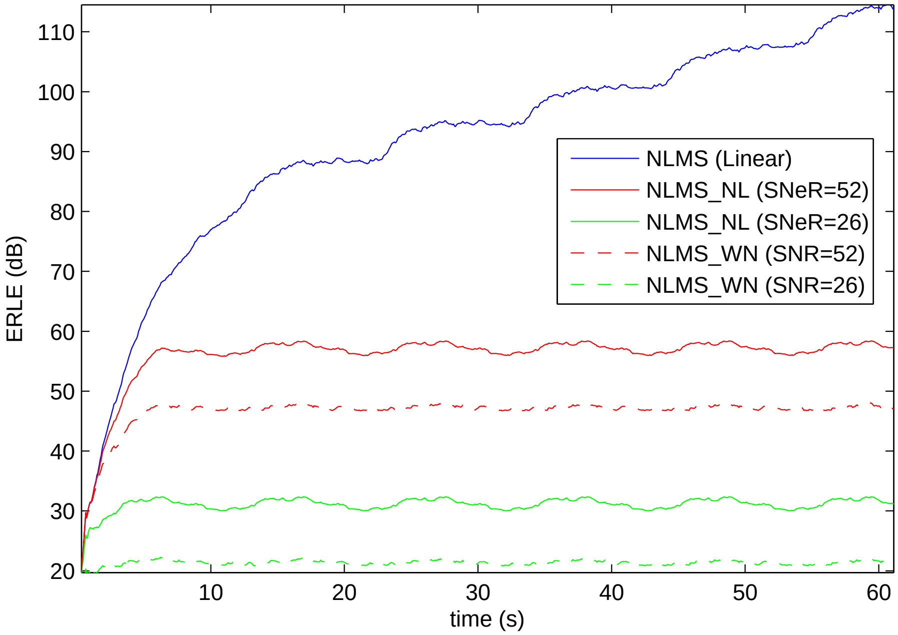


Figure 4.7: Convergence performance with non-linear (NL) and white noise (WN) perturbations for the NLMS algorithm plotted as ERLE against time.

converge in practice to a higher minimum error; the ERLE level is in fact reached faster simply because it is lower. Looking, for instance, at the profile for LMS algorithm, its convergence time decreases from 23 s to less that 5 s for an SNeR varying between 130 and 25 dB, but at the same time the ERLE achieved by LMS collapses by 45 dB (see Figure 4.5). It is nevertheless an important result that echo cancellation algorithms operating in non-linear environments provide less echo reduction but their maximum level of echo reduction is reached relatively quickly. Accordingly, fast converging algorithms such as APA can be of less interest in non-linear environments as the argument to use such algorithms due to their reduced convergence time may no longer hold.

Comparative noise and non-linear assessment

The plots in Figure 4.6 show the absolute convergence time in seconds but do not give an impression of the dynamic performance and, as already discussed, nor do they reflect the ERLE that is eventually achieved. They are thus potentially misleading and for this reason we present in Figure 4.7 a plot of ERLE against time, here for the NLMS algorithm to better illustrate its dynamic and absolute performance. Figure 4.7 shows the ERLE against time with linear echo only and added non-linear echo or noise at an SNeR of 52 dB and SNeR of 26 dB respectively.

These plots show that higher levels of perturbation result in lower levels of ERLE. In the case of linear echo (top profile) convergence is slow and is not even reached during the 60 s illustrated. Crucially, though, the ERLE is much higher than it is for non-linear and noise perturbations. However, in these cases the algorithm converges faster, but to a lower level (i.e. ∼55 dB for non-linear echo with an SNeR of 52 dB and ∼20 dB at 26 dB SNeR, cf. ∼45 dB for noise with an SNR of 52 dB and ∼25 dB at 26 dB SNR). Hence consideration of the convergence time or maximum obtained ERLE are not sufficient on their own to properly appreciate the performance of each approach. Similar profiles were obtained for specifics APA, FBLMS adaptive filters and show an identical trend to that shown here for the NLMS algorithm albeit to different levels of ERLE. Finally, since all algorithms are shown to converge reasonably quickly in noise and non-linear environments it is of questionable advantage to focus effort on more computationally efficient algorithms; efforts are better directed toward the development of more robust algorithms. Indeed, more stable and straight forward algorithms, such as NLMS, are arguably of more interest for mobile terminal applications than their less stable and more computationally demanding alternatives such as APA.

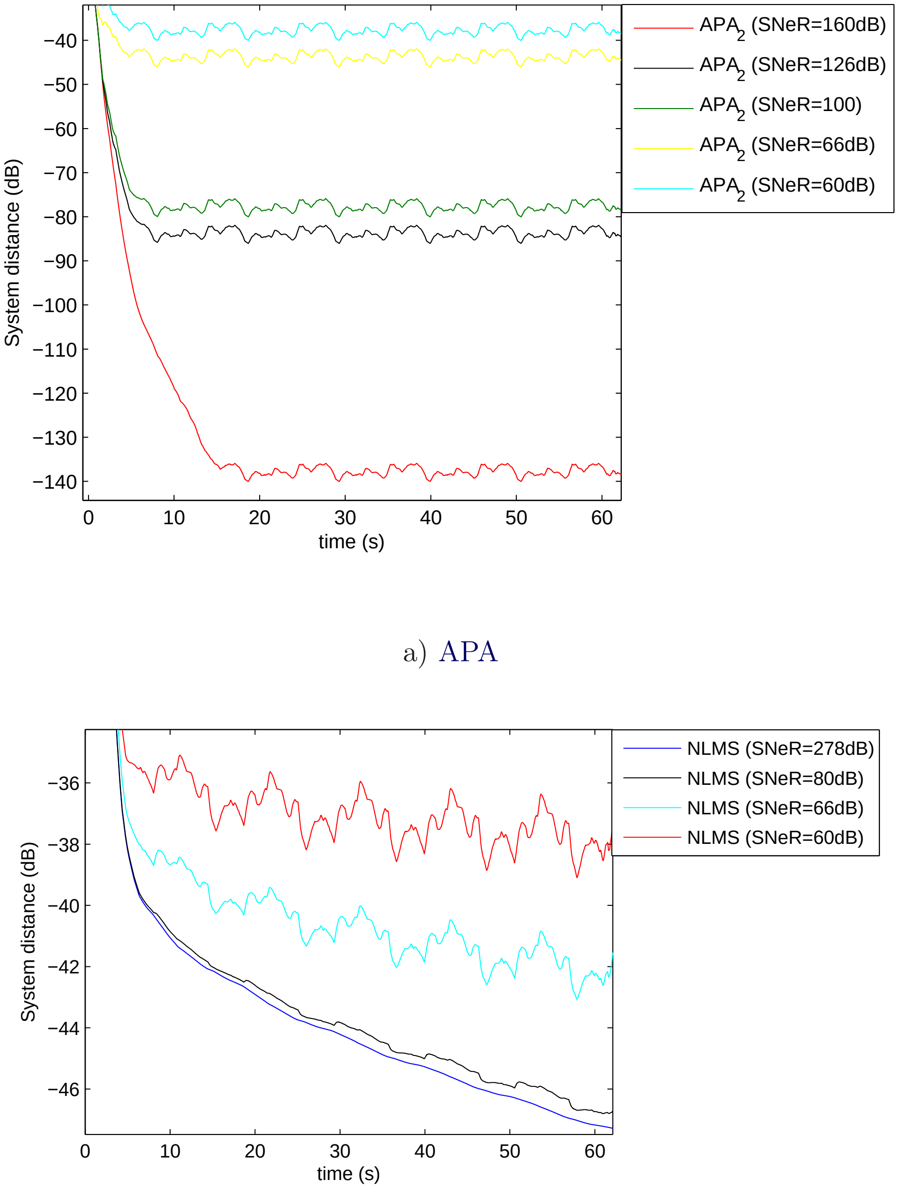
4.3.3 Estimation of linear echo path

Algorithm assessment

The assessment of performance with linear echo is commonly measured according to the SD. It is less appropriate in the case of non-linear echo as the SD shows only how the linear EP is estimated by the adaptive filter. In linear echo environments, the SD indicates how effective is the echo cancellation. In the case of non-linear echo, the SD indicates only how well the linear component is estimated but does not necessarily reflect the level of echo attenuation actually achieved.

Plotted in Figures 4.8 (a) and 4.8 (b) is the evolution of the SD in dB against time for APA and NLMS algorithms respectively. We first observe that APA results in better estimation in the presence of low level non-linearities, but less accurate estimation when non-linearities increase. The NLMS shows slower convergence than APA but estimates are closer to the 256 dB of SNeR case until the SNeR become lower than 80 dB. The linear condition SD of the NLMS is similar to that 256 dB of SNeR reason why it is not shown here. This shows that the estimation of the linear component of the echo is more robust when using NLMS rather than APA in highly non-linear environments. The behaviour of the NLMS algorithm is similar to that of LMS (results not shown here). The FBLMS SD shows that it is more affected than other algorithms. This is explained by the same reasons given previously, the effect of block processing and also by the fact that it is adapted by blocks of M samples.

One could easily assume that linear echo cancellers estimate the linear component of the echo, but this assumption is not supported by these results. Indeed the SD increases when non-linearities increase. This means that, in practice, echo cancellers do not converge to a reliable estimate of the optimal Wiener solution of the EP in linear condition. This observation is of particular interest as many



a) APA

b) NLMS

Figure 4.8: SD over time in non-linear environments for (a) NLMS and (b) APA algorithms. In this case only the third order non-linear component is used ($\alpha = 0, \beta$) but similar behaviour is observed when second order non-linearities are introduced.

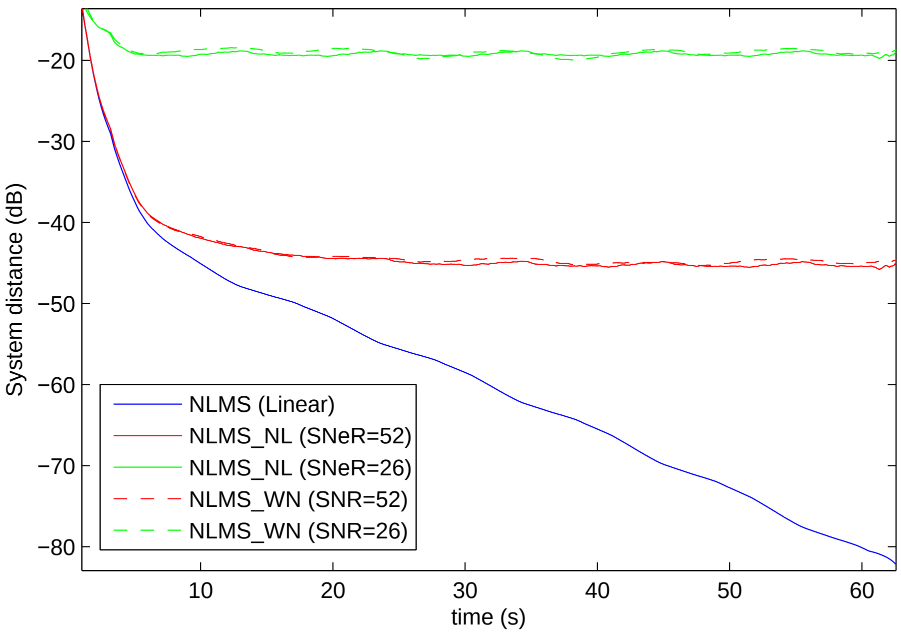


Figure 4.9: Plots of SD (in dB) against time (in seconds) for the NLMS algorithm. Profiles are illustrated for linear echo and also for linear echo with either non-linear echo or added noise at two different levels.

algorithms assume that a non-linear system can be accurately modelled by a cascade of a linear echo canceller and post cancellation of the residual non-linear echo [Hoshuyama & Sugiyama 2006a, Kuech & Kellermann 2007]. According to such assumptions, the linear AEC should approximate the linear optimal solution. Our experiment contradicts this assumption and leads to questions regarding the use of such approaches. The deviation of the linear AEC estimate from the linear optimal Wiener solution is discussed in the next section.

Comparative noise and non-linear assessment

Figure 4.9 shows the behaviour of the NLMS SD as a function of time. Whilst there are differences in exact SD values, the order of the profiles and general trends are indicative of performance for APA and FBLMS. In general, the better the SD, the better the ERLE. However, upon comparison of Figures 4.7 and Figure 4.9 we observe an apparent disparity. Figure 4.7 shows that performance with non-linear echo is generally better than that under additive noise with the same SNR, whereas Figure 4.9 shows almost no differences. This is due to the fact that the SD is only equivalent to ERLE under the condition of total linearity. The ERLE reflects the global performance according to the residual error, whereas the SD reflects the accuracy of the echo path estimate $\hat{h}(n)$. Equivalent values of SD show that linear echo can be attenuated equally well with either non-linear echo or noise perturbations. The differences in the ERLE, however, show that non-linear echo perturbations are better attenuated than noise. This is due to the fact that, in non-linear environments, some of the non-linearities are indeed effectively attenuated by the adaptive filter even if the residual error is still higher than in the linear situation. This is explained by the fact that adaptive filters aim to reduce the correlation (increase the orthogonality) between the error and the input signal. Since non-linear echo is correlated with the input signal it can also be attenuated, albeit only slightly. This is not the case with additive noise. This does not imply that adaptive filters are better in non-linear environments than they are in noisy environments as the adaptive filter does not aim to reduce the noise, but rather the echo signal which includes the non-linear component. It nevertheless shows that non-linearity cannot be assumed to be similar in nature as additive noise. In the next section we try to illustrate the implications of correlation, the relation to convolution and the potential of modelling non-linear environments as time-varying systems with the assumption of a time invariant EP (or an EP which varies more slowly than the speech signal).

4.4 Discussion

This section presents a mathematical analysis of the effects observed in reported experimental results. Under the assumption of a time invariant EP we propose a time variant model of the system that takes into account non-linear components which are correlated with the far-end signal. We then explain the reasons why the performance of APA and FBLMS algorithms decrease faster in non-linear environments than it does for NLMS.

4.4.1 From time invariant to time variant echo path

We propose to derive the Wiener solution of the echo path estimate in non-linear environments. As in the experimental work non-linearities are assumed to be generated only by the loudspeaker. In this case the echo signal is given by:

$$\begin{aligned} d(n) &= h(n) * (x(n) + g(x(n))) \\ &= h(n) * x(n) + h(n) * g(x(n)) \end{aligned} \tag{4.10}$$

where $g(x)$ is a non-linear function of x corresponding to loudspeaker effect. In our experimental test, $g(x) = \alpha x^2 + \beta x^3$ but the analysis provided here is not limited to a polynomial model of $g(x)$. In the presence of the non-linear echo component $h(n) * g(x(n))$ the Wiener solution under the assumption of (short-time) stationarity is given by [Haykin 2002]:

$$h^0_{nl} = \mathbf{R}^{-1}\mathbf{p} + \mathbf{R}^{-1}\mathbf{p}_{h*g(x),x} \tag{4.11}$$

where h_{nl}^{0} is the optimal solution in non-linear environments. $\mathbf{R}$, $\mathbf{p}$ and $\mathbf{p}_{h*g(x),x}$ are the auto-correlation matrix of $x(n)$, the cross-correlation vector between $x(n)$ and $d(n)$ and the cross-correlation vector between $h(n)*g(x(n))$ and $x(n)$ respectively.

If $g(x(n))$ and $x(n)$ are completely decorrelated then $g(x(n))$ can be considered as uncorrelated noise and lead to $\mathbf{p}_{h*g(x),x}$ being equal to zero. In this case we have the same optimal solution in linear and non-linear environments ($h_{nl}^{0} = h^{0}$). But the most appropriate assumption is that the non-linearities are well correlated with the far-end speech signal and leads to $\mathbf{p}_{h*g(x),x}$ being non-zero in short-time process.

As observed in experimental results reported in Section 4.3.3 the correlation may explain the difference between the ERLE in non-linear and noise environments. In the general case this correlation can be assumed regarding the observation that, in general, some distortions are harmonic distortions which, with the short-time process, can introduce correlation. Since this correlation cannot be considered as perfect we can then decompose $g(x(n))$ into two components: $[g(x(n))]_{//}$ which is assumed to be *perfectly* correlated with $x(n)$ and $[g(x(n))]_{\perp}$ which is completely uncorrelated with $x(n)$ and which, therefore, play the same role as uncorrelated noise. In this case Equation 4.11 can be written as:

$$\begin{aligned} h_{nl}^{0} &= \mathbf{R}^{-1}\mathbf{p} + \mathbf{R}^{-1}\mathbf{p}_{h*g,x} \\ &= \mathbf{R}^{-1}\mathbf{p} + \mathbf{R}^{-1}\big(\mathbf{p}_{h*[g(x)]_{//},x} + \underbrace{\mathbf{p}_{h*[g(x)]_{\perp},x}}_{=0}\big) \\ &= \mathbf{R}^{-1}\mathbf{p} + \mathbf{R}^{-1}\mathbf{p}_{h*[g(x)]_{//},x} \end{aligned} \tag{4.12}$$

If $[g(x(n))]_{//}$ exists then there also exists an $h_{//}(n)$ so that:

$$[g(x(n))]_{//} = h_{//}(n) * x(n) \tag{4.13}$$

Equation 4.13 can be viewed as a linear prediction analysis which can be written as:

$$\begin{aligned} g(x(n)) &= [g(x(n))]_{//} + [g(x(n))]_{\perp} \\ &= h_{//}(n) * x(n) + [g(x(n))]_{\perp} \end{aligned} \tag{4.14}$$

where $[g(x(n))]_{\perp}$ is a prediction error. We do not assume, however, that the correlated component $[g(x(n))]_{//}$ is stronger than the decorrelated component $[g(x(n))]_{\perp}$ which in practice depends on the type of distortions. If we convolve the two sides of Equation 4.13 by $h(n)$ we obtain:

$$\begin{aligned} h(n) * [g(x(n))]_{//} &= h(n) * h_{//}(n) * x(n) \\ &= h_{//}(n) * h(n) * x(n) \end{aligned} \tag{4.15}$$

Equation 4.15 shows that the cross-correlation $p_{h*g,x}(n)$ can be written as a function of the cross-correlation $p(n)$ as $p(n) * h_{//}(n)$. Hence the optimal solution in non-linear environments can be written as:

$$\begin{aligned} h^{0}_{nl} &= h^{0} + h^{0} * h_{//} \\ &= h^{0} + h^{0} * h_{//} \\ &= (\delta(n) + h_{//}) * h^{0} \end{aligned} \tag{4.16}$$

where $\delta(n)$ is the Dirac function. As $h_{//}$ is time variant due to speech non-stationarity Equation 4.16 is extended to a more general relation given as:

$$h_{nl}(n) = (\delta(n) + h_{//}(n)) * h(n) \tag{4.17}$$

Equation 4.17 shows that, even when $h(n)$ is a Linear Time Invariant (LTI) system, the linear path tracked by the AEC will be time variable and is represented by $h_{nl}(n)$. This also shows that the variability introduced by $h_{//}(n)$ depends on the speech characteristics. The assumption here is that the relation between the non-linear component and the linear component is time varying and may fluctuate around a mean value depending on the correlation between the non-linear component and the far-end signal. Equation 4.17 shows that in frame-by-frame process the optimal solution deviates from the linear optimal solution. The global optimal solutions for the linear (h^{0}) and non-linear (h^{0}_{nl}) case, however, are the same if the mean of $h_{//}(n)$ is equal to zero. Hence the LEMS becomes a time varying filter due to variation in the speech signal characteristics that will affect $h_{//}(n)$. This introduces a major problem since it is in general assumed that the echo path variations are independent from the input, which is not the case here. This also shows the difficulty in applying statistical analysis in this situation as the properties of the speech signal must be taken into account.

The analysis provides an explanation to the results on the comparison between the non-linearity and the noise and we will be extended next to the comparison of different algorithms. As we have shown that the LEMS can be considered as time varying due to the correlated part of the non-linear component and the far-end signal we can now explain the reason why APA and FBLMS performance decrease faster than NLMS.

4.4.2 Effect of the time varying EP

Here we provide explanations regarding the effect of the time varying EP on the APA and the FBLMS algorithms.

Affine projection algorithm

Figure 4.10 illustrates a third order APA procedure at time sample 10 where k represents the delay position of the input vector data which corresponds to a delayed error. If only $k = 0$ is used this corresponds to the basic NLMS as explained in Section 2.4.2. This illustration aims to show that the condition where $\hat{\mathbf{h}}(10)$ minimises all the errors $(e_{10}(8), e_{10}(9), e_{10}(10))$ requires that $\mathbf{h}(8)$ and $\mathbf{h}(9)$ are identical

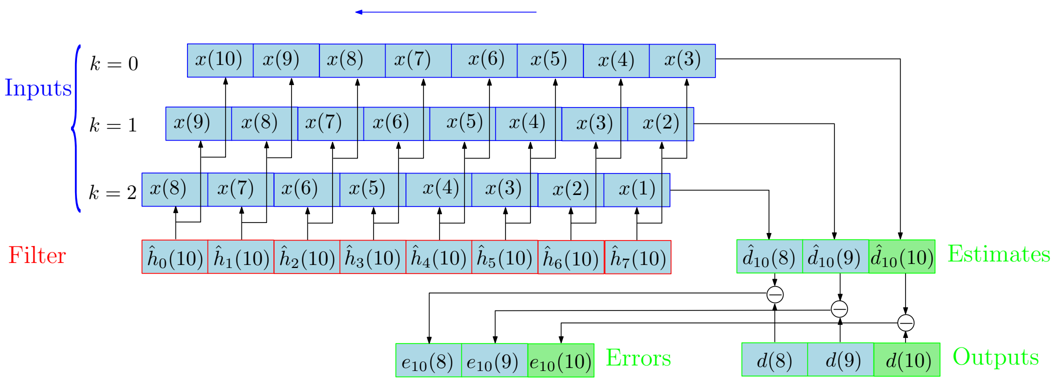

Inputs
$k=0$
$x(10)$ $x(9)$ $x(8)$ $x(7)$ $x(6)$ $x(5)$ $x(4)$ $x(3)$
$k=1$
$x(9)$ $x(8)$ $x(7)$ $x(6)$ $x(5)$ $x(4)$ $x(3)$ $x(2)$
$k=2$
$x(8)$ $x(7)$ $x(6)$ $x(5)$ $x(4)$ $x(3)$ $x(2)$ $x(1)$
Filter
$\hat{h}_0(10)$ $\hat{h}_1(10)$ $\hat{h}_2(10)$ $\hat{h}_3(10)$ $\hat{h}_4(10)$ $\hat{h}_5(10)$ $\hat{h}_6(10)$ $\hat{h}_7(10)$
$\hat{d}_{10}(8)$ $\hat{d}_{10}(9)$ $\hat{d}_{10}(10)$
Estimates
$e_{10}(8)$ $e_{10}(9)$ $e_{10}(10)$
Errors
$d(8)$ $d(9)$ $d(10)$
Outputs

Figure 4.10: Illustration of an third order APA procedure at time sample $n = 10$.

to $\mathbf{h}(10)$. This means in a general sense that the EP should be time invariant. We can simplify Equation 2.15 as:

$$\Delta h(n) = \mathbf{K}(n)\mathbf{e}(n) \tag{4.18}$$

where $\mathbf{K}(n)$ is given by $\mu\mathbf{X}^T(n)\left(\mathbf{X}(n)\mathbf{X}^T(n) + \xi\mathbf{I}_N\right)^{-1}$ represents the gain applied to the error. In circumstances of non-linearity the error vector $\mathbf{e}(n)$ is given by:

$$e(n-k) = d(n-k) - \hat{\mathbf{h}}^T(n)\mathbf{x}(n-k) \tag{4.19}$$

Equation 4.19 can be rewritten as:

$$\begin{aligned} e(n-k) &= \mathbf{h}^T_{nl}(n-k)\mathbf{x}(n-k) - \hat{\mathbf{h}}^T(n)\mathbf{x}(n-k) \qquad (4.20)\\ &= [(\mathbf{u} + \mathbf{h}_{//}(n-k)) * \mathbf{h}(n)]^T\mathbf{x}(n-k) - \hat{\mathbf{h}}^T(n)\mathbf{x}(n-k)\\ &= (\mathbf{h}(n) - \hat{\mathbf{h}}(n))^T\mathbf{x}(n-k) + \underbrace{(\mathbf{h}_{//} * \mathbf{h})^T(n-k)\mathbf{x}(n-k)}_{\text{residual error due to LTI assumption}}\\ &= e_l(n-k) + e_{tv}(n-k) \end{aligned}$$

where "*" is the notation for the convolution operator used her to keep the simplicity of the equation and $\mathbf{u} = [1, 0, \cdots, 0]^T$ with same length as $\mathbf{h}_{//}(n)$. Note that $\hat{\mathbf{h}}(n)$ is not the same estimate as in the linear case since the estimation is perturbed by the presence of non-linearity. It is assumed, however, to be a close estimate of the current $h_{nl}(n)$ which is different to $h_{nl}(n-k)$. Hence $\mathbf{e}(n)$ is given by the summation of two vectors $\mathbf{e}_l(n)$ and $\mathbf{e}_{tv}(n)$, where $\mathbf{e}_l(n)$ is the error vector when the (LTI) assumption holds and $\mathbf{e}_{tv}(n)$ is a perturbation error vector when the assumption of time invariance does not hold (as in non-linear case). As the past directions and the next direction are not the same in the steady-state period the algorithm will introduce more perturbation compared to the case where only the current error sample $e(n)$ direction is used as in the NLMS. This will introduce a perturbation error which corresponds to one sample $e_{tv}(n)$ instead of a perturbation vector. The above analysis thus explains why NLMS introduces less perturbations.

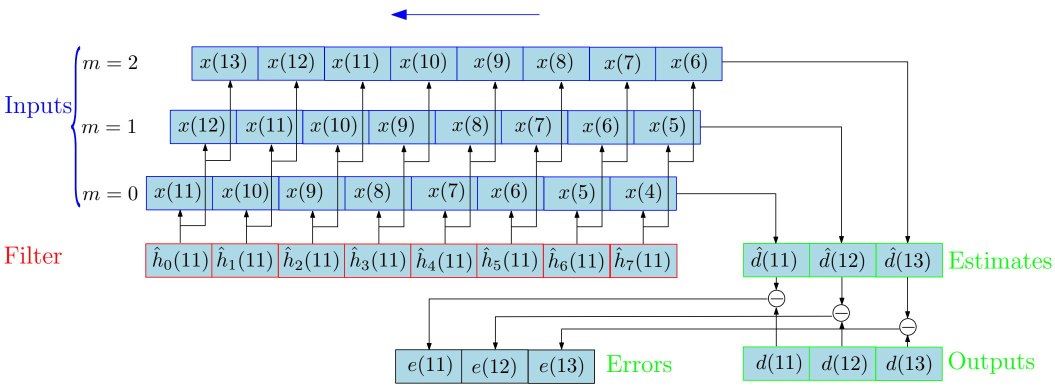


Figure 4.11: Illustration of BLMS procedure at time sample $n = 11$ with a block length $B = 3$.

In conclusion we may expect more perturbation if the order of the APA increases since the bigger the value of K (APA order), the higher the difference between $h_{nl}(n-K)$ and $h_{nl}(n)$.

Frequency block LMS

Figure 4.11 illustrates the time domain equivalent process of the FBLMS which is referred to as BLMS. A block length of 3 is assumed here for illustration only (efficient FBLMS is obtained when $B = N$ as used in the experimental tests). Compared to APA we remark that, for the BLMS the more m increases the more we exploit future samples meaning in real applications we introduce more delay, whereas for the APA the more k increases the more we use past samples which does not introduce delay.

The FBLMS has a similar problem as the APA, as a block-by-block process where the efficient block length corresponds to that of the EP length. In fact, for the FBLMS the error increases as a block error is computed and it is assumed that, inside this block, the system is stationary, which is not true when the EP is time variant. This will increase the error and make frequency domain fast convolution inefficient. Considering the BLMS we can write the gradient as:

$$\Delta\mathbf{h}(n,B) = \mu \sum_{m=0}^{B-1} e(nB+m)\mathbf{x}(nB+m) \tag{4.21}$$

In non-linear environment Equation 4.21 can be written as:

$$\begin{aligned}\Delta\mathbf{h}(n,B) &= \mu \sum_{m=0}^{B-1} e(nB+m)\mathbf{x}(nB+m) \qquad (4.22)\\ &= \mu \sum_{m=0}^{B-1} \mathbf{h}_{nl}^{T}(nB+m)\mathbf{x}(nB+m) - \hat{\mathbf{h}}^{T}(n)\mathbf{x}(nB+m)\end{aligned}$$

In Equation 4.22 since $\mathbf{h}_{nl}(nB+m)$ is time variant it is no longer possible to use the frequency domain fast convolution since it relies on stationarity. To use frequency domain convolution it is required that the filter is at least constant over the length of the Discrete Fourier Transform (DFT) thereby explaining the requirement of stationarity when using the DFT. This is only acceptable for an EP which is slowly time-variant compared to the block length. As in the APA we can derive the error introduced by each sample in the block due to the assumption of LTI:

$$e_{tv}(nB+m) = (\mathbf{h}_{//} * \mathbf{h})^{T}(nB+m)\mathbf{x}(nB+m) \tag{4.23}$$

The time variant EP explains also the problem with the FBLMS and shows that such systems are inefficient in non-linear environments. Compared to the APA, the FBLMS is highly affected as the block length of the FBLMS should be sufficiently long to satisfy the condition where it becomes less complex than the NLMS. The APA order is in general less than 10, due to an increase in complexity, in addition to the fact that the FBLMS is updated only every B samples instead of every sample. The second problem assumed with the FBLMS is that, in the frequency domain, the input frequency bins are assumed to be decorrelated, but in the presence of non-linearity this independence may be affected due to speech harmonicity. Hence all the bins will be affected by $DFT[(\mathbf{h}_{//} * \mathbf{h})^{T}(nB+m)\mathbf{x}(nB+m)] = \mathcal{H}_{//}(f) \times \mathcal{H}(f)\mathcal{X}(f)$ which means that the effect mainly depends on $\mathcal{H}_{//}(f)$. For speech signals it should be expected that $\mathcal{H}_{//}(f)$ has a greater effect on the position of speech harmonics meaning that the FBLMS will be highly perturbed as the harmonics of speech signal are generally high level.

APA and FBLMS results reported above show their inefficiency in non-linear environments. Another approach that is not presented in our test results is the Recursive Least Square (RLS). It also shows poor performance in presence of non-linearity [Niemistö & Mäkelä 2003b] as expected under tracking conditions [Haykin 2002, Eweda 1994].

Non-linear and noise effects

We have seen that the ERLE is better in non-linear conditions than in noise condition. These observations explained on account of the assumption that non-linearities are correlated with the far end speech signal. Hence we can write the microphone signal under the two conditions as:

$$y(n) = x(n) * h(n) + n(n) \text{ (noise)} \tag{4.24}$$

$$y(n) = x(n) * h(n) + g(x(n)) * h(n) \text{ (non-linear)} \tag{4.25}$$

with the condition that the SNR is equal to the SNeR meaning that $\frac{E\{x^2(n)\}}{E\{n(n)^2\}}$ equals $\frac{E\{x^2(n)\}}{E\{(g(x(n))*h(n))^2\}}$. With the assumption that the non-linearity is correlated with the far-end signal $x(n)$ we can write Equation 4.25 as:

$$\begin{aligned} y(n) &= x(n) * h(n) + g(x(n)) * h(n) & (4.26) \\ &= x(n) * h_{nl}(n) + [g(x(n))]_{\perp} & \\ & & (4.27) \end{aligned}$$

As $[g(x(n))]_{\perp}$ has lower energy than $g(x(n))$ this may result in a lower effect on linear AEC as compared to the noise. However, we have observed that the average SD in non-linear and noise environments are not overly dissimilar. This can be explained by the instability of the speech characteristics so that the mean overtime of $h_{//}(n)$ is equal to zero, hence $E\{h_{nl}(n) = (\delta(n) + h_{//}(n)) * h(n)\}$ tends to h^0 which is the linear optimal solution. This can be written as:

$$E\{h_{nl}(n)\} = E\{(\delta(n) + h_{//}(n)) * h(n)\} \tag{4.28}$$

As $h_{//}(n)$ depends on the signal characteristics and not on the environment after the loudspeaker we can assume that it is independent from $h(n)$. Hence Equation 4.28 can be written as:

$$\begin{aligned} E\{h_{nl}(n)\} &= (\delta(n) + E\{h_{//}(n)\}) * E\{h(n)\} & (4.29) \\ &= (\delta(n) + \underbrace{E\{h_{//}(n)\}}_{\approx 0}) * E\{h(n)\} & \\ &\approx E\{h(n)\} & \end{aligned}$$

This shows that, if the relation between non-linear and the linear components is not stable we can expect as given in Figure 4.9 that the SD of the linear AEC in the noise and non-linear environments to be similar in the mean. Thus we can reduce in a certain interval a part of the non-linear echo component without improving the estimate of linear echo. Note that the expectation used in Equation 4.29 is assumed to be done on a longer time process than that for the derivation of the Wiener solution in Equation 4.11.

This approach also shows that echo post-processing in non-linear conditions should take into account the problem of echo path deviations introduced by the correlation between non-linear and linear components. This will affect the assumption that the AEC can converge to the optimal solution and also the correlation assumption of the residual non-linearities and the linear component. In fact this explains that, when the linear AEC follows the variable path, the non-linear component may be reduced whereas a part of the linear component cannot be removed.

4.4.3 Frequency domain approach and echo post-filtering

This approach can also be analysed in the frequency domain as reported in [Mossi *et al.* 2010c]. The frequency domain filter can be shown to be given by:

$$\mathcal{H}_{nl}(f) = (1 + \mathcal{H}_{//}(f))\mathcal{H}(f) \tag{4.30}$$

where from [Mossi *et al.* 2010c] $\mathcal{H}_{//}(f)$ is given by:

$$\mathcal{H}_{//}(f) = \frac{\gamma_{G(x(n),f),X(f)}}{\gamma_{X(f)}} \tag{4.31}$$

where $\gamma_{G(x(n),f),X(f)}$ is the cross spectral density between the non-linearities and the far-end signal and $\gamma_{X(f)}$ is the spectral density of the far-end signal. Note that, as a frame-level approach, $\mathcal{H}_{//}(f)$ is time dependent. This shows that, when the frequency bins of the linear and non-linear components overlap, the Wiener solution will not converge to the linear optimal solution.

With the frequency domain solution in Equation 4.30 we can easily understand the results reported in [Yemdji *et al.* 2010] where we use different frequency domain Wiener filter approaches for echo suppression and for residual echo suppression. When the Wiener solution is applied in a linear environment its provides less echo reduction compared to when it is applied in a non-linear environment. This shows that a part of the non-linear component can be reduced and, since the Wiener solution does not take into account the phase difference, echo suppression approaches become more robust in reducing the non-linear component.

In [Yemdji *et al.* 2010] it is also reported results where a linear AEC is combined with a post-filter based on the frequency domain Wiener approach referred as residual echo suppression. It is shown that, after the AEC the difference in ERLE between the linear and non-linear environment is about 10 dB but, after post-filtering, it is only about 6 dB meaning that the post-filter can reduce more non-linearities compared to the AEC. This can be explained by the fact that the AEC is more perturbed by the variability of the echo path introduced by the non-linear component which is correlated with the far-end signal. This is not the case for the post-filtering which estimates a new filter for each frame. However, the presence of an uncorrelated component in the non-linear component leads to the fact that we have better results when we use a post-filter in linear conditions than non-linear conditions. A better ERLE is thus expected in linear environments than in non-linear environments.

This theoretical analysis also shows that the solution proposed in [Hoshuyama & Sugiyama 2006a, Hoshuyama & Sugiyama 2006c] is compatible to the approach described above. The approach in [Hoshuyama & Sugiyama 2006a] uses the linear estimate of the echo signal to reduce the non-linearities. This approach shows the effectiveness of their results in removing the correlated non-linear component. This also shows that, when a non-linear post-filter is used one should take into-account the correlation between the linear echo component and the non-linear echo component in the estimation procedure. If not respected this may lead to the over or under estimation of the non-linear component or residual linear component.

4.5 Conclusions

This section provides an assessment of linear AEC performance in non-linear environments modelled by a polynomial approximation. We compare the performance of four common standard algorithms. Experimental results show that APA achieves similar performance to NLMS in highly non-linear environments. The performance of FBLMS collapses even for relatively small non-linearities. We also show that, in the presence of non-linearities, the linear echo component is not well estimated by conventional approaches to AEC.

In noisy environments, however, there is little difference between each approach and, being less computationally demanding than the other approaches, FBLMS is an appealing solution in this case. We also show that, as the level of perturbations increase, performance decreases in both non-linear and noisy environments. Nevertheless, the echo canceller seems to be more robust to non-linearities than noise with a similar SNR (with the exception of the FBLMS algorithm). We show that the linear component of the EP is under estimated but is as accurate in the case of non-linear echo as it is in noisy environments, again with a similar SNR. In addition, as the non-linear component is correlated with the far-end signal a fraction of non-linearities are effectively attenuated. Noise, in contrast, is neither correlated, nor attenuated.

Finally we show how non-linear echo cancellation can be addressed as through time varying filter estimation and that this approach has potential to bring improvements in non-linear environments. Given the correlation between the input speech signal and non-linear echo, this model illustrates why echo cancellers are less perturbed by non-linear echo than they are by additive noise. An important consideration is that the effectiveness of such a model is based only on the existence of $[g(x(n))]_{//}$ and not on the model itself, even if, depending on the non-linearities, it may or may not be the dominant component. This is generally the case with speech due to its harmonicity. This leads us to question the common application of linear AEC to cancel the linear component in non-linear environments.

The fact that linear AEC performance decreases in non-linear environment shows the requirement of developing algorithms that take into account non-linearities. However, this requires to define the main sources of non-linearities and their characteristics. This is the objective of the next section.

Chapter 5

Static modelling of the loudspeaker

The objective in this chapter is to define the main sources of non-linearity in the Loudspeaker Enclosure Microphone System (LEMS) and to propose an appropriate model, in this case static. The LEMS is described as a cascade of three different systems: the down-link path, the acoustic channel and the up-link path. Each part is classified according to the literature as linear or non-linear. It is well known for example that the loudspeaker is the main source of non-linearities. Measurements made to compare the distortions introduced in the down-link to those introduced by the up-link confirm that the loudspeaker introduces significantly more non-linearities than the microphone. To understand these non-linearities we review the electrodynamic loudspeaker literature and present their main sources. Finally, based on our literature review we propose a discrete, static model for the loudspeaker which is presented in [Mossi *et al.* 2010d].

5.1 LEMS components

The most determinant parameters of the Acoustic Echo Cancellation (AEC) are related to the LEMS which itself involves different components. The LEMS can be divided into three main paths. The down-link path involves the components after the AEC reference input to the loudspeaker. The acoustical channel is based on the characteristics of the near-end environment and groups together all influences on the loudspeaker signal which is coupled to the microphone. The up-link path is composed of the components from the microphone to the reference echo point of the AEC. All these components are described further below.

5.1.1 Down-link path

The Down-Link (DL) path is generally composed of the Digital-to-Analog Converter (DAC), an analogue amplifier and a loudspeaker. The DAC is assumed to introduce low distortion in the original signal as quantization noise. The amplifier in the analogue domain can introduce clipping distortion. Most of the distortions introduced by the amplifier are generally clipping distortions due to the low electrical power of the mobile. The clipping effect can in fact be well modelled but is generally complex due to the hysteresis effect [Pillonnet *et al.* 2008, Burrow & Grant 2001]. The hysteresis effect introduces memory in the distortion which then increases the complexity of the model. A simplification involves modelling amplifier distortion as a simple hard clipping. This hard clipping is studied further in Section 6.3.1.

The loudspeaker is also a source of distortion; in fact it is one of the LEMS devices which has received the most attention in the literature. Many papers have focused on the source of loudspeaker distortion and have proposed solutions to model them. This is explained by the fact that, instead of being part of the AEC problem, modelling of loudspeaker distortion is itself a wide area of research. Hence, non-linear AEC sometimes relies on solutions proposed in the loudspeaker area to propose a model for the LEMS. In Section 5.3 we will present the most widely used model of the electro-dynamic loudspeaker and the different source of non-linearities that have been investigated in the literature.

5.1.2 Acoustic channel of near-end environment

The acoustic channel is generally assumed to be linear but it has been shown in [Birkett & Goubran 1994] that it may introduce some non-linearities due to enclosure vibrations. These non-linearities are very difficult to model and resemble as noise-like distortion and may severely reduce the Echo Return Loss Enhancement (ERLE) of linear AEC [Birkett & Goubran 1994]. Results in [Birkett & Goubran 1994] show, however, that non-linearities from the loudspeaker have less effect compared to those of the enclosure. Another constraint that is introduced by the enclosure vibration is echo path changes. Even if we assume that non-linearities can be well modelled, enclosure vibration will introduce Echo Path (EP) instability and so a simple adaptive filter cannot easily track such non-linearities. Enclosure distortions are not considered in our study and the acoustic channel is assumed to be linear.

5.1.3 Up-link path

The up-link path is composed of three components: microphone, amplifier and Analog-to-Digital Converter (ADC). The microphone is a simple transducer which converts the acoustic signal to the electrical domain. The microphone is sometimes based on similar transduction elements as the loudspeaker, it is nevertheless considered to introduce little distortions. Microphone distortions have accordingly been for less studied [Ravaud *et al.* 2009]. Research in the microphone field is mainly focused on the directivity problem than non-linearity. In any case microphone signals are generally low level and do not introduce significant distortion. In general the up-link amplifier is also used in the range where it can be considered as linear but distortion will nevertheless be introduced by the ADC is comparable to quantization noise.

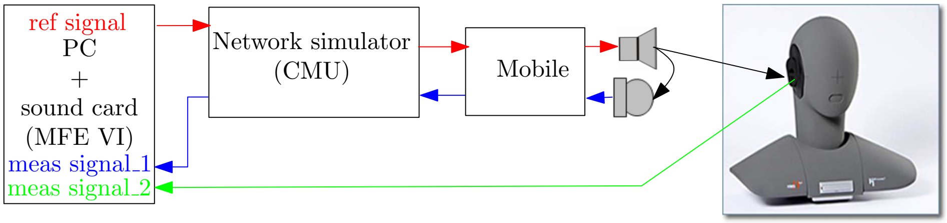


Figure 5.1: Global system, receiving direction

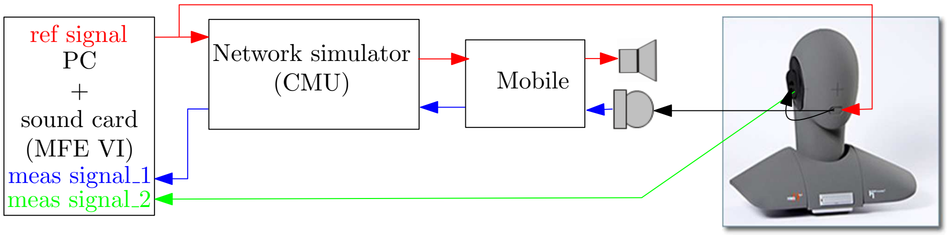


Figure 5.2: Global system, sending direction

5.2 Analysis of real device distortion

This section presents an analysis of the LEMS based on mobile phone measurements. The objective is to define components that generate more distortions. Hence, they can be further modelled.

Real device measurements consist of verifying the assumption that the loudspeaker introduces more non-linearities than the microphone. In this measurement the mobile phone is fixed. Signals are sent directly to the loudspeaker of the terminal or the mouth of a mannequin and recorded with the mobile phone microphone or a reference microphone in the ear of the mannequin. The signals sent to the loudspeaker are referred to as receiving direction signals as illustrated in Figure 5.1 whereas those from the mouth of the mannequin are referred to as sending direction signals as illustrated in Figure 5.2.

5.2.1 Experimental set-up

The systems used for all of our experiments are illustrated in Figure 5.1 and 5.2. A Personal Computer (PC) is used to store and record all audio data that is sent to, or received from a mobile terminal via an MFE VI sound card [HEAD acoustics 2008] and a network simulator [ROHDES&SCHWARTZ 2008]. In Figure 5.1 a signal is played by the PC, transmitted through the network simulator to the mobile and then played by mobile terminal loudspeaker. The loudspeaker output is then recorded with an independent, high-quality microphone mounted in the ear of a mannequin [ITU 1996] and with the mobile microphone which sends its signal back to the PC via the same network simulator. As illustrated in Figure 5.2 the signal played by the PC is sent to the mannequin and played by the loudspeaker in the mouth. The signal is again recorded by the two microphones.

The mobile terminal is placed at a distance of 20 cm from the mouth, i.e. in hands-free mode rather than handset mode, and all speech enhancement processes are deactivated. Since we aim to verify the source of distortions in the LEMS we first verified the linearity of all other system, or channel elements. The sampling frequency of the input signals is 48 kHz. When using the loudspeaker this is converted in the network simulator to 8 kHz according to GSM specifications then recorded at 48 kHz at the ear of the mannequin and at 8 kHz at the mobile microphone.

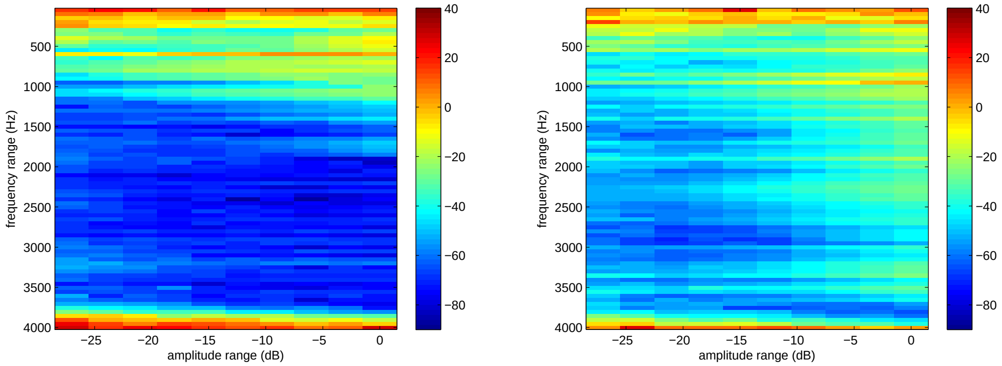
System linearity

In addition to the non-linear distortion introduced by the loudspeaker, various other non-linear signal processing algorithms, such as the speech codec (here the Enhanced Full-Rate codec), may also contribute distortions and thus corrupt the model of distortions introduced specifically by the loudspeaker. Therefore, it is necessary to determine amplitude and frequency ranges where the other system elements can be considered to behave linearly. Any distortions under these conditions can thus be reliably attributed to the loudspeaker only. To determine the linear range we conducted some non-intrusive tests where artificial, pure sinusoidal signals were sent to the mobile terminal but were recorded in digital form immediately before the loudspeaker. Signals with different amplitudes and frequencies were considered. By comparing the single sinusoidal input to the output we can easily observe any non-linear behaviour and thus determine amplitude and frequency ranges for which the system can be assumed linear. Our experimental results show that the system is effectively linear for the full amplitude range between the frequencies of 200 Hz and 3700 Hz.

5.2.2 Device measurements

The non-linear behaviour of the loudspeaker and microphone is thus observed by repeating the same experiment described above but where signals are recorded after the loudspeaker or the mannequin. Here we consider single sinusoidal test signals with one of 10 different amplitudes in the range of 0 dB (full-scale) to -27 dB with a step size of -3 dB and one of 80 different frequencies within the range of 50 Hz to 4000 Hz with a step size of 50 Hz. Each of these signals may be denoted by $A_{i,ref}e^{2j\pi f_{i,ref}}$ where $A_{i,ref}$ is the amplitude and $f_{i,ref}$ is the frequency. This amounts to a total of 800 test signals. In order to observe the resulting harmonics the output signals are transformed into the frequency domain. Measured signals are then used to compute the total harmonic distortion.

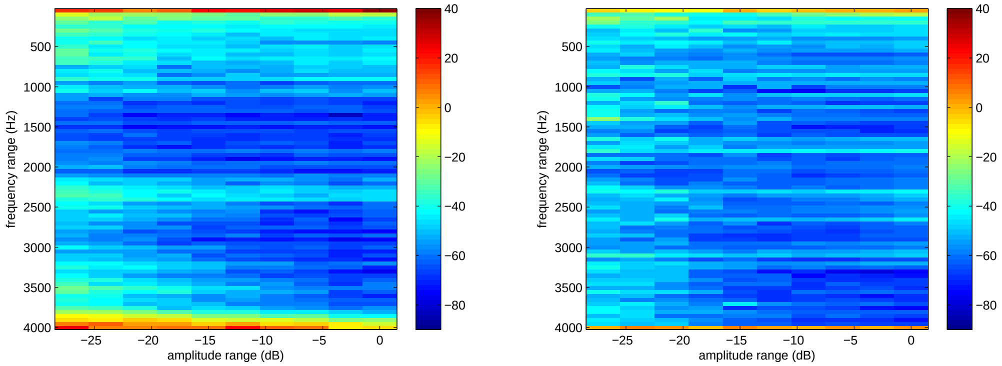
Preliminary experiments showed that loudspeaker non-linearities are sufficiently modelled by considering up to 6^{th} order harmonics. This is explained by the use of a 4000 Hz sampling frequency meaning that many higher frequency harmonics will not be recorded by the mobile microphone, which is limited in most current mobile phone terminals to frequencies lower than 4000 Hz. The THD is computed over 6



(a) mobile microphone

(b) mannequin ear

Figure 5.3: THD in dB measured in the receiving direction



(a) mobile microphone

(b) mannequin ear

Figure 5.4: THD in dB measured in the sending direction

harmonics and is given by:

$$THD = \frac{\sum_{p=2}^{6} \left\| A_{i,ref}^{2} e^{2j\pi p * f_{i,ref}} \right\|}{\left\| A_{i,ref}^{2} e^{2j\pi f_{i,ref}} \right\|} \tag{5.1}$$

where $A_{i,ref}^{2} e^{2j\pi p * f_{i,ref}}$ is the estimated level of the p^{th} harmonic in the measured signal. The THD is computed over all frequencies and amplitudes.

Receiving and sending direction THDs are illustrated in Figure 5.3 and 5.4 respectively. Upon their comparison, we observe that the receiving direction signal is the more disturbed illustrated by greater amount of yellow and red. The loudspeaker thus introduces more distortion. We also observe that, in the receiving direction, distortion is greater at higher amplitudes. In the sending direction, however, some low distortions are present for low level signals. These small distortions can be explained by the fact that low level signals are very sensible to estimation error and noise.

When comparing the distortions in the mobile phone microphone (Figure 5.3 (a) and 5.4 (a)) to that of the mannequin ear (Figure 5.3 (b) and 5.4 (b)), we observe that the mobile phone microphone introduces more distortions at lower frequencies (under 1000 Hz) than the reference microphone. For frequencies in the range of 1000 Hz to 3500 Hz, however, the reference microphone signal seems to be more distorted. This is explained by the limited sampling frequency of the mobile microphone. As the ear microphone and the mobile microphone have different sampling frequencies it is therefore normal that the ear microphone signal is the more distorted. In fact for the mobile terminal microphone all second harmonics generated above 2000 Hz and third harmonics generated above 1400 Hz will be filtered as the mobile phone sampling frequency is limited to 4000 Hz. The reference microphone provides a clearer picture of the loudspeaker distortions. It shows that, even if it is true that loudspeaker distortions are generally localized in low frequencies with high level signal they can be significant until 2500 Hz, as shown in Figure 5.3 (b).

Figures 5.3 (a) and (b) show that the recorded signal from the mobile microphone has more distortion in the lower frequencies (< 1000 Hz). This difference can be explained by the effect of the acoustic path which is different for the microphone. As explained previously the middle range difference is due to the sampling frequency of the mobile microphone.

Figure 5.4 (a) shows that the mobile phone microphone can be considered as linear even if we can observe that it is not as perfect as the reference microphone (Figure 5.4 (b)). More distortion is also noticed for lower frequencies (< 200 Hz) than higher frequencies (> 3700 Hz). These distortions are generally introduced by the channel before the loudspeaker and should not be taken into account.

These measurements show that the loudspeaker introduces more non-linearity than the microphone and justifies why this work focuses on loudspeaker distortions. To understand these non-linearities a study of the loudspeaker is required but is beyond the scope of this work. Hence we present an overview of the electro-dynamic loudspeaker model and the source of non-linearities in the literature review.

5.3 Electro-dynamic loudspeaker

This section presents a literature review of the electro-dynamic loudspeaker and an appropriate model. Loudspeakers are transducers that are used to convert electrical signal to acoustic signal. Different types of transduction exist and lead to different types of loudspeaker including e.g. piezo-electric, electromagnetic, electrostatic and electro-dynamic.

The most widely used loudspeaker is the electro-dynamic loudspeaker which is based on the principle of electromagnetic induction. Its popularity is due to its low-cost and robustness, reliability in the corresponding human ear frequency range. This popularity has been boosted by the development of neodymium magnets which

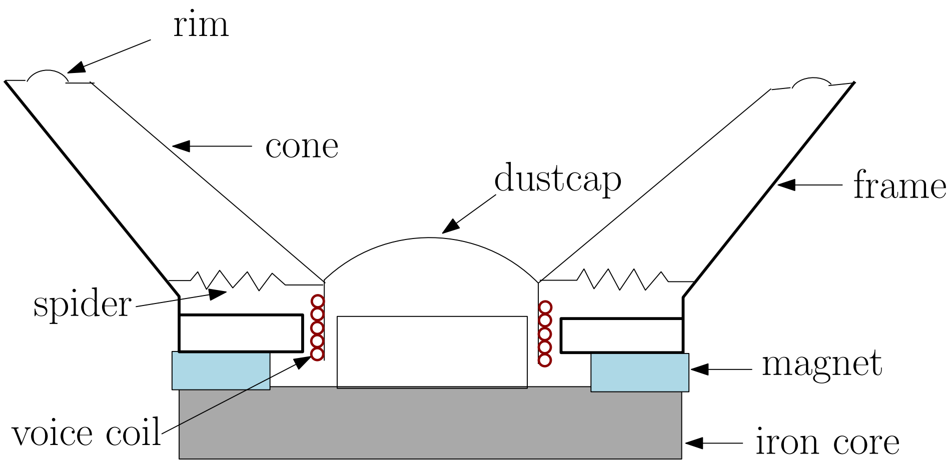


Figure 5.5: Electro-dynamic loudspeaker.

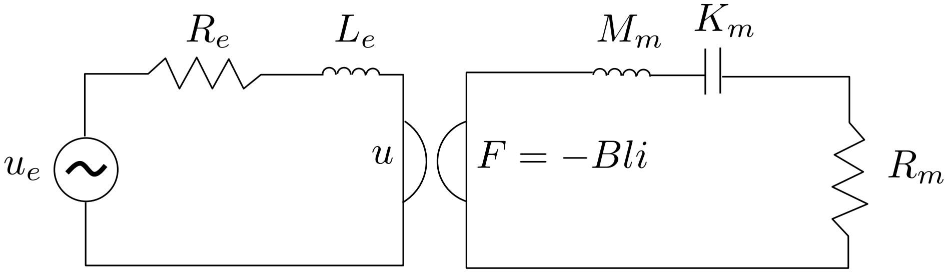


Figure 5.6: Thiele and Small model of the electro-dynamic loudspeaker.

allows very light speakers and greater efficiency which is of particular appeal in the mobile phone market. The electro-dynamic loudspeaker represents 99% of the loudspeaker market [Quaegebeur 2007].

An illustration of a typical electrodynamic loudspeaker is illustrated in Figure 5.5. Speakers used in mobile phones, however, do not typically include a rim. A discussion of non-linearities and more specifications can be found in [Bright 2002].

5.3.1 Electro-dynamic model

The electro-dynamic loudspeaker is based on electro-magnetic induction. An induction force is imposed on an element traversed by a current in a magnetic field. The transduction elements include the magnet, the voice coil and the cone. The magnet generates the magnetic field necessary for the induction. The voice coil is fixed on "fine" aluminium paper which is connected to the diaphragm. The diaphragm is supported by the frame via suspensions; the spider for the inner suspension and the rim for the outer suspension.

When the voice coil is traversed by an electrical current, due to the presence of the magnetic field a force is generated (Lorentz force). This results in movement of the voice coil and the diaphragm and results in change in acoustical pressure.

This transduction is typically modelled as an electro-mechanical system proposed by Thiele and Small [Schurer 1997, Quaegebeur 2007]. The model is illustrated in Figure 5.6.

The elements in Figure 5.6 can be classified in the electrical or mechanical domain. A voltage u_e is applied to the loudspeaker. R_e and L_e are respectively the resistance and self-inductance of the voice coil. u is the voltage of the self-inductance when the voice coil is traversed by a current i. In the mechanical part Bl is the force factor which, in the Lorentz case and according to the loudspeaker geometry, gives the induction as $F = -Bli$. M_m represents the mechanical moving mass, K_m the stiffness of the spider and rim and R_m the mechanical damping. This model may include an acoustical impedance as given in [Schurer 1997] which is generally needed to make the difference between a vented cabinet and a closed cabinet model. This extension is generally required for a state space loudspeaker control and is not the objective in this thesis due to reasons explained later. The model has different extensions regarding whether the model is used with a closed cabinet or vented cabinet where the effect of pressure in the back of the loudspeaker is not the same. But this is beyond the topic of our work and we stay with the general model given by:

$$\begin{array}{rcc} u_e(t) & = & R_e i(t) + L\frac{di(t)}{dt} + Bl\frac{dx}{dt} \\ Bli(t) & = & M_m\frac{d^2x(t)}{dt^2} + R_m\frac{dx(t)}{dt} + K_m x(t) \end{array} \tag{5.2}$$

Non-linearities stem from the electrical domain and the mechanical domain, both of which are described below.

5.3.2 Electrical non-linearities

Electrical non-linearities are those which arise with electrical parameters, however, their effects may affect in the mechanical parameters.

Non-linearities of the force factor

In reality Equation 5.2 is an approximation based on the assumption of an uniform magnetic field which is indeed rarely the case. In fact, when the voice coil moves, a part of it may be far from the magnet and make the magnetic field to be non-uniform. In this case the force factor Bl cannot be assumed linear. Two non-linear models have been proposed. A polynomial expansion model given by:

$$Bl(x) = Bl_0 + Bl_1 x + Bl_2 x^2 \cdots \tag{5.3}$$

and a Gaussian model given by:

$$Bl(x) = Bl_0\, \mathrm{e}^{-\mu(x-x_0)^2} \tag{5.4}$$

The latter is assumed to have a better approximation of the real model but is practically expensive and less popular.

Non-linearities of the self-inductance

When the self-inductance moves and is traversed by a current this will create another magnetic field that is opposed to the movement. This can be formulated as a self-inductance which depends on the current and affects the voltage of the self-inductance. Hence, the voltage of the self-inductance is given by:

$$\begin{aligned} u(t) &= \frac{dL_e(x(t))i(t)}{dt} \\ &= \frac{dL_e(x(t))}{dt}i(t) + L_e(x(t))\frac{di(t)}{dt} \\ &= i(t)\frac{dL_e(x(t))}{dx}\frac{dx(t)}{dt} + L_e(x(t))\frac{di(t)}{dt} \end{aligned} \tag{5.5}$$

due to the dependency of the movement of the self-inductance another additional force in the mechanical parameter appears which is derived from the energy:

$$\begin{aligned} F_r(x,t) &= \frac{\partial W_l}{\partial x} \\ &= \frac{\partial(\frac{1}{2}i^2 L_e)}{\partial x} \\ &= \frac{1}{2}i^2\frac{dL_e}{dx} \end{aligned} \tag{5.6}$$

$F_r(x,t)$ is the reluctance force which affects the mechanical part and is opposed to the induction force $F = -Bli$ which is due to Eddy current. As the voltage at the self-inductance and a reluctance force are generated this introduces some additional component in the electrical and mechanical parts respectively of the Thiele and Small model. These non-linearities are also modelled by a polynomial expansion:

$$L_e(x) = L_0 + L_1 x + L_2 x^2 \cdots \tag{5.7}$$

5.3.3 Mechanical non-linearities

Mechanical non-linearities arise due to variations in the mechanical parameters such as stiffness and mass deformation due to the movement of the diaphragm.

Non-linear stiffness

In the linear case the stiffness is assumed to be linear, but in reality the stiffness is excursion dependent. Hence, like the self-inductance, it is also modelled with a polynomial expansion series as:

$$K_m(x) = K_0 + K_1 x + K_2 x^2 \cdots \tag{5.8}$$

The model of the stiffness is difficult when all other parameters that may influence the stiffness such as repeatability, temperature and others, are taken into account.

Other sources of non-linearity

Mechanical clipping is generally avoided but may arise if the excursion reaches the limit of the diaphragm and extension. Mass variation non-linearities also affect the mechanical part of the loudspeaker as the resulting force which depends on the mass will become excursion dependent. Some other non-linearities are also introduced in the acoustical part such as Doppler distortion and non-linear wave propagation.

Non-linear analysis approaches

Many approaches have been proposed for loudspeaker modelling, i.e. the state space and Volterra models. The state space model usually requires an access to the loudspeaker components. The control of such models requires a preliminary knowledge of the parameters and the behaviour of the distortions. The advantage of such a model is mainly the number of parameters which are smaller, but is not well fitted to AEC applications. As it requires knowledge of different parameters and if these parameters are not well defined, it may introduce instability. In [Kajikawa 2011] it is shown that the Volterra model may have better performance for the non-linearity reduction of the loudspeaker. The Volterra filter has been widely used for non-linear identification and the capability to model loudspeakers has been demonstrated in many papers. Next we present two different approaches to model loudspeaker's non-linearity.

5.4 Loudspeaker distortion modelling

Loudspeakers convert electrical signals into sound but may introduce distortions. With the miniaturization of mobile terminals the linearity of the loudspeaker is often adversely affected and, at sufficient levels, the associated non-linear distortion can become disturbing for the near-end listener. Linearity is also important for digital signal processing (DSP) algorithms which assume linear conditions. Therefore, without appropriate compensation, the performance of all downstream processes, will also be adversely affected, e.g. as in echo cancellation [Mossi *et al.* 2010a].

One approach to mitigate such distortion involves loudspeaker linearisation techniques which all rely on non-linear modelling of the loudspeaker. Modelling typically involves an electro-acoustic and mechanical study of the loudspeaker to characterise its behaviour in non-linear conditions. These approaches, however, are generally too complex due to the high number of parameters which need to be estimated and the complex relationship between the electro-acoustic and mechanical properties as described in Section 5.3. The general conclusion of such studies show that loudspeakers are adequately characterised using a Volterra series for weak non-linearities and researchers have proposed many different loudspeaker models via such approaches [Frank 1994, Gao & Snelgrove 1990].

We present two non-linear loudspeaker models which are both based on practical studies of input-output characteristics. The first model is based on frequency-domain, harmonic distortion modelling whereas the second approach is based on parallelized polynomial filters to model harmonic distortion. Both models are derived from the same set of empirical observations and are compared to real system outputs in order to demonstrate their effectiveness in predicting non-linear distortion in speech signals.

This section presents a system set-up for loudspeaker modelling which is used to collect practical examples of non-linear loudspeaker distortion from real mobile terminals. This data is used to derive the two non-linear loudspeaker models. The section ends with an assessment of the two approaches by comparing loudspeaker outputs for real speech signals to those generated according to each of the two models.

5.4.1 System characterization

Here we describe the experimental test that was used to acquire the empirical observations from which the two models are derived.

Set-up for loudspeaker modelling

The system used for the loudspeaker characterization is as illustrated in Figure 5.1 and the set-up is similar to that presented in Section 5.2.1. The difference is that, for the mobile loudspeaker characterization, only the reference microphone is used to record the signals. In these experiments the mobile terminal is placed in close proximity to the reference microphone, i.e. in handset mode rather than hands-free. Handset mode is used to reduce the effect of the acoustic channel which may have different effects on the fundamental components and their harmonics and lead to an acoustic-channel-dependent characterization. As in the previous set-up the other devices are deactivated (e.g. speech enhancement) and the linearity condition of the channel without the loudspeaker is again necessary (see Section 5.2.1).

A similar approach is again used i.e. the THD computation described in Section 5.2.2. The same sinusoids are sent to the loudspeaker and recorded at the mannequin ear. Again the measured signals are transformed into the frequency domain in order to isolate and characterise the different harmonic components. Then, according to the same quantized frequency scale, the amplitudes at the output are set into a matrix, one for each input amplitude. Each matrix element thus gives the amplitudes at the output for each of the 80 fundamental reference frequencies and their generated harmonics. These matrices characterise the non-linear behaviour of the loudspeaker and are the basis of the models that are described next.

Two models are described here: one is based upon a frequency domain approach and the other is based upon a polynomial approach.

5.4.2 Frequency domain model

The matrix model is based on the assumption that speech signals may be represented as a sum of sinusoids and thus that the non-linear effect of the loudspeaker may be

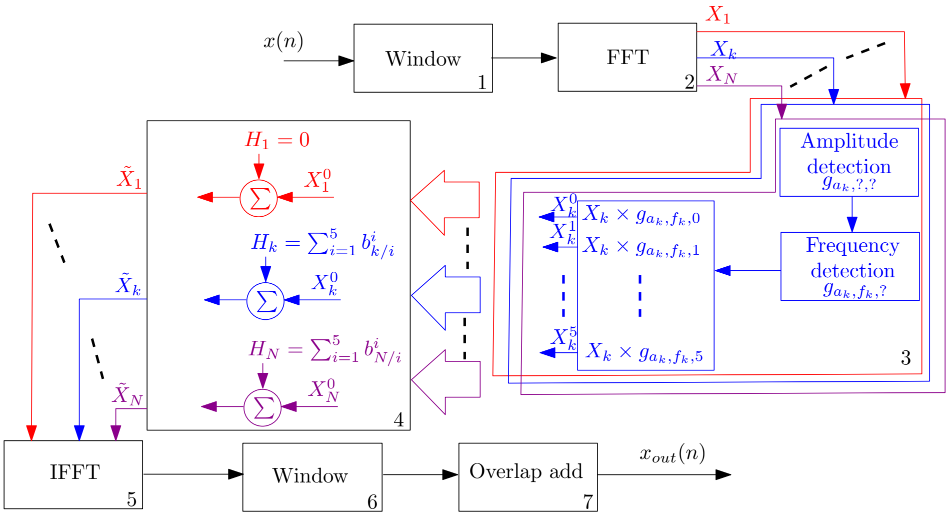


Figure 5.7: The frequency domain model. The input signal is windowed and transformed into the frequency domain where harmonic distortions are introduced according to the amplitude-dependent matrices. The notation $b_{k/i}^i$ represents the harmonic generated by the i^{th} frequency bin that will impact the k^{th} fundamental bin

modelled as the summed distortion of individual sinusoids. The decomposition into sinusoids is performed with the discrete Fourier transform (DFT) and the entire model is constructed in the frequency domain.

An overview of the system is illustrated in Figure 5.7. The input signal is first windowed into successive overlapping frames of length 40 ms with a frame rate of 48 kHz, corresponding to a frame overlap of 75%. Each frame is transformed into the frequency domain where each component is denoted by $X_i = A_i e^{2j\pi f_i}$ and where i is the DFT bin, A_i is the amplitude and f_i is the frequency. Then, for each frequency f_i, we determine the nearest quantised sinusoidal reference frequency $f_{i,ref}$, in addition to the nearest reference amplitude $A_{i,ref}$, i.e. we identify the '*closest*' or most applicable reference matrix. As explained in Section 5.2.1, each reference sinusoid at the input leads, at the output, to (i) a sinusoid at frequency $f_{i,ref}$ and amplitude $A_{i,ref}(0)$ and (ii) 5 harmonics at frequencies $(k+1) \cdot f_{i,ref}$ with corresponding amplitudes $A_{i,ref}(k)$, for $k = 1...5$. $A_{i,ref}(0)$ and $A_{i,ref}(k)$ are obtained directly from the matrices described in the previous subsection. We assume that, if $A_i \approx A_{i,ref}$ and $f_i \approx f_{i,ref}$, then $\frac{A_i(k)}{A_i} \approx \frac{A_{i,ref}(k)}{A_{i,ref}}$, and hence we obtain the fundamental and harmonics generated by $A_i e^{2j\pi f_i}$ using cross-multiplication with A_i and is given by:

$$\begin{aligned} A_i(k) &= \frac{A_{i,ref}(k)}{A_{i,ref}} \times A_i \\ &= g_{a_i,f_i,k} \times A_i \end{aligned} \tag{5.9}$$

where $g_{a_i,f_i,k} = \frac{A_{i,ref}(k)}{A_{i,ref}}$ is the gain applied to the k-th harmonic for an input signal of amplitude a_i and frequency f_i. This process corresponds to the 3^{rd} block in

Figure 5.7. By combining all of the harmonics generated by each of the reference signals (block 4 in Figure 5.7) we obtain an approximation of the non-linear distortion in the frequency domain. Finally, a time domain signal is then resynthesized by applying an inverse DFT with overlap-and-add.

5.4.3 Polynomial model

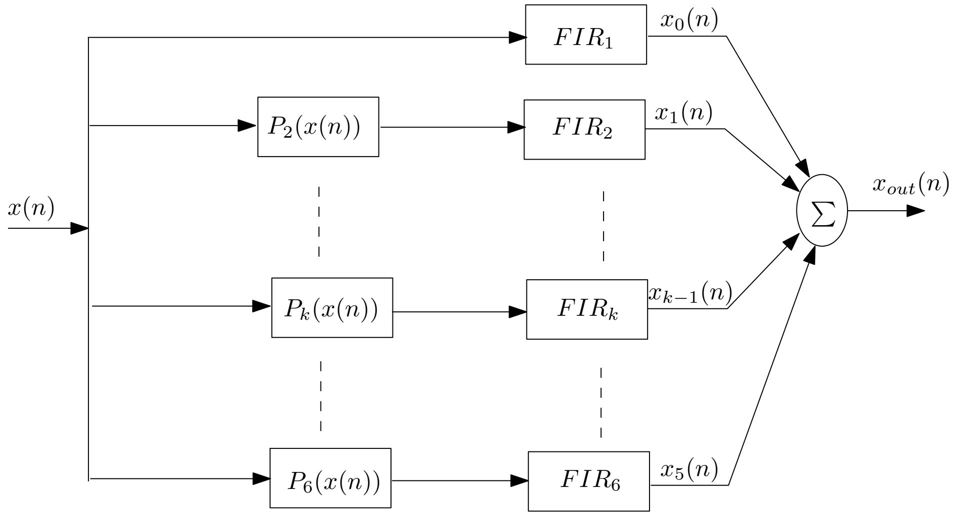


Figure 5.8: The polynomial model. The signal is processed in each stage by different polynomial and FIR filters

The so-called polynomial model is based upon a combination of polynomial and FIR filters. In contrast to the frequency domain model the idea here is to generate the different harmonics in the time domain according to different polynomial filters. The system is illustrated in Figure 5.8 where the polynomial filters are given by $P_k(x(n))$. Six parallelized branches aim to compute the linear response, $x_0(n)$, and the non-linear harmonics, $x_k(n)$. All signals are summed together with the original input signal to give the output $x_{out}(n)$.

The polynomial filter coefficients are determined according to the relationship between a cosine function at multiple frequencies and a cosine function at multiple powers:

$$cos(2\pi n \times f) = \sum_{i=0}^{n} \alpha_i cos^i(2\pi f). \tag{5.10}$$

Using trigonometric properties we determine the value of α_i for $n = 1, ..., 6$ (one fundamental frequency and five harmonics). These values correspond to the different coefficients in the polynomial model given by the Chebyshev polynomials as:

$$\begin{aligned}
P_1(x) &= x \\
P_2(x) &= 2x^2 - 1 \\
P_3(x) &= 4x^3 - 3x \\
P_4(x) &= 8x^4 - 8x^2 + 1 \\
P_5(x) &= 16x^5 - 20x^3 + 5x \\
P_6(x) &= 32x^6 - 48x^4 + 18x^2 - 1.
\end{aligned} \tag{5.11}$$

Without added filtering the amplitude of the generated harmonics is independent of the input frequency and so an additional bank of FIR filters is used to adjust their amplitudes.

If, for example, a particular range of input frequencies do not lead to any significant energy at the k-th harmonic, then a high-pass FIR filter, FIR_k, with high attenuation is applied to the output of the polynomial filter $P_k(x(n))$. For $k = 1$ the FIR filter is the impulse response which characterizes the coupling between the loudspeaker and the microphone in the ear of the mannequin.

To estimate the FIR filter coefficients we use reference signals to compute the gains, in a similar manner to that described at the beginning of Section 5.4.1 in Subsection "set-up for loudspeaker modelling". Filter gains are computed per harmonic using frame-by-frame Fast Fourier Transform (FFT)s of the input $(A_{i,ref}e^{2j\pi f_{i,ref}})$ and each individual output harmonic $(A_{i,ref}(k)e^{2j\pi k f_{i,ref}})$. Filter gains are then determined according to their average ratio:

$$G_k(f_l) = \frac{\overline{|A_{i,ref}(l)e^{j2\pi l f_i}|^2}}{\overline{|A_{i,ref}e^{j2\pi f_i}|^2}}, \tag{5.12}$$

where f_l is the frequency of the harmonic equal to $(k+1) \cdot f_i$. The FIR filter is then the minimum phase filter which reflects the determined gain profile. After the estimation of all filter coefficients the system output is easy to compute. The input signal is passed through each combined polynomial and FIR filtering stage and the sum of the resulting signals gives the system output.

5.4.4 Constraints and limitations

Before we assess each of the two models we describe the limitations of each approach and their potential accuracy. The limits are defined by the complexity of the model, i.e. the size of the harmonic matrix. For the frequency domain model this translates directly to the number of harmonics considered, which has a direct impact on system accuracy. The bigger the matrix the better the accuracy, but the more complex the model. For the polynomial model, accuracy depends on the number of stages and the length of the FIR filters. Increasing in the number of parameters will increase the complexity but less so than for the frequency domain model.

Finally, in the two approaches described above, intermodulation distortions are not considered. In the frequency domain model they are completely ignored. Some intermodulation distortions are generated with the polynomial model (though they were not fully considered directly in the design and polynomial parameter estimation). The only effect in this case is that they cannot be controlled independently from the harmonics.

5.4.5 Experimental work

To compare the two models we assess each of them with real speech signals that are played at the loudspeaker of a mobile terminal and recorded at the ear of the mannequin as described in Sections 5.4.2 and 5.4.3. The signals measured at the ear are compared to the results obtained according to the two models described above. Three different metrics are used to assess model accuracy. First, signals are assessed in the time domain in terms of the segmental Signal-to-Estimate Ratio (SER) given by:

$$SER(m) = 10 \times log_{10}\left(\frac{\sum_{i=m\times N}^{(m+1)\times N} x_{real}^2(i)}{\sum_{i=m\times N}^{(m+1)\times N} x_{model}^2(i)}\right) \tag{5.13}$$

where x_{real} is the speech signal recorded at the ear of the mannequin and x_{model} is the distorted speech predicted according to the model. Performance is also assessed in the frequency and cepstral domains through the log-spectral and cepstral distances. The Log-spectral Distance (LsD) is given by:

$$LsD(m) = \sqrt{E\{(L_{x_{real}}(m) - L_{x_{model}}(m))^2\}} \tag{5.14}$$

where

$$L_{x_s}(m) = 20 \cdot log_{10}(FFT[x_s(m \cdot N), ..., x_s((m+1) \cdot N)]) \tag{5.15}$$

and the Cepstral Distance (CD) is given by:

$$CD(m) = \sqrt{\sum_N [C_{x_{real}}(m) - C_{x_{model}}(m)]^2} \tag{5.16}$$

where

$$C_{x_s}(m) = IFFT\{ln|FFT[x_s((m \cdot N))...x_s((m+1) \cdot N)]|\} \tag{5.17}$$

The CD is intended to give a more perceptually-related assessment, or at least one which is better correlated to subjective assessment than the log-spectral distance. In Equations 5.15 and 5.17 the index s refers to measurements or estimates from real experiments or a model respectively. Measurements come from consecutive frames of 20 ms in length. For all experiments reported here performance is evaluated using a dataset of 3 speech signals with a total length of 1 minute.

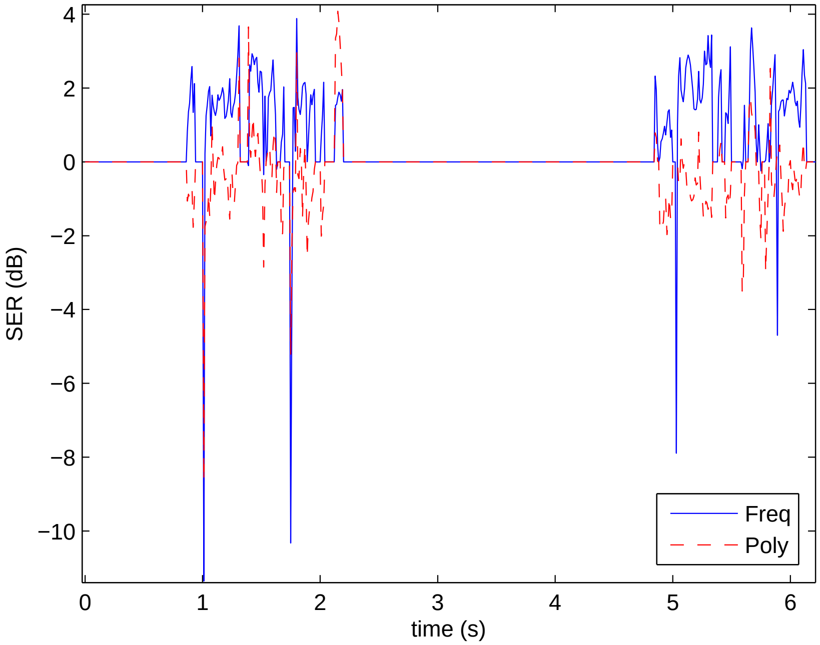


Figure 5.9: SER against time for the two loudspeaker models. The frequency domain distortion model underestimates the real output whereas the polynomial model overestimates the real output (An SER of 0 dB indicates accurate estimates).

Time domain assessment

The SER provides an impression of global system performance and, when plotted against time, profiles illustrate variation in the error against time between modelled and ground-truth distortions. Figure 5.9 shows a profile for an example speech signal which typifies performance across the whole speech dataset. The solid blue profile illustrates performance for the frequency domain model and the dashed red profile illustrates performance for the polynomial model. On average the two systems give similarly accurate distortion estimates: despite some deviations the SER for both models is generally within a margin of $+/-2$ dB. Figure 5.9 also shows that the polynomial model generally overestimates the distortion (SER< 0) whereas the frequency domain model generally underestimates the distortion (SER> 0). This can be explained by the complete absence of intermodulation harmonic estimation in the frequency domain model, leading to lower energies in x_{model} than in x_{real}. In contrast, the polynomial model leads to an overestimation of intermodulation, and consequently more energy in x_{model} than in x_{real}.

Overall, the two models lead to approximately the same amount of error with a mean absolute SER of 1.33 dB and 1.28 dB for the frequency domain and polynomial models respectively, for the complete speech dataset. The profiles in Figure 5.9 contain some significant troughs, especially for the frequency domain model (around

1, 1.8, 5 and 6 s for instance). Listening tests reveal that they typically occur only during speech/non-speech transitions, i.e. at 1, 1.8 and 5 seconds in Figure 5.9. This can be explained by the fact that the frequency domain model generates harmonics from the speech signal either side of the transition. Considering a silence/speech transition this leads to a form of pre-echo as the harmonics are generated for the entire frame being processed. This is the classical pre-echo effect inherent in frequency domain processing. Informal listening tests show that these transitions are generally less perturbing with the polynomial model, despite important differences that can still be noticed in the SER measurement during such periods.

Spectral and cepstral domain assessment

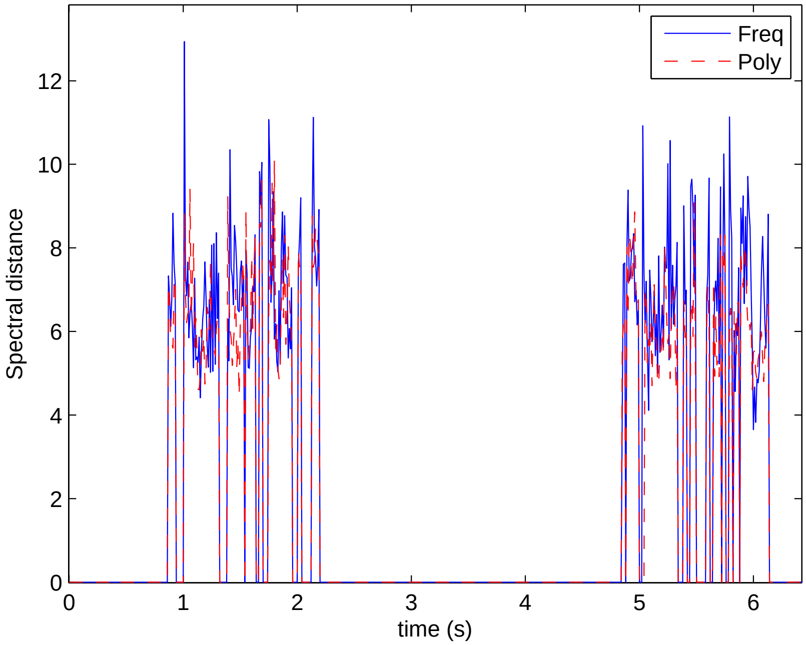


Figure 5.10: Frequency domain assessment with the log-spectral distance

In order to give an assessment that is more reflective of human perception we also computed LsD and CD to assess model accuracy.

Figures 5.10 and 5.11 show profiles for LsD and CD respectively, for each of the two models. As for time domain measurements with the SER, the two models show similar performance. LsD for both frequency domain and the polynomial models are relatively close (averages of 7.11 dB cf. 7.10 dB across the entire speech datasets). As illustrated in Figure 5.11, the CD between modelled and ground-truth distortions is reasonably similar. The global mean of the CD for this typical example is about 0.52 for the frequency domain model and 0.50 for the polynomial model. We found similar averages between 0.5 and 0.7 for the whole speech dataset.

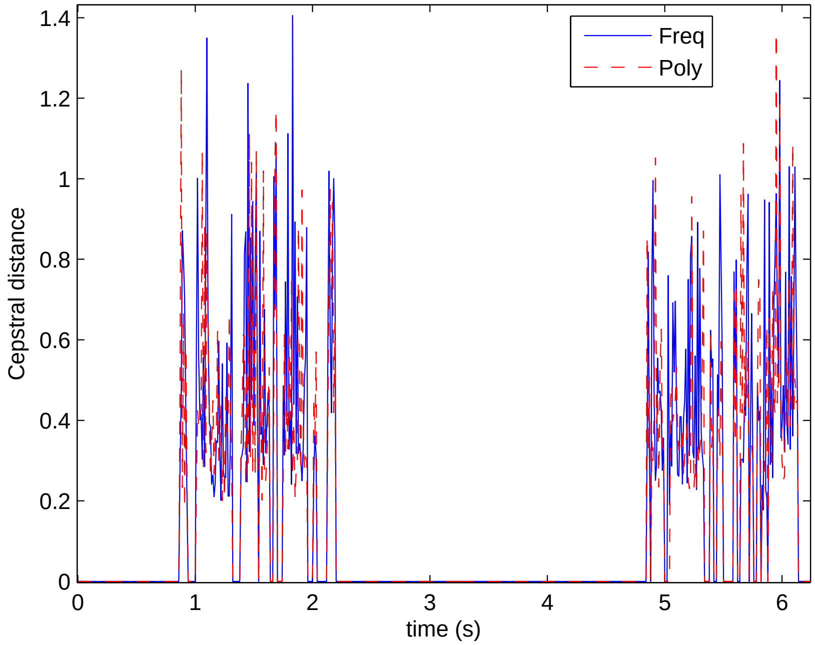


Figure 5.11: Frequency domain assessment with the cepstral distance

There are noticeable peaks in the LsD profiles. These peaks correspond to the peaks in the SER profiles, i.e. during transitions. The CD profiles, however, show more erratic behaviour. Even if the CD remains relatively low, such erratic behaviour can be explained by the fact that the CD better reflects human perception and is hence more sensitive to perceptual distortion than the other distances considered. The peaks appear during different periods for the two models. Even if the mean distances are similar, the CD reflects the fact that the deviations between modelled and real signals sound different for both models: the kind of deviation introduced by the polynomial model does not appear for the same kind of speech signal as for the frequency domain model. Listening tests confirm this assessment. On one hand, the polynomial model interferes with the timbre of the signal, sometimes overly exaggerating certain frequencies compared to real recorded signals. On the other hand the deviations introduced by the frequency domain model are more noticeable during transitions, even within the speech signal, for instance during transitions between voiced and unvoiced speech. In any case the CD is relatively small and the variation over time is not that high. This indicates that the two models give a good approximation of non-linear system behaviour. This conclusion is confirmed by listening tests during which the differences are audible, but the model outputs are comparable to the real recorded signals.

5.5 Conclusion

Using real measurements recorded using a real mobile phone we compare the distortions from the loudspeaker and the microphone and show that the loudspeaker is the main source of non-linearities. According to these results we make an overview of the electro-mechanical model of the loudspeaker and present the main sources of non-linearities. We then present two models of non-linear harmonic distortion in mobile terminal loudspeakers. Both models may be used to give relatively accurate predictions of loudspeaker behaviour, through a fixed set of coefficients determined empirically, and can be seen as a good first approximation of small loudspeakers. Nevertheless the models do not match perfectly with reality and thus there remains some potential for improvement. The lack of reliable intermodulation modelling seems to be the main drawback of both approaches.

The fact that these models are static is also a drawback in AEC applications as they are device dependent and cannot follow the changes that arise along time in the loudspeaker parameters. Hence, these solutions particularly the time domain model are used to develop non-linear AEC approaches that will be presented in the next chapter.

Chapter 6

Adaptive non-linear AEC

Non-linearities generally degrade the performance of most echo cancellation algorithms which are based on the assumption of linearity and thus the problem of non-linear echo cancellation has emerged as an increasingly important problem.

There are two main approaches to tackle the problem of non-linearities in the acoustic path. The first approach is based on non-linear post filtering to suppress the residual non-linear echo [Hoshuyama & Sugiyama 2006c]. In general the post-filter is preceded by a conventional linear adaptive filter. However, non-linearities have an adverse effect on linear filtering which impacts upon non-linear post processing and thus degrades global performance. The second, more popular approach is based on the use of a Volterra series and non-linear adaptive filtering [Stenger & Rabenstein 1998, Fermo *et al.* 2000]. Whilst there is less dependence on the performance of linear filtering the approach typically suffers from slow convergence. This lead us to focus on new structures which can utilise the model of the loudspeaker developed previously in Chapter 5. The model of the loudspeaker is used as a compensator for non-linearity pre-processing.

A Cascaded Structure (CS) is chosen here due to the lower number of parameters required compared to the Volterra filter in a parallel approach. The pre-processor proposed here is used in two different approaches. The first approach consists of pre-processing the linear Acoustic Echo Cancellation (AEC) input to emulate the loudspeaker non-linearity effects. The second consists of linearising the loudspeaker by pre-processing its input and thus allow to support the use of linear AEC.

The Volterra filter is discussed first as the baseline non-linear AEC approach. The Volterra filter is presented with an overview of its characteristics. We propose to first investigate the Volterra filter identification, then we focus on the special case of a Volterra filter derived from the concatenation of a non-linear system and a linear system. This approach is used in this thesis since, as described in Chapter 5, the main non-linearities are introduced by the loudspeaker and we consider that the rest of the Loudspeaker Enclosure Microphone System (LEMS) is linear. Then we introduce the CS which is split into two sections. The first presents a baseline cascaded structure developed in [Mossi *et al.* 2011a] whereas the second present an improved version where we combine the basic model of the loudspeaker with a clipping compensator and decorrelation pre-processing of the linear AEC input [Mossi *et al.* 2012]. We then present the loudspeaker pre-processing approach where linearisation pre-processing is applied to the loudspeaker input signal [Mossi *et al.* 2011b]. Finally, the last section presents a summary of the different algorithms which are assessed in the next chapter.

6.1 Volterra series approach

In this section we present the general second order Volterra filter and then focus on its application to acoustic echo cancellation. The objective is to highlight some characteristics of the Volterra filter which are already known in non-linear AEC applications and the limitations of the structure. We then justify the investigation of pre-processing approaches for non-linear AEC in the case of loudspeaker non-linearity.

The Volterra filter is an extension of the Taylor series to non-linear systems with memory. In general, eletrical and mechanical systems have memory and cannot be well modelled by Taylor series which is in generally used to model memoryless systems. The Volterra series takes cross term effect from past samples into account. This makes the Volterra filter more reliable for system modelling and is the most widely used non-linear model. Another feature which contributes to its appeal is the linearity in parameters which allows easy exploitation of linear system algorithms. The Volterra filter is widely used but the foundation of the Volterra series for engineering applications is reported in few papers such as [Schetzen 2006, Boyd *et al.* 1984, Boyd 1985].

This section is organized in two parts. The first part introduces the general Volterra series. In the second part we describe the application of the Volterra to non-linear AEC.

6.1.1 Volterra filter identification

While it may be readily extended to higher orders, we focus on quadratic Volterra filter identification. The objective is to understand the issues which arise when trying to identify the Volterra kernel. The output of the Volterra filter is given by:

$$y(n) = h_0 + \sum_{p=1}^{P} \sum_{l_1=0}^{N_p-1} \cdots \sum_{l_p=0}^{N_p-1} h_p(l_1, \cdots, l_p) x(n-l_1) \cdots x(n-l_p) \tag{6.1}$$

where $h_p(l_1, \cdots, l_p)$ is the $p-th$ order Volterra kernel, P represents the order of the Volterra filter (which is generally equal to 2 for complexity reasons) and N_p represents the memory of the $p-th$ non-linear kernel whose size corresponds to N_p^p. The particular kernel of the Volterra filter is the $0-th$ order kernel which corresponds to a constant value of the system (generally not used) and the 1^{st} order kernel which corresponds to the linear system filter. The Volterra filter can be considered as a Multiple Inputs Single Output (MISO) system where each input corresponds to a kernel and the output is the sum of the different kernel outputs. In most systems the Volterra filter is limited to the quadratic kernel such as on-line system identification. When the Volterra filter is limited to the quadratic kernel

$$\begin{array}{cccccc}
 & x(n) & x(n-1) & x(n-2) & x(n-3) & x(n-4) \\
x(n) & h_2(0,0) & h_2(0,1) & h_2(0,2) & h_2(0,3) & h_2(0,4) \\
x(n-1) & h_2(1,0) & h_2(1,1) & h_2(1,2) & h_2(1,3) & h_2(1,4) \\
x(n-2) & h_2(2,0) & h_2(2,1) & h_2(2,2) & h_2(2,3) & h_2(2,4) \\
x(n-3) & h_2(3,0) & h_2(3,1) & h_2(3,2) & h_2(3,3) & h_2(3,4) \\
x(n-4) & h_2(4,0) & h_2(4,1) & h_2(4,2) & h_2(4,3) & h_2(4,4)
\end{array}$$

Figure 6.1: Matrix representation of the second order Volterra kernel, $\mathbf{h}_Q^T(n)\mathbf{x}_Q(n)(N_Q = 4)$

(second order P=2) without a constant component it is given by:

$$y(n) = \underbrace{\sum_{m=0}^{N-1} h_1(m)x(n-m)}_{\mathbf{h}_1^T(n)\mathbf{x}(n)} + \underbrace{\sum_{l_1=0}^{N_Q-1}\sum_{l_2=0}^{N_Q-1} h_Q(l_1,l_2)x(n-l_1)x(n-l_2)}_{\mathbf{h}_Q^T(n)\mathbf{x}_Q(n)} \tag{6.2}$$

where all the parameters are defined as in Section 3.2.1. Here the quadratic kernel is denoted with the index Q to differentiate it from further notations used later. The quadratic kernel can be represented by a 2-dimensional matrix as illustrated in Figure 6.1. The figure shows the taps of the quadratic Volterra kernel where the output of the kernel is given by the summation two words taps, $h_Q(l_1, l_2)$ multiplied with the corresponding input signal samples at row, $x(n-l_1)$ and line, $x(n-l_2)$. In the figure the taps are divided into two colours, black (lower triangular taps) and blue (upper triangular and diagonal taps). This approach is used here to show that, in AEC applications it is not required to estimate the overall Volterra kernel, as will be explained latter. By transposing the matrix $\mathbf{h}_Q(n)$ in Figure 6.1 the output value will not change. This shows that the identifiability of filter $\mathbf{h}_Q(n)$ is not guaranteed when using error minimization techniques. This is due to the fact that, by switching $h_Q(l_1, l_2)$ and $h_Q(l_2, l_1)$ the result will not change. To show that let us suppose that $y_2(n, l_1, l_2)$ and $y_2(n, l_2, l_1)$ are given as:

$$y_2(n, l_1, l_2) = x(n-l_1)x(n-l_2)h_Q(l_1, l_2)$$

$$y_2(n, l_2, l_1) = x(n-l_2)x(n-l_1)h_Q(l_2, l_1)$$

$$y_2(n, l_1, l_2) + y_2(n, l_2, l_1) = x(n-l_1)x(n-l_2)(h_Q(l_1, l_2) + h_Q(l_2, l_1)) \tag{6.3}$$

As these two elements $(y_2(n, l_1, l_2), y_2(n, l_2, l_1))$ are summed in the final output $y(n)$ of Equation 6.2, their position do not affect the output as given in Equation 6.3. Additionally, whatever the couple $(t_Q(l_1, l_2), t_Q(l_2, l_1))$ such that $t_Q(l_1, l_2) + t_Q(l_2, l_1) = h_Q(l_1, l_2) + h_Q(l_2, l_1)$ satisfy the error minimization criteria. This poses a problem of identifiability of the Volterra kernel in general. But, as is generally the case for applications such as echo cancellation, proper identification is not an issue. Hence, as this identification issue does not change the value of the output, the performance in AEC applications will not be affected. The advantage of such a property is that it reduces complexity meaning that the filter can be reduced to around half of its elements. This is done by setting $t_Q(l_2, l_1) = 0$ and $t_Q(l_1, l_2) = h_Q(l_1, l_2) + h_Q(l_2, l_1)$. This approach is generally used to reduce the complexity of the Volterra filter in many approaches to non-linear system identification based on Volterra filtering [Kuech & Kellermann 2004, Zeller & Kellermann 2010a].

In AEC applications the Volterra filter is used as a model of the LEMS, even if some of the elements (echo path, microphone) of the LEMS may be assumed as linear, meaning that the Volterra filter may represent a concatenation of linear and non-linear systems (see section 3.1). It is generally used in a Parallel Structure (PS) where it can be considered as an MISO system whose parameters need to be identified. Hence, the first and second order adaptive identification procedure are as given in Equation 3.8.

$$\begin{aligned} \hat{\mathbf{h}}_1(n+1) &= \hat{\mathbf{h}}_1(n) + \mu_1 e(n)\mathbf{x}(n) \\ \hat{\mathbf{h}}_Q(n+1) &= \hat{\mathbf{h}}_Q(n) + \mu_Q e(n)\mathbf{x}_Q(n) \end{aligned} \tag{6.4}$$

where $\mathbf{h}_Q = [h_Q(i,j)]$ with $i,j = 0$ to $N_Q - 1$ is an $N_Q \times N_Q$ matrix as represented in Figure 6.1. These Equations show how the filter can be estimated using linear adaptive filtering approaches. Another approach consists of sub-dividing the quadratic kernel into sub-filters where each filter corresponds to a row of the matrix $\mathbf{h}_Q$ in Figure 6.1. In this case the adaptation process is given by:

$$\hat{\mathbf{h}}_{l_1}(n+1) = \hat{\mathbf{h}}_{l_1}(n) + \mu_{l_1} x(n-l_1)\mathbf{x}(\mathbf{n})e(n) \tag{6.5}$$

where l_1 is the index of the quadratic kernel matrix $\mathbf{h}_Q$ row vectors which vary from 0 to $N_Q - 1$. For each l_1, $\mathbf{h}_{l_1}(n)$ is given by:

$$\mathbf{h}_{l_1}(n) = [h_Q(l_1, 0), h_Q(l_1, 1), \cdots, h_Q(l_1, N_Q - 1)]^T \tag{6.6}$$

which corresponds to the rows of the matrix $\mathbf{h}_Q$. In this procedure it is easier to choose the step-size μ_{l_1} than the global identification procedure in Equation 6.4 where the normalization is done for the overall matrix.

6.1.2 Volterra filter for non-linear AEC

Figure 6.2 illustrates a simplified Volterra filtering approach for non-linear AEC. The loudspeaker is modelled by two filters: $h_1(n)$ which represents the linear filter and $h_Q(n)$ which represents the quadratic kernel. The Volterra filter is used for AEC and it is assumed to model the full LEMS. This includes the cascade of a non-linear system (loudspeaker) and a linear system (acoustic channel and up-link path) can be globally modelled as one non-linear system. The linear filter $\hat{h}_l(n)$ of the AEC should converge to the concatenation of $h(n)$ and $h_1(n)$ and the quadratic kernel to a two dimensional matrix as shown in Figure 6.2 (left) representing the concatenation of $h_Q(n)$ and $h(n)$. This approach has some impacts that have been observed in previous works such as [Stenger *et al.* 1999b]

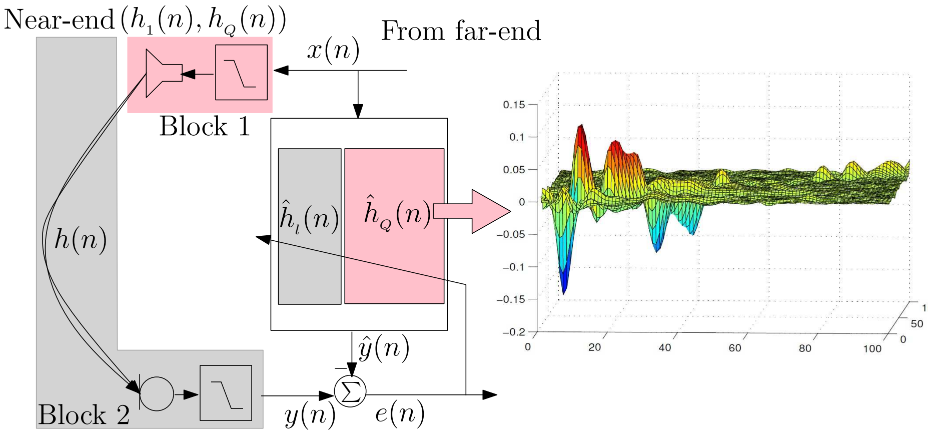
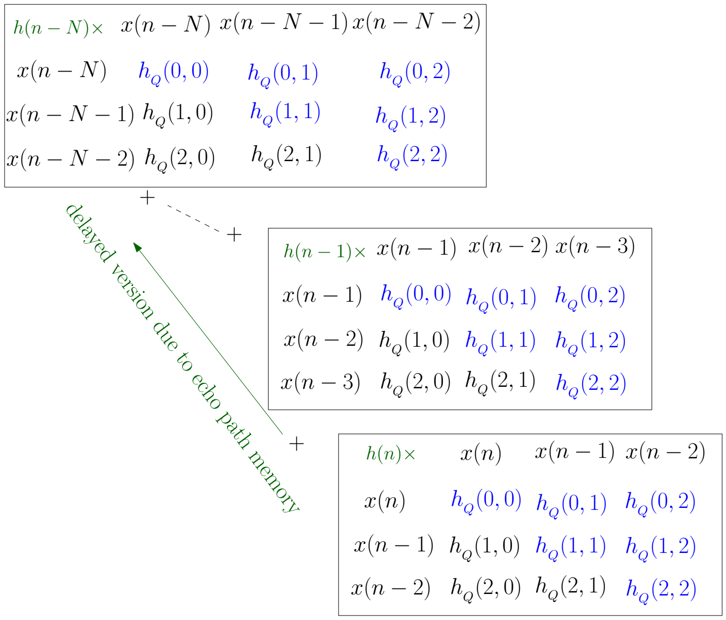


Figure 6.2: Volterra approach to non-linear echo cancellation. $\hat{h}_1(n)$ and $\hat{h}_Q(n)$ aim to estimate respectively $h_1(n) * h(n)$ and $h_Q(n) * h(n)$.



$$\begin{array}{cccc} h(n-N)\times & x(n-N) & x(n-N-1) & x(n-N-2) \\ x(n-N) & h_Q(0,0) & h_Q(0,1) & h_Q(0,2) \\ x(n-N-1) & h_Q(1,0) & h_Q(1,1) & h_Q(1,2) \\ x(n-N-2) & h_Q(2,0) & h_Q(2,1) & h_Q(2,2) \end{array}$$

$+$ $+$

$$\begin{array}{cccc} h(n-1)\times & x(n-1) & x(n-2) & x(n-3) \\ x(n-1) & h_Q(0,0) & h_Q(0,1) & h_Q(0,2) \\ x(n-2) & h_Q(1,0) & h_Q(1,1) & h_Q(1,2) \\ x(n-3) & h_Q(2,0) & h_Q(2,1) & h_Q(2,2) \end{array}$$

$+$

$$\begin{array}{cccc} h(n)\times & x(n) & x(n-1) & x(n-2) \\ x(n) & h_Q(0,0) & h_Q(0,1) & h_Q(0,2) \\ x(n-1) & h_Q(1,0) & h_Q(1,1) & h_Q(1,2) \\ x(n-2) & h_Q(2,0) & h_Q(2,1) & h_Q(2,2) \end{array}$$

Figure 6.3: Visualisation of the quadratic Volterra kernel of the loudspeaker, $\mathbf{h}_Q(n)$ concatenated with a linear channel $h(n)$. Each delayed version of the quadratic kernel, $\mathbf{h}_Q(n)$ is multiplied by the corresponding delayed filter coefficient of $h(n)$. As in the matrix representing $\mathbf{h}_Q(n)$ in Figure 6.1 we observe that even in the equivalent matrix of $\bar{\mathbf{h}}_Q(n)$ we still have the symmetry meaning that only taps in blue need to be estimated.

which proposes the truncation of the quadratic kernel to remove null coefficients which correspond to the delay between the loudspeaker and microphone. In [Kuech & Kellermann 2002, Kuech & Kellermann], Kuech proposes to simplify the Volterra filter to the elements around the diagonal using simplified Multi Mem-

$$\begin{array}{cccccccccc}
 & x(n) & x(n-1) & x(n-2) & x(n-3) & x(n-4) & --- & x(n-N) & x(n-N-1) & x(n-N-2) \\
x(n) & \bar{h}_Q(0,0) & \bar{h}_Q(0,1) & \bar{h}_Q(0,2) & 0 & 0 & --- & 0 & 0 & 0 \\
x(n-1) & \bar{h}_Q(1,0) & \bar{h}_Q(1,1) & \bar{h}_Q(1,2) & \bar{h}_Q(1,3) & 0 & --- & 0 & 0 & 0 \\
x(n-2) & \bar{h}_Q(2,0) & \bar{h}_Q(2,1) & \bar{h}_Q(2,2) & \bar{h}_Q(2,3) & \bar{h}_Q(2,4) & & & & \\
x(n-3) & 0 & \bar{h}_Q(3,1) & \bar{h}_Q(3,2) & \bar{h}_Q(1,3) & \bar{h}_Q(3,4) & & & & \\
x(n-4) & 0 & 0 & \bar{h}_Q(4,2) & \bar{h}_Q(4,3) & \bar{h}_Q(4,4) & & & \vdots & \vdots \\
\vdots & \vdots & \vdots & & & & & & & \\
 & & & & & & & \bar{h}_Q(N-2,N) & 0 & 0 \\
 & & & & & & \bar{h}_Q(N-1,N-1) & \bar{h}_Q(N-1,N) & \bar{h}_Q(N-1,N+1) & 0 \\
x(n-N) & 0 & 0 & 0 & & \bar{h}_Q(N,N-2) & \bar{h}_Q(N,N-1) & \bar{h}_Q(N,N) & \bar{h}_Q(N,N+1) & \bar{h}_Q(N,N+2) \\
x(n-N-1) & 0 & 0 & 0 & & 0 & \bar{h}_Q(N+1,N-1) & \bar{h}_Q(N+1,N) & \bar{h}_Q(N+1,N+1) & \bar{h}_Q(N+1,N+2) \\
x(n-N-2) & 0 & 0 & 0 & & & 0 & \bar{h}_Q(N+2,N) & \bar{h}_Q(N+2,N+1) & \bar{h}_Q(N+2,N+2)
\end{array}$$

Figure 6.4: The equivalent matrix $\bar{\mathbf{h}}_Q(n)$ representing of the concatenation of a quadratic kernel $\mathbf{h}_Q(n)$ follow by a linear system $h(n)$. This result includes some null components as it is assumed that the length of the quadratic kernel is lower than that of the linear system.

ory Decomposition (MMD) approaches and Proportionate NLMS (PNLMS) by assuming a sparse Volterra filter. This is supported by the fact that the quadratic kernel of the LEMS dominant taps are around the main diagonal. We focus on this characteristic here and show the effect of loudspeaker non-linearities on Volterra filter models of the LEMS. We show here that a larger matrix dimension is required but, due to the fact that the non-linearities are generated by the loudspeaker only, the coefficients around the main diagonal are significant.

Now we assume that the system is composed of a cascaded non-linear system with a 1^{st} order kernel $h_1(n)$ of length N_1, a quadratic kernel $h_Q(n)$ with a memory length N_Q and a linear system, $h(n)$ representing the acoustic channel and up-link path. Since the linear system $h_1(n)$ does not affect the non-linear component, we focus our attention on the quadratic kernel of the concatenated system $((h_1, h_Q)(n) * h(n))$ which represents the global non-linear component, $\bar{h}_Q(n) = h_Q(n) \circledast h(n)$ ($\circledast$: represents here an operator corresponding to the result of the concatenation of two systems, when the systems are both linear this is equivalent to the convolution operator), and referred to here as the equivalent system.

Figure 6.3 illustrates the quadratic kernel $h_Q(n)$ convolution with the linear system $h(n)$. It shows that the equivalent non-linear system $\bar{h}_Q(n)$ corresponds to delayed versions of the quadratic kernel $h_Q(n)$ weighted by the coefficient of $h(n)$. The resulting equivalent kernel is given in Figure 6.4. We observe that, if the memory of the non-linear system (loudspeaker) is smaller compared to that of the linear system of the LEMS, meaning that $N_Q \ll N$, only the diagonal components are significant in the global LEMS. They will nevertheless have a quadratic kernel length of about $N + N_Q$ with the same non-linear memory length (N_Q) as the initial (loudspeaker) non-linear system. In AEC the assumption that $N_Q \ll N$, generally holds due to the fact that the LEMS system is mainly dominated by the characteristics of the near-end environment, i.e. the room reverberation time. To obtain the equivalent matrix elements $\bar{h}_Q$ according to $h_Q(n)$ and $h(n)$ we first express the resulting echo component as:

$$\begin{aligned}\bar{y}_Q(n) &= \sum_{m=0}^{N-1} h(m) y_Q(n-m) \qquad (6.7)\\ &= \sum_{m=0}^{N-1} h(m) \sum_{l_1=0}^{N_Q-1} \sum_{l_2}^{N_Q-1} h_Q(l_1,l_2) x(n-l_1-m) x(n-l_2-m)\end{aligned}$$

Supposing that $\bar{l}_1 = m + l_1$ and $\bar{l}_2 = m + l_2$, Equation 6.7 can be written as:

$$\bar{y}_Q(n) = \sum_{m=0}^{N-1} \sum_{\bar{l}_1=m}^{N_Q-1+m} \sum_{\bar{l}_2=m}^{N_Q-1+m} h(m) h_Q(\bar{l}_1 - m, \bar{l}_2 - m) x(n-\bar{l}_1) x(n-\bar{l}_2) \qquad (6.8)$$

We can then write the new equivalent coefficients as:

$$\bar{h}_Q(\bar{l}_1, \bar{l}_2) = \sum_{m=0}^{N-1} \sum_{\bar{l}_1=m}^{N_Q-1+m} \sum_{\bar{l}_2=m}^{N_Q-1+m} h(m) h(\bar{l}_1 - m, \bar{l}_2 - m) \qquad (6.9)$$

Equation 6.9 shows that the equivalent quadratic kernel of the Volterra filter which represents the LEMS has a memory length $\bar{N}_2$ equal to $N_Q + N$ where N_Q is the memory for the loudspeaker quadratic kernel and N is the length of the linear filter. This means that the Volterra filter has a length. This is deduced from the definition of $\bar{l}_i = l_i + m$, where the maximum of l_i is $N_Q - 1$ and that of m is $N - 1$. The summation indexes $\bar{l}_1$ and $\bar{l}_2$ in Equation 6.9 shows that the resulting filter cross term memory is equal to N_Q as the range of $\bar{l}_i$, $i = 1, 2$ is $(N_Q - 1 + m) - m + 1 = N_Q$.

The convolution of the quadratic kernel with a linear filter does not affect the symmetry conditions in the resulting equivalent quadratic kernel so the resulting equivalent matrix can be simplified to its upper or lower part including the diagonal elements without decreasing the estimation efficiency. In practice the quadratic kernel Volterra filter to be estimated has only significant elements on its N_Q diagonal elements, the rest of the matrix elements are negligible in the case of loudspeaker non-linearity.

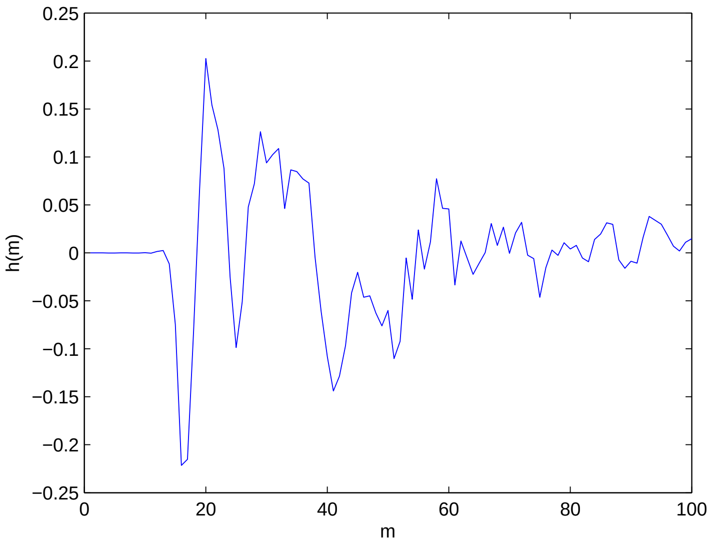


Figure 6.5: Linear channel representing the concatenation of the acoustical channel and up-link path

Example of Volterra estimation

Here we show the effect of using a non-linear system based on a quadratic Volterra model followed by the linear filter. A simplified version of a non-linear model is used. As the linear part of the non-linear system will only affect the linear part of the global system, we are more focused on the resulting quadratic kernel. Hence the non-linear system is combined with a filter $h_1(n) = 1$ and quadratic non-linearities of memory length $N_Q = 10$.

$$y_Q(n) = \sum_{l_1=0}^{9} \sum_{l_2=0}^{l_1} h_Q(l_1, l_2) x(n - l_1) x(n - l_2) \tag{6.10}$$

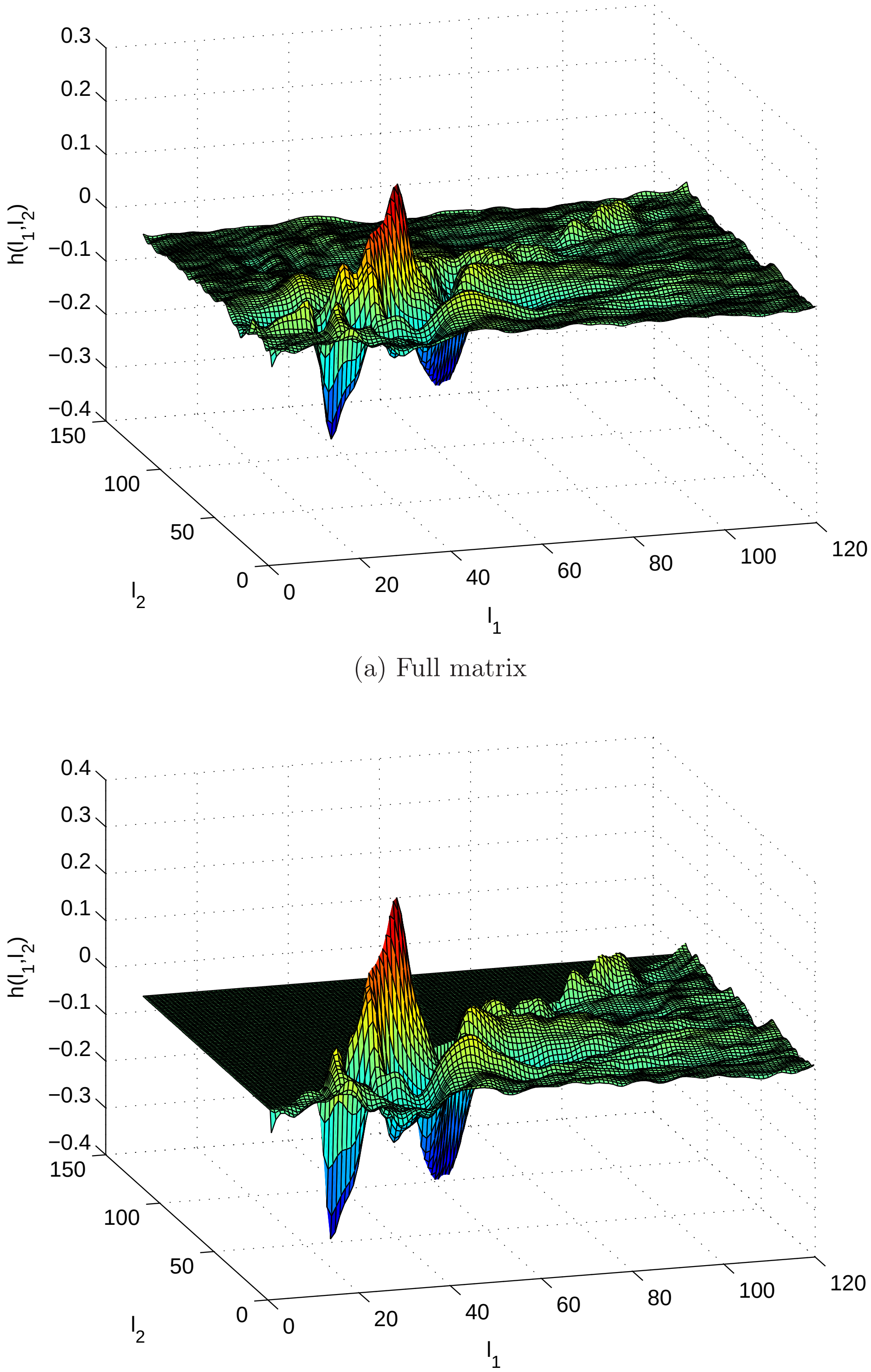
Equation 6.10 differs from the general Equation 6.2 in that it does not take into account the difference between l_1 and l_2 (coefficients are not given here as they are not needed). The linear filter is illustrated in Figure 6.5, this illustration helps to perceive the effect of $h(n)$ on the shape of the resulting quadratic kernel.

The resulting echo signal is then given by:

$$d(n) = h(n) * (x(n) + d_2(n)) \tag{6.11}$$

Using the estimation procedure given in Equation 6.5 we obtain the results illustrated in Figures 6.6 and 6.7 for the quadratic kernel with a speech and random signal respectively.

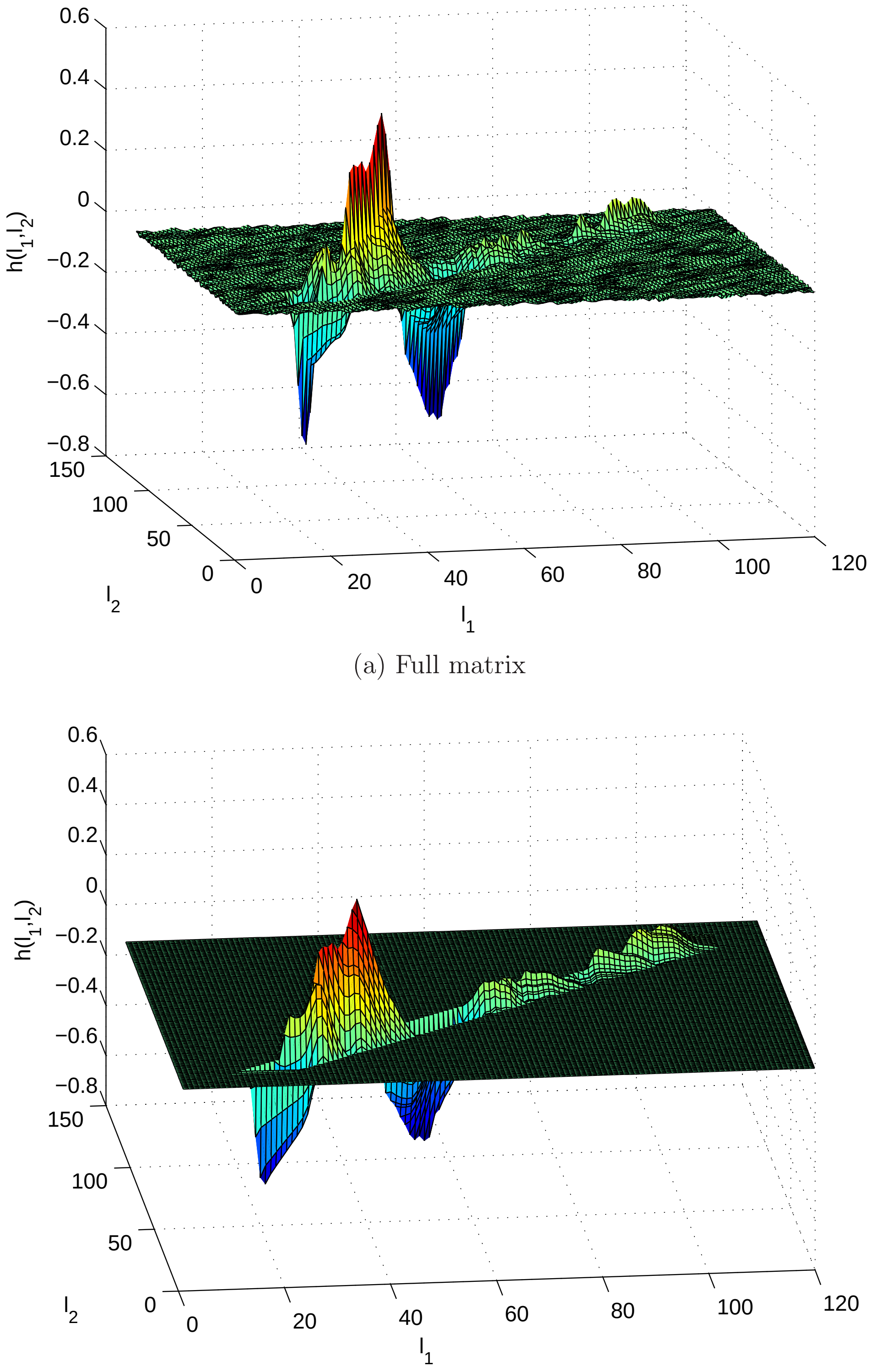
Upon comparison of Figures 6.6 and 6.7 we observe that, for random signal the matrix is more accurately identified than in the case of a speech signal. This is



(a) Full matrix

(b) Triangular matrix

Figure 6.6: Estimated taps of the Volterra filter given by Equation 6.5 with a speech signal.



(a) Full matrix

(b) Triangular matrix

Figure 6.7: Estimated taps of the Volterra filter given by Equation 6.5 with a random signal. (b) Triangular matrix shows that the quadratic filter follows the shape of the linear filter but is more longer due to the length of the quadratic filter of the loudspeaker.

shown by the fact that elements far from the diagonal are closer to zero for the random signal case than with the speech case. This shows that, as for the linear system the problem of speech correlation affects the estimation of the Volterra filter parameters. We also observe more non-zeros taps in Figure 6.7 (a) which are far from the main diagonal whereas in the triangular matrix shown in Figure 6.7 (b) most of the taps far from the diagonal are close to zero. As explained previously, if the sum of two numbers $t(l_1, l_2) + t(l_2, l_1)$ is equal to that of $h(l_1, l_2) + h(l_2, l_1)$ they are also solution for the error minimization (see Equation 6.3). The triangular matrix approach shows the case where $t(l_1, l_2)$ is forced to zero so that $t(l_2, l_1)$ is constrained to estimate $h(l_1, l_2) + h(l_2, l_1)$. Even if it is not immediately noticeable on Figures 6.6 and 6.7 we also observe that the taps which are not on the main diagonal are (twice) higher in the triangular case than in the full matrix case.

In Figures 6.6 and 6.7 we observe that the shape of the diagonal is comparable to that of the linear filter in Figure 6.5. This means that it is required to the quadratic kernel diagonal length to be at least equal to that of the linear filter. However, only its diagonal elements are significant meaning that the computation is significantly increased compared to the performance gained with the Volterra filter.

The fact that the dimension of the resulting kernel is equal to $N_Q + N$ is also seen on Figure 6.7 (b) in which some taps above $l_1 < 100$ (100: length of the linear filter) are slightly different from zero and in which some taps $l_1 < 110$ are equal to zero. This is expected from Equation 6.9 but is not perceptible with the speech signal due to the correlation effect.

Regarding Equation 6.9 and the observations in Figure 6.7 we remark that the resulting quadratic kernel of the LEMS has two dimensions of memory, the channel memory determined by $N_Q + N$ and the non-linearity memory equal to that of the quadratic filter in the loudspeaker N_Q. This poses some constraints in the application of the Volterra filter in non-linear AEC.

As is well known, the main problem in AEC is the length of the acoustic path in general ranges from 50 to 2000 taps. When the linear channel is longer it will be difficult to use a Volterra filter for identification on account of the computational demand. This is explained by the fact that most of the solutions developed to speed up the convergence rate of the conventional Least Mean Square (LMS) algorithm are complex even in the linear echo case. The simplification of the Volterra filter to diagonal matrix implementations is then required and explains the reason behind some proposed solutions such as MMD and the truncation proposed by [Stenger *et al.* 1999b].

The most challenging constraint of using Volterra filters for non-linear AEC is the variability of the acoustic channel. If the channel is static then the Volterra filter will be a good solution as it will not require tap re-estimation. Hence, according to Equation 6.9 in case of Echo Path Change (EPC) (which affect $h(n)$) all the kernels need to be re-estimated and this poses a problem of re-convergence. The initial convergence poses less problem as the initial taps are in general set to zero. But, when the filter has already converged, if an echo path change arises it may require more time to re-converge than the convergence after initialization. This is explained by the fact that each kernel will introduce some error which may affect the other kernels and slow down convergence. In addition to this problem as seen in Figure 6.6 the estimation is more difficult when the input is a speech signal.

The work developed here for the quadratic kernel Volterra model can be easily extended to higher than 2^{nd} order Volterra filters. Here we present only the second order solution as the Volterra filter in AEC is in general limited to the quadratic.

As we have observed it is practically difficult to use Volterra solution when the Echo Path (EP) is long. This problem is solved by using the cascaded structure to non-linear echo cancellation. This approach has some limits but requires less coefficients than the parallel Volterra approach. Cascaded structure is presented in the next section using the time domain model of the loudspeaker reported in Section 5.4.

6.2 Cascaded structure

In general the Volterra model takes a unified approach to estimate the overall LEMS which is a PS. This involves the simultaneous tracking of non-linearities and changes in the acoustical channel, i.e. the path between the loudspeaker and microphone. This is potentially inefficient since the same acoustic path is estimated by each Volterra sub-filter. Since the kernels inputs are correlated, convergence is typically slow. Here we propose a method that can improve the convergence of the system using a cascaded LEMS model. This approach uses a pre-processor which aims to model the loudspeaker non-linearities in series with a conventional linear adaptive filter to model the time varying acoustic channel. The linear adaptive filter is thus applied to a single input signal, which estimates the loudspeaker output, instead of being applied in parallel to the inputs of each sub-filter as in the Volterra model. Similar approaches to pre-processing based on clipping or polynomial models have already been proposed in [Nollett & Jones 1997, Stenger & Kellermann 2000, Costa *et al.* 2003, Guerin *et al.* 2003]. The pre-processor is based upon the loudspeaker model in Section 5.4.3. The time domain model uses parallel polynomial filters followed by a linear Finite Impulse Response (FIR) filter to model the loudspeaker non-linearities and can be considered equivalent to power filters [Kuech & Kellermann 2006]. However, the model proposed model in Section 5.4.3 is static, is thus dependent to the specific device and does not track slow variations which might occur over time.

This section is divided into three parts, we first present the model of the system and introduce signal and parameter notations. We also describe how all the different system modules interact. In the second part we develop the estimation procedure of the different parameters. Finally, we present the problem of local minima when using the cascaded structure.

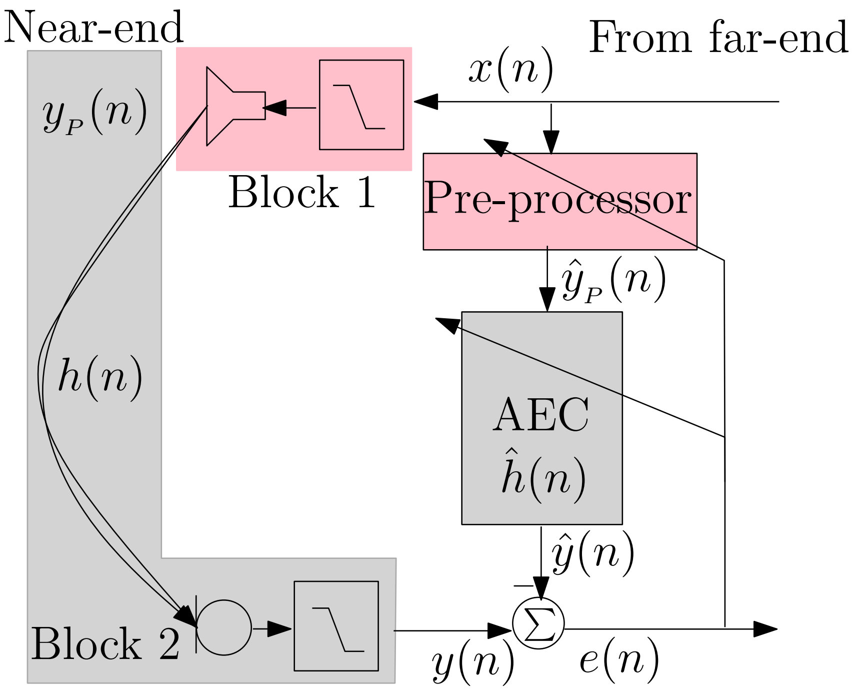


Figure 6.8: The LEMS is divided into two blocks. The first corresponds to the non-linear model whereas the second block is a linear model.

6.2.1 System model

In this section we present a general model of the LEMS. We also review the power filter presented in [Kuech & Kellermann 2006] and the CS proposed here.

The general LEMS illustrated in Figure 6.8 can be divided into two different blocks. The first involves the down-link components and includes the amplifier and loudspeaker. With small components (amplifier, loudspeaker), shorter impulse responses and lower variability is safely assumed. The second block involves the acoustic channel and the up-link components. The acoustic channel which, in the absence of significant non-linearities, can be well-modelled by a linear filter [Breining *et al.* 1999]. The acoustical channel has a significantly longer impulse response and also a higher degree of time variability, thus filtering approaches are generally adaptive in nature [Breining *et al.* 1999]. The up-link components include the microphone and amplifier. This part introduces less distortion and is generally assumed to be linear [Stenger & Kellermann 2000, Guerin *et al.* 2003, Kuech & Kellermann 2006].

In view of their different characteristics and in contrast to the PS approaches, the idea here is to treat each block of the system according to its distinct feature. The first block is distinctly non-linear whereas the second block is predominantly linear. It is therefore desirable to use just two filters: one to represent the down-link path, which is assumed to have a short impulse response and be the principle source of non-linearities, and a second filter to represent both the acoustical channel

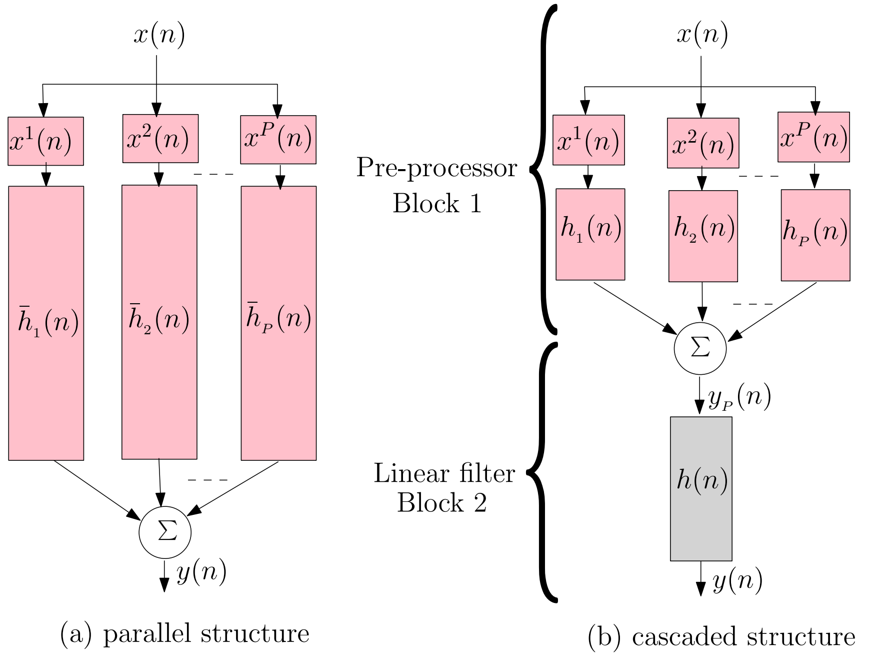


Figure 6.9: Different structures for the power filter model. (a) PS of the LEMS uses P longer sub-filters ($\bar{h}_p(n)$). (b) CS of the LEMS, P lower sub-filters ($h_p(n)$) in the pre-processor and the power filter model with P longer sub-filter ($\bar{h}_p(n)$).

and the up-link path. The second block is dominated by the characteristics of the acoustical channel: a longer impulse response and higher variability. This strategy leads to a cascaded structure of the LEMS as illustrated in Figure 6.8 which includes a separate pre-processor and linear adaptive filter for AEC.

With such an approach conventional linear adaptive filters are well suited to the second block. Being non-linear the down-link path is more troublesome but polynomial models [Mossi *et al.* 2010d] are appropriate. A polynomial loudspeaker model as in [Mossi *et al.* 2010d] is used here, so that its combination with a linear filter (Figure 6.9 (a)) is comparable to the power filter model for non-linear AEC (Figure 6.9 (b)). Here the sub-filters of the power filter model are a combination of the pre-processor sub-filters $\mathbf{h}_p(n)$ and the linear filter $\mathbf{h}(n)$ leading to the equality $\bar{h}_p(n) = h(n) * h_p(n)$. For each sub-filter $\bar{h}_p(n)$ we need at least the same number of taps as $h(n)$ to model the LEMS with power filters. With more taps and high variability in the acoustic channel it becomes difficult to track the LEMS in this way which thus explains why the Volterra model is difficult to use in practice. An orthogonalization procedure was introduced in [Kuech & Kellermann 2006] to improve the performance when the length of $\bar{h}_p$ is too large. The orthogonalization effect is explained in the following section which includes a detailed description of our approach.

6.2.2 Parameter estimation

In this section we present our approach to non-linear AEC with emphasis on the estimation of the loudspeaker model. Filter estimation is performed according to the Minimum Mean Square Error (MMSE) criterion. The Mean Square Error (MSE) is given by:

$$E\{e^2(n)\} = E\{(y(n) - \hat{y}(n))^2\}$$

where $y(n)$ is the echo signal and $\hat{y}(n)$ is the estimated echo signal given by:

$$\hat{y}(n) = \hat{\mathbf{h}}^T(n)\hat{\mathbf{y}}_P(n)$$

$\hat{\mathbf{h}}(n)$ is an N-column vector which represents the echo path and $\hat{\mathbf{y}}_P(n) = [\hat{y}_P(n), \cdots, \hat{y}_P(n-N+1)]$ is an N-column vector which contains the loudspeaker output estimates given by:

$$\hat{y}_P(n) = \sum_{p=1}^{P} \hat{\mathbf{h}}_p^T(n)\mathbf{x}_p(n)$$

$\hat{\mathbf{h}}_p(n)$ is the estimated filter vector of length N_p and $\mathbf{x}_p(n) = [x^p(n), \cdots, x^p(n - N_p + 1)]^T$. The error can thus be written as:

$$e(n) = y(n) - \hat{\mathbf{h}}^T(n) \sum_{p=1}^{P} \hat{\mathbf{h}}_p^T(n)\mathbf{X}_p(n) \tag{6.12}$$

where $\mathbf{X}_p(n) = [\mathbf{x}_p(n), \cdots, \mathbf{x}_p(n-N+1)]$. As Equation 6.12 contains too many unknowns we need to assume that $\hat{y}_P(n) = y_P(n)$, i.e. that the estimate is equal to the true value. The MMSE solution of $\hat{\mathbf{h}}(n)$ is then given by:

$$\hat{\mathbf{h}} = \mathbf{R}_{y_P}^{-1}\mathbf{p}_{y,y_P}$$

where $\mathbf{p}_{y,y_P}$ is the cross-correlation between the microphone signal and the output of the loudspeaker and $\mathbf{R}_{y_P}$ is the auto-correlation of the loudspeaker output $E\{y_P^T(n)y_P^T(n)\}$. This solution thus depends on knowledge of the loudspeaker output which will be discussed later in this section.

Here we derive an estimate of the pre-processor sub-filters while assuming that only the filter $\hat{\mathbf{h}}_k$ is unknown whereas the others are known ($\hat{\mathbf{h}} = \mathbf{h}$ and $\hat{\mathbf{h}}_{p \neq k} = \mathbf{h}_{p \neq k}$). The MMSE solution is given by:

$$\begin{aligned}
\frac{\partial E\{e(n)^2\}}{\partial \mathbf{h}_k} &= \frac{\delta E\{(y(n) - \mathbf{h}^T(n)\sum_{p=1}^{P} \hat{\mathbf{h}}_p^T(n)\mathbf{X}_p(n))^2\}}{\delta \mathbf{h}_k} \\
&= E\{\mathbf{X}_k(n)\hat{\mathbf{h}}^T(n)\big(y(n) - \mathbf{h}^T(n)\sum_{p=1}^{P} \hat{\mathbf{h}}_p^T(n)\mathbf{X}_p(n)\big)\}
\end{aligned}$$

If we pose $\mathbf{X}_k(n)\hat{\mathbf{h}}^T(n) = \tilde{\mathbf{y}}_k(n)$,$\tilde{\mathbf{y}}_k(n)$ has a length of N_k then:

$$\begin{aligned}\frac{\partial E\{e(n)^2\}}{\partial \mathbf{h}_k} &= E\{\tilde{\mathbf{y}}_k(n)\big(y(n) - \hat{\mathbf{h}}^T(n)\sum_{p=1}^{P}\mathbf{h}_p^T(n)\mathbf{X}_p(n)\big)\} \\ &= \mathbf{p}_{y,\tilde{y}_k} - \mathbf{p}_{Y_{p\neq k},\tilde{y}_k} - \mathbf{h}_k\mathbf{R}_{\tilde{y}_k}\end{aligned}$$

where $\mathbf{p}_{y,\tilde{y}_k}$ is the cross-correlation between the echo signal and the corresponding output and where $\mathbf{p}_{Y_{p\neq k},\tilde{y}_k}$ is the cross-correlation between the other sub-filter outputs and the output of the sub-filter k. The estimate of the filter $\mathbf{h}_k$ in the MMSE sense is given by:

$$\hat{\mathbf{h}}_k = \mathbf{R}_{y_k}^{-1}\big(\mathbf{p}_{y,y_k} - \mathbf{p}_{Y_{p\neq k},\tilde{y}_k}\big) \tag{6.13}$$

Equation 6.13 shows that the estimation of the pre-processor sub-filters are dependent due to their inter-correlation. Since pre-processor sub-filters use different power expansions of the same signal this may lead to a degradation of the estimation as a direct consequence of the inter-correlation. To overcome this limitation an orthogonalization procedure introduced in [Kuech & Kellermann 2006] shows that better performance is achieved when the sub-filter inputs are orthogonal. Orthogonalization leads to $\mathbf{p}_{y,\tilde{y}_k} = \mathbf{p}_{y_k,\tilde{y}_k}$ and $\mathbf{p}_{Y_{p\neq k},\tilde{y}_k} = \mathbf{0}$ so that the filter parameters become independent. In the proposed model we did not use orthogonalization since, with fewer taps in the pre-processor filters, it does not improve performance. As shown in [Kuech & Kellermann 2006] a further bias correction would be needed to improve performance and would lead to an overly complex solution in our case.

As we are in a short-term stationary environment adaptive filters are a necessity. The LMS adaptive filter can easily be derived using an approach similar to that described in [Nollett & Jones 1997, Stenger & Kellermann 2000, Haykin 2002]. The LMS algorithm for the sub-filter $\mathbf{h}_k(n)$ is given by:

$$\begin{aligned}\hat{\mathbf{h}}_k(n+1) &= \hat{\mathbf{h}}_k(n) + \frac{1}{2}\mu_k\frac{\delta e(n)^2}{\delta h_k} \\ &= \hat{\mathbf{h}}_k(n) + \mu_k\frac{\delta e(n)}{\delta h_k}e(n) \\ &= \hat{\mathbf{h}}_k(n) + \mu_k\mathbf{X}_k(n)\hat{\mathbf{h}}^T(n)e(n)\end{aligned} \tag{6.14}$$

whereas the linear filter is given by:

$$\hat{\mathbf{h}}(n+1) = \hat{\mathbf{h}}(n) + \mu\hat{\mathbf{Y}}_p(n)e(n) \tag{6.15}$$

Equations 6.14 and 6.15 show that the linear filter and the pre-processor filter estimates are dependent. The problem of the dependency between filters is discussed in [Stenger & Kellermann 2000] where the authors suggest that linear filter adaptation is done before adaptation of the pre-processor. Here we start with the linear filter $\hat{\mathbf{h}}(n)$ and the sub-filters $\hat{\mathbf{h}}_1(n)$ and $\hat{\mathbf{h}}_2(n)$, since their inputs are the least correlated.

The updating process of the sub-filter in Equation 6.14 of the CS is very complex. This is due to the fact that the adaptation procedure of sub-filters in Equation 6.14 requires the linear filter. This also means that the estimate of the linear filter is important for the pre-processor. Another challenging aspect of the down-link path non-linearities is the clipping effect. The clipping is challenging due to the fact that it may be time variable. In this case the pre-processor parameters may highly change which may perturb the overall system. Note that this also in the case of Volterra procedure will introduce a change of all the kernel taps and will be inefficient. To overcome these situations we propose in the next section three improvements of this baseline CS. Before introducing the proposed improvements we will first explain the problem of local minimum that arises in CS and the solution proposed in [Guerin *et al.* 2003].

6.2.3 Global and local minima

The fact that the CS approaches suffer from presence of local minima (MMSE sense) have been reported in many papers on non-linear AEC. Here we give some explanations related to the presence of local minima and the solution proposed in [Guerin *et al.* 2003]. In CS the pre-processor output, $\hat{y}_P(n)$ is not assumed to be an efficient estimate of the output $y_P(n)$. The requirement is that $\hat{y}_P(n)$ to approximate $h_r(n) * y_P(n)$ (where $h_r(n)$ is a linear filter, $\hat{y}_P(n) \approx h_r(n) * y_P(n)$) what we refer to linearly related. However, this is acceptable if the linear filter that follows the pre-processor can compensate the linearity between the loudspeaker and the pre-processor outputs.

Hence, if the output of the pre-processor corresponds to $h_r(n) * y_P(n)$ the linear filter $\hat{h}(n)$ should converge to $h_s(n) * h(n)$ with $h_r(n) * h_s(n) = \delta(n)$. If $h_s(n)$ exists then the system converges to a global minimum. However, if $h_s(n)$ cannot be estimated ($h_r(n)$ not invertible) then it is a local minimum. In [Guerin *et al.* 2003] it is proposed to use $\hat{h}_1(n) = 1$ ($h_r(n) = h_1^{(-1)}(n)$: the inverse of $h_1(n)$, in this case it constraints the filter to converge to one minimum but not necessarily a global minimum. It becomes a global minimum if and only if $h_s(n)$ the inverse of $h_1(n)$ can be sufficiently estimated otherwise it is a local minimum. Nevertheless, even when it is a local minima it has the advantage to constrain the system to converge to only one solution. This is important as it avoids fluctuation around different minimas in the presence of perturbations.

Following the idea in [Guerin *et al.* 2003] to constrain the system to one solution we apply a smaller step-size on the estimation of $h_1(n)$ compared to the other sub-filters. But, sub-filters ($p > 1$) also need to use smaller step-sizes compared to that of the linear filter for stability reason. This will initially work as the constraint proposed in [Guerin *et al.* 2003] but can improve the solution if the inverse of $h_1(n)$ does not exist, however, it may converge a bit slower compared to the solution in [Guerin *et al.* 2003].

6.3 Improved cascaded structure

This section focuses on improving the CS to non-linear AEC developed in Section 6.2. The CS has shown to give good tracking performance [Mossi *et al.* 2011b] and in this section we present three directions to improve CS. First, we have developed various modifications to the original work in [Nollett & Jones 1997, Guerin *et al.* 2003] to significantly improve computational efficiency of the pre-processor.

Second, we investigate the use of separate models of the amplifier and loudspeaker within the pre-processor. These two components typically exhibit different characteristics and thus independent models are more appropriate: a clipping model for the amplifier and a power-filter model for the loudspeaker.

Third, we have investigated the use of Decorrelation Filtering (DF). This aims to counter the increase in correlation caused by pre-processor filtering and the presence of non-linearities. DF is also known to improve the convergence of AEC based on Normalized-LMS (NLMS) algorithms [Breining *et al.* 1999, Hänsler & Schmidt 2004] when the input signal is highly correlated. Even if alternative linear AEC algorithms, such as the Recursive Least Square (RLS) algorithm, tend to deliver faster convergence, tracking performance is known to be inferior to that of the NLMS algorithm in certain non-stationary environment [Breining *et al.* 1999, Haykin 2002]. With DF, NLMS algorithms are generally preferred on account of lower complexity, and better stability and tracking performance.

The remainder of this section is organized as follows. In Section 6.3.1 we present an overview of the proposed system model which aims to give an overview of signals, parameters and their relationships. In Section 6.3.2 procedures to estimate the different parameters are used in the model.

6.3.1 System model

In this section we review the non-linear AEC model presented in Section 6.2 and outline the essence of the contributions presented. As illustrated in Figure 6.10 the approach is composed of a non-linear pre-processor (1) and a group of interconnected modules combining DF and linear AEC.

Pre-processor and clipping model

The pre-processor is used to model the characteristics of the down-link path, i.e. the amplifier and the loudspeaker. As illustrated in Figure 6.10 (top) the far-end signal $x(n)$ is first processed to obtain an output signal $\hat{y}_P(n)$ which is an estimate of the loudspeaker output. As in the CS the loudspeaker is assumed to be the main source of non-linearity.

In general, due to limited power, the amplifier may introduce clipping distortion for high level signals. Clipping distortion is modelled here as in [Nollett & Jones 1997, Stenger & Kellermann 2000] using a hard clipping model

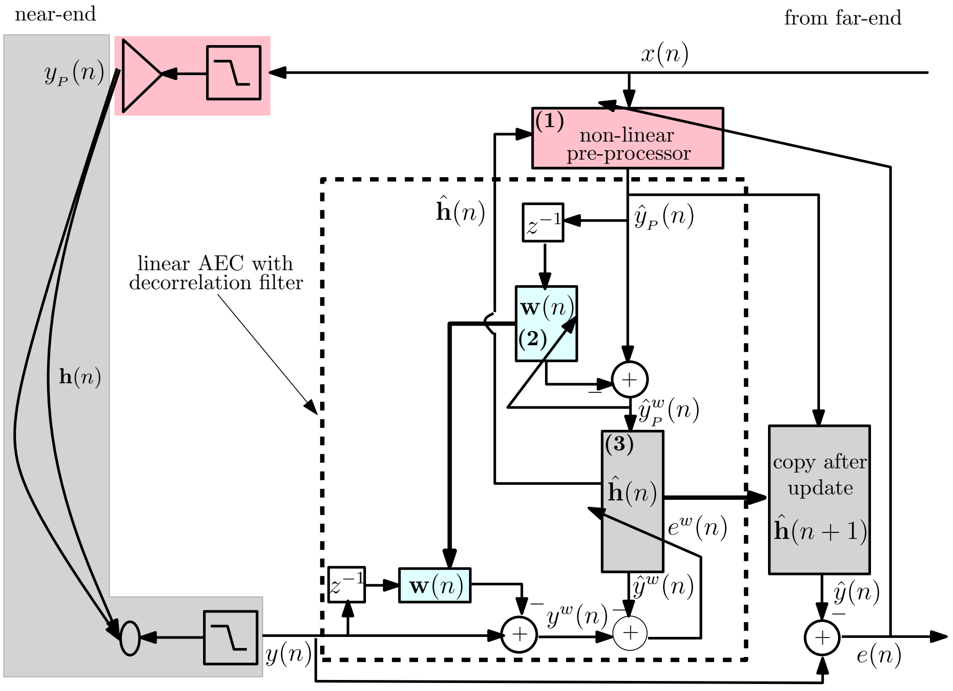


Figure 6.10: The non-linear AEC system is composed of a pre-processor that models the down-link path, a decorrelation filter $\mathbf{w}(n)$ and a linear AEC $\mathbf{h}(n)$.

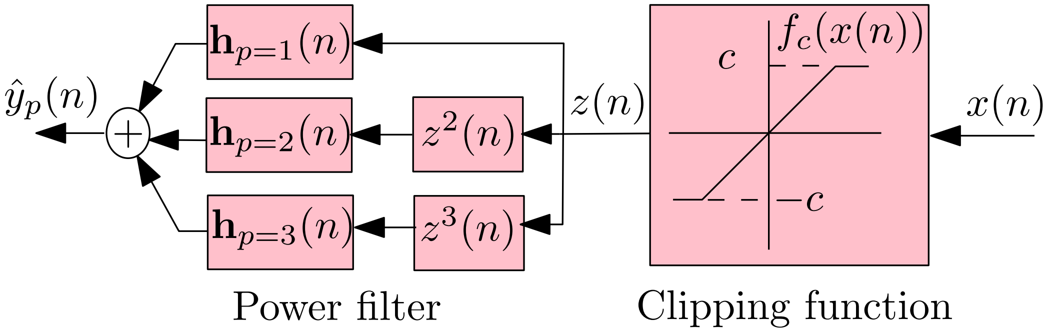


Figure 6.11: Pre-processor of the non-linear AEC system: a concatenation of a clipping compensator to model the amplifier and a power filter model of the loudspeaker.

which is a function with a parameter c. As illustrated in Figure 6.11 the clipping function is given as:

$$z(n) = f_c(x(n)) = \begin{cases} sign(x(n))c & \text{if } |x(n)| \geq c \\ x(n) & \text{if } |x(n)| < c \end{cases} \tag{6.16}$$

where $c \geq 0$ is the absolute value of the clipping level.

The loudspeaker is also assumed to be non-linear and is modelled with a power filter also illustrated in Figure 6.11. The output $z(n)$ of the clipping function is processed by the power filter to obtain an estimate $\hat{y}_P$(n) of the loudspeaker output. The output $\hat{y}_P(n)$ of the power filter is a summation of the different sub-filter outputs $\mathbf{h}_{p=1,2,3}(n)$ which are filtered versions of the input signal at different powers. The pre-processor output $\hat{y}_P(n)$ is thus given by:

$$\hat{y}_P(n) = \sum_{p=1}^{P} \underbrace{\mathbf{h}_p^T(n)\mathbf{z}_p(n)}_{=\hat{y}_p(n)} \tag{6.17}$$

where $P = 3$ is the number of pre-processor sub-filters and $\mathbf{z}_p(n) = [z^p(n), z^p(n-1), \cdots, z^p(n-N_p)]^T$ is the input signal to the sub-filter $\mathbf{h}_p(n)$ with N_p taps and output $\hat{y}_p(n)$. The down-link path is assumed to have a low memory (short impulse response) and is static or changes slowly (compared to the acoustic channel) [Stenger & Kellermann 2000, Guerin *et al.* 2003, Mossi *et al.* 2011b].

Decorrelation filtering and linear AEC

The adaptive decorrelation filter (block 2 in Figure 6.10) is represented by the adaptive filter $\mathbf{w}(n)$ and is applied to the pre-processor output. Duplicate filtering is applied to the echo signal $y(n)$ so that the echo path estimate will still converge to $\mathbf{h}(n)$ [Breining *et al.* 1999, Hänsler & Schmidt 2004]. As in [Widrow 1971, Haykin 2002] the output is given by:

$$\hat{y}_P^w(n) = \hat{y}_P(n) - \mathbf{w}^T(n)\hat{\mathbf{y}}_P(n-1), \tag{6.18}$$

and according to classical linear prediction analysis, $\hat{y}_P^w(n)$ is a decorrelated signal.

The linear AEC module (block 3 in Figure 6.10) represents the concatenation of the acoustic channel and the up-link path. On account of the decorrelation filter the linear AEC operates on $\hat{y}_P^w(n)$. The output of the linear AEC module in the decorrelated link is given by:

$$\hat{y}^w(n) = \mathbf{h}^T(n)\hat{\mathbf{y}}_P^w(n)$$

where $\hat{\mathbf{y}}_P^w(n) = [\hat{y}_P^w(n), \hat{y}_P^w(n-1), \cdots, \hat{y}_P^w(n-N-1)]^T$. The real echo estimate $\hat{y}(n)$ is then obtained using the updated version of $\hat{\mathbf{h}}(n)$ filter, $\hat{\mathbf{h}}(n+1)$ which is applied to $\hat{y}_P(n)$ as illustrated in Figure 6.10. The use of $\hat{\mathbf{h}}(n+1)$ to compute $\hat{y}(n)$ has the advantage to take into account the new update information in the estimation of the echo. In fact the decorrelation filter has shown to provide fast convergence. By introducing the strategy of using the updating filter we improve the filter estimation as we take into account the new information. This procedure is efficient in echo only period. When a near-end signal is present it requires the adaptation to be paused. This procedure is similar to certain frequency domain Wiener filtering where the filter gains are estimated with the current samples and then applied to the current microphone signal. Higher step-size (close to 1) should be avoided as they will introduce more attenuation in presence of noise. As will be seen in the assessment a better loudspeaker model is required to reach better echo reduction.

6.3.2 Parameter estimation

Though the estimation of the different modules parameters are individually straight forward, the integration of the clipping compensator and adaptive decorrelation filter into the basic CS of Section 6.2 requires more investigation to make these estimators efficient. The basic CS is presented here starting with a description of our baseline system and an approach to reduce its complexity.

Cascaded structure

The cascaded power filter and linear AEC system are presented in detail in [Mossi *et al.* 2011b] and thus we give here the essential baseline estimation procedures with minimal detail only. Ignoring the clipping compensator in Figure 6.11, i.e. by assuming that $x(n) = z(n)$, the pre-processor estimate is obtained according to:

$$\hat{\mathbf{h}}_p(n+1) = \hat{\mathbf{h}}_p(n) + \bar{\mu}_p(n)\underbrace{[\hat{\mathbf{h}}^T(n)\mathbf{Z}_p(n)]^T e(n)}_{=\Delta\mathbf{h}_p(n)} \tag{6.19}$$

where $\mathbf{Z}_p(n) = [\mathbf{z}_p(n), \mathbf{z}_p(n-1), \cdots, \mathbf{z}_p(n-N-1)]^T$ and where $\mathbf{z}_p(n)$ is an input vector with length N_p where the normalized step size $\bar{\mu}_p(n) = \frac{\mu_p}{\left\|\mathbf{h}^T(n)\mathbf{Z}_p(n)\right\|^2+\xi}$ and where ξ is a regularization factor to avoid division by zero. The estimate of the linear filter $\mathbf{h}(n)$ is given by:

$$\hat{\mathbf{h}}(n+1) = \hat{\mathbf{h}}(n) + \mu_l(n)\hat{\mathbf{y}}_P(n)e(n), \tag{6.20}$$

where $\bar{\mu}_l(n) = \frac{\mu_l}{\left\|\hat{\mathbf{y}}_P(n)\right\|^2+\xi}$.

Complexity reduction

We propose here an approach to reduce sub-filter estimation complexity which aims to offset the extra computation introduced through Clipping Compensation (CC). Computation of the gradient $\Delta\mathbf{h}_p(n)$ in Equation 6.19 is rather complex as the calculation of $\mathbf{Z}_p(n)\hat{\mathbf{h}}^T(n)$ requires $N_p \times N$ multiplications. A more efficient approximation can be obtained if, for all but the first coefficient of the gradient $\Delta\mathbf{h}_p(n)$, $\hat{\mathbf{h}}(n)$ is replaced by a previously calculated echo path estimate. Thus, instead of:

$$\begin{aligned}\hat{\mathbf{h}}^T(n)\mathbf{Z}_p(n) \quad &= \quad [\hat{\mathbf{h}}^T(n)\mathbf{z}_p(n), \cdots, \underbrace{\hat{\mathbf{h}}^T(n)\mathbf{z}_p(l)}_{=\tilde{z}_p(l)}, \\ &\qquad \cdots, \hat{\mathbf{h}}^T(n)\mathbf{z}_p(n-N_p-1)]\end{aligned}$$

where $\tilde{z}_p(l) = \hat{\mathbf{h}}^T(n)\mathbf{z}_p(l)$, which depends on the current estimate $\hat{\mathbf{h}}(n)$, we use:

$$\tilde{z}_p(l) = \hat{\mathbf{h}}^T(l)\mathbf{z}_p(l)$$

which depends on $\hat{\mathbf{h}}(l)$ calculated in previous iterations. This approximation does not require any computation for $l < n$ and leads to:

$$\begin{aligned}\hat{\mathbf{h}}^T(n)\mathbf{Z}_p(n) &= [\hat{\mathbf{h}}^T(n)\mathbf{z}_p(n), \cdots, \underbrace{\hat{\mathbf{h}}^T(l)\mathbf{z}_p(l)}_{=\tilde{z}_p(l)}, \\ &\quad \cdots, \hat{\mathbf{h}}^T(n-N_p-1)\mathbf{z}_p(n-N_p-1)]\end{aligned}$$

Complexity is thus reduced by a factor of N_p per sub-filter with the added advantage of reacting faster to changes in the echo path. The only drawback is that convergence is somewhat slower. Note that a similar simplification can be applied to other CS as they use similar adaptation of pre-processor, for example those in [Nollett & Jones 1997, Guerin *et al.* 2003].

Clipping compensation

In this section we first present the cascade of the power filter and linear AEC algorithm according to [Mossi *et al.* 2011b]. Then we show how Clipping Compensation (CC) can be efficiently incorporated into the global model.

The proposed approach combines the clipping system proposed in [Nollett & Jones 1997, Stenger & Kellermann 2000] with the cascaded model presented in [Mossi *et al.* 2011b]. We show here that the CC can be implemented with a complexity comparable to the system presented in [Stenger & Kellermann 2000] where no pre-processor is used. We again use the LMS approach to derive an adaptive clipping level estimator. The model presented here is based on a hard clipping model [Nollett & Jones 1997] (which could easily be extended to soft clipping) as given in Equation 6.16. To derive a gradient for the estimator according to the LMS approach we need to incorporate the clipping function within an expression for the error $e(n)$ thus leading to:

$$e(n) = y(n) - \mathbf{h}^T(n)\sum_{p=1}^{P}\mathbf{h}_p^T(n)\underbrace{[f_c(\mathbf{X}(n))]_p}_{=\mathbf{Z}_p(n)}$$

where $[f_c(\mathbf{X}(n))]_p$ indicates that the function $f_c(x(n))$ is applied to each element of the matrix $\mathbf{X}(n) = [\mathbf{x}(n), \mathbf{x}(n-1), \cdots, \mathbf{x}(n-N)]$ where $\mathbf{x}(n) = [x(n), x(n-1), \cdots, x(n-N_p)]$.

Applying the LMS approach we derive the clipping level estimator using the derivative of the error with respect to c which leads to:

$$\hat{c}(n+1) = \hat{c}(n) + \mu_c\mathbf{h}^T(n)\sum_{p=1}^{P}\mathbf{h}_p^T(n)\underbrace{[\frac{\partial f_c}{\partial c}(\mathbf{X}(n))]_p}_{=[\dot{\mathbf{Z}}(n)]_p}e(n) \qquad (6.21)$$

where $\frac{\partial f_c}{\partial c}(x(n))$ is the derivative of $f_c(x(n))$ according to c and $\dot{\mathbf{Z}}(n)$ is the derivative of $\mathbf{Z}(n)$ according to c. From Equation 6.16 we see that $\frac{\partial f_c}{\partial c}(x(n))$ is equal to:

$$\frac{\partial f_c}{\partial c}(x(n)) = \begin{cases} sign(x(n)) & \text{if } |x(n)| \leq c \\ 0 & \text{elsewhere} \end{cases} \tag{6.22}$$

We observe that the gradient in Equation 6.21 is highly complex due to the cascaded pre-processor. To simplify its calculation, we note that the derivative $\frac{\partial f_c}{\partial c}(x(n))$ is independent from the sub-filter order p which explains the use of $[\dot{\mathbf{Z}}(n)]_p$ instead of $\dot{\mathbf{Z}}_p(n)$ in which case the derivative depends on p. Hence,

$$\text{if } \exists k / \ [f_{\hat{c}}(x(n))]_k \rightarrow [f_c(x(n))]_k \Rightarrow \forall\ p\ [f_{\hat{c}}(x(n))]_p \rightarrow [f_c(x(n))]_p,$$

where the convergence is in the sens of $\hat{c}$ converges to c. Meaning that if we can minimize the gradient in one sub-filter it will be automatically minimised in the other sub-filters. This means that we can reasonably ignore some of the pre-processor sub-channels. Here we consider only the sub-channel 1 as it presents the strongest component of the echo signal. We thus consider $z(n)$ to be composed of a linear component $z_l(n)$ and a non-linear distortion component $z_d(n)$ so that $z(n) = z_l(n) + z_d(n)$. We then suppose that the distortions within the power filter generated by $z_d(n)$ for $p \geq 2$ are negligible, i.e.

$$\underbrace{(|z(n)| - c)^p}_{z(n) \geq c, p \geq 2} \approx 0, \tag{6.23}$$

so that they can be safely ignored in the compensation.

In fact, as we suppose that only the linear part ($p = 1$) is affected by the clipping, the error minimization that leads $\hat{c}(n)$ to converge to c will also minimize the error in the non-linear part ($p \geq 2$) as $\hat{c}$ is also applied to the non-linear part. This means that the approximation in Equation 6.23 will be more effective when $\hat{c}(n)$ converges so that it can reach its optimal value in the minimum mean square error sense. This approximation implicitly assumes that $f_c(x(n))_{p \geq 2}$ is independent from c and leads to $(\frac{\partial f_c}{\partial c}(x(n)))_{p \geq 2}$ being equal to zero. Equation 6.21 is thus simplified to:

$$\hat{c}(n+1) = \hat{c}(n) + \mu_c \mathbf{h}^T(n) \mathbf{h}_1^T(n) \frac{\partial f_c}{\partial c}(\mathbf{X}(n)) e(n) \tag{6.24}$$

A second source of complexity relates to the cascade of the two filters $h(n) * h_1(n)$ in Equation 6.24. In fact it is possible to use the estimates $(\hat{h}(n) * \hat{h}_1(n))$ but, in practice, they must be highly accurate otherwise Equation 6.24 will be ineffective and give poor performance. Another problem encountered using $\hat{h}(n) * \hat{h}_1(n)$ is that it leads to a more complex system since, for each iteration, $N \times N_1$ multiplications are required to compute the convolution. To overcome this problem we need to constrain one of the filters to be equal to $\delta(n)$ (Dirac function). In practice it is easier to set $\hat{h}_1(n) = \delta(n)$ as used in [Guerin *et al.* 2003] so that $h(n) * h_1(n) \approx \hat{h}(n)$. We can then rewrite Equation 6.24 as:

$$\hat{c}(n+1) = \hat{c}(n) + \mu_c \hat{\mathbf{h}}^T(n) \frac{\partial f_c}{\partial c}(\mathbf{x}(n)) e(n) \tag{6.25}$$

which is less complex and amenable to real-time implementation. If instead we were to constrain $\hat{h}_1(n)$ to be equal to $\delta(n)$ then it will affect the estimate of the sub-filters $p \geq 2$ and the linear AEC. In this case the linear filter will converge to $h_1(n) * h(n)$ and the sub-filter $\hat{h}_p(n)$ will converge to $h_1^{-1}(n) * h_p(n)$.

Finally note that, in terms of implementation the pre-processor is not significantly different to the system presented in Section 6.2.2. The only change is that the first order sub-filter $\hat{\mathbf{h}}_1(n)$ is set to 1 and is not adaptive.

Decorrelation filtering

Conventional, fixed approaches to decorrelation are not appropriate here due to the use of pre-processing to which the decorrelation filter must adapt. Adaptive decorrelation is thus necessary but is inevitably more complex. Using the LMS criteria to minimize the decorrelation filter output $\hat{y}_P^w(n)$ we obtain an adaptive estimate of $\mathbf{w}(n)$ according to:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu_w(n)\hat{\mathbf{y}}_P(n-1)\hat{y}_P^w(n)$$

where $\mu_w(n) = \frac{\mu}{\left\|\hat{\mathbf{y}}_P(n-1)\right\|^2+\xi}$ and where $\mu \leq 1$.

We now consider the effect of Decorrelation Filtering (DF) on other system elements. First $\hat{\mathbf{y}}_P(n)$ in Equation 6.20 is replaced by $\hat{\mathbf{y}}_P^w(n)$ and similarly $e(n)$ is replaced by $e^w(n)$ as in Figure 6.10. The input to the linear AEC module is thus decorrelated and so convergence is improved. Second, on account of adaptive pre-processing, the signal $\hat{y}_P(n)$ in Equation 6.17 will be highly non-stationary. It is then necessary to apply lower step-sizes to sub-filter estimation in order to reduce the non-stationarity in $\hat{y}_P(n)$ and thus to improve DF. For this reason the decorrelation filter is of short order so that it can reliably follow variations in pre-processing. The decorrelation filter also has secondary benefits. In Equation 6.19 we see that sub-filter estimation uses the linear AEC estimate $\hat{h}(n)$ and will now be more accurate (faster convergence). The pre-processor estimate is then itself more accurate and will converge faster, resulting in more stable sub-filter estimation $\hat{\mathbf{h}}_p(n)$. Note also that, due to the presence of CC, decorrelation filter estimation should be paused during intervals in which CC is applied, i.e. when $z(n) = \hat{c}(n)$, since in these intervals a constant-level pre-processor output may disturb estimation. A solution involves changing the decorrelation filter step size to $\bar{\mu}_w(n) = (\neg\frac{\partial f_c}{\partial c}(x(n)))\cdot\mu_w(n)$ where $\neg$ is the logic 'NOT' and where $\frac{\partial f_c}{\partial c}(x(n))$ is as given in Equation 6.22. Hence $\neg\frac{\partial f_c}{\partial c}(x(n))$ is equal to 0 when $z(n) = \hat{c}(n)$ and equal to $\delta(n)$ otherwise.

The proposed improvement enhances the performance of the CS and can be easily extended to different pre-processor models. Nevertheless just as with PSs, the CS has the drawback to introduce some distortions in the processed microphone signal. This effect is due to the limitation of the microphone sampling frequency which suppresses higher frequency distortions already included in the non-linear model of the AEC. To avoid such an effect re-sampling or low pass filtering was reported in [Frank 1996, Niemistö & Mäkelä 2003a] to suppress higher frequencies generate by the non-linear system of the AEC. Here we investigate another approach which consists of moving the pre-processor from the AEC path to the down-link path.

6.4 Loudspeaker pre-processing

In this section we present an approach to non-linear AEC. This approach is based on the same LEMS model as in the CS which involves a non-linear model of the down-link path and a linear model of the acoustic channel and up-link path. Using this model we propose here an approach that uses Loudspeaker Pre-processing (LP) and linear AEC to improve performance of an otherwise classical approach to linear AEC. The proposed approach relates to an on-line linearisation pre-processing algorithm that adapts to long-term variations in the loudspeaker characteristics. This feature contrasts with fixed pre-processor algorithms which have been reported previously.

In this section we focus on non-linear adaptive filtering based on LP where the loudspeaker input is pre-processed by a non-linear filter, referred to here as a linearisation pre-processor. It aims to compensate for non-linearities that are subsequently introduced by the loudspeaker so that, when combined, the linearisation pre-processor and loudspeaker form a linear system. LP then permits the use of conventional linear AEC. In terms of echo reduction performance is improved and, as the linearisation pre-processor relies only on the loudspeaker characteristics, it does not need to be re-initialized when the acoustic environment changes. This is a distinct benefit over alternative post-loudspeaker approaches which depend fundamentally on the acoustic path and thus suffer from convergence issues when the echo path changes.

In practice, however, LP is rarely used with AEC since there is no direct access to the loudspeaker output. A solution proposed in [Furuhashi *et al.* 2006] renders the system dependent to the device and is based upon an inverse, static model of the loudspeaker. Transducer characteristics are dynamic, however, and in practice such solutions can sometimes even increase distortion instead of reducing non-linear echo. The solution proposed here uses an on-line LP approach which enables the tracking of long-term variation.

This section is organized similarly as the previous section. We present the LP approach and define the parameters involved. Then we describe the overall system, its operation and behaviour.

6.4.1 System model

The overall system is illustrated in Figure 6.12. In contrast to the CS approach, the pre-processor is now in the downlink path. Here the system is composed of an adaptive pre-processor which aims to linearise the loudspeaker output and a linear AEC module which tracks the acoustic path (acoustic channel + down-link devices).

The LP system is illustrated in Figure 6.13 and corresponds to the down-link path in Figure 6.12, including the pre-processor and the loudspeaker. The far-end signal $x(n)$ forms the input to the pre-processor and the linear acoustic echo

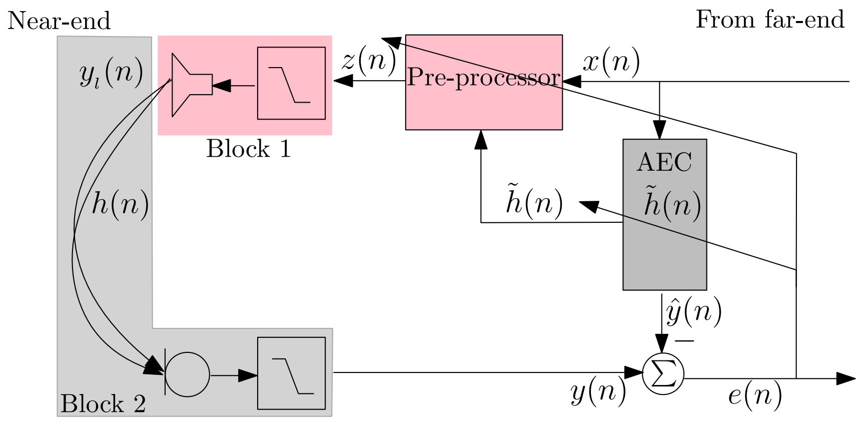


Figure 6.12: LP and acoustic echo cancellation where the LEMS is divided into two blocks. The first is a non-linear model and the second is a linear model.

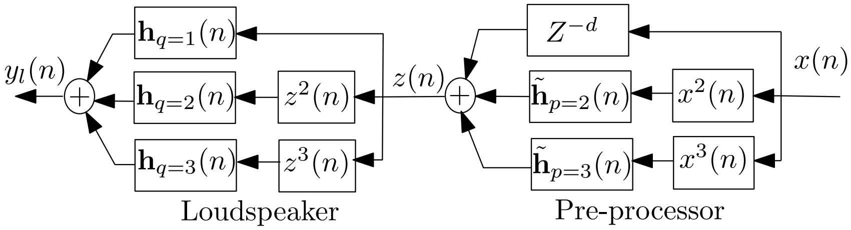


Figure 6.13: Loudspeaker linearisation system.

canceller. According to the non-linear power model the discrete loudspeaker output $y_l(n)$ can be written as:

$$y_l(n) = \sum_{q=1}^{Q} \mathbf{h}_q(n)\mathbf{z}_q^T(n) \tag{6.26}$$

where the vector $\mathbf{z}(n)$ is the input, $\mathbf{h}_q(n)$ is the sub-filter applied to the q^{th} power of $z(n)$ and Q is the maximum signal exponent. The vector $\mathbf{z}_q(n)$ is given by:

$$\mathbf{z}_q(n) = [z^q(n), z^q(n-1), \cdots, z^q(n-M_q-1)]^T \tag{6.27}$$

where M_q is the length of each filter $\mathbf{h}_q(n)$. It is always independent of q for all work reported here.

There is no need to process the linear component of $x(n)$ (top path) which is instead delayed by d samples and corresponds to the processing delay of the non-linear components ($p \geq 2$). Together they are used to generate an output that aims to compensate for the non-linearities which are subsequently introduced by the loudspeaker [Frank 1994, Lashkari 2005]. The pre-processor output signal can be written as:

$$z(n) = x(n) + \sum_{p=2}^{P} \tilde{\mathbf{h}}_p(n)\mathbf{x}_p^T(n) \tag{6.28}$$

where $z(n)$ is the output of the pre-processor, the vector $\mathbf{x}_p(n)$ is the far-end signal vector and $\tilde{\mathbf{h}}_p(n)$ is the sub-filter of the p^{th} power input. The objective is to obtain a loudspeaker output such that $y_l(n) \approx \mathbf{h}_1(n)\mathbf{x}^T(n)$ where $\mathbf{h}_1(n)$ is the linear loudspeaker impulse response. If this approximation is reached then the resulting echo signal can be estimated using a conventional linear adaptive AEC filter such as the LMS algorithm.

6.4.2 Parameter estimation

In this section we present the proposed non-linear AEC algorithm which is based on the well-known LMS approach.

AEC filtering

To derive the estimate of AEC filter and linearisation pre-processor we need to express the AEC system error ($e(n)$ in Figure 6.12) according to the different model parameters. According to Equation 6.26 and Equation 6.28 the discrete loudspeaker output can be rewritten as:

$$\begin{aligned}
y_l(n) &= \sum_{q=1}^{Q} \mathbf{h}_q(n) \left\{ x(n) + \sum_{p=2}^{P} \tilde{\mathbf{h}}_p(n)\mathbf{x}_p^T(n) \right\}_q \\
&= \sum_{q=1}^{Q} \mathbf{h}_q(n) \left\{ \mathbf{x}_{1*q}^T(n) + \sum_{p=2}^{P} \tilde{\mathbf{h}}_p(n)\mathbf{x}_{p*q}^T(n) \right\} \\
&= \mathbf{h}_1(n)\mathbf{x}_1^T(n) + \sum_{q=2}^{Q} \mathbf{h}_q(n)\mathbf{x}_q(n) \\
&+ \sum_{p=2}^{P} \mathbf{h}_1(n)[\tilde{\mathbf{h}}_p(n)\mathbf{X}_{p*(q=1)}^T(n)]^T \\
&+ \underbrace{\sum_{q=2}^{Q}\sum_{p=2}^{P} \mathbf{h}_q(n)[\tilde{\mathbf{h}}_p(n)\mathbf{X}_{p*q}^T(n)]^T}_{\text{neglected}(p*q\geq 4)}
\end{aligned}$$

where $\mathbf{X}_{p*q}(n)$ is an $M_p \times M_q$ matrix form of the signal $x^{(p*q)}(n)$ given by:

$$\mathbf{X}_{p*q}(n) = [\mathbf{x}_{p*q}(n), \mathbf{x}_{p*q}(n-1), \cdots, \mathbf{x}_{p*q}(n - M_p - 1)]$$

where $\mathbf{x}_{p*q}(n)$ is a vector of length M_q defined as in Equation 6.27. M_p and M_q are respectively the length of filters $\tilde{\mathbf{h}}_p(n)$ and $\mathbf{h}_q(n)$. We assume that the highest order terms are negligible and that the non-linearity can be modelled sufficiently with $P = 3$. Experiments performed by other authors and with real loudspeakers show that the performance benefit obtained from the inclusion of higher order terms does not justify the extra complexity [Kuech & Kellermann 2006, Furuhashi *et al.* 2006, Frank 1994]. The loudspeaker output can therefore be approximated as:

$$\begin{aligned} y_l(n) &= \mathbf{h}_1(n)\mathbf{x}_1^T(n) + \sum_{q=2}^{Q} \mathbf{h}_q(n)\mathbf{x}_q(n) \\ &+ \sum_{p=2}^{P} \mathbf{h}_1(n)[\tilde{\mathbf{h}}_p(n)\mathbf{X}^T_{p\cdot(q=1)}(n)]^T \end{aligned}$$

The output of the loudspeaker is convolved with the acoustic path $\mathbf{h}(n)$ (acoustic channel + up-link):

$$\begin{aligned} y(n) &= \mathbf{h}(n)[\mathbf{h}_1(n)\mathbf{X}_1^T(n)]^T + \sum_{q=2}^{Q} \mathbf{h}(n)[\mathbf{h}_q(n)\mathbf{X}_q^T(n)]^T \\ &+ \sum_{p=2}^{P} (\mathbf{h}(\mathbf{n}) * \mathbf{h_1}(\mathbf{n}))[\tilde{\mathbf{h}}_p(n)\mathbf{X}^T_{p*(q=1)}(n)]^T \end{aligned}$$

The AEC output is given by:

$$\hat{y}(n) = \tilde{\mathbf{h}}(n)\mathbf{x}^T(n)$$

and the error between the echo and its estimate is given by:

$$e(n) = y(n) - \hat{y}(n). \tag{6.29}$$

The error is used to obtain an adaptive estimate of the linear filter [Haykin 2002]. We assume that the linear echo component is dominant and thus that we have direct access to it. Using the LMS approach the adaptation of the AEC filter is given by:

$$\tilde{\mathbf{h}}(n+1) = \tilde{\mathbf{h}}(n) + \mu e(n)\mathbf{x}(n) \tag{6.30}$$

and, after sufficient iterations, $\tilde{\mathbf{h}}(n)$, will converge to $\mathbf{h}_l(\mathrm{n})$, where $\mathbf{h}_l(n)$ is the linear filter such that its convolution with $x(n)$ gives the linear echo component. Note from the first term of Equation 6.29, which represents the linear echo component, that $h_l(n) = h_1(n) * h(n)$.

Linearisation processing

In the same way as for the AEC filter the sub-filters of the linearisation pre-processor are estimated using the LMS approach, leading to:

$$\tilde{\mathbf{h}}_p(n+1) = \tilde{\mathbf{h}}_p(n) + \mu \frac{\delta e^2(n)}{\delta \tilde{\mathbf{h}}_p(n)}$$

By deriving the square of the error with respect to $\tilde{\mathbf{h}}_{p=2,3}(n)$ we obtain:

$$\tilde{\mathbf{h}}_p(n+1) = \tilde{\mathbf{h}}_p(n) + \mu e(n)\mathbf{h}(n)[\mathbf{h}_1(n)\mathbf{X}^T_{p\cdot(q=1)}(n)]^T \tag{6.31}$$

In Equation 6.31 the filter $h_{l}(n) = h(n) * h_{1}(n)$ is unknown. To overcome this problem an estimate $\tilde{\mathbf{h}}(n)$ in Equation 6.30 is used and leads to:

$$\tilde{\mathbf{h}}_{p}(n+1) = \tilde{\mathbf{h}}_{p}(n) + \mu_n e(n)\tilde{\mathbf{h}}(n)\mathbf{x}^{T}_{p\cdot(q=1)}(n) \tag{6.32}$$

where μ_n is a normalized step-size equal to $\frac{\mu}{|\tilde{\mathbf{h}}(n)\mathbf{X}^{T}_{p\cdot(q=1)}(n)|^2}$ with $0 < \mu \leq 1$. Equation 6.32 provides a solution for the linearisation of the loudspeaker in non-linear echo environments.

From Equation 6.32, we see that the pre-processor updating process uses the estimate of the AEC, $\tilde{\mathbf{h}}(n)$, meaning that the linear component should be dominant. As the estimate $\tilde{\mathbf{h}}(n)$ of the AEC is used to estimate the sub-filters, $\tilde{\mathbf{h}}_{p=2,3}(n)$, it is important to ensure that the pre-processor still depends only on the loudspeaker characteristics. This means that the sub-filter estimates should converge to a fixed filter which depends only on $\mathbf{h}_{q=1,2,3}(n)$ (loudspeaker characteristics). The independence of the pre-processor to $\mathbf{h}(n)$ (acoustic path) is needed to ensure stability to changes in the echo path characteristics. We thus extend Equation 6.29 to:

$$\begin{aligned} e(n) = {} & \underbrace{\big(\mathbf{h}_{l}(n) - \tilde{\mathbf{h}}(n)\big)\mathbf{x}^{T}_{1}(n)}_{\text{linear component}} \\ & + \underbrace{\mathbf{h}_{l}(n)\sum_{q=2}^{Q}\big([\mathbf{h}^{(-1)}_{1}(n)\mathbf{h}_{q}(n)] + \tilde{\mathbf{h}}_{p}(n)\big)\mathbf{x}^{T}_{q\cdot(q=1)}(n)}_{\text{non-linear component}} \end{aligned} \tag{6.33}$$

Equation 6.33 shows that, if the linear component $\tilde{\mathbf{h}}(n)$ is an estimate of $\mathbf{h}_{l}(n)$, the first term (linear component) in Equation 6.33 goes to zero.

To minimize the second term (non-linear component) the estimate of each filter $\tilde{h}_{p}(n)$ should converge to $-h^{(-1)}_{1}(n) * h_{q=p}(n)$ with $h^{(-1)}_{1}(n) * h_{1}(n) = \delta(n)$ (where $\delta(n)$ is the Dirac function). This shows that $\tilde{\mathbf{h}}_{p}(n)$ is independent of the acoustic path. Thus, with a reliable estimate of the pre-processor, the updating process can be frozen without degrading the performance of the overall system. This is potentially beneficial in terms of reduced computational complexity and for robustness in adverse environments.

6.5 Summary of the different non-linear algorithms

Here we present a summary of the different non-linear AEC approaches. We classified them into three categories: Parallel Structure (PS), Cascaded Structure (CS) and Loudspeaker Pre-processing (LP). This summary serves as reference for the assessment presented in the next chapter.

6.5.1 Parallel structure

PS refers to algorithms that model overall LEMS as one non-linear system. We will use two models here the Volterra filter and the power filter. This latter corresponds to the PS version of the CS that we have presented previously. The Volterra filter is limited to the quadratic kernel. Hence, one filter is used to estimate the linear component and a second filter for the quadratic kernel output.

PS Volterra filter algorithm

- **linear filter**
  - $\hat{\mathbf{h}}_1(n+1) = \hat{\mathbf{h}}_1(n) + \mu_1 e(n)\mathbf{x}(n)$
  - complexity: $2 \times N$ multiplications
- **quadratic filter**
  - $\hat{\mathbf{h}}_Q(n+1) = \hat{\mathbf{h}}_Q(n) + \mu_Q e(n)\mathbf{x}_Q(n)$
  - complexity: $N^2 + N$ multiplications
- complexity: $N^2 + 3 \times N$

The power filter model is referred to PS in comparison to the CS version used in this thesis. It estimates different sub-filters which use different power expansion of the far-end signal as input.

PS algorithm

- **linear filter and non-linear sub-filter**
  - $\hat{\mathbf{h}}_p(n+1) = \hat{\mathbf{h}}_p(n) + \mu_1 e(n)\mathbf{x}_p(n)$
  - p is the power of the input, $p = 1$ corresponds to the linear filter, $P = 3$ is the number of sub-filters $(h_p(n))$
  - complexity: $2 \times P \times N$ multiplications

6.5.2 Cascaded structure

The CS uses a pre-processor to estimate loudspeaker parameters followed by a linear filter which is used to estimate the rest of the LEMS (acoustic channel and up-link devices).

CS algorithm

- **pre-processor**
  - **linear filter (acoustic channel and up-link devices)**
    - $\hat{\mathbf{h}}(n+1) = \hat{\mathbf{h}}(n) + \mu e(n)\mathbf{x}(n)$
    - complexity: $2 \times N$
  - **sub-filters (basic pre-processor elements)**

        - $\hat{\mathbf{h}}_p(n+1) = \hat{\mathbf{h}}_p(n) + \bar{\mu}_p(n)[\hat{\mathbf{h}}^T(n)\mathbf{X}_p(n)]^T e(n)$
        - p is the power of the input, $P = 3$ is the number of pre-processor sub-filters ($h_p(n)$)
        - complexity: $P \times (\bar{N} + 2 \times N_p)$ (assuming N_p is the same for all sub-filters, $\bar{N} = N$ but in practice it is better to truncate the filter $\hat{h}(n)$ to use $\bar{N} = N/2$ which corresponds to earlier taps which are significant and more robust to noise)
    - **clipping compensation (CC)**
        - $\hat{c}(n+1) = \hat{c}(n) + \mu_c \hat{\mathbf{h}}^T(n)\frac{\partial f_c}{\partial c}(\mathbf{x}(n))e(n)$
        - c is the clipping level. When the CC is used the algorithm is referred to Cascaded Structure with Clipping Compensation (CS + CC).
        - complexity: $\bar{N}$ multiplications

- **decorrelation filtering (DF)**
    - $\mathbf{w}(n+1) = \mathbf{w}(n) + \mu_w(n)\hat{\mathbf{y}}_P(n-1)\hat{y}_P^w(n)$
    - $\mathbf{y}_P(n-1)$ is the output of the pre-processor. $\hat{y}_P^w(n)$ is the prediction error. When the decorrelation filter is used the system is referred to Cascaded Structure with Clipping Compensation and Decorrelation Filtering (CS + CC + DF), without the clipping compensator it is referred to Cascaded Structure with Decorrelation Filtering (CS + DF).
    - complexity: $3 \times N_w + N + \Delta$ (Δ: process delay)

- complexity:
    - CS: $P \times (\bar{N} + 2 \times N_p) + 2 \times N$
    - CS + CC: $(P-1) \times (\bar{N} + 2 \times N_p) + 2 \times N + \bar{N}$
    - CS + DF: $(P-1) \times (\bar{N} + 2 \times N_p) + 3 \times N + 3 \times N_w$
    - CS + CC + DF: $(P-1) \times (\bar{N} + 2 \times N_p) + 3 \times N + \bar{N} + 3 \times N_w$

6.5.3 Loudspeaker pre-processing

The LP algorithm is a pre-processing approach where the pre-processor aims to linearise the output of the loudspeaker. It has similar elements as the basic CS.

LP algorithm

- **pre-processor**
    - **linear filter (acoustic channel and up-link devices)**
        - $\hat{\mathbf{h}}(n+1) = \hat{\mathbf{h}}(n) + \mu e(n)\mathbf{x}(n)$
        - complexity: $2 \times N$ multiplications
    - **sub-filters (basic pre-processor elements)**

  - $\hat{\mathbf{h}}_p(n+1) = \hat{\mathbf{h}}_p(n) + \bar{\mu}_p(n)[\hat{\mathbf{h}}^T(n)\mathbf{X}_p(n)]^T e(n)$
  - $p > 1$ is the power of the input, $P = 3$ is the number of pre-processor sub-filters ($h_p(n)$) with $h_1(n) = Z^d$ (d: processing delay in sub-filters 2 and 3).
  - complexity: $(P-1) \times (N + 2 \times N_p)$ multiplications

- complexity: $(P-1) \times (N + 2 \times N_p) + 2 \times N$

6.6 Conclusions

This chapter presents the application of Volterra filter to non-linear AEC for the special case of loudspeaker non-linearity. We show that the equivalent model of the global LEMS has a quadratic kernel with longer memory but the non-linearities memory is equivalent to that of the loudspeaker. This supports the assumption that the Volterra kernel is in general sparse and the fact that the significant taps are around the main diagonal.

We also show that the CS can be an efficient solution when the acoustic channel is assumed to be time variant. We propose three approaches to improve the baseline CS. These improvements aim to reduce the complexity of the iterative estimation procedure, efficiently incorporate a clipping estimator and improved linear AEC performance with a decorrelation filter.

As the parallel and cascaded structures both introduce distortion in the microphone signal we investigate the use of pre-processing in the Down-Link (DL) path. Based on LP this approach solves the problem of additional distortion in the microphone signal. Being focused on the loudspeaker properties it does not require re-initialisation. The drawback in this approach is that it introduces some distortion in the loudspeaker at lower frequencies.

In the next chapter we present the assessment of each solution with synthesised and real recorded data.

Chapter 7

Non-linear AEC assessment

This chapter presents an assessment of the different algorithms with synthetized and real data. With synthetized data we focus on analysing the characteristics of the different algorithms as the environment is known. The performance of linear Acoustic Echo Cancellation (AEC), the different versions of the Cascaded Structure (CS), the Parallel Structure (PS) and the Loudspeaker Pre-processing (LP) performance are all compared. The Echo Return Loss Enhancement (ERLE) and System Distance (SD) are used as metrics for the assessment. The ERLE, which measures echo reduction, is used in all assessments, however the SD which measures the performance of the algorithms to estimate the real echo path is used only for the LP assessment to assess linearisation performance.

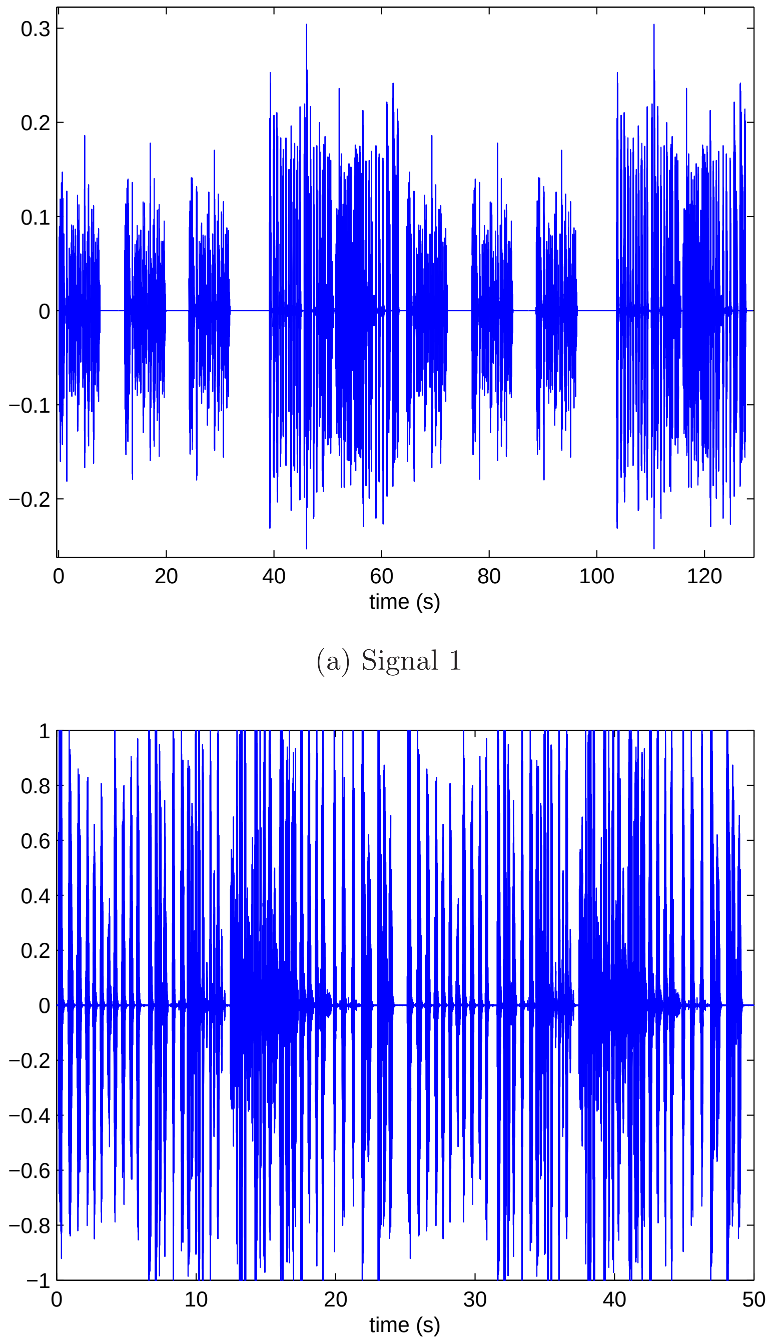
The analysis of real data is then used to assess the different algorithms where no a priori is available. In this case we additionally assess the PS Volterra filter as no assumption is made on the non-linearity model. However, the LP, which requires the signal to be processed before the loudspeaker cannot be assessed with recorded data and is not presented in this section. The ERLE is used for all assessments.

7.1 Analysis with synthetized data

The assessment performed in this section aims to reveal the behaviour of the different algorithms in conditions where the non-linearity model matches with the real system. The robustness in noise conditions is also presented using a signal-to-noise ratio of 50 and 30 dB. Before the analysis of the results we first describe the far-end speech signals and then the Loudspeaker Enclosure Microphone System (LEMS) which is the important element in AEC applications. The LEMS characteristics are presented as in Section 5.1. We use a non-linear model to simulate the loudspeaker and a linear model to simulate the concatenation of the acoustic channel and the up-link devices according to conclusions of Chapter 5. Latter, when clipping distortion is assessed, we additionally simulate a hard clipping effect which is applied before the loudspeaker model.

7.1.1 Simulation parameters

The different parameters used to build the synthetized environments are described here. Additionally the different algorithms and their parameters are also presented.



(a) Signal 1

(b) Signal 2

Figure 7.1: Signals recorded from a smart-phone and used for the simulation

Table 7.1: Characteristics of the speech signals

Characteristics	Signal 1	Signal 2
Long term energy (rms) [dBov]	-31.927	-14.444
Active speech level [dBov]	-30.295	-13.633
RMS peak-factor [dB]	31.836	14.444
Active peak factor [dB]	30.203	13.633
Activity factor [%]	68.665	82.966

Speech signals

Two speech signals are used in this analysis as far-end signals. They are illustrated in Figure 7.1. The difference between these two signals lies principally in their loudness. The first signal, which is of moderate, non-saturated volume is referred to as signal 1 whereas the second is a loud, saturated signal referred to as signal 2. These two signals are considered here to show the effect of non-linearities according to signal loudness, as signal loudness is known to be the main cause of non-linearity. The two signals were recorded at the loudspeaker input point of a smart-phone and used here as far-end signals. The characteristics of each signal are given in Table 7.1. Characteristics are estimated using the active level estimator (actlev function) of the software tools for speech and audio coding standardization produced by the ITU Telecommunication Standardization Sector (ITU-T) [ITU-T 2011, ITU-T 2010, ITU-T 2009]. Comparing their long term energy we notice 17.5 (dBov) less energy in Signal 1 than in Signal 2. There is also a high degree of inactivity in Signal 1 than in Signal 2. These difference are clearly evident in Figure 7.1. The two signals are applied to a synthetized LEMS to generate the echo signals. The following subsections present the different components and parameters which are assumed to have an influence on the LEMS.

Down-link devices

- **Amplifier (clipping distortion)**:
  Clipping distortion is assumed to be generated by the amplifier. Here we simulate a hard limiter which is applied only for the purposes of clipping compensation assessment presented in Section 7.1.4 where the limited value is fixed to 0.5. This means that all samples whose amplitude is above 0.5 are set to 0.5 with the same sign of the original sample. The objective of this assessment is to observe the interest of using a clipping compensator in the presence of clipping distortion.

- **Loudspeaker** The loudspeaker is modelled by a power filter as described in Section 6.3. Here 3 sub-filters are assumed with a length equal to 50 taps. Sub-filters parameters are measured with signals recorded from real devices. A random signal is sent to a loudspeaker and recorded at the ear of the mannequin as described in Section 5.4.1 (see Figure 5.1). Then, assuming a power filter model of the loudspeaker, a Least Square (LS) procedure is used to estimate the model parameters. This approach neglects the filter between the loudspeaker output and the microphone which we assume is reasonable in handset mode.

Estimated sub-filters are then used in simulations to model the loudspeaker in our synthetized environment. The output of the loudspeaker is obtained by convolving each sub-filter with the corresponding input signal. Hence we can generate a synthetized loudspeaker output signal that will be convolved with the rest of the LEMS elements to generate the echo signal.

Acoustic channel and up-link devices

The rest of the LEMS is composed of the acoustic channel and the microphone which are assumed to be linear and modelled by one linear filter. Assuming that changes may arise in the near-end environment three linear filters are used in this work. They come from the Aachen Impulse Responses (AIR) database [Jeub *et al.* 2009, Jeub *et al.* 2010]. The AIR database contains different impulse responses measured in different conditions. The three used here correspond to impulse responses measured in a kitchen of size 5.20 m × 2.60 m, an office room (5.00 m × 6.40 m) and a lecture room (10.80 m × 10.90 m). All impulse responses have a sampling frequency of 48 kHz. For our simulation we down-sample these three impulse responses to a sampling frequency of 8 kHz which corresponds to the sampling frequency of narrow band communication.

The three impulse responses are truncated to 200 taps. They are then successively used to model the Echo Path (EP) (acoustic channel and microphone). The transition where we change the EP from one impulse response to another impulse response corresponds to echo path changes. These echo path changes happen in general when the speaker is moving or when a change arises in the environment.

7.1.2 Algorithms

Here we describe the algorithms used in this assessment and their respective parameters. While significant experiment has been conducted with different parameters, we provide here a subset which is representative of general trends and which illustrates the main differences.

Linear AEC

The linear AEC is based on the Normalized-LMS (NLMS) algorithm and the parameters are chosen so that it can provide a good performance and less disturbance during all the test especially when the noise level increases. The parameter step-size is hence set to $\mu = 0.5$ and the regularization factor $\xi = 0.1$. The length is chosen according to the concatenation of the loudspeaker linear component which is equal to 50 taps (length of the linear impulse response) and the impulse responses (acoustic channel and up-link devices) which have 200 taps. Hence the linear AEC filter length has 250 taps.

Parallel structure

The parallel structure (PS) is described in Sections 6.2.1 and 6.5.1. It is based on a power filter model which used 3 sub-filters of the same length. The first sub-filter is used to estimate the linear component of the echo. The second and third sub-filters are used to estimate the non-linear echo component. Each sub-filter is estimated using an NLMS approach. In this case each sub-filter needs to have a minimum length equal to that of the echo path which also involves the loudspeaker model, i.e. 250 taps.

The parallel structure uses the 250 taps in each sub-filter. The linear part of the parallel structure ($h_1(n)$) is parameterised identically to the linear AEC with same step-size and regularization factor. The other sub-filters are parameterised to provide a better result Signal 2 than Signal 1 since the former correspond to the high non-linear case. As expected their step-size is lower compared to $h_1(n)$ for stability reasons. Sub-filter step-sizes are equal to $\mu_{p=2,3} = 0.1$ with a regularization factor $\xi_{p=2,3} = 0.1$.

Cascaded structure

The cascaded structure (CS) which is the focus in this work is used to divide the LEMS system into two blocks: a first non-linear block (pre-processor) which models the loudspeaker and a second linear filter which models the remainder of the LEMS which is assumed to be linear.

The pre-processor uses a power filter model of the loudspeaker and, as for the PS, 3 sub-filters. As described in Sections 6.2.1 and 6.5.2 they are based on an NLMS algorithm. The outputs of the sub-filters are summed up and form the input to the linear filter. This linear filter is also based on a NLMS algorithm.

This system has $P = 3$ sub-filters where each of them has 50 taps. Each sub-filter is parameterised with a small step-size ($\mu_1 = 0.001$, $\mu_{p=2,3} = 0.01$) and a small regularization factor ($\xi_{p=1,2,3} = 0.0001$). Smaller step-sizes are used to ensure system stability and that of the first sub-filter ($p = 1$) is chosen smaller to avoid fluctuation around different solutions. The linear filter of the cascaded structure is parameterised similarly to the linear AEC with 250 taps and a step-size equal to $\mu = 0.5$.

Improved cascaded structure

The improved cascaded structure is described in Sections 6.3 and 6.5.2. It combines Decorrelation Filtering (DF) and Clipping Compensation (CC). Decorrelation filtering is applied to the output of the pre-processor of the CS. Then the decorrelated signal forms the input to the linear AEC. This procedure uses an adaptive linear prediction analysis based on a NLMS algorithm to decorrelate the pre-processor output. This allows fast convergence of the linear filter and by using the updated version of the filter in the reconstruction of the true echo signal, it also improves the tracking capability.

The cascaded structure and decorrelation filter use the parameters as the CS only we add the DF process and set the first sub-filter to a delta function. The filter length should be low order and is equal to 3 with a step-size equal to $\mu_w = 0.001$. The second branch of the decorrelation filter is just a duplication of the estimated filter and does not require any parameterization.

CC is based on the estimation of one parameter which represents the clipping level. This parameter estimation is based as well on adaptive estimation.

The Cascaded Structure with Clipping Compensation (CS + CC) are implemented as separate blocks which aim to enhance the CS in the basic presence of clipping distortion. All the parameters are kept similar to those used for the basic cascaded structure except that $h_1(n)$ is again set to a delta function. The step-size of the CC is again small and equal to $\mu_c = 0.01$ with a regularization factor equal to 0.1.

The global CS which combines DF and CC in one module is referred to as the Cascaded Structure with Clipping Compensation and Decorrelation Filtering (CS + CC + DF). The CS + CC + DF combines the two systems with their original parameterisation. However, in this case the DF is controlled by the CC, as explained in Section 6.3. When a clipping effect is detected the decorrelation procedure update is paused.

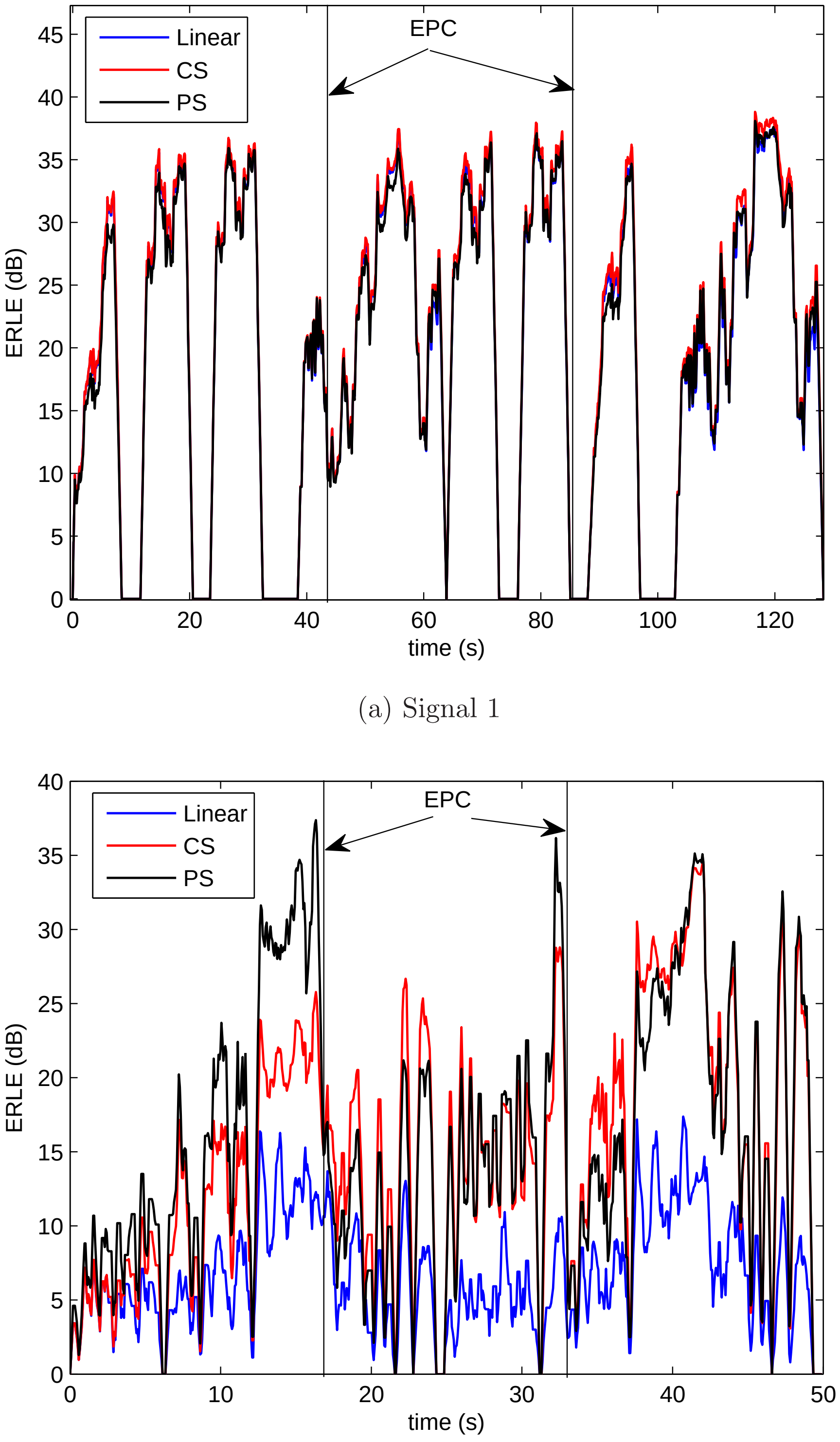
Loudspeaker pre-processing

As with the CS approach, the loudspeaker pre-processing LP approach is based on the use of a pre-processor, however in this case the objective of the pre-processor is not to emulate the loudspeaker but to linearise the loudspeaker output. The objective here is to generate some non-linearities that are opposed in phase to those generated by the loudspeaker at its output. In contrast to the CS pre-processor it uses two sub-filters whose estimation is also based on a NLMS algorithm.

The loudspeaker pre-processor has similar characteristics to the CS. The linear filter of the LP is parameterised as the linear AEC with as step-size 0.5 and 250 taps. The sub-filters have 50 taps. Their step-sizes are similar to that of the CS ($\mu_{p=2,3} = 0.01$ for all sub-filters) and their regularization factors are $\xi_{p=2,3} = 0.1$.

In the simulation we first present an assessment of the linear AEC, the cascaded approach and its parallel version as given in Figure 6.9. Assessment are performed for different signal-to-noise ratios without control of the step-size and for two different speech signals; one of a low level signal (Signal 1) and another loud signal (Signal 2) . They are used together to show the impact of varying loudness in AEC approaches as it can be a source of non-linearity.

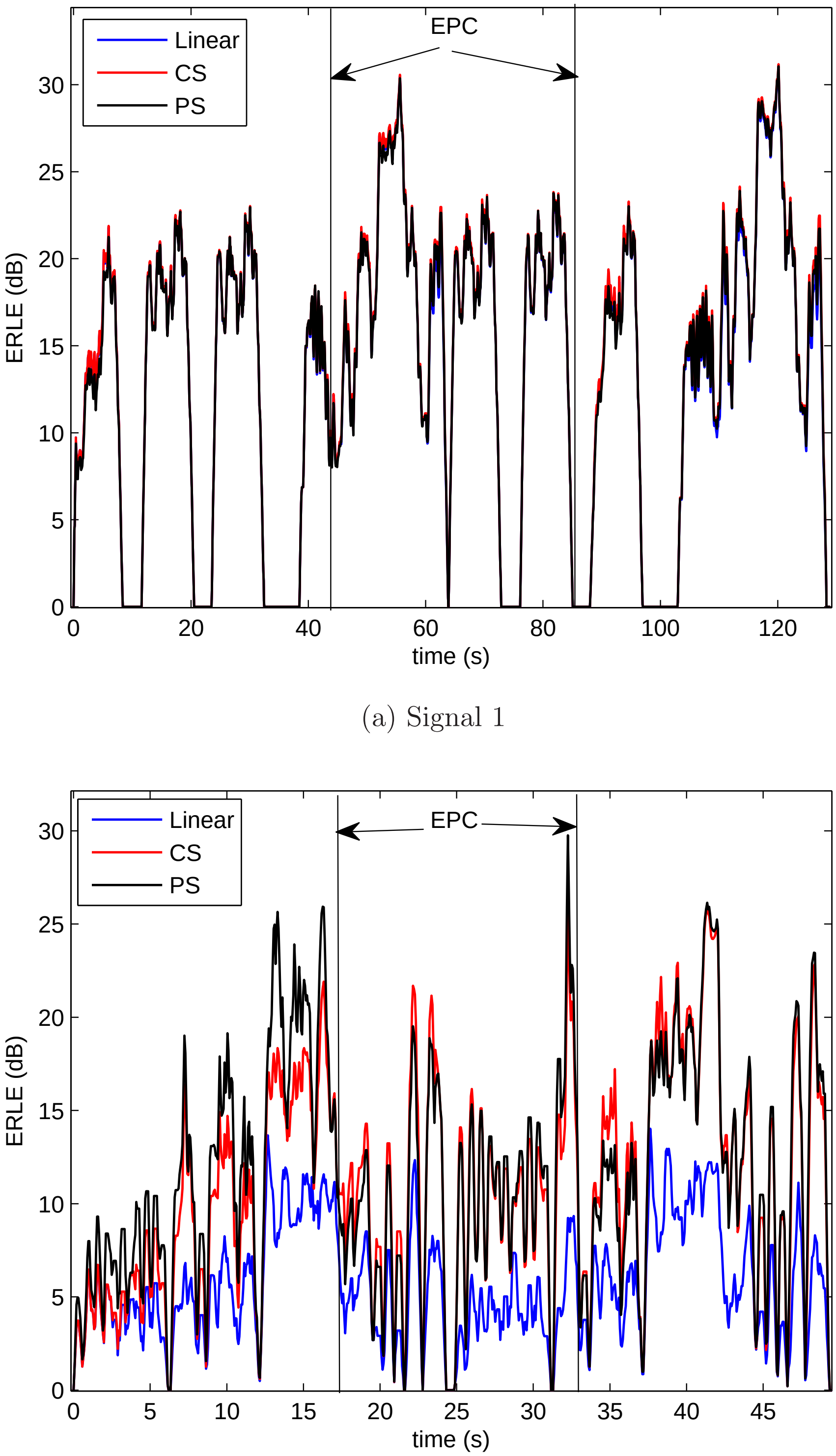
In the LP we show that when loudspeaker parameters are static, as is the case for simulations presented here we can pause the pre-processor updates without losing much in performance.



(a) Signal 1

(b) Signal 2

Figure 7.2: ERLE over time for linear AEC (NLMS) and non-linear AEC (Cascaded structure (CS) and Parallel structure (PS)) with SNR of 50 dB. The lower level signal (a) introduces less distortion than the higher level signal (b).



(a) Signal 1

(b) Signal 2

Figure 7.3: ERLE over time for linear AEC (NLMS) and non-linear AEC (Cascaded structure (CS) and a Parallel structure (PS)) with a SNR of 30 dB. The lower level signal (a) introduces less distortion than the higher level signal (b).

7.1.3 Assessment parallel and cascaded structures

In this section the linear AEC, PS and CS approaches are assessed in two different noise conditions. The first condition has an SNR of 50 dB and the second an SNR of 30 dB. The assessment is based on the ERLE metric. We focus on the analysis of echo reduction and convergence behaviour. Two aspects of convergence are mentioned here: initial convergence, which corresponds to the beginning of the process with filter taps initialized to zero, and the convergence after echo path changes.

SNR of 50 dB

Figure 7.2 illustrates the ERLE profiles over time for the linear and non-linear AEC approaches using the two different speech signals. We observe different behaviour for the two signals. Results for Signal 1 are illustrated in Figure 7.2 (a) in which we observe better performance with the linear AEC and the CS. Results for Signal 2 are illustrated in Figure 7.2 (b). Here the PS structure shows better performance. We also observe that with Signal 1 (Figure 7.2 (a)) the linear AEC algorithm and CS provide better convergence than the PS. This is explained by the fact that the non-linearities in this case are respectively lower so that, at the beginning of the process the system behaves as linear. Also the presence of the noise may perturb the estimation of non-linearities. Difference in convergence is also observed during the two echo path changes around 43 and 86 s.

For Signal 2 in Figure 7.2 (b), the signal is louder and often saturated. We observe that the convergence behaviour changes completely. At the beginning of the process the PS provides better convergence than the CS and linear AEC. This is expected as lower step-sizes are used in the CS pre-processor. From the beginning until 5 s the linear AEC and CS show similar behaviour. Afterwards, however, the CS starts to provide better performance than the linear AEC. This shows the effect of applying lower step-sizes to the CS pre-processor. Low step-sizes provide a stability but also lead to slower convergence.

Still referring to Signal 2 and Figure 7.2 (b), after approximately 15 s, before the first Echo Path Change (EPC) the PS gives 10 dB more ERLE than the CS and 20 more than the linear AEC algorithm. This is explained by the slow convergence of the CS or presence of local minima. In contrast to observations for Signal 1, we observe with Signal 2 that the convergence at the beginning of the process and after the two EPCs is not similar. In fact, when EPCs occur around 16.5 and 33 s we observe that the CS provides better convergence which is due to the lower step-size used in the pre-processor. Whereas these lower step-sizes reduce initial convergence they increase robustness against EPCs. Echo path changes normally affect the pre-processor as well but will be imperceptible with low step-sizes.

Other tests with different parameters have shown that we can reach a better ERLE than those illustrated in Figure 7.2 (a) for Signal 1 with the PS but these same parameters will induce divergence with Signal 2. Higher regularization factors can provide better stability but result in a lower ERLEs for the PS than for the CS and the linear AEC algorithm.

SNR of 30 dB

Figure 7.3 illustrates the ERLE over time for each of the different algorithms using the same speech signals. But now for an SNR of 30 dB we observe that when the noise level increases the ERLE is lower in all cases. This is expected as the algorithms are all limited by the noise level. For Signal 1 in Figure 7.3 (a) we again observe faster convergence with the linear AEC and CS than for the PS. However, the differences in convergence at the beginning and after EPCs are less noticeable with increased noise. The linear AEC algorithm and CS now provide similar performance to the PS. This shows that, depending on the noise level, non-linearities may have less impact than noise which is shown in Chapter 5. Nevertheless the CS shows slightly better echo reduction than other algorithms.

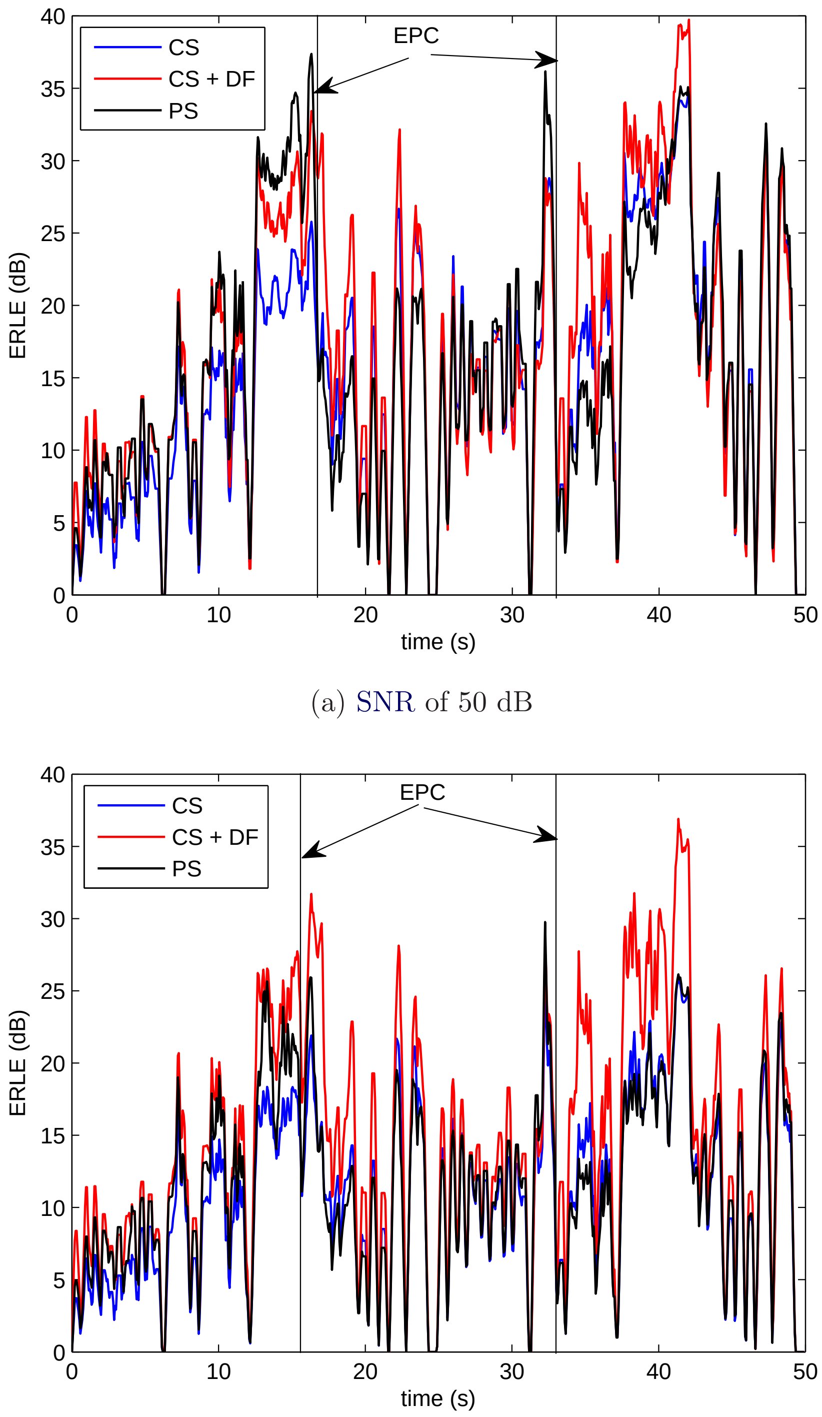
Figure 7.3 (b) shows that even when the noise level increases and under higher levels of non-linearity the PS provides better performance. We observe the same effect in Figure 7.2 (b), around 15 s before the first EPC the ERLE for the PS is about 7 dB higher than that of the CS. When EPCs arise around 16.5 and 33 s the CS again converges more quickly but the difference compared to the PS is reduced by the increase in noise (Figure 7.2 (b) and Figure 7.3 (b)). Normally, the CS is expected to be more affected by noise. In fact as the pre-processor depends on the linear filter estimate, when the noise increases the linear filter estimate is less accurate and degrades the pre-processor. Here only the earlier taps of the linear filter are used which is supposed to be more robust to noise and explains the fast convergence of the CS in this case.

The CS approach is efficient under conditions where non-linearities are low in which case it provides comparable results to the linear filter. Experiments with the second signal (Signal 2) shows that the pre-processor is more robust against EPCs when the signal level is high but that it converges slowly at the beginning of the process. Increased noise, however, affects the convergence of the pre-processor estimate by affecting the linear filter estimate which hence leads to reduce the performance. However, it is shown that the PS offers faster convergence and better echo reduction than the CS when the non-linearities are high except during EPCs.

Now we focus on signal 2 and show that decorrelation filtering DF can improve convergence of the CS and hence provide better echo reduction. Then we present the CC assessment which assumes clipping distortion introduced before the loudspeaker.

7.1.4 Improved cascaded structure assessment

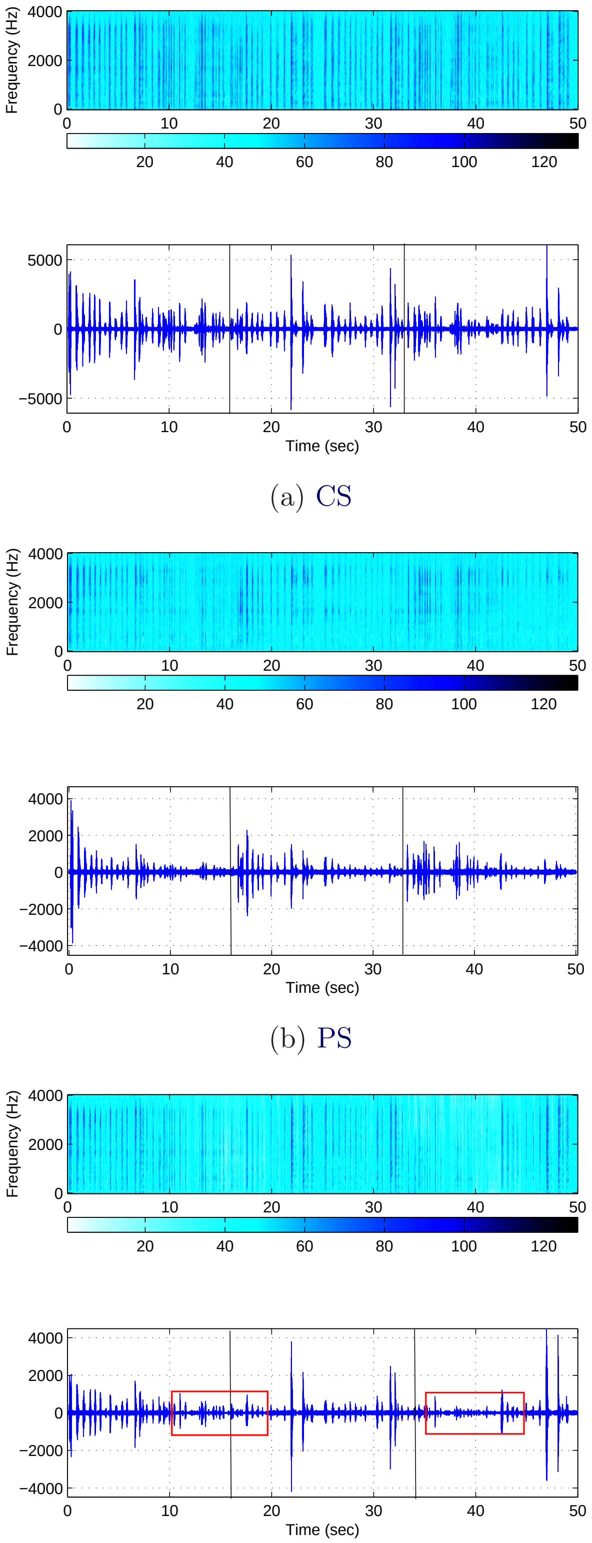
The assessment of the improved cascaded structure presents first the improvement brought by the use of DF. Whereas the second step focuses on the use of CC and the combination of CS + CC + DF. Assessments presented are based on the Signal 2 where CS shows slow convergence.



(a) SNR of 50 dB

(b) SNR of 30 dB

Figure 7.4: ERLE over time of non-linear AEC (Cascaded structures (CS, CS + DF) and Parallel structure (PS)) with SNR of 50 and 30 dB.

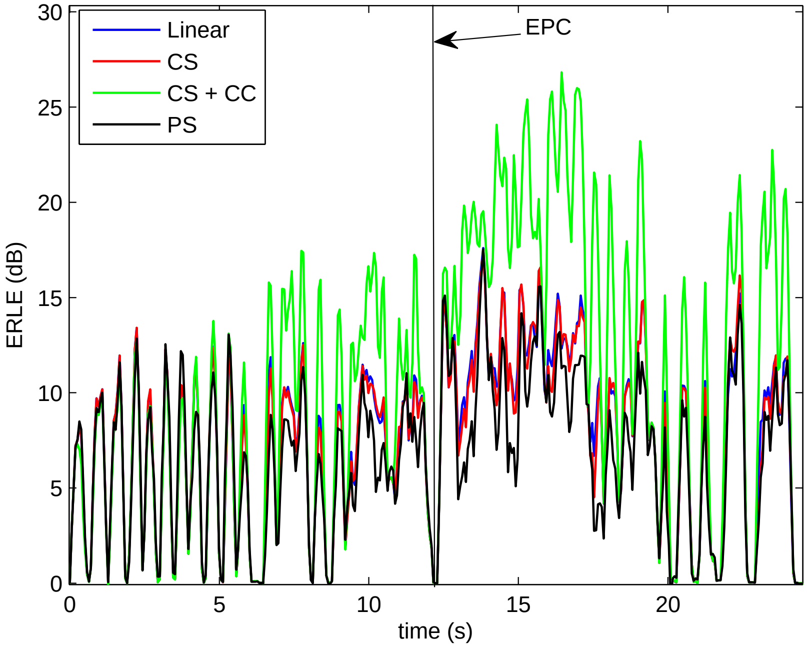


(a) CS

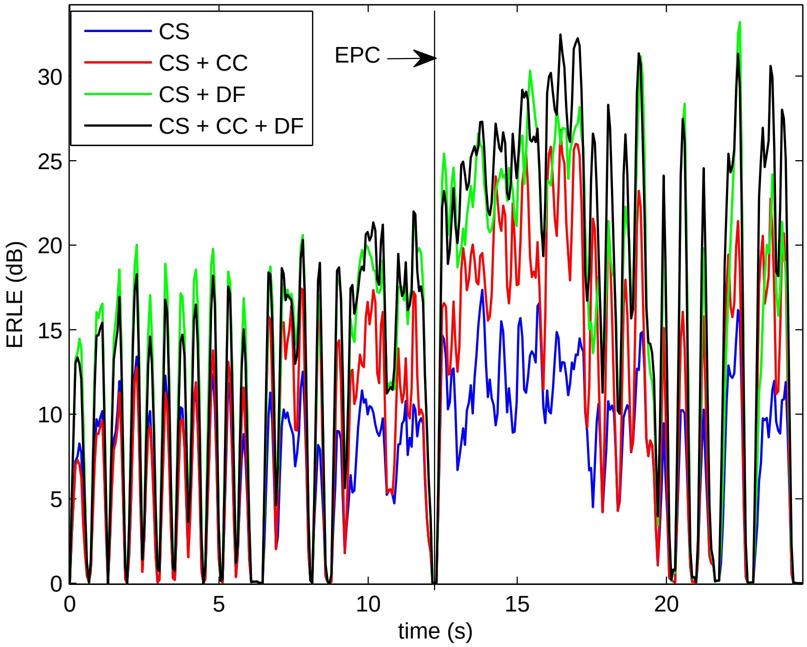
(b) PS

(d) CS + DF

Figure 7.5: Output error of Cascaded structures (CS, CS + DF) and Parallel structure (PS)) with SNR of 30 dB. The vertical lines show the different echo path change and the red rectangles show where CS + DF attenuated the error. Note that the error was scaled by $2 * 10^6$ to make visible the noise attenuation.



(a) Signal 2



(b) Signal 2

Figure 7.6: ERLE over time of linear AEC (NLMS) and non-linear AEC. (a) Linear AEC, Cascaded structures (CS, CS + CC) and Parallel structure (PS) with an SNR of 30 dB. (b) Cascaded structures (CS, CS + CC, CS + CC + DF).

Decorrelation filtering

Figure 7.4 illustrates the ERLE over time for the CS, the CS + DF and the PS. The CS and PS results are the same as those given in Figures 7.2 (b) and 7.3 (b). Results are shown for Signal 2 and SNRs of 50 and 30 dB. Upon comparison of the ERLE profiles in Figure 7.4 (a) we observe that the CS + DF converges faster than the CS. This is explained by the use of a decorrelated signal in the linear filter. We observe that during EPCs around 16.5 and 33 s the CS + DF reacts faster than the CS and PS. We see that just after the EPC the CS + DF has about 10 dB more ERLE than the CS and PS around 16.5 s.

In Figure 7.4 (b), where the SNR is equal to 30 dB, we observe that the CS + DF still provides a better initial convergence also after EPC. However, the ERLE of the CS + DF is much higher than that of the other algorithms when the SNR is equal to 30 dB compared to 50 dB of SNR. Hence we focus on a representation of the error to observe if the noise was not affected by the DF procedure.

Figure 7.5 shows the output errors in the time domain and corresponding spectra for the three algorithms. As expected we observe that the error attenuation of the different algorithms is in line with the ERLE. We also observe around 22, 32 and 48 s some peaks in the CS and CS + DF error which are not well observable with the ERLE. These effects may be related to the shape of the original signal or a mismatch in the estimators. However, the peaks in the CS + DF error signal are smaller than that in the CS error. Additionally we see that the error is slightly attenuated (red blocks) in certain periods when using the CS + DF. This is due to the echo signal estimation procedure which uses the updated version of the filter in high noise conditions. This implies that a control is required in high noise conditions even if no spectral domain distortion is introduced by this approach. We also observe that the PS error has higher energy in high frequencies after EPCs which is explained by the convergence time required in sub-filters 2 and 3. This shows that after EPC the PS is less able to remove the non-linearities than CS + DF.

These results show that the DF improves the convergence (especially initial convergence) of the CS. In the following section we present a general model of the non-linear AEC which combines DF and CC and is referred to as CS + CC + DF. However, before assessing this solution we first focus on the improvement of the CC when using a CS based non-linear AEC. The CS approach, which additionally uses a CC, is referred to as CS + CC and is presented next.

Clipping compensation combined with decorrelation filtering

Here the objective is to assess the clipping compensation and the combined system with decorrelation filtering. We aim to show the importance of using a clipping compensator when clipping distortion arises. In this case the clipping parameter is set to 0.5 and we focus on clipping compensation results for Signal 2. As the level of Signal 1 does not reach the clipping level, results show comparable performance with the CS and CS + CC, as expected.

Figure 7.6 illustrates the ERLE of the different linear and non-linear approaches. Figure 7.6 (a) shows the ERLE of the linear AEC, CS, CS + CC and PS. We observe that, as all pre-processor modules use low step-sizes, the linear filter performance with CC increases slowly. We also see that performance is robust to EPCs which is important in AEC applications. Even if the CS and PS are based on a non-linear model, they do not achieve better performance than the linear AEC. This can be explained by the fact that clipping is not well estimated by a power series. Hence, in such conditions a simple linear system may be preferable, even if clipping distortion may not arise as often as in this simulation case. We observe that, after the echo path change at around 12.5 s, fast convergence is observed with the CC. This means that clipping compensation will improve robustness to EPCs as did the CS without clipping distortion.

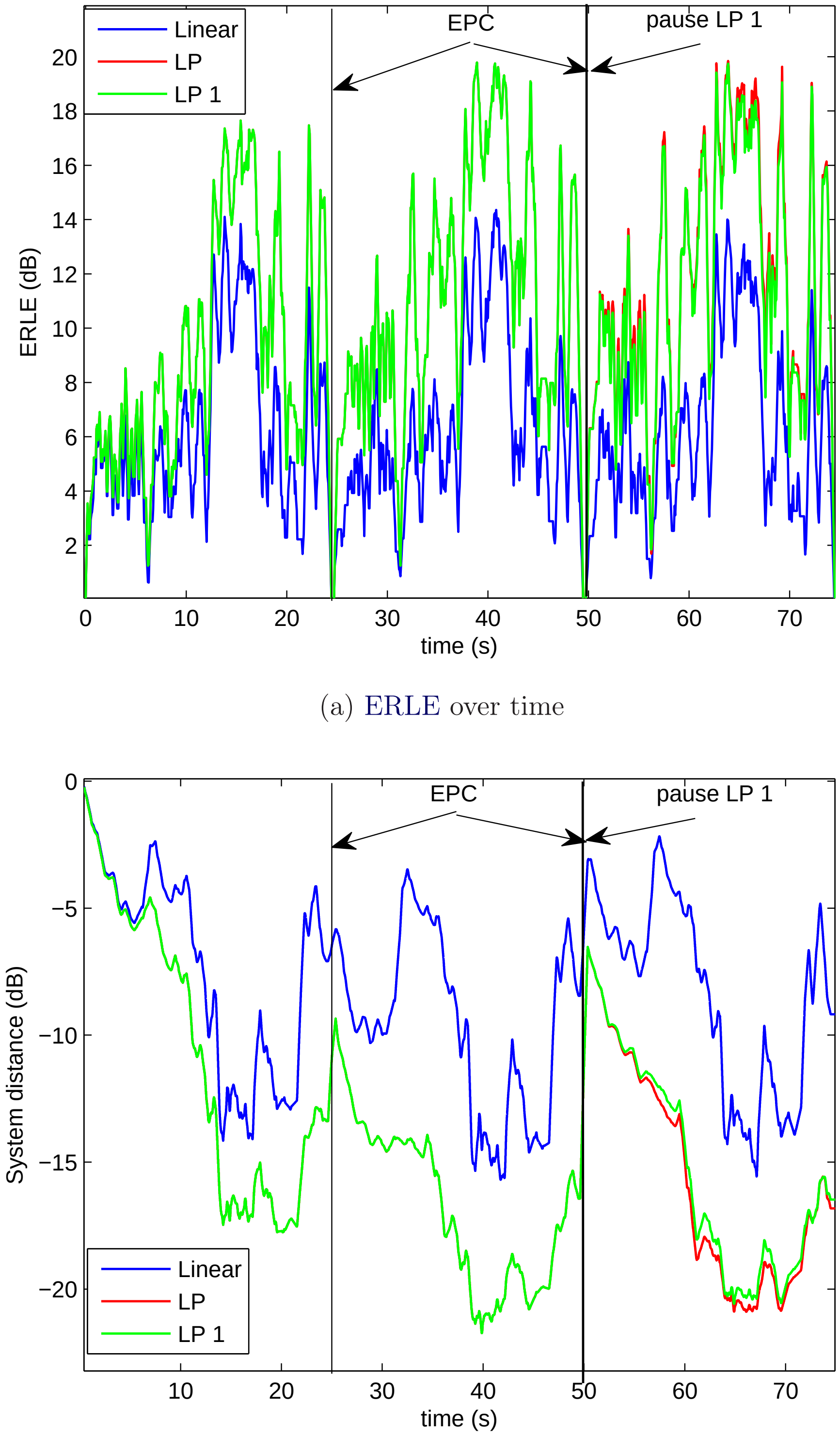
Figure 7.6 (b) shows the ERLE profiles of the improved CS approaches. We observe that CS + DF provides better performance, however, more attenuation of the noise signal may happen as the model is less accurate. We also observe that, when the system is well modelled with the CS + CC + DF, we have better performance. The improvement of CS + CC + DF is mainly shown around 15 s after the echo path change. This shows that it is better to have a more general model of non-linearities than CS + CC or CS + DF.

Better performance is shown initially for the CS + DF which is understandable as the CC converges very slowly to be robust during EPCs. After around 7 s we notice a difference between CS + CC and CS in Figure 7.6 (a) at this same period we observe that the CS + CC + DF starts to provide better performance compared to CS + DF. This shows that the CS + DF can be improved by CC. In fact the use of the CC better models non-linearities and helps DF to be more efficient.

7.1.5 Loudspeaker pre-processing assessment

The loudspeaker pre-processing (LP) assessment shows similar behaviour to that of linear AEC and the low level signal as we have seen for the CS. Here the LP is compared with the linear AEC with a similar level of noise. The SNR of the linear AEC is given by the power ratio between the echo signal (linear and non-linear components) and of that of the background noise. This SNR is the same for CS and PS. However, the SNR of the LP is the power ratio between the echo signal (where the non-linear component is already attenuated) and that of the background noise. Hence, the SNR refers here to that of the linear AEC. As LP may introduce distortion in the far-end signal the ERLE is not sufficient for assessment and the system distance (SD) is additionally used here to assess the linearisation process. Compared to the assessment of the CS and PS the ERLE of the LP take into account only the suppression of linear echo whereas that non-linearities are assessed by the SD.

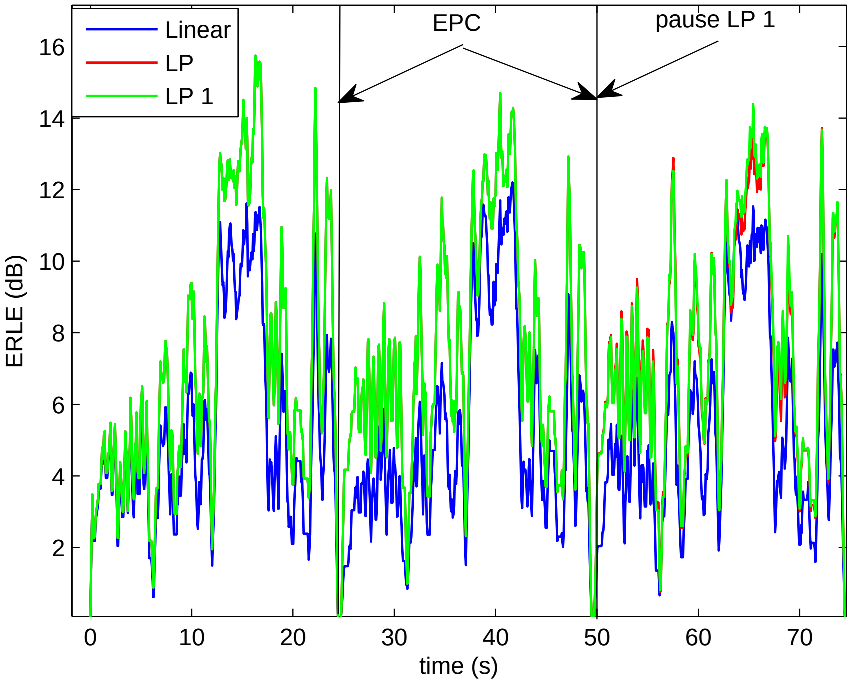
Linearisation performance is assessed using the SD between the linear filter, which results from the cascade of the loudspeaker and the acoustic path ($h_l(n) = h_1(n) * h(n)$), and the AEC filter $\hat{\mathbf{h}}(n)$. Note that the convolution of $\mathbf{h}_l(n)$ with the



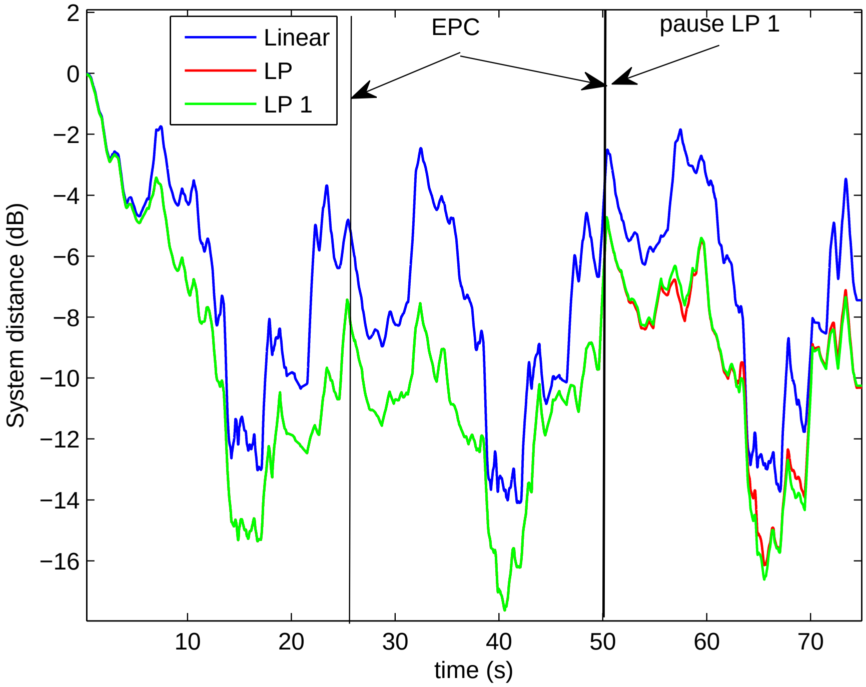
(a) ERLE over time

(b) SD over time

Figure 7.7: Comparative performance of linear AEC (NLMS) and loudspeaker pre-processing (LP, LP 1) for an SNR of 50 dB. The SD shows that the loudspeaker pre-processing has an accurate estimation of the far-end linear component.



(a) ERLE over time



(b) SD over time

Figure 7.8: Comparative performance of linear AEC (NLMS) and loudspeaker pre-processing (LP, LP 1) for an SNR of 30 dB. SD shows that loudspeaker pre-processing reaches an accurate estimate of the far-end linear component.

far-end signal gives the linear echo component. In this case the system distance is given by $SD(n) = |\frac{\mathbf{h}_l(n)-\hat{\mathbf{h}}(n)}{\mathbf{h}_l(n)}|$. We compare the system distance of the linear AEC (NLMS) algorithm both with and without linearisation pre-processing.

In this assessment we also focus on the independence of the pre-processing regarding the EP. Hence, after the second EPC, the pre-processor is paused in Loudspeaker Pre-processing 1 (LP1). To show that longer signals are used by replicating half of Signal 2 to have better convergence of the pre-processor which updating process is paused after the second EPC.

SNR equal to 50 dB

Figure 7.7 illustrates the ERLE and SD against time for Signal 2 which is used as far-end signal with SNR of 50 dB.

Echo reduction: The ERLE profiles in Figure 7.7 (a) shows that the loudspeaker pre-processing helps the system to converge faster than that of the linear AEC. We observe that, until after 8 s the LP convergence is slow. After approximately 15 s we observe a difference of ERLE about 10 dB compared to the linear AEC. We also observe that when the EPC arises the LP converges faster than the linear AEC which is a similar behaviour as with CS. After the second EPC we observe that the LP is slightly better than LP1 for which pre-processing adaptation has been paused. This means that the pre-processor continues to converge. The fact that these parameters are independent from the EPC allow an off-line estimation of these parameters in ideal conditions. Adaptation needs only to be performed occasionally and the parameters do not need to be reset upon each use (call).

Linearisation performance: Results are presented in Figure 7.7 (b) and show the system distance against time for each configuration. We observe that LP achieves better accuracy and shows an SD reduction in the order of 5 to 10 dB more than the linear AEC. This confirms the results obtained with the ERLE and the fact that the loudspeaker pre-processor provides an accurate estimate of the far-end signal at the output of the loudspeaker. An interest point is when we observe the EPC period around 25 and 50 s, we see that the two systems have a peak that shows the abrupt path change. Due to efficient parameter estimation the LP reconverges quickly whereas the linear AEC is perturbed in its re-convergence by the presence of non-linearities. This deviation of the linear AEC is in general due to noise and is explained in Section 4.4 which describes how non-linearities may deviate the estimated path from the linear Wiener solution. We observed during periods around 35 and 60 s that the linear AEC is highly disturbed whereas the LP approach still converges. This difference is more important at 60 s LP and LP1 converge whereas the linear AEC is highly disturbed. We also observe that after the second EPC the difference between the LP and LP1 profiles is small meaning that the pre-processor has converged. This cannot be done with the PS as all the parameters are environment dependent.

SNR equal to 30 dB

Figure 7.8 illustrates the ERLE and SD profiles for an SNR of 30 dB. As expected we see that all metrics show decreased LP performance.

Echo reduction: We observe in Figure 7.8 that the LP still gives better performance than linear AEC. Initially, and around the different EPCs at 25 and 50 s ERLE profiles show that the LP converges very quickly. The last part shows that, when the pre-processor is paused, LP1 can be better than LP. This can be explained by the fact that the pre-processor has not converged so that, when the pre-processor is paused in LP1 the constraint will be on the linear filter of the LP1 which tries to have a more efficient estimate according to the current state of the pre-processor. This means that the pre-processor has not completely converged but the linear filter can find the best estimates that gives the minimum error according to the current state of the pre-processor.

Linearisation process: Figure 7.8 (b) shows the SD profiles for the different algorithms. Upon comparison of the LP SD for 30 dB SNR and the linear AEC for 50 dB SNR we observe that the LP can reach a better linear echo component estimation than the linear AEC in lower SNR. This characteristic is better shown in terms of SD than ERLE as the latter is biased due to the pre-processing of the far-end signal. This characteristics is still dependent on the noise level and the level of non-linearities. The SD for the linear AEC and LP in Figure 7.8 (b) are closer. This can be explained by the slower convergence of the pre-processor and the presence of more noise. We also observe that the LP shows a better convergence than the linear AEC when EPCs arise. When the LP1 pre-processor is paused we observe that, during high level signal period, at around $55-60$ s the LP is better than LP1 whereas afterwards LP1 shows better performance. This can be explained by the fact that when the pre-processor is paused the linear filter reaches a lower error even if this is not a global minimum. Note that the bias in the linear path estimation is not only due to the pre-processor estimation accuracy. It is normal to expect that bias in presence of noise even when the pre-processor is accurate.

Results for a synthetized environment show that improvements to CS are brought by the DF and CC. The resulting system CS + CC + DF provides better convergence than the PS and allows for efficient echo reduction. On the other hand the LP is as to be efficient as the CS to EPCs and can be paused without significant impact on performance. However, this procedure is shown in a simulated environment where the loudspeaker is static. Real time processing tests are required to verify this procedure. In the following we report an assessment with real data recorded with a smart-phone.

7.2 Analysis with real data

Two different experiments are reported in this section. Both involve data recorded from a real smart-phone. The first aims to assess tracking performance whereas the second aims to assess clipping compensation performance.

7.2.1 Data

Signals used for this test are extracted from the same database used for analysis with synthetic set-up. In the first experiment tracking performance is assessed using recordings of real mobile devices in hands-free mode with abrupt echo path changes and an interval containing high-level signals in order to induce clipping. During the interval between 0 and 30 s the phone is placed on a table before being taken in hand from 30 to 40 s and then placed again on the table from 40 s until the end. In the second approach a loud signal is applied to the loudspeaker to assess increases in non-linear distortion.

7.2.2 Algorithms

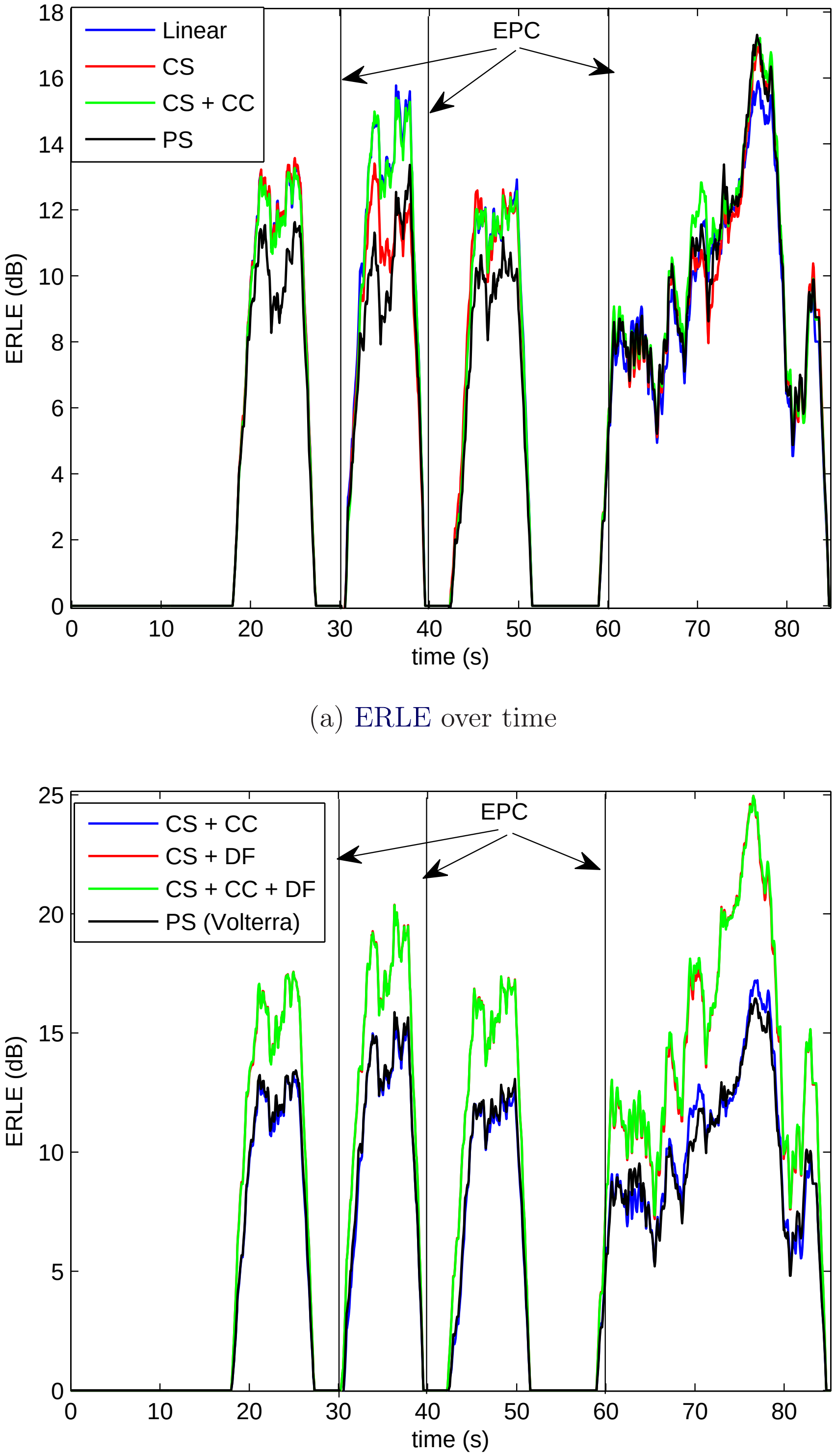
The same algorithms presented in the synthetized data analysis are used, with the exception of LP which requires online processing. We additionally assess the PS Volterra approach by assuming no a priori on the type of non-linearity. The same parameters used in the previous synthetized environment are kept with the exception of the length of the sub-filters which are reduced to 10 taps and lead to the use of 210 taps in the linear AEC and the PS sub-filters. This is to avoid increase error when the pre-processor is unknown. The volterra filter also has 210 taps in the first order kernel (linear filter) with the same parameters as the synthetized environment linear AEC. The quadratic kernel of the Volterra filter has 5050 taps which is equivalent to $\frac{(N^2-N)}{2} + N$ when the half matrix is used. The step-size of the quadratic kernel is equal to 0.1 and a regularization factor to 1. These parameters have been shown to ensure its stability along the different tests and provide better ensemble results.

7.2.3 Tracking performance

Figure 7.9 (a) illustrates ERLE profiles for the different solutions. We observe that the linear AEC and the CS + CC obtain better performance than the other solutions. This can be explained by the low signal level which introduces less distortion so that linear AEC and CS are more efficient. This behaviour has been observed in low level signals even in the simulation and confirms the comparable performance of CS to linear AEC in linear environments. We observe that the convergence of the PS is slow due to the low level of non-linearity that is not easy to estimate.

After the first EPC we observe better convergence for the CS than the PS, as before. This shows that the CS convergence is not negligible even in a real test case. We also observe that at around 32 s the CS is also perturbed whereas the CS + CC is stable. This can be explained by the fact that the integration of the clipping compensation in the pre-processor leads to a better stability. This difference is due to the constraint on sub-filter 1 in the CS + CC. In fact when the system is quasi linear, as in this case, we can expect the constraint $h_1(n) = 1$ to obtain better results than the CS.

The second EPC around 40 s shows similar behaviour for each of the different algorithms even if we observe in this case that the CS is more stable and better



(a) ERLE over time

(b) ERLE over time

Figure 7.9: ERLE against time for the linear AEC (NLMS) and non-linear AEC approaches with Signal 1, with EPCs.

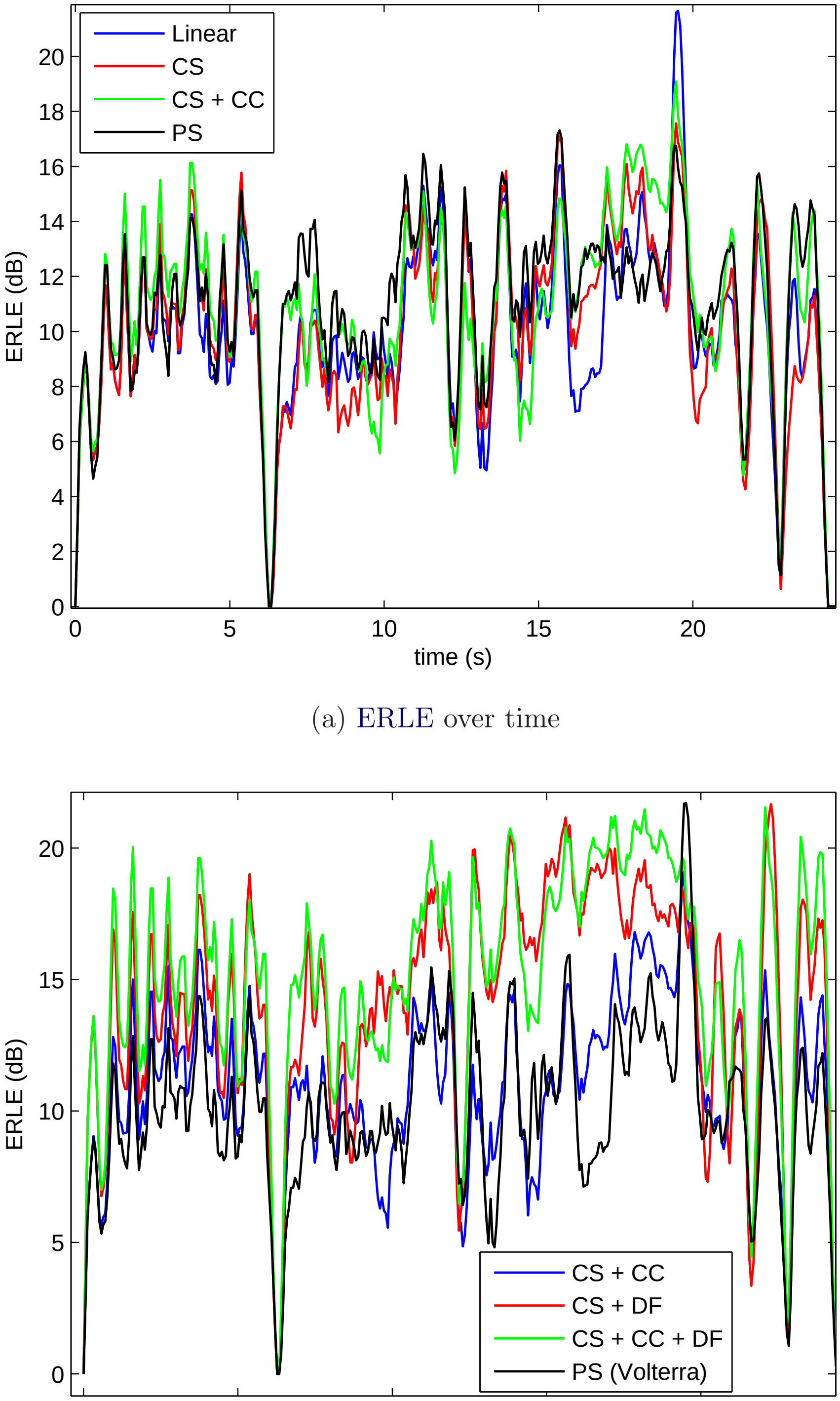
than the PS, which has slower convergence. To reduce the convergence time of the PS it is possible to increase the step-size of sub-filters 2 and 3, but in this case the system will be less stable. As a common error is used in the updating process of the different sub-filters, when the system is close to being linear, sub-filters 2 and 3 with high step-sizes may introduce more perturbations.

The third EPC is followed by a longer and louder period speech signal compared to the latter three parts. This period shows a change in algorithm behaviour. We again observe fast convergence in the case of the PS. We also observe that for the linear AEC the ERLE is slightly under that of non-linear systems. In certain periods we observe that the CS + CC has better performance which may be explained by some slightly short period of clipping distortion. Here the PS reaches similar performance to the CS which is explained by the loudness of the signal.

Performance for the more complex solutions are presented and the CS + CC is used as it has shown to be the best solution in the algorithms presented in Figure 7.9 (a). The CS + CC shows similar performance to the PS Volterra solution, even after EPC, whereas PS is expected to be lower. This can be explained by the decorrelation between the first order component and the quadratic component. This decorrelation helps the quadratic Volterra filter to behave as a linear filter in a linear environment but, if a third order is considered, the behaviour will change due to the correlation of first order and third order components. In the PS presented in the previous case (Figure 7.9 (a)) a third order component is used which is correlated with the first order component thus convergence is slower. However, we observe that in the loud signal period the CS + CC is better than the PS Volterra solution. This can be explained by the slow convergence of the Volterra solution or a mismatch between the real system and the Volterra filter. CS + DF and CS + CC + DF have better ERLE than the other systems, which is predictable due to decorrelation filtering. We observe that algorithms which use DF present similar performance, for certain periods the combination of DF and CC provide better performance than the CS + DF only. A significant improvement is obtained when using DF and 5 dB of difference of ERLE is observed compared to the other systems without decorrelation filtering.

7.2.4 Loud signal assessment

Figure 7.10 illustrates the performance of the different systems with a loud signal. Figure 7.10 (a) shows the behaviour of the linear AEC, the CS, CS + CC and the PS. We observe that the CS + CC converges more quickly at the beginning of the process than the other algorithms. This can be explained by the presence of some clipping distortion. Between 6 and 13 s better performance is provided by the PS which can be explained by a period without clipping distortion. Then, in a second period around 17 s CS + CC shows better performance which can also be explained by a period of clipping distortion. Around 19 s we observe a peak in the profile for the linear AEC which can be due to many reasons such as changes in the non-linear behaviour of the real system or a strong linear component. Except for this in general



(a) ERLE over time

(b) ERLE over time

Figure 7.10: ERLE against time for the linear AEC (NLMS) and non-linear AEC approaches with Signal 2.

the ERLE for the linear AEC is lower compared to the other systems but above that of the CS around 10 s.

Figure 7.10 (b) shows that the CS + CC is better than the Volterra approach at the beginning of the process but around the same period (11 – 16 s) during which PS in Figure 7.10 (a) shows better performance, we observe that the Volterra filter is also better than CS + CC. Then, at the period when the clipping effect arises around 17 s we observe better performance of the CS + CC compared to the Volterra filter. This shows that, in short periods, the Volterra filter cannot model a clipping distortion. Comparing the CS + DF and CS + CC + DF, we can observe some similarity with the CS + CC and Volterra filters. In most periods where the CS + CC is better than the Volterra filter, the CS + CC + DF is also better than CS + DF. This shows that the decorrelation filter can provide better performance when the model well-fits the real system. This is normal as the more the output of the pre-processor is linearly close to the loudspeaker output, the more the improvement brought by the decorrelation filtering. Here we also observe that the systems with decorrelation filtering have 5 to 9 dB more ERLE than systems without decorrelation filtering.

The real data analysis shows that as with simulated data the CS still present better convergence during EPCs. Some unexpected behaviours are observed with real data which can be explained by a mismatch with the non-linear model that may happen in real system. However, we observe that the DF procedure increases the performance of the CS even with real data, and can achieve better with CC. These tests show that the CS + CC + DF provides better results compared to the Volterra filter which is the baseline system generally used in non-linear AEC.

7.3 Conclusions

This chapter demonstrates the performance for each of the different algorithms. A simulated environment, where parameters are under full control, is used to assess the different algorithms. The simulated environment shows that where non-linearities are low the CS provides comparable performance to linear AEC. Under such conditions linear AEC and CS have faster convergence than the PS. Better convergence is explained by the fact that at the beginning of the process the system behaves like a linear system. The behaviour of PS is explained by some perturbations introduced by sub-filters 2 and 3.

However, when non-linear distortions are significant, as is the case with loud signals the PS shows faster convergence due to the ability to estimate non-linear component than CS which uses small step-sizes in the pre-processor. When EPCs arise the PS has more difficulty to re-converge than the CS which exploit the reliable output of the pre-processor. Hence the basic CS provides better performance with time variable EP non-linearities than PS, whereas the latter is more efficient in echo reduction in static environments.

In a second step we also show that the convergence and echo reduction of the CS can be improved using DF. The CS + CC scheme effectively tackles clipping which may arise with loud signal. With clipping compensation the CS + CC shows better performance than CS but the most efficient is the CS + CC + DF which combines clipping compensation and decorrelation filtering and is shown to outperform all other algorithms for all different conditions.

The LP is also shown to provide better performance than linear AEC by reducing the non-linear component. If the loudspeaker characteristics are static then the pre-processing can be paused without any significant loss in performance.

Finally we have used data recorded from a smart-phone with both a low level signal and a loud signal. Experiments with the low level signal show that linear AEC and CS have better performance especially if the mobile is moved in which case re-adaptation is required. When the signal is loud the PS can provide comparable performance to CS. The CS + CC shows better improvement with loud signals where clipping distortion is expected. We have observed, similarly to some simulation results, that the CS + CC is better in periods where clipping arises but sometimes the PS performs better in periods without clipping distortion and EPCs. Better results are provided by systems with decorrelation filtering. However, the best result is generally provided by the system combining decorrelation filtering and clipping compensation.

The assessment have shown that the CS structure is an efficient solution for non-linear systems with EPC. They can behave as linear systems in linear environments and provide faster convergence and better robustness to EPC in highly non-linear conditions. These two advantages are significant when considering the fact that the EPC problem is a challenging issue in linear AEC and more difficult for non-linear AEC.

This work has shown in terms of echo reduction that CS + CC + DF generally outperforms the different algorithms. This is due to its ability to model different types of distortion and to converge quickly.

Chapter 8

Conclusions and future work

8.1 Conclusions

This thesis relates to the problem of Acoustic Echo Cancellation (AEC) and specifically that of non-linearity. Following a review of the state-of-the-art solutions in linear and non-linear AEC are presented. Non-linear solutions are subdivided into four categories: the parallel structure, cascaded structure, loudspeaker pre-processing and non-linear echo post-filtering. We present an analysis of the different non-linear structures and assess the environments under which they are expected to be efficient. We show that in a time invariant environment the Parallel Structure (PS) is more robust and can provide better results than the Cascaded Structure (CS). On the other hand, if the environment is subject to some variability such as echo path changes then the CS is more efficient as it can more easily follow the changes to the environments.

Our contribution starts with an analysis of the effect of non-linearities on linear AEC. This analysis shows that solutions such as the Adaptive Projection Algorithm (APA) and Frequency Block LMS (FBLMS), which are known to provide better performance in linear environments, are far less efficient in non-linear environments. We then propose a theoretical analysis of the effects of non-linearity on linear AEC. The analysis is based on the Wiener solution of the echo path in non-linear conditions, with the assumption that non-linear and linear components are correlated in the case of speech signals. The theoretical analysis explains the behaviour of linear AEC in non-linear conditions. It also shows that some non-linear echo post-filtering may under or over estimate residual echo in the presence of non-linearity. This analysis demonstrates that, in non-linear environments, the simple Normalized-LMS (NLMS) is relatively robust and can be an appropriate choice compared to alternative, more complex solutions such as APA. Nonetheless, the loss in performance of linear AEC shows that efficient, specific solutions are required in non-linear environments.

Before investigating non-linear AEC solutions we first characterise the source of non-linearities in mobile phone environments. Measurements in real mobile devices confirm that the loudspeaker is the main source of non-linearities, as is widely acknowledged in the non-linear AEC literature. Based on these observations we propose two non-linear loudspeaker models: a frequency domain and a time domain model. Both models show a certain accuracy with real signals but the frequency domain model is significantly more complex than the time domain model. Fixed models are also not appropriate and so we propose to adaptively estimate the parameters of the time domain model.

Since the loudspeaker is the main source of non-linearities we adopt a non-linear Loudspeaker Enclosure Microphone System (LEMS) structure. This structure assumes a non-linear filter (pre-processor) representing the loudspeaker followed by a linear filter representing the concatenation of the acoustic channel and microphone responses. We show that only a small number of taps around the diagonal of the Volterra quadratic kernel are significant, as confirmed in the literature.

Regarding the high number of parameters required in the PS and the LEMS structure we decided to focus our work on a cascaded structure. We first propose to adapt the cascaded structure to the time domain loudspeaker model. The CS is based on two adaptive filters: a non-linear adaptive filter which estimates loudspeaker parameters (referred to as a pre-processor) and a second filter which is assumed to be linear.

The improved version of the CS advances the state-of-the-art. New developments relate to computationally efficient pre-processing and clipping compensation which aim to improve non-linear modelling and decorrelation filtering which aims to improve linear filter tracking performance.

In addition we propose a Loudspeaker Pre-processing (LP) approach where a pre-processing is applied before the loudspeaker to linearise its output. A linear AEC can be efficiently used in this case, as the LEMS to be estimated becomes a linear function. A drawback of this solution is that the LP introduces at high frequency distortion in the far-end signal.

We finally compare the linear AEC and the different non-linear AEC solutions (PS and CS) in two situations: a synthetized scenario and a real data scenario. The objective with the synthetized analysis is to compare the behaviour of the different approaches in an environment where the characteristics are known a priori. The subsequent use of real data validate our findings for mobile terminals.

It is shown that, when non-linearities are low, linear AEC provides comparable performance to CS. The amount of echo reduction achieved with the PS is slightly lower than that of linear AEC. We have also shown that linear AEC and CS converge faster than the PS at the beginning of a call and during echo path changes. When the system is highly non-linear, we observe different behaviour with each algorithm. The PS shows better convergence and echo reduction compared to the CS and linear AEC at the beginning of a call. However, when echo path changes arise (in which case the filter taps are not initialized to zero) the CS shows faster re-convergence than the PS, as expected from our analysis.

Tests with the improved CS structure show that Decorrelation Filtering (DF) improves the convergence of the CS even in highly non-linear conditions. The CS with DF shows faster convergence, better tracking and more echo reduction than the CS and PS. However, it is shown that, in noisy conditions the Cascaded Structure with Decorrelation Filtering (CS + DF) may attenuate noise. This leads to the requirement of a control based on noise power.

In the presence of clipping distortion better echo reduction is obtained when the CS incorporates Clipping Compensation (CC). The Cascaded Structure with

Clipping Compensation (CS + CC) provides better performance than CS and PS. We also show that when CS + CC is combined with DF faster convergence can be achieved. As a result, the full combination of Cascaded Structure with Clipping Compensation and Decorrelation Filtering (CS + CC + DF) provides the best overall performance.

The LP approach is assessed with linear AEC. LP provides better echo reduction and more efficient linearisation than the linear AEC alone. It is also shown that when the loudspeaker characteristics are static, as in the simulation, the pre-processor adaptation can be paused without any significant decrease in LP performance. This approach therefore has similar potential to off-line pre-processor estimation proposed in the literature.

The interest of the simulation is to determine the properties of the different approaches in a controlled environment. Our experiments show that the algorithms react as expected in such simulated environments. The simulation is nevertheless insufficient on their own in AEC assessment as different behaviours may arise in practice and the real system is never perfectly modelled as assumed in simulations. Signals recorded in real scenarios with a smart-phone are then used to assess the different algorithms. These results are difficult to clearly interpret since the real environment is not fully known, however, some of our expectations are confirmed.

We observe that when the far-end signal is of moderate volume (does not include clipping) linear AEC and the CS converge faster than the PS. In contrast when the signal is loud the PS shows better echo reduction than the CS and linear AEC. It is also shown that the Volterra filter provides comparable performance to the CS + CC. In these tests the CS + CC + DF shows the best results among the different systems and can be seen as an efficient choice for non-linear echo cancellation even in real environments. In the absence of clipping distortion CS and DF can be combined on their own to reduce the computational load.

Regarding these different tests results, the different algorithms can be classified according to two environments: a static environment where the most efficient solution is the PS in terms of echo reduction but with slightly low performance in quasi linear conditions. The PS Volterra solution is particularly complex so that CS + DF and CS + CC + DF are good compromise in performance versus computational load. In the case where the environment is variable, such as with echo path changes the best structure is the CS. This structure can furthermore be combined with DF to improve convergence. If clipping distortion is expected then the CS + CC + DF is the most efficient.

The LP was not assessed with real data as we would have had to develop a real-time system for such analysis. This is a limitation for recommendations as synthetized environment results do not always reflect results in real condition. Nevertheless, from simulations we can expect the LP to be at least better than CS. On the other hand the LP can also be combined with DF to improve convergence, even if it is potentially problematic to further combined LP and CC regarding their structure.

Finally, we focus on approaches based on loudspeaker pre-processing. We propose some improvements to the CS and an online loudspeaker linearisation to make more efficiency conventional linear AEC.

8.2 Perspectives

In this section we introduce some perspectives that can be investigated to improve non-linear AEC performance.

Non-linear acoustic echo post-filtering

In Section 4.4, the analysis of linear AEC behaviour in non-linear conditions shows that non-linearities can be considered similar in nature to the combination of correlated and uncorrelated components. This subdivision can be exploited to derive a two steps non-linear echo suppression.

correlated component: The correlated component induces the linear AEC to behave as in a non-stationary environment. Hence, it is required the use of linear AEC with fast tracking capability to follow the changes introduced by the correlated non-linear component. Fast-tracking, adaptive filters have previously received great deal of attention in the literature principally for time-variant systems. Even if some solutions for fast tracking already exist such as the Extended RLS (E-RLS) introduced in [Haykin *et al.* 1997], this task is challenging since the AEC will be perturbed by the uncorrelated component.

uncorrelated component: As the linear AEC cannot remove the uncorrelated component a post-filtering approach will be required to remove that component. The uncorrelated residual echo can be treated as noise but requires a further statistical analysis, such as power spectral density estimation, to be reliably tackled.

Multi-microphone non-linear AEC

Multi-microphone systems have been investigated in different domains, however, they are less investigated for non-linear AEC. This situation may be explained by the fact that the well-known Volterra solution in non-linear AEC will be highly complex. In this case the complexity of the PS in single microphone is multiplied by the number of microphones whose signals are treated. However, the CS and LP are more efficient in such circumstance since only one pre-processor is required to treat non-linearities of the different channels.

In Section 6.2.3 it is shown that CS is subject to local minima. This problem can be investigated by using multi-microphone approaches. Hence, the pre-processor can be constrained to satisfy different error minimization in the different channels in order to reduce the effect of local minima. These solutions can also rely on approaches developed in multi-microphone source localization and de-reverberation techniques.

Bibliography

[Addington & Schodek 2005] D. Michelle Addington and Daniel L. Schodek. Smart Material and New Technologies. Architectural Press, 2005. (Cited on page 15.)

[Arezki *et al.* 2006] M. Arezki, F. Ykhlef, A. Guessoum, P. Meyrueis and D. Berkani. *Stability of the fast recursive least square algorithms with perfectly predictable signals.* In Industrial Technology, 2006. ICIT 2006. IEEE International Conference on, pages 698 –703, dec. 2006. (Cited on page 26.)

[Avargel & Cohen 2009] Y. Avargel and I. Cohen. *Adaptive Nonlinear System Identification in the Short-Time Fourier Transform Domain.* Signal Processing, IEEE Transactions on, vol. 57, no. 10, pages 3891 –3904, oct. 2009. (Cited on page 43.)

[Azpicueta-Ruiz *et al.* 2009] L.A. Azpicueta-Ruiz, M. Zeller, J. Arenas-Garcia and W. Kellermann. *Novel schemes for nonlinear acoustic echo cancellation based on filter combinations.* In Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on, pages 193 –196, april 2009. (Cited on page 42.)

[Azpicueta-Ruiz *et al.* 2011] L.A. Azpicueta-Ruiz, M. Zeller, A.R. Figueiras-Vidal, J. Arenas-Garcia and W. Kellermann. *Adaptive Combination of Volterra Kernels and Its Application to Nonlinear Acoustic Echo Cancellation.* Audio, Speech, and Language Processing, IEEE Transactions on, vol. 19, no. 1, pages 97 –110, jan. 2011. (Cited on page 42.)

[Beaugeant *et al.* 1998] Christophe Beaugeant, Valérie Turbin, Pascal Scalart and André Gilloire. *New optimal filtering approaches for hands-free telecommunication terminals.* Signal Processing, vol. 64, no. 1, pages 33 – 47, 1998. <ce:title>Acoustic Echo and Noise Control</ce:title>. (Cited on page 47.)

[Bendersky *et al.* 2008] D.A. Bendersky, J.W. Stokes and H.S. Malvar. *Nonlinear residual acoustic echo suppression for high levels of harmonic distortion.* In Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on, pages 261 –264, 31 2008-april 4 2008. (Cited on page 48.)

[Benesty & Gansler 2001] J. Benesty and T. Gansler. *A robust fast recursive least squares adaptive algorithm.* In Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on, volume 6, pages 3785 –3788 vol.6, 2001. (Cited on page 26.)

[Benesty & Gay 2002] Jacob Benesty and Steven L. Gay. *An improved PNLMS algorithm.* In Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on, volume 2, pages II–1881 –II–1884, may 2002. (Cited on page 29.)

[Birkett & Goubran 1994] A.N. Birkett and R.A. Goubran. *Acoustic echo cancellation for hands-free telephony using neural networks.* In Neural Networks for Signal Processing [1994] IV. Proceedings of the 1994 IEEE Workshop, pages 249 –258, sep 1994. (Cited on pages 45, 46 and 74.)

[Birkett & Goubran 1995a] A.N. Birkett and R.A. Goubran. *Acoustic echo cancellation using NLMS-neural network structures.* In Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on, volume 5, pages 3035 –3038 vol.5, may 1995. (Cited on pages 45, 46 and 51.)

[Birkett & Goubran 1995b] A.N. Birkett and R.A. Goubran. *Limitations of hands-free acoustic echo cancellers due to nonlinear loudspeaker distortion and enclosure vibration effects.* In Applications of Signal Processing to Audio and Acoustics, 1995., IEEE ASSP Workshop on, pages 103 –106, oct 1995. (Cited on pages 4, 44, 50 and 58.)

[Boyd *et al.* 1984] S. Boyd, Leon O. Chua and Charles A. Desoer. *Analytical Foundations of Volterra Series.* Technical report UCB/ERL M84/14, EECS Department, University of California, Berkeley, 1984. (Cited on page 94.)

[Boyd 1985] S. P. Boyd. *Volterra Series: Engineering Fundamentals.* PhD thesis, University of California, Berkeley, 1985. (Cited on page 94.)

[Breining *et al.* 1999] C. Breining, P. Dreiscitel, E. Hansler, A. Mader, B. Nitsch, H. Puder, T. Schertler, G. Schmidt and J. Tilp. *Acoustic echo control. An application of very-high-order adaptive filters.* Signal Processing Magazine, IEEE, vol. 16, no. 4, pages 42 –69, jul 1999. (Cited on pages 21, 25, 28, 29, 52, 105, 110 and 112.)

[Bright 2002] A. Bright. Active control of loudspeakers: an investigation of practical applications. Report // Acoustic Technology, Technical University of Denmark. Ørsted DTU, Acoustic Technology, Technical University of Denmark, 2002. (Cited on page 79.)

[Burnett *et al.* 1988] T.D. Burnett, N.P. Morgan, J. Noble and E.V. Stansfield. *Echo cancellation in mobile radio environments.* In Digitized Speech Communication via Mobile Radio, IEE Colloquium on, pages 7/1 –7/4, dec 1988. (Cited on page 3.)

[Burrow & Grant 2001] S. Burrow and D. Grant. *Efficiency of low power audio amplifiers and loudspeakers.* In Consumer Electronics, 2001. ICCE. International Conference on, pages 322 –323, 2001. (Cited on page 73.)

[Burton *et al.* 2009] T.G. Burton, R.A. Goubran and F. Beaucoup. *Nonlinear System Identification Using a Subband Adaptive Volterra Filter.* Instrumentation and Measurement, IEEE Transactions on, vol. 58, no. 5, pages 1389 –1397, may 2009. (Cited on page 43.)

[Callender & Cowan 1990] C.P. Callender and C.F.N. Cowan. *Numerically stable fast recursive least squares algorithms for adaptive filtering using interval arithmetic.* In Digital and Analogue Filters and Filtering Systems, IEE Colloquium on, pages 5/1 –5/3, may 1990. (Cited on page 26.)

[Challa *et al.* 2007] D. Challa, S.L. Grant and A. Mohammad. *Variable Regularized Fast Affine Projections.* In Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on, volume 1, pages I–89 –I–92, april 2007. (Cited on page 25.)

[Chen *et al.* 2006] K. Chen, J. Lu and B. Xu. *A method to adjust regularization parameter of fast affine projection algorithm.* In Signal Processing, 2006 8th International Conference on, volume 1, 16-20 2006. (Cited on page 25.)

[Cioffi & Kailath 1984] J. Cioffi and T. Kailath. *Fast, recursive-least-squares transversal filters for adaptive filtering.* Acoustics, Speech and Signal Processing, IEEE Transactions on, vol. 32, no. 2, pages 304 – 337, apr 1984. (Cited on page 26.)

[Costa *et al.* 2003] J.-P. Costa, A. Lagrange and A. Arliaud. *Acoustic echo cancellation using nonlinear cascade filters.* In Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference on, volume 5, pages V – 389–92 vol.5, april 2003. (Cited on page 104.)

[Ding 2000] Heping Ding. *A stable fast affine projection adaptation algorithm suitable for low-cost processors.* In Acoustics, Speech, and Signal Processing, 2000. ICASSP '00. Proceedings. 2000 IEEE International Conference on, volume 1, pages 360 –363 vol.1, 2000. (Cited on page 25.)

[Duttweiler 2000] D.L. Duttweiler. *Proportionate normalized least-mean-squares adaptation in echo cancelers.* Speech and Audio Processing, IEEE Transactions on, vol. 8, no. 5, pages 508 –518, sep 2000. (Cited on page 29.)

[Eneman & Moonen 2003] K. Eneman and M. Moonen. *Iterated partitioned block frequency-domain adaptive filtering for acoustic echo cancellation.* Speech and Audio Processing, IEEE Transactions on, vol. 11, no. 2, pages 143 – 158, mar 2003. (Cited on page 43.)

[Enzner & Vary 2006] Gerald Enzner and Peter Vary. *Frequency-domain adaptive Kalman filter for acoustic echo control in hands-free telephones.* Signal Processing, vol. 86, no. 6, pages 1140 – 1156, 2006. <ce:title>Applied Speech and Audio Processing</ce:title>. (Cited on page 42.)

[Eweda 1994] E. Eweda. *Comparison of RLS, LMS, and sign algorithms for tracking randomly time-varying channels.* Signal Processing, IEEE Transactions on, vol. 42, no. 11, pages 2937 –2944, nov 1994. (Cited on page 69.)

[Farhang-Boroujeny 1998] B. Farhang-Boroujeny. Adaptive filters: theory and applications. Wiley, 1998. (Cited on pages 23, 31 and 33.)

[Fermo *et al.* 2000] A. Fermo, A. Carini and G.L. Sicuranza. *Analysis of different low complexity nonlinear filters for acoustic echo cancellation.* In Image and Signal Processing and Analysis, 2000. IWISPA 2000. Proceedings of the First International Workshop on, pages 261 –266, 2000. (Cited on pages 42, 51 and 93.)

[Frank 1994] W.A. Frank. *MMD-an efficient approximation to the 2nd order Volterra filter.* In Acoustics, Speech, and Signal Processing, 1994. ICASSP-94., 1994 IEEE International Conference on, volume iii, pages III/517 –III/520 vol.3, apr 1994. (Cited on pages 42, 45, 82, 118 and 120.)

[Frank 1995] W.A. Frank. *An efficient approximation to the quadratic Volterra filter and its application in real-time loudspeaker linearization.* Signal Processing, vol. 45, no. 1, pages 97 – 113, 1995. (Cited on pages 42 and 51.)

[Frank 1996] W.A. Frank. *Sampling requirements for Volterra system identification.* Signal Processing Letters, IEEE, vol. 3, no. 9, pages 266 –268, sep 1996. (Cited on pages 47 and 116.)

[Frenzel & Hennecke 1992] R. Frenzel and M.E. Hennecke. *Using prewhitening and stepsize control to improve the performance of the LMS algorithm for acoustic echo compensation.* In Circuits and Systems, 1992. ISCAS '92. Proceedings., 1992 IEEE International Symposium on, volume 4, pages 1930 –1932 vol.4, may 1992. (Cited on page 28.)

[Fu & Zhu 2008] Jing Fu and Wei-Ping Zhu. *A Nonlinear Acoustic Echo Canceller Using Sigmoid Transform in Conjunction With RLS Algorithm.* Circuits and Systems II: Express Briefs, IEEE Transactions on, vol. 55, no. 10, pages 1056 –1060, oct. 2008. (Cited on page 45.)

[Furuhashi *et al.* 2006] H. Furuhashi, Y. Kajikawa and Y. Nomura. *Realization of Nonlinear Acoustic Echo Cancellation by Subband Parallel Cascade Volterra Filter.* In Intelligent Signal Processing and Communications, 2006. ISPACS '06. International Symposium on, pages 837 –840, dec. 2006. (Cited on pages 43, 47, 117 and 120.)

[Gansler *et al.* 2000] T. Gansler, J. Benesty, S.L. Gay and M.M. Sondhi. *A robust proportionate affine projection algorithm for network echo cancellation.* In Acoustics, Speech, and Signal Processing, 2000. ICASSP '00. Proceedings. 2000 IEEE International Conference on, volume 2, pages II793 –II796 vol.2, 2000. (Cited on page 29.)

[Gao & Snelgrove 1990] X.Y. Gao and W.M. Snelgrove. *Adaptive linearization schemes for weakly nonlinear systems using adaptive linear and nonlinear FIR filters*. In Circuits and Systems, 1990., Proceedings of the 33rd Midwest Symposium on, pages 9 –12 vol.1, aug 1990. (Cited on page 82.)

[Gay & Tavathia 1995] S.L. Gay and S. Tavathia. *The fast affine projection algorithm*. In Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on, volume 5, pages 3023 –3026 vol.5, may 1995. (Cited on page 25.)

[Guerin *et al.* 2003] A. Guerin, G. Faucon and R. Le Bouquin-Jeannes. *Nonlinear acoustic echo cancellation based on Volterra filters*. Speech and Audio Processing, IEEE Transactions on, vol. 11, no. 6, pages 672 – 683, nov. 2003. (Cited on pages 45, 46, 50, 51, 104, 105, 109, 110, 112, 114 and 115.)

[Hänsler & Schmidt 2004] E. Hänsler and G. Schmidt. Acoustic echo and noise control: a practical approach. Adaptive and learning systems for signal processing, communications, and control. Wiley-Interscience, 2004. (Cited on pages 3, 14, 20, 22, 23, 26, 28, 30, 31, 33, 54, 110 and 112.)

[Haykin *et al.* 1997] S. Haykin, A.H. Sayed, J.R. Zeidler, P. Yee and P.C. Wei. *Adaptive tracking of linear time-variant systems by extended RLS algorithms*. Signal Processing, IEEE Transactions on, vol. 45, no. 5, pages 1118 –1128, may 1997. (Cited on pages 26 and 154.)

[Haykin 2002] S. Haykin. Adaptive filter theory. Prentice-Hall information and system sciences series. Prentice Hall, 2002. (Cited on pages 17, 19, 20, 23, 24, 25, 26, 28, 30, 64, 69, 108, 110, 112 and 120.)

[HEAD acoustics 2008] HEAD acoustics. *HQS-mobile Rev.04*. Head acoustic standard documentation, pages 13–20, June 2008. (Cited on page 75.)

[Hoshuyama & Sugiyama 2006a] O. Hoshuyama and A. Sugiyama. *An Acoustic Echo Suppressor Based on a Frequency-Domain Model of Highly Nonlinear Residual Echo*. In Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on, volume 5, page V, may 2006. (Cited on pages 63 and 71.)

[Hoshuyama & Sugiyama 2006b] Osamu Hoshuyama and Akihiko Sugiyama. *EVALUATIONS OF AN ECHO SUPPRESSOR BASED ON A FREQUENCY-DOMAIN MODEL OF HIGHLY NONLINEAR RESIDUAL ECHO*. In IWAENC 2006, International Workshop on Acoustic Echo and Noise Control, September 12-14, 2006, Paris, France, 09 2006. (Cited on page 47.)

[Hoshuyama & Sugiyama 2006c] Osamu Hoshuyama and Akihiko Sugiyama. *Nonlinear Acoustic Echo Suppressor Based on Spectral Correlation between*

Residual Echo and Echo Replica. IEICE Trans. Fundam. Electron. Commun. Comput. Sci., vol. E89-A, pages 3254–3259, November 2006. (Cited on pages 48, 71 and 93.)

[Houacine 1991] A. Houacine. *Regularized fast recursive least squares algorithms for adaptive filtering.* Signal Processing, IEEE Transactions on, vol. 39, no. 4, pages 860 –871, apr 1991. (Cited on page 26.)

[ITU-T 2009] ITU-T. *ITU-T Software Tool Library 2009 User's Manual.* Technical report, Nov 2009. (Cited on page 127.)

[ITU-T 2010] ITU-T. *ITU-T G.191: Software tools for speech and audio coding standardization.* Technical report, March 2010. (Cited on page 127.)

[ITU-T 2011] ITU-T. *ITU-T P.56 : Objective measurement of active speech level.* Technical report, Dec 2011. (Cited on page 127.)

[ITU 1996] ITU. *ITU-T P.58: Objective measuring apparatus, Head and torso simulator for telephonometry.* Technical report, Aug 1996. (Cited on page 75.)

[Jeub *et al.* 2009] M. Jeub, M. Schafer and P. Vary. *A binaural room impulse response database for the evaluation of dereverberation algorithms.* In Digital Signal Processing, 2009 16th International Conference on, pages 1 –5, july 2009. (Cited on page 128.)

[Jeub *et al.* 2010] M. Jeub, M. Schäfer, H. Krüger, C. Nelke, C. Beaugeant and P. Vary. *Do We Need Dereverberation for Hand-Held Telephony?* In Proceedings of 20th International Congress on Acoustics, ICA 2010, Aug 2010. (Cited on page 128.)

[Kajikawa 2011] Y. Kajikawa. *Linearization ability evaluation of nonlinear filters employing dynamic distortion measurement.* In Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on, pages 409 – 412, may 2011. (Cited on page 82.)

[Kuech & Kellermann] Fabian Kuech and Walter Kellermann. *Proportionate NLMS Algorithm for Second-Order Volterra Filters and its Application to Nonlinear Echo Cancellation.* (Cited on pages 42 and 97.)

[Kuech & Kellermann 2002] F. Kuech and W. Kellermann. *Nonlinear line echo cancellation using a simplified second order Volterra filter.* In Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on, volume 2, pages II–1117 –II–1120, may 2002. (Cited on pages 41 and 97.)

[Kuech & Kellermann 2004] F. Kuech and W. Kellermann. *A novel multidelay adaptive algorithm for Volterra filters in diagonal coordinate representation [nonlinear acoustic echo cancellation example].* In Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on, volume 2, pages ii – 869–72 vol.2, may 2004. (Cited on page 96.)

[Kuech & Kellermann 2005] F. Kuech and W. Kellermann. *Partitioned block frequency-domain adaptive second-order Volterra filter*. Signal Processing, IEEE Transactions on, vol. 53, no. 2, pages 564 – 575, feb 2005. (Cited on page 43.)

[Kuech & Kellermann 2006] Fabian Kuech and Walter Kellermann. *Orthogonalized power filters for nonlinear acoustic echo cancellation*. Signal Processing, vol. 86, no. 6, pages 1168 – 1181, 2006. <ce:title>Applied Speech and Audio Processing</ce:title>. (Cited on pages 50, 51, 104, 105, 106, 108 and 120.)

[Kuech & Kellermann 2007] F. Kuech and W. Kellermann. *Nonlinear Residual Echo Suppression using a Power Filter Model of the Acoustic Echo Path*. In Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on, volume 1, pages I–73 –I–76, april 2007. (Cited on pages 48 and 63.)

[Kuech *et al.* 2005] F. Kuech, A. Mitnacht and W. Kellermann. *Nonlinear acoustic echo cancellation using adaptive orthogonalized power filters*. In Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on, volume 3, pages iii/105 – iii/108 Vol. 3, march 2005. (Cited on pages 42 and 48.)

[Kuech *et al.* 2006] F. Kuech, M. Zeller and W. Kellermann. *Input Signal Decorrelation Applied to Adaptive Second-Order Volterra Filters in the Time Domain*. In Digital Signal Processing Workshop, 12th - Signal Processing Education Workshop, 4th, pages 348 –353, sept. 2006. (Cited on page 42.)

[Lashkari 2005] K. Lashkari. *A Modified Volterra-Wiener-Hammerstein Model for Loudspeaker Precompensation*. In Signals, Systems and Computers, 2005. Conference Record of the Thirty-Ninth Asilomar Conference on, pages 344 –348, 28 2005-nov. 1 2005. (Cited on page 118.)

[Lee & Mathews 1993] J. Lee and V.J. Mathews. *A fast recursive least squares adaptive second order Volterra filter and its performance analysis*. Signal Processing, IEEE Transactions on, vol. 41, no. 3, pages 1087 –1102, mar 1993. (Cited on page 42.)

[Loganathan *et al.* 2011] P. Loganathan, E.A.P. Habets and P.A. Naylor. *A proportionate adaptive algorithm with variable partitioned block length for acoustic echo cancellation*. In Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on, pages 73 –76, may 2011. (Cited on page 29.)

[Mäkelä & Niemistö 2003] Tuomo Mäkelä and Riitta Niemistö. *Effects of harmonic components generated by polynomial pre-filter in acoustic echo controls*. In Proc. the 8th International Conference on Acoustic Echo and Noise Control, IWAENC, Sept 2003. (Cited on page 47.)

[Malik & Enzner 2011] S. Malik and G. Enzner. *Fourier expansion of hammerstein models for nonlinear acoustic system identification*. In Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on, pages 85 –88, may 2011. (Cited on page 42.)

[Mansour & Gray 1981] D. Mansour and Jr. Gray A. *Frequency domain non-linear adaptive filter*. In Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '81., volume 6, pages 550 – 553, apr 1981. (Cited on page 42.)

[Mathews & Lee 1988] V.J. Mathews and J. Lee. *A fast recursive least-squares second order Volterra filter*. In Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on, pages 1383 –1386 vol.3, apr 1988. (Cited on page 42.)

[Mathews 1991] V.J. Mathews. *Adaptive polynomial filters*. Signal Processing Magazine, IEEE, vol. 8, no. 3, pages 10 –26, jul 1991. (Cited on page 42.)

[Mathews 1995a] V.J. Mathews. *Adaptive Volterra filters using orthogonal structures*. In Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on, volume 2, pages 957 –960 vol.2, may 1995. (Cited on page 42.)

[Mathews 1995b] V.J. Mathews. *Orthogonalization of correlated Gaussian signals for Volterra system identification*. Signal Processing Letters, IEEE, vol. 2, no. 10, pages 188 –190, oct 1995. (Cited on page 42.)

[Mboup *et al.* 1992] M. Mboup, M. Bonnet and N. Bershad. *Coupled adaptive prediction and system identification: a statistical model and transient analysis*. In Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on, volume 4, pages 1 –4 vol.4, mar 1992. (Cited on page 28.)

[Mboup *et al.* 1994] M. Mboup, M. Bonnet and N. Bershad. *LMS coupled adaptive prediction and system identification: a statistical model and transient mean analysis*. Signal Processing, IEEE Transactions on, vol. 42, no. 10, pages 2607 –2615, oct 1994. (Cited on page 28.)

[Mossi *et al.* 2010a] M.I. Mossi, N.W.D. Evans and C. Beaugeant. *An assessment of linear adaptive filter performance with nonlinear distortions*. In Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on, pages 313 –316, march 2010. (Cited on pages 7, 49, 52 and 82.)

[Mossi *et al.* 2010b] M.I. Mossi, C. Yemdji, N. Evans and C. Beaugeant. *A comparative assessment of noise and non-linear echo effects in acoustic echo cancellation*. In Signal Processing (ICSP), 2010 IEEE 10th International Conference on, pages 223 –226, oct. 2010. (Cited on pages 7 and 49.)

[Mossi *et al.* 2010c] Moctar Mossi, Christelle Yemdji, Nicholas W D Evans and Christophe Beaugeant. *Acoustic echo cancellation in non-linear and noisy environment.* In Research report RR-10-240, 08 2010. (Cited on pages 7, 70 and 71.)

[Mossi *et al.* 2010d] Moctar I Mossi, Christelle Yemdji, Nicholas W D Evans, C Hergoltz, Christophe Beaugeant and P Degry. *New models for characterizing mobile terminal loudspeaker distortions.* In IWAENC 2010, International Workshop on Acoustic Echo and Noise Control, August 30-September 2nd, 2010, Tel Aviv, Israel, 08 2010. (Cited on pages 8, 73 and 106.)

[Mossi *et al.* 2011a] M.I. Mossi, C. Yemdji, N. Evans, C. Beaugeant and P. Degry. *Robust and low-cost cascaded non-linear acoustic echo cancellation.* In Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on, pages 89 –92, may 2011. (Cited on pages 8 and 93.)

[Mossi *et al.* 2011b] M.I. Mossi, Christelle Yemdji, Nicholas Evans and Christophe Beaugeant. *Non-linear acoustic echo cancellation using online loudspeaker linearization.* In Applications of Signal Processing to Audio and Acoustics (WASPAA), 2011 IEEE Workshop on, pages 97 –100, oct. 2011. (Cited on pages 9, 93, 110, 112, 113 and 114.)

[Mossi *et al.* 2012] Moctar Mossi, Christelle Yemdji, Nicholas W D Evans, Christophe Beaugeant, Fabrice M Plante and Fatimazahra Marfouq. *Dual amplifier and loudspeaker compensation using fast convergent and cascaded approaches to non-linear acoustic echo cancellation.* In ICASSP 2012, 37th International Conference on Acoustics, Speech and Signal Processing, March 25-30, 2012, Kyoto, Japan, Kyoto, JAPAN, 03 2012. (Cited on pages 8 and 93.)

[Niemistö & Mäkelä 2003a] R. Niemistö and T. Mäkelä. *Effects of harmonic components generated by polynomial pre-filter in acoustic echo control.* In Acoustic Echo and Noise Control, 2003 IWAENC-03, Workshop Proceedings., 2003 International Workshop on, Sept 2003. (Cited on page 116.)

[Niemistö & Mäkelä 2003b] R. Niemistö and T. Mäkelä. *On Performance of Linear Adaptive Filtering Algorithms in Acoustic Echo Control in Presence of Distorting Loudspeakers.* In Acoustic Echo and Noise Control, 2003 IWAENC-03, Workshop Proceedings., 2003 International Workshop on, Sept 2003. (Cited on page 69.)

[Nollett & Jones 1997] B. S. Nollett and D. L. Jones. *Nonlinear Echo Cancellation for Hands-Free Speakerphones.* In Nonlinear Signal and Image Processing, 1997 NSIP-97. Workshop Proceedings., 1997 IEEE-EURASIP International Workshop on, 1997. (Cited on pages 45, 46, 104, 108, 110 and 114.)

[Paleologu *et al.* 2010] C. Paleologu, J. Benesty and S. Ciochina. *Sparse Adaptive Filters for Echo Cancellation.* Synthesis Lectures on Speech and Audio Processing, vol. 6, no. 1, pages 1–124, 2010. (Cited on page 29.)

[Pillonnet *et al.* 2008] G. Pillonnet, R. Cellier, E. Allier, N. Abouchi and A. Nagari. *A topological comparison of PWM and hysteresis controls in switching audio amplifiers.* In Circuits and Systems, 2008. APCCAS 2008. IEEE Asia Pacific Conference on, pages 668 –671, 30 2008-dec. 3 2008. (Cited on page 73.)

[Quaegebeur 2007] N. Quaegebeur. *Vibrations non linéaires et rayonnement acoustique de structures minces de type haut-parleur.* PhD thesis, ENSTA - Ecole Doctorale de l'Ecole Polytechnique, October 2007. (Cited on pages 45, 79 and 80.)

[Ravaud *et al.* 2009] R. Ravaud, G. Lemarquand and T. Roussel. *Experimental measurement of the nonlinearities of electrodynamic microphones.* Applied Acoustics, vol. 70, no. 9, pages 1194 – 1199, 2009. (Cited on page 74.)

[Reed & Hawksford 2000] M.J. Reed and M.O. Hawksford. *Efficient implementation of the Volterra filter.* Vision, Image and Signal Processing, IEE Proceedings -, vol. 147, no. 2, pages 109 –114, apr 2000. (Cited on page 42.)

[ROHDES&SCHWARTZ 2008] ROHDES&SCHWARTZ. *R&S CMU200 Universal Radio Communication Tester, Specifications.* Data Sheet version 09.00, Oct 2008. (Cited on page 75.)

[Rupp 1993] M. Rupp. *A comparison of gradient-based algorithms for echo compensation with decorrelating properties.* In Applications of Signal Processing to Audio and Acoustics, 1993. Final Program and Paper Summaries., 1993 IEEE Workshop on, pages 12 –15, oct 1993. (Cited on page 29.)

[Sayed 2008] A.H. Sayed. Adaptive filters. Wiley-Interscience, 2008. (Cited on page 23.)

[Schetzen 2006] M. Schetzen. The volterra and wiener theories of nonlinear systems. A Wiley - Interscience publication. KRIEGER PUBLISHING COMPAGNY, 2006. (Cited on page 94.)

[Schurer 1997] Hans Schurer. *Linearization of Electroacoustic Transducers.* PhD thesis, Universiteit Twente, Enschede, November 1997. (Cited on pages 45 and 80.)

[Shi *et al.* 2007] Kun Shi, G.T. Zhou and M. Viberg. *Compensation for Nonlinearity in a Hammerstein System Using the Coherence Function With Application to Nonlinear Acoustic Echo Cancellation.* Signal Processing, IEEE Transactions on, vol. 55, no. 12, pages 5853 –5858, dec. 2007. (Cited on page 46.)

[Shi *et al.* 2008a] Kun Shi, Xiaoli Ma and G.T. Zhou. *Acoustic Echo Cancellation Using a Pseudocoherence Function in the Presence of Memoryless Nonlinearity.* Circuits and Systems I: Regular Papers, IEEE Transactions on, vol. 55, no. 9, pages 2639 –2649, oct. 2008. (Cited on page 46.)

[Shi *et al.* 2008b] Kun Shi, Xiaoli Ma and G.T. Zhou. *Adaptive Acoustic Echo Cancellation in the Presence of Multiple Nonlinearities.* In Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on, pages 3601 –3604, 31 2008-april 4 2008. (Cited on page 46.)

[Shi *et al.* 2008c] Kun Shi, Xiaoli Ma and G.T. Zhou. *A Residual Echo Suppression Technique for Systems with Nonlinear Acoustic Echo Paths.* In Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on, pages 257 –260, 31 2008-april 4 2008. (Cited on page 48.)

[Shi *et al.* 2009] Kun Shi, Xiaoli Ma and G. Tong Zhou. *An efficient acoustic echo cancellation design for systems with long room impulses and nonlinear loudspeakers.* Signal Processing, vol. 89, no. 2, pages 121 – 132, 2009. (Cited on page 46.)

[Slock & Kailath 1988] D.T.M. Slock and T. Kailath. *Numerically stable fast recursive least-squares transversal filters.* In Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on, pages 1365 –1368 vol.3, apr 1988. (Cited on page 26.)

[Stenger & Kellermann 2000] Alexander Stenger and Walter Kellermann. *Adaptation of a memoryless preprocessor for nonlinear acoustic echo cancelling.* Signal Processing, vol. 80, no. 9, pages 1747 – 1760, 2000. (Cited on pages 45, 46, 50, 51, 104, 105, 108, 110, 112 and 114.)

[Stenger & Rabenstein 1998] A Stenger and Rudolf Rabenstein. *An Acoustic Echo Canceller with Compensation of Nonlinearities.* In Proc. EUSIPCO 98, Isle of Rhodes, pages 969–972, 1998. (Cited on pages 41 and 93.)

[Stenger *et al.* 1999a] A Stenger, Walter Kellermann and Rudolf Rabenstein. *Adaptation of Acoustic Echo Cancellers Incorporating a Memoryless Nonlinearity.* In Proc. IEEE Workshop on Acoustic Echo and Noise Control (IWAENC'99, pages 168–171, 1999. (Cited on page 46.)

[Stenger *et al.* 1999b] A. Stenger, L. Trautmann and R. Rabenstein. *Nonlinear acoustic echo cancellation with 2nd order adaptive Volterra filters.* In Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on, volume 2, pages 877 –880 vol.2, mar 1999. (Cited on pages 41, 96 and 103.)

[Tanaka *et al.* 1999] M. Tanaka, S. Makino and J. Kojima. *A block exact fast affine projection algorithm.* Speech and Audio Processing, IEEE Transactions on, vol. 7, no. 1, pages 79 –86, jan 1999. (Cited on page 25.)

[Thomas 1971] E.J. Thomas. *Some Consideration on the Application of the Volterra Representation of Nonlinear Networks to Adaptive Echo Candellers.* The Bell System Technical Journal, vol. 50, no. 8, pages 2797 –2805, oct 1971. (Cited on page 41.)

[Van de Kerkhof & Kitzen 1992] L.M. Van de Kerkhof and W.J.W. Kitzen. *Tracking of a time-varying acoustic impulse response by an adaptive filter.* Signal Processing, IEEE Transactions on, vol. 40, no. 6, pages 1285 –1294, jun 1992. (Cited on page 21.)

[Vary & Martin 2006] P. Vary and R. Martin. Digital speech transmission: enhancement, coding and error concealment. John Wiley, 2006. (Cited on pages 3, 20, 22, 23, 52, 54 and 58.)

[Vondrasek & Pollak 2005] M. Vondrasek and P. Pollak. *Methods for Speech SNR Estimation: Evaluation Tool and Analysis of VAD Dependency.* Radioengineering, vol. 14, no. 1, pages 11–25, Apr 2005. (Cited on page 51.)

[Wada & Juang 2012] T.S. Wada and Biing-Hwang Juang. *Enhancement of Residual Echo for Robust Acoustic Echo Cancellation.* Audio, Speech, and Language Processing, IEEE Transactions on, vol. 20, no. 1, pages 175 –189, jan. 2012. (Cited on page 48.)

[Widrow 1966] B. Widrow. *Adaptive Filters 1 : Fundamentals.* Technical report 6764-6, Stanford Electronics Laboratories, Stanford University, December 1966. (Cited on page 20.)

[Widrow 1971] B. Widrow. Aspect of Network and System Theory, chapter Adaptive Filters. Holt, Rinehart and Winston, Inc, 1971. (Cited on page 112.)

[Yamada *et al.* 2002] I. Yamada, K. Slavakis and K. Yamada. *An efficient robust adaptive filtering algorithm based on parallel subgradient projection techniques.* Signal Processing, IEEE Transactions on, vol. 50, no. 5, pages 1091 –1101, may 2002. (Cited on page 25.)

[Yasukana *et al.* 1988] H. Yasukana, I. Furukawa and Y. Ishiyama. *Characteristics of acoustic echo cancellers using sub-band sampling and decorrelation methods.* Electronics Letters, vol. 24, no. 16, pages 1039 –1040, aug 1988. (Cited on page 28.)

[Yemdji *et al.* 2010] Christelle Yemdji, Moctar Mossi, Nicholas W D Evans and Christophe Beaugeant. *Efficient low delay filtering for residual echo suppression.* In EUPSICO 2010, 18th European Signal Processing Conference, August 23-27, 2010, Aalborg, Denmark, Aalborg, DENMARK, 08 2010. (Cited on page 71.)

[Zeller & Kellermann 2007] M. Zeller and W. Kellermann. *Iterated Coefficient Updates of Partitioned Block Frequency Domain Second-Order Volterra Filters for Nonlinear AEC*. In Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on, volume 3, pages III–1425 –III–1428, april 2007. (Cited on page 43.)

[Zeller & Kellermann 2008] M. Zeller and W. Kellermann. *Framewise Repeated Coefficient Updates for Enhanced Nonlinear AEC by Diagonal Coordinate Volterra Filters*. In Hands-Free Speech Communication and Microphone Arrays, 2008. HSCMA 2008, pages 196 –199, may 2008. (Cited on page 43.)

[Zeller & Kellermann 2010a] M. Zeller and W. Kellermann. *Advances in identification and compensation of nonlinear systems by adaptive volterra models*. In Signals, Systems and Computers (ASILOMAR), 2010 Conference Record of the Forty Fourth Asilomar Conference on, pages 1940 –1944, nov. 2010. (Cited on pages 43 and 96.)

[Zeller & Kellermann 2010b] M. Zeller and W. Kellermann. *Multirate Implementation of Aliasing-free Adaptive Volterra Filters by Interpolation of Higher-Order Kernel Inputs*. In ITG-Fachtagung Sprachkommunikation, Bochum, Germany, October 2010. (Cited on page 47.)

[Zeller *et al.* 2009] M. Zeller, L.A. Azpicueta-Ruiz and W. Kellermann. *Online estimation of the optimum quadratic kernel size of second-order Volterra filters using a convex combination scheme*. In Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on, pages 2965 –2968, april 2009. (Cited on page 43.)

[Zeller *et al.* 2010] M. Zeller, L.A. Azpicueta-Ruiz, J. Arenas-Garcia and W. Kellermann. *Efficient adaptive DFT-domain Volterra filters using an automatically controlled number of quadratic kernel diagonals*. In Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on, pages 4062 –4065, march 2010. (Cited on page 43.)

[Zeller *et al.* 2011] M. Zeller, L.A. Azpicueta-Ruiz, J. Arenas-Garcia and W. Kellermann. *Adaptive Volterra Filters With Evolutionary Quadratic Kernels Using a Combination Scheme for Memory Control*. Signal Processing, IEEE Transactions on, vol. 59, no. 4, pages 1449 –1464, april 2011. (Cited on page 43.)

[Zhou *et al.* 2006] Dayong Zhou, V. DeBrunner, Yan Zhai and M. Yeary. *Efficient Adaptive Nonlinear Echo Cancellation, Using Sub-band Implementation of the Adaptive Volterra Filter*. In Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on, volume 5, page V, may 2006. (Cited on page 43.)