UNIVERSITE BLIDA 1

Faculté de Technologie

Département d'Électronique

THESE DE DOCTORAT

en Electronique

Acoustic Echo Cancellation with Double-Talk Detection and Acoustic Channel Variation

Par

Islam HASSANI

Soutenu le 04 juillet 2021, devant le jury composé de :

A.GUESSOUM	Professeur, U.Blida 1	Président
M.DJENDI	Professeur, U.Blida 1	Examinateur
A.DAAMOUCHE	Professeur, IGEE (U.Boumerdes)	Examinateur
R.BENDOUMIA	M.C.A, U.Blida 1	Examinateur
A.BENALLAL	Professeur, U.Blida 1	Rapporteur
M.AREZKI	M.C.B, U.Blida 1	Co-Rapporteur

Blida, 2021

Acknowledgement

I would like to express my great respect, warm feelings and thanks to my supervisor, ***Prof. Ahmed BENALLAL****, for providing the opportunity to research this interesting topic, for his sympathy and valuable advice and the direction he had shown throughout the course of this project. Also, I would like to express my sincere gratitude to my co-supervisor,* ***Dr. Madjid AREZKI*** *for his inspirations, encouragement and guidance during this project.*

Thanks also go to the jury: ***Prof. Abderrezak GUESSOUM*** *as a president;* ***Prof. Mohamed DJENDI****;* ***Prof. Abdelhamid DAAMOUCHE*** *and* ***Dr. Rédha BENDOUMIA*** *as examiners who accepted to evaluate this work.*

Many thanks to ***Dr. Mahfoud HAMIDIA*** *and* ***Dr. Mohamed El Amine SLIMANI*** *from USTHB-Algiers for their continuous helps and advices.*

Dedication

To my parents, especially my father for his support and encouragement,

To my brothers "Med Nadjib, Med Tayeb and Yasser"

To my wife

To my best friends

To those who will be happy for me

ملخص

في هذه الأطروحة، الهدف الرئيسي هو التحديد التكيفي لاستجابات النبضات الصوتية الطويلة، في سياق إلغاء الصدى الصوتي. من خلال خوارزميات سريعة وأقل تعقيدا وفي حالة القناة الصوتية المتغيرة وسيناريو التحدث المزدوج. تم تقديم اسهامين رئيسيين، الأول هي اقتراح خوارزميتين على أساس تحديد عضوية المجموعة، والغرض من ذلك تحسين سرعة التقارب وقدرة التتبع والفعالية ضد الضوضاء الإضافية. الاسهام الثاني هي دمج كاشف الحديث المزدوج على أساس الارتباط المتبادل في الخوارزميتين المقترحتين لمعالجة مشكل الصدى الصوتي في حالة وجود الحديث المزدوج. توضح النتائج التي نم الحصول عليها أداء أفضل مع أقل تعقيد حسابي في حالة الحديث الفردي، وأيضا في حالة التحدث المزدوج، تقدم الخوارزميات المقترحة أداء جيدا باستخدام كاشف الحديث المزدوج مقارنة بالخوارزميات الموجودة.

Abstract

In this thesis, the main goal is the adaptive identification of the long acoustic impulse responses, in the context of the acoustic echo cancellation (AEC), by fast and low complexity algorithms and in the case of a variable acoustic channel and double-talk scenario. Two major contributions were introduced; the first one is to propose two algorithms based on the set-membership identification (SMI), the purpose is to improve the convergence speed and the tracking capability with robustness against additive noise. The second contribution is the integration of double-talk detector (DTD) based on normalized cross-correlation (NCC) in the two proposed algorithms to tackle the problem of the the acoustic echo in the case of the presence of double-talk. The obtained results demonstrate better performances with lower computational complexity in the single-talk situation, also in the double-talk situation the proposed algorithms present good performances using DTD compared to the existing algorithms.

Résumé

Dans cette thèse, le but principal est l'identification adaptative des réponses impulsionnelles acoustiques longues, dans le cadre de l'annulation d'écho acoustique, par des algorithmes rapides de complexité réduite et dans le cas d'un canal acoustique variable et d'un scénario de double parole. Deux contributions majeures ont été introduites ; la première est de proposer deux algorithmes basés sur l'identification d'appartenance à l'ensemble (SMI), le but est d'améliorer la vitesse de convergence et la capacité de poursuite avec robustesse au bruit additif. La deuxième contribution est l'intégration du détecteur de double parole (DTD) basé sur l'intercorrélation normalisée (NCC) dans les deux algorithmes proposés pour traiter le problème de l'écho acoustique dans le cas de la présence de la double parole. Les résultats obtenus démontrent de meilleures performances avec une complexité de calcul réduite en situation de parole unique, et également en situation de double parole les algorithmes proposés présentent de bonnes performances en utilisant le DTD en comparant avec des algorithmes existants.

Table of contents

Acknowledgement……………………………………………………………………………i
Dedication……………………………………………………………………………………..ii
Abstract……………………………………………………………………………………… iii
Table of contents……………………………………..………………………………… iv
List of figures……………………………………………………..……………………......vii
List of tables………………………..…………………………………………….……… x
List of abbreviations…………………………………………………..……….……. xi
List of symbols…………………………………………..………………………………. xiii

General introduction……………………………………………………………………….1
Chapter 1: Acoustic Echo and adaptive filtering……………………………………...…4
1.1 Introduction…………………………………………………………………………………4
1.2 Acoustic Impulse Response (IR) of a room………………………………...…………4
1.3 Acoustic Echo Cancellation system…………………………...………………………….5
1.4 Adaptive filtering ……………………………………………………………………………..7
1.4.1. Basic principle and applications of the adaptive filter……………………….…………..7
1.4.1.a System identification…………………………………………………………………8
1.4.1.b Prediction…………………………………………………………………………….8
1.4.1.c. Inverse modeling………………………………………………………………………9
1.4.1.d Interference cancellation……………………………………………………………10
1.4.2 Adaptation process of an adaptive filter………………………………………………..10
1.4.3 The criteria of the choice of the adaptive filtering algorithms…………………………12
1.5 The Wiener filtering…………………………………………………………….………...12
1.6 The principle of deterministic gradient algorithm……….………………………..……….14
1.7 The stochastic gradient algorithm LMS……..……………………………………….………15
1.8 The Normalized LMS (NLMS) algorithm…………………………………………….…….16
1.9 The Recursive Least Square (RLS) algorithm……………………………………….…...17
1.10 Fast Recursive Least Square (FRLS) algorithms…………………………………….…...19

1.11 The Fast NLMS algorithm……………………………………………………………….…...26

1.12 Conclusion……………………………………………………………………….….…...30

Chapter 2: Set-Membership adaptive filtering algorithms…………………………………31

2.1 Introduction……………………………………………………………………………….31

2.2 Set-Membership NLMS (SM-NLMS) algorithm……………………………………....…...31

2.3 The SM-NLMS with Adaptive Error Bound (SMAEB-NLMS) algorithm………………...34

2.4 The SM-NLMS with Robust Error Bound (SMREB-NLMS) algorithm…………...….........36

2.5. The proposed SM Fast NLMS (SM-FNLMS) algorithm…………..…...............................37

2.5.1 Performance evaluation of the proposed SM-FNLMS algorithm…………..…..........39

2.5.2 The effect of the error bound on the SM-FNLMS algorithm…………..…..................42

2.6 The proposed Improved SM-FNLMS (ISM-FNLMS) algorithm…………..….....................43

2.6.1 The derivation of the proposed ISM-FNLMS algorithm…………..…........................43

2.6.2 Comparative performances for stationary input signal…………..….........................46

2.6.3 Comparative performances with noisy signals…………..…......................................49

2.6.4 Comparative performances for non-stationary systems…………..…..........................51

2.6.5 Comparative performances with speech inputs…………..….......................................54

2.6.6 Computational complexity evaluation…………..…...55

2.7 Conclusion…………..…..58

Chapter 3 Acoustic Echo Cancellation with Double-Talk Detection…………..….............59

3.1 Introduction…………..…..59

3.2 Double-Talk Detection…………..…...59

3.2.1 General structure of DTD…………..…...59

3.2.2 The Geigel algorithm…………..…...60

3.2.3 The cross-correlation algorithm…………..…..61

3.2.4 The Normalized Cross-Correlation (NCC) algorithm…………..…..............................62

3.2.5 DTD based on the signal envelope…………..…..63

3.2.6 Fast DTD based on signal envelope…………..…..65

3.2.7 DTD based on the Holder inequality…………..…..66

3.2.8 DTD using the singular value decomposition…………..….......................................69

3.2.9 DTD based on joint energy and cross-correlation estimation…………..…................70

3.3 The AEC with DTD…………..…..71

3.4 Experiments on AEC during DT…………..…..74

3.5 Conclusion…………..…..78

General conclusion and perspectives…………..…...79

Bibliography…………..…..81

Author contribution…...…..86

List of figures

Figure 1.1 Sound reflections in a room ………………………………………………..……….5

Figure 1.2 Basic AEC setup in a teleconferencing system...……………………………..…….6

Figure 1.3 Principle of an adaptive filter..…………………………………………..………….7

Figure 1.4 The bloc diagram of system identification.………………………………...…….8

Figure 1.5 The bloc diagram of direct error prediction …..……………………………...…….9

Figure 1.6 The bloc diagram of inverse modeling…….…………………………………………9

Figure 1.7 The bloc diagram of interference cancellation …….……………………...……...10

Figure 1.8 Wiener filtering scheme ………………………………………………...…...13

Figure 2.1 Coefficient vector updating for the SM-NLMS algorithm…………………………...32

Figure 2.2 The used normalized input signals. (a) WGN AR-20 signal with a length of 256000 samples, (b) USASI noise signal with a length of 131072 samples, (c) Speech signal with a length of 108208 samples……………………………………………………………………….39

Figure 2.3 The used car impulse response of $L = 1024$…………………………......................40

Figure 2.4 The Comparison of the initial convergence and the steady-state NMSE. Input: WGN-AR(20) signal with Car system of length $L = 1024$. No output noise…………………………41

Figure 2.5 The Comparison of the initial convergence and the steady-state NMSE. Input: USASI signal with Car system of length $L = 1024$. No output noise …………………………………41

Figure 2.6 The Comparison of the initial convergence and the steady-state MSE. Input: USASI signal with Car system of length $L = 1024$. SNR=50 dB ……………….……….……………42

Figure 2.7 Learning curves of the SM-FNLMS, error bound ζ effect ………………….............43

Figure 2.8 The used real acoustic impulse responses ……………………...........................46

Figure 2.9 Comparison of the initial convergence and the steady-state NMSE. Input: WGN-AR(20) signal with Car system of length $L = 256$. No output noise ………………................47

Figure 2.10 Comparison of the initial convergence and the steady-state NMSE. Input: WGN-AR(20) signal with VCN system of length $L = 1024$. No output noise ……………..……......48

Figure 2.11 Comparison of the initial convergence and the steady-state NMSE. Input: USASI signal with Car system of length $L = 256$. No output noise …………………..49

Figure 2.12 Comparison with noisy input: USASI signal with car system of length $L = 256$. Output noise: SNR=15 dB. Threshold $\zeta = 0.1$. $\sigma_y^2 = 0.38$…………………..50

Figure 2.13 Comparison with noisy input: USASI signal with car system of length $L = 256$. Output noise: SNR=30 dB. Threshold $\zeta = 0.1$. $\sigma_y^2 = 0.38$…………………..50

Figure 2.14 Comparison with noisy input: USASI signal with car system of length $L = 256$. Output noise: SNR=50 dB. Threshold $\zeta = 0.0032$. $\sigma_y^2 = 0.38$……………………......................51

Figure 2.15 Comparison of the initial convergence and the re-convergence after an abrupt change in the impulse response of car system of length $L = 256$. Input: USASI noise. No output noise.…………………...52

Figure 2.16 Comparison of the initial convergence and the re-convergence after an abrupt change in the impulse response of VCN system of length $L = 1024$. Input: USASI noise. No output noise.…………………...52

Figure 2.17 Variable artificial gain used for the tracking experiment. The variation is made between $n = 51200$ and $n = 71680$…………………………………...53

Figure 2.18 . Comparison of tracking time-varying system of length $L = 256$. Input: USASI noise. No output noise …………………………………..53

Figure 2.19 . Comparison of tracking time-varying system of length $L = 1024$. Input: USASI noise. No output noise …………………………………..54

Figure 2.20 . MSE evaluation Input: Speech signal with car system of length $L = 256$. Output noise: SNR=50dB …………………………………...55

Figure 2.21 . MSE evaluation Input: Speech signal with VCN system of length $L = 1024$. Output noise: SNR=50dB …………………………………...55

Figure 2.22 . Comparison of SM algorithms with car system of length $L = 256$.Input USASI noise. Output noise: SNR=30dB. Threshold $\zeta = 0.025$………...56

Figure 2.23 . Comparison of SM algorithms with VCN system of length $L = 1024$.Input USASI noise. Output noise: SNR=30dB. Threshold $\zeta = 0.025$………..57

Figure 3.1 . General structure of the AEC with DTD …………………………………..……….59

Figure 3.2 . Basic structure of the AEC with DTD based on NCC ……………………….…….62

Figure 3.3 . Bloc diagram of the AEC with DTD based on the signal envelope [43] …….…….64

Figure 3.4 . The flowchart of the DTD based on joint signal energy and cross-correlation estimation [56]………………………………………………………………………………………… 71

Figure 3.5 . The used signals, $x(n)$: Far-end signal (in Blue), $s(n)$:Near-end signal DT (in red)……………………………………………………………………………………………….75

Figure 3.6 . NM of ISM-FNLMS, FNLMS and NLMS with DTD in car system of $L = 256$……………………………………………………………………………………………….76

Figure 3.7 . Output waveforms of ISM-FNLMS, FNLMS and NLMS with DTD in car system of $L = 256$……………………………………………………………………………………….76

Figure 3.8 . NM of ISM-FNLMS, FNLMS and NLMS with DTD in VCN system of $L = 1024$………………………………………………………………………………..…………….77

Figure 3.9 . Output waveforms of ISM-FNLMS, FNLMS and NLMS with DTD in VCN system of $L = 1024$…………………………………………………………………….…………….77

List of tables

Table 1.1 The LMS algorithm ……………………………………………………………….15

Table 1.2 The RLS algorithm with complexity of $O(L^2)$…………...………………………….19

Table 1.3 The FRLS algorithm with complexity of $(7L)$…………...………………………….25

Table 1.4 The FNLMS algorithm with complexity of $(2L)$…………...……………………….29

Table 2.1 The SM-NLMS algorithm …………...…………………………………..……….33

Table 2.2 The SMAEB-NLMS algorithm …………...……………………………..……….35

Table 2.3 The SMREB-NLMS algorithm …………...……………………………..……….37

Table 2.4 The proposed SM-FNLMS algorithm …………...………………..……..……….38

Table 2.5 The proposed ISM-FNLMS algorithm …………...………………..……..……….45

Table 2.6 Computational complexity evaluation USASI Noise, CAR AIR, N=256, ($\boldsymbol{\sigma_y^2 = 0.38}$, with no noise)…………...………………..……..……………………………………..…….57

Table 2.7 Computational complexity evaluation USASI Noise, VCN AIR, N=1024, ($\boldsymbol{\sigma_y^2 = 0.43}$, with no noise)………………………………………………………………..………..…….58

Table 3.1 FNLMS algorithm with DTD based on NCC …………...…………………..……….71

Table 3.2 The ISM-FNLMS algorithm with DTD based on NCC …………...…………..…….73

List of abbreviations

AEC	**A**coustic **E**cho **C**ancellation
AIR	**A**coustic **I**mpulse **R**esponse
ANSI	**A**merican **N**ational **S**tandards **I**nstitute
AP	**A**ffine **P**rojection
CC	**C**ross **C**orrelation
DT	**D**ouble **T**alk
DTD	**D**ouble **T**alk **D**etector
FAEST	**F**ast **A** posteriori **E**rror **S**equential **T**echnique
FIR	**F**inite **I**mpulse **R**esponse
FK	**F**ast **K**alman
FNLMS	**F**ast **N**ormalized **L**east **M**ean **S**quare
FPGA	**F**ield **P**rogrammable **G**ate **A**rray
FRLS	**F**ast **R**ecursive **L**east **S**quare
FTF	**F**ast **T**ransversal **F**ilter
GMM	**G**aussian **M**ixture **M**odel
IR	**I**mpulse **R**esponse
ISM-FNLMS	**I**mproved **S**et **M**embership **FNLMS**
LMS	**L**east **M**ean **S**quare
LS	**L**east **S**quare
MAE	**M**ean **A**bsolute **E**rror
MECC	**M**icrophone **E**rror **C**ross **C**orrelation
MSE	**M**ean **S**quare **E**rror
NCC	**N**ormalized **C**ross **C**orrelation
NLMS	**N**ormalized **LMS**

NM	**N**ormalized **M**isalignment
NMSE	**N**ormalized **MSE**
PU	**P**artial **U**pdate
REB	**R**obust **E**rror **B**ound
RLS	**R**ecursive **L**east **S**quare
SM	**S**et **M**embership
SMAEB-NLMS	**S**et **M**embership **A**daptive **E**rror **B**ound **NLMS**
SMF	**S**et **M**embership **F**iltering
SM-FNLMS	**S**et **M**embership **FNLMS**
SMFTF	**S**implified **M**odified **FTF**
SMI	**S**et **M**embership **I**dentification
SM-NLMS	**S**et **M**embership **NLMS**
SMREB-NLMS	**S**et **M**embership **R**obust **E**rror **B**ound **NLMS**
SNR	**S**ignal to **N**oise **R**atio
SS-MSE	**S**teady **S**tate **MSE**
ST-MSE	**A**verage **St**eady State **MSE**
SVD	**S**ingular **V**alue **D**ecomposition
UP	**U**pdate **P**robability
USASI	**U**nited **S**tates of **A**merica **S**tandards **I**nstitute
VCN	**V**ideo **C**onfere**n**ce
WGN-AR20	**W**hite **G**aussian **N**oise filtered by **A**uto **R**egressive model of **20**
WLS	**W**eighted **L**east **S**quare

List of symbols

In this thesis, mathematical representation of symbols is:

- Scalar quantities are represented by small letters "x".
- Vectors are represented by small letters in **bold** "**x**".
- Matrices are represented by **C**apital letters in **bold** "**X**".

$\mathbf{h}$	Unknow impulse response vector
$\mathbf{x}(n)$	Input signal vector
$y(n)$	Echo signal
$(.)^T$	Transposition of $(.)$
$(.)^{-1}$	Inverse of $(.)$
$d(n)$	Desired signal (Microphone signal)
$v(n)$	Noise signal
$\mathbf{w}(n)$	Coefficient vector of the adaptive filter
$\hat{y}(n)$	Estimated echo signal
$e(n)$	Error signal
$J[.]$	Cost function with respect to $[.]$
$\lvert . \rvert$	The absolute value
$\lVert . \rVert$	Euclidian norm
λ_i, β	Forgetting factors
$E[.]$	Mathematical expectation
$\mathbf{w}_{\mathrm{opt}}$	Optimal coefficient vector of Wiener
$\nabla_{\mathbf{w}}$	The gradient operator with respect to $\mathbf{w}(n)$
$\mathbf{R}_{L,xx}(n)$	Autocorrelation matrix of the input signal $x(n)$

$\mathbf{p}_{L,xd}(n)$	Cross-correlation vector between $x(n)$ and $d(n)$
$\frac{\partial\,(.)}{\partial \mathbf{w}(n)}$	Partial derivative with respect to $\mathbf{w}(n)$
μ_i	Step-size of the corresponding “i” algorithm
$trace(.)$	Trace of matrix
σ_i^2	Variance of the corresponding “i” signal
c_0	Regularization parameter
$O(.)$	Big O operator (Order of (.))
$\epsilon(n)$	A posteriori error
$\mathbf{c}(n)$	Kalman gain
$\tilde{\mathbf{c}}(n)$	Dual Kalman gain
$e^f(n)$	A priori forward prediction error
$\epsilon^f(n)$	A posteriori forward prediction error
$e^b(n)$	A priori backward prediction error
$\epsilon^b(n)$	A posteriori backward prediction error
$\tilde{\mathbf{C}}_{L+1}^{L+1}$	Extended Kalman gains of order $(L+1)$
$\gamma(n)$	Likelihood variable
ζ	Error bound
$diag\ (\mathbf{x})$	Diagonal matrix with **x** on its diagonal
$median(x)$	Median value of x
$< x(n) >$	Time average of $x(n)$
$\log_{10}(.)$	Logarithm base 10
T	Threshold parameter
$\xi_i(n)$	Decision variable of the corresponding “i” method

General introduction

In the last decade, there is an ever-increasing use of teleconferencing systems, that allow the meeting of a correspondent or a group of correspondents and exchange information without physical displacement. Unfortunately, a non-robust teleconferencing system suffers from some degradations of the voice intelligibility due to several phenomena such as: reverberation in the acoustic medium, presence of acoustic noise and the presence of acoustic echo. Acoustic echo can be generated in a bidirectional communication between two rooms. The transmitted signal by a room is re-transmitted to the same room because of the coupling between the speaker and the microphone in this room; so, if the transmission introduces a significant delay of the order of 30 milliseconds the people present in this room re-hear their own voices. This kind of problem is solved using a robust Acoustic Echo Cancellation (AEC) system, which should ideally remove everything that comes from the system speaker. AEC system is based on the estimation of the echo path which represents the local environment by generating the estimate of the echo signal that is extracted from the microphone. The AEC can be considered as one of system identification applications because it uses an adaptive filter in the estimation of the impulse response to be identified. The identification of the Acoustic Impulse Response (AIR) for the AEC in teleconferencing system can be ideal, if some effects are well examined such as the long length of the impulse response, fast variations in time of the echo path and non-stationarity of the input signal (speech signal).

Several adaptive filtering algorithms have been proposed. There exist two major families of adaptive algorithms: stochastic gradient algorithms and recursive least square algorithms, in the stochastic gradient algorithm we can find the Least Mean Square (LMS) and Normalized LMS (NLMS) algorithms, in the Recursive Least Square (RLS) algorithms, we have RLS and fast versions of RLS (FRLS) algorithms. All these algorithms are based on the minimization of an objective function which results to the convergence to an optimal solution. The choice of the adaptive algorithm is done according several important criteria, such as: the convergence speed, tracking capability in the case of a variable impulse response, final steady-state error, robustness in noisy environment and the computational complexity.

NLMS algorithms have a low computational complexity, but they are limited in the convergence speed, especially in the case of highly correlated input signal, contrary to RLS algorithms that have a high convergence rate with high computational complexity. A suitable solution in this case, is to use FRLS algorithms that meet the low complexity around NLMS algorithm complexity with high convergence rate as RLS algorithms. Several techniques to reduce the computational complexity have been proposed in the literature. In a real time teleconferencing system, the far-end signal and the near-end signal can be present, simultaneously in some periods, this situation is called the Double-Talk (DT) and can cause a degradation of the adaptive filter performances. A Double-Talk Detector (DTD) should be integrated on the AEC system to avoid the divergence of the adaptive filter, by controlling the adaptation of the filter coefficients.

The objective of our work is to develop efficient algorithms for AEC, in order to obtain a higher convergence speed with a lower computational complexity. As a first step, we have proposed a fast convergence algorithm with a lower computational complexity, by incorporating the Set-Membership technique which result to a better convergence rate compared to the original one. Then, as a second contribution, we did some improvements to this latter to deal with a situation of acoustic echo path variations, the novel algorithm presents good performances in terms of convergence speed, computational complexity and tracking capability. In the last part, we have studied the performances of the proposed algorithms in the DT scenario by adding a DTD method to control the filter adaptation with fast detection.

This thesis contains three chapters, organized as follows:

The first chapter introduces basic definitions of the AEC, adaptive filtering application and adaptive filtering algorithms from LMS to FRLS algorithms.

The second chapter aims to present SM based adaptive filtering algorithms, the first proposed SM-FNLMS algorithm which is derived from the FNLMS algorithm and the second proposed ISM-FNLMS algorithm which is derived from SM-FNLMS algorithm with some improvements on the SM condition.

The third chapter is dedicated to define DTD structures in the AEC application, then a state of art of different DTD methods proposed in the literature. Experiments on AEC with DTD are conducted in the last part of this chapter, to show the performances of the proposed algorithms in the case of DT.

Chapter 1:

Acoustic Echo and Adaptive filtering

1.1 Introduction

Acoustics can be defined as the scientific knowledge that studies sound including its production, transmission and effects. It is therefore not limited to the phenomena responsible for auditory sensation alone. It is distinguished from optics by the mechanical nature of sound waves rather than their electromagnetic nature.

The purpose of this first chapter is to give general notions about the acoustic echo and the techniques to eliminate it. Thus, a description of the acoustic echo and the acoustics of the rooms is first presented in a brief manner. Then the principle of an application of the acoustic echo cancellation AEC in the single-channel case is initiated. Finally, the basic adaptive algorithms, from deterministic gradient to stochastic gradient algorithms are described, such as Least Mean Square (LMS), Normalized LMS (NLMS), Recursive Least Square (RLS) and Fast RLS algorithms.

1.2. Acoustic impulse response (IR) of a room

To define the acoustic impulse response, a probe (receiver) is placed in an empty room, and a sound is launched by a medium equipped with speakers (source), the receiver picks up the direct sound and several reflections on the walls of a room.

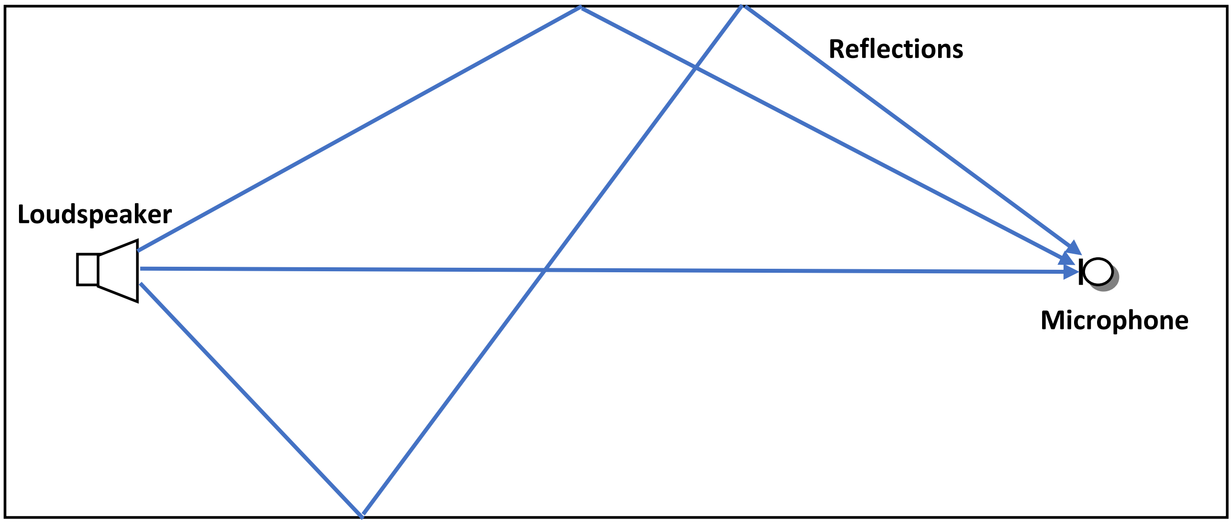

Figure 1. 1 : Sound reflections in a room.

Figure (1.1) shows the propagation of sound in a room containing a few objects, each reflection is delayed and attenuated. The microphone placed far from the speaker picks up the different sounds produced by these reflections. In linear acoustics theory [1], these reflections are modeled by a Finite Impulse Response (FIR) filter. In practice, depending on the dimensions of the room, the size of IR varies from a few coefficients to several thousand coefficients.

1.3. Acoustic Echo Cancellation system

The origin of the acoustic echo comes from the use of new so-called "hands-free" telecommunications systems. The acoustic coupling between the speaker and the microphone will generate the acoustic echo in mobile phones, hands-free, hearing aids and teleconferencing systems. The echo is linked to the reflection of the signal emitted by the telephone loudspeaker on the wall of the room or the car cabin and picked up by the microphone of the same system. consequently, the distant speaker is in the situation where he hears his own voice again with certain delay. This echo causes a reduction in the quality of speech in conversations.

In this work we will deal with the problem of acoustic echo in car and teleconferencing systems.

The principle of an echo cancellation is shown in figure.1.2. the echo signal received on the microphone is obtained by the convolution between the room impulse response $\mathbf{h}$ and the input signal $\mathbf{x}(n)$, then the echo signal is:

$$y(n) = \mathbf{h}^{\mathrm{T}}\mathbf{x}(n) \tag{1.1}$$

where $\mathbf{h} = [h_0\ h_1\ , \dots\ , h_{L-1}]^T$ is the unknown filter vector, L is the length of the echo path, the superscript $(.)^T$ denotes transpose of a vector and $\mathbf{x}(n) = [x(n)\ x(n-1), \dots \dots, x(n-L+1)]^T$ is the vector contains the last L samples of the far-end speech signal $x(n)$. The desired signal is:

$$d(n) = y(n) + v(n) \tag{1.2}$$

where $v(n)$ is the background noise and $\mathbf{w}(n)$ is a finite impulse response (FIR) adaptive filter.

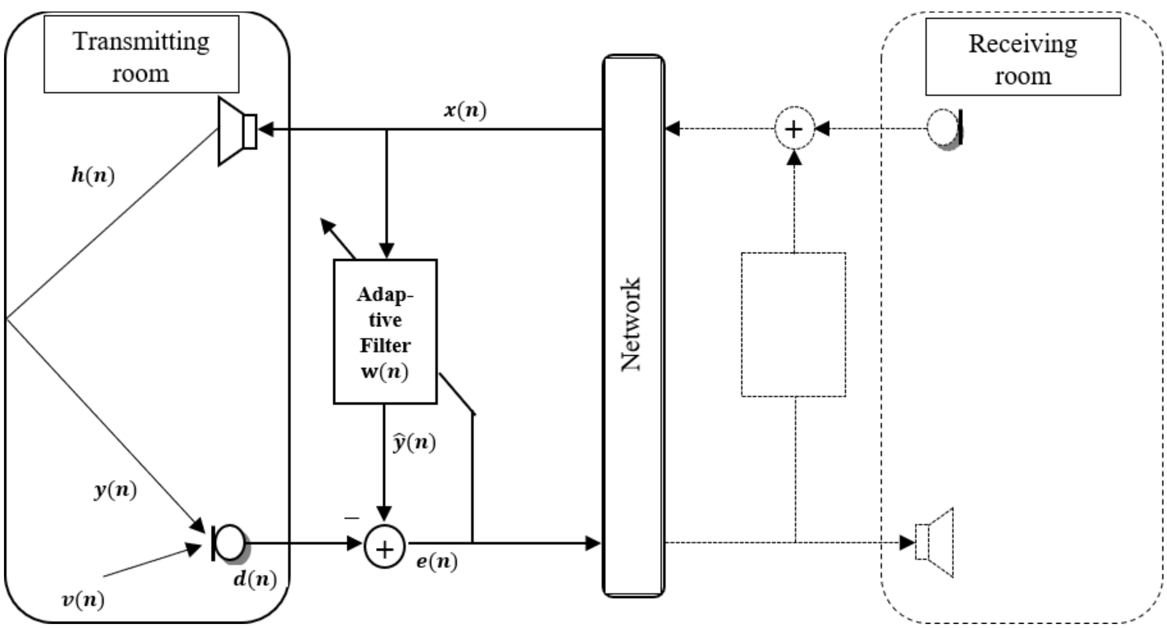

Figure.1.2. Basic AEC setup in a teleconferencing system.

The estimated echo $\hat{y}(n)$ is created by the convolution of the coefficient vector of adaptive filter $\mathbf{w}(n) = [w_0(n)\ w_1(n)\ , \dots \dots, w_{L-1}(n)]^T$, with the received input signal $\mathbf{x}(n)$. The estimated echo is:

$$\hat{y}(n) = \mathbf{w}^T(n-1)\mathbf{x}(n) \tag{1.3}$$

The adaptive algorithms estimate **h** by $\mathbf{w}(n)$ using a priori estimation error, is written, for each n:

$$e(n) = d(n) - \hat{y}(n) \qquad (1.4).$$

1.4. Adaptive filtering

An adaptive filter is a digital filter whose coefficients change themselves according to external signals (figure 1.3) [2]. It is used whenever an environment is poorly known or changing, or to suppress disturbances located in the frequency domain of the useful signal, which conventional filters fail to do.

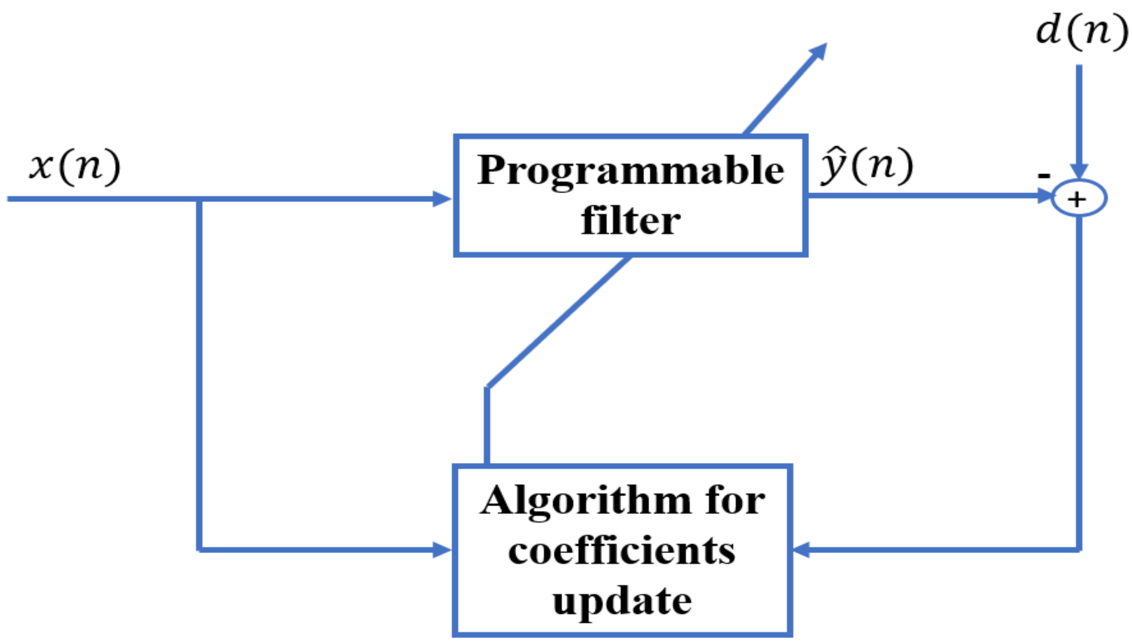

Figure 1. 3 : Principle of an adaptive filter.

1.4.1.Basic principle and applications of the adaptive filter

The principle of the adaptive filter is described as follows: the input signal $\mathbf{x}(n)$ is convolved with the filter $\mathbf{w}(n)$, the result of this convolution gives $\hat{y}(n)$ which is compared with the desired signal $d(n)$, the difference between the desired signal $d(n)$ and the signal at the output of the filter $\hat{y}(n)$ gives the error signal $e(n)$, makes it possible to update the coefficients of the adaptive filter $\mathbf{w}(n)$ (figure 1.4).

Adaptive filters have applications in different fields with specific objectives [2], among these applications we cite:

- ✓ System identification.
- ✓ Prediction.
- ✓ Inverse modeling.
- ✓ Noise cancellation.

1.4.1.a. System identification

One of the most important application is to determine the model of the system to be identified. The system identification is achieved by minimizing $e(n)$ which results to $\hat{y}(n) \approx d(n)$, as shown in figure 1.4.

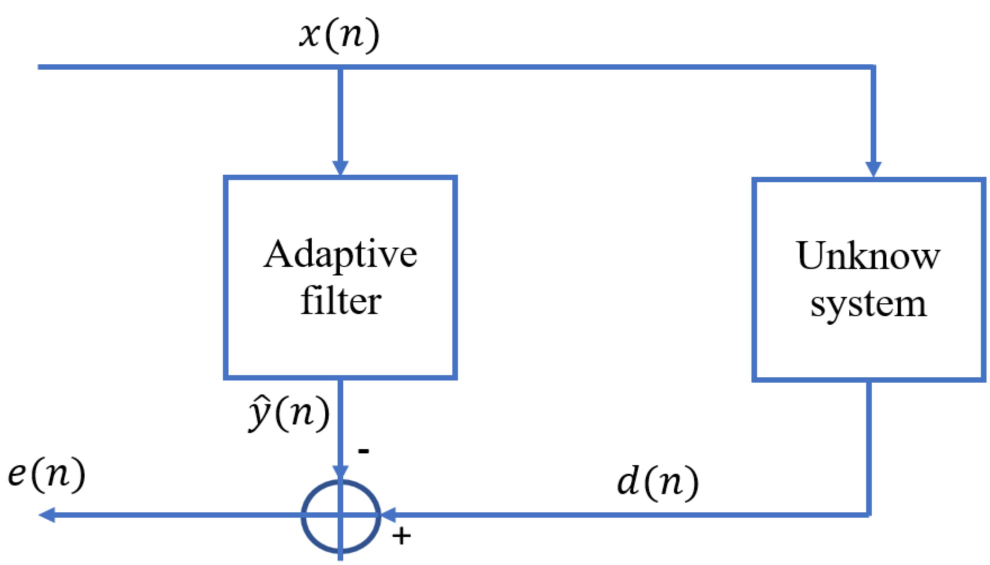

Figure 1. 4: The bloc diagram of system identification.

1.4.1.b. Prediction

In a such system, we can predict the output $\hat{y}(n)$ using the input vector $\mathbf{x}(n-1)$. Figure 1.5 shows the prediction process that generates an output signal $\hat{y}(n)$ which is the predicted value obtained by the adaptive filter using past observation of the input signal $x(n)$. This method is used in several application such as: speech coding and speech synthesis.

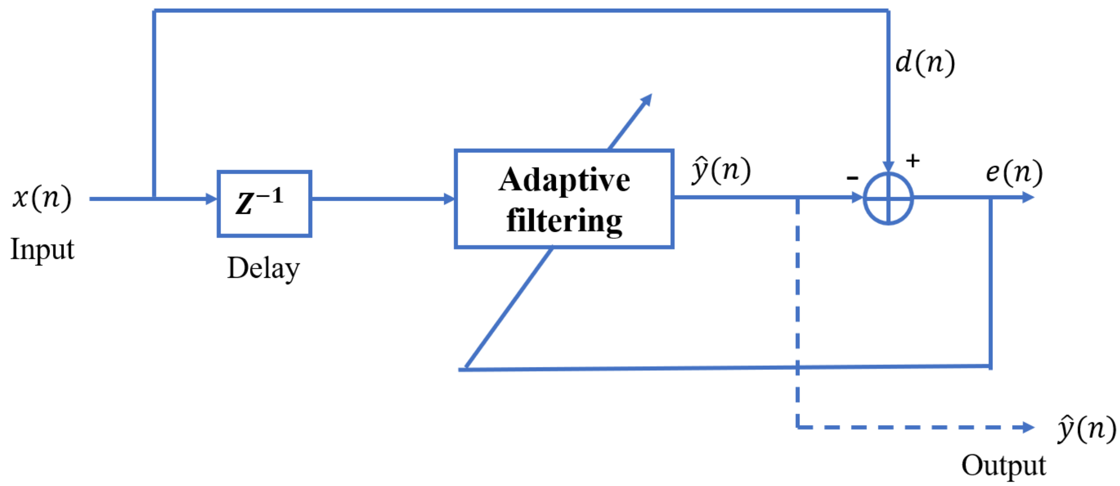

Figure 1. 5 : The bloc diagram of direct error prediction.

1.4.1.c. Inverse modeling

Figure 1.6 represent the basic principle of inverse modeling; it can be made by inserting an adaptive filter with the system to be identified to perform the estimate.

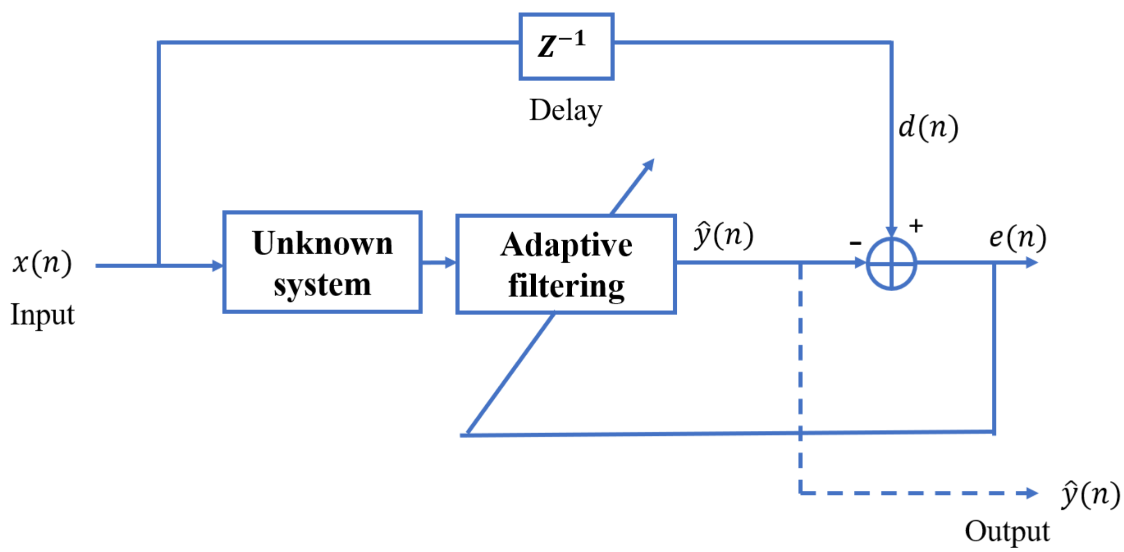

Figure 1. 6 : The bloc diagram of inverse modeling.

1.4.1.d. Noise cancellation

As shown in the figure 1.7, a noise cancellation system has two inputs: primary and reference. The primary input receives a signal $x(n)$ from the signal source that is corrupted by the presence of noise $v_1(n)$ uncorrelated with the signal. The reference input receives a noise $v_2(n)$ uncorrelated with the signal but correlated in some way with the noise $v_1(n)$. The noise $v_2(n)$ passes through a filter to produce an output $\hat{y}(n)$ that is a close estimate of primary input noise. This noise estimate is subtracted from the corrupted signal to produce an estimate of the signal at $e(n)$, the noise cancellation system output.

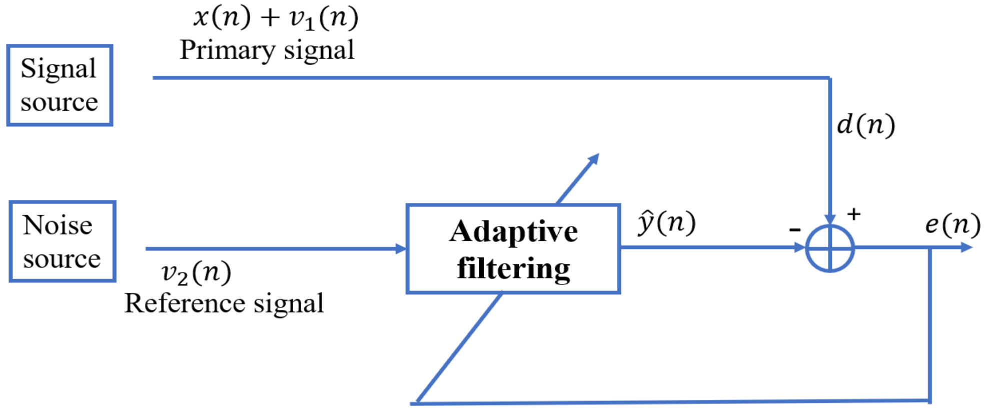

Figure 1. 7: The bloc diagram of noise cancellation.

1.4.2. Adaptation process of an adaptive filter

The adaptation process is the procedure used for updating the filter coefficients, it requires two preliminary stages:

- **Choice of objective function**

The objective function (cost) $J[e(n)]$ can be defined as a function that must satisfy the properties of optimization and non-negativity. Taking into account the complexity of the algorithm as a criterion, we can cite the most used objective functions in the establishment of adaptive filtering algorithms.

- **Least Square (LS)**

$$J[e(n)] = \sum_{k=1}^{n}[e(k)]^2 \qquad (1.5).$$

- **Recursive Least Square (RLS)**

$$J[e(n)] = \frac{1}{n+1}\sum_{k=0}^{n}[e(k)]^2 \quad (1.6).$$

- **Weighted Recursive Least Square (WRLS)**

$$J[e(n)] = \sum_{k=0}^{n}\lambda^{n-k}[e(k)]^2 \quad (1.7).$$

Where λ is a forgetting factor $0 < \lambda \leq 1$.

- **Mean Square Error (MSE)**

$$J[e(n)] = E[|e(n)|^2] \quad (1.8).$$

- **Mean Absolute Error (MAE)**

$$J[e(n)] = E[|e(n)|] \quad (1.9).$$

- **The algorithm of minimization of the cost function**

The algorithm of minimization is defined as a procedure used to adjust the parameters of the adaptive filter in order to minimize the objective function.

The adaptive filtering algorithm allows to calculate the coefficients of the filter $\mathbf{w}(n)$ so that the difference between the desired signal $d(n)$ and the current output of the filter $\hat{y}(n)$ is minimized according to a predefined statistical criterion. In general, the adaptation algorithm is presented in the vector form in the following equation [2]:

$$\begin{bmatrix} Vector\ of \\ New \\ coefficients \\ of\ the\ filter \end{bmatrix} = \begin{bmatrix} Vector\ of \\ Old \\ coefficients \\ of\ the\ filter \end{bmatrix} + (step\ size)\begin{pmatrix} Sample\ of \\ error\ signal \end{pmatrix}\begin{bmatrix} Vector \\ of\ input \\ signal \end{bmatrix} \quad (1.10).$$

Several adaptive filtering algorithms have been proposed in the literature, to update the coefficients of the adaptive filter, in order to minimize the error between the desired signal and the estimated signal and to converge around the optimal solution. We cite for example, stochastic gradient algorithms, also called Least Mean Square (LMS) [3], Recursive Least Square (RLS) [4], Affine Projection (AP) [5]. As well as adaptive algorithms in the frequency domain like frequency domain LMS algorithms [6].

1.4.3. The criteria of the choice of adaptive filtering algorithms:

In the performance comparison between the algorithms of identification by adaptive filtering, several performance criteria can be used to evaluate objectively the performance of the new algorithms proposed in the literature, the choice between them is based on the criteria of following performances:

- **Convergence speed**

The convergence speed represents the number of iterations required by an algorithm to reach the smallest error in the steady state. A fast convergence speed allows the algorithm to adapt quickly the stationary environment of unknown statistics.

- **Steady-state Mean Square Error (SS-MSE)**

It is the minimum value of the error, when the algorithm reaches its final MSE.

- **Tracking capability**

Tracking capability is present when an adaptive filtering algorithm operates in a non-stationary environment (People or object movement). The algorithm must be able to track quickly the statistical variations in the environment.

- **Computational complexity**

It is the number of the arithmetic operations required to do a complete iteration of the algorithm; the objective is to facilitate calculations with maintaining (ensuring) the best performance of the filter.

1.5. The Wiener filtering

The Wiener filtering, first proposed by Norbert Wiener and published in 1949, it is suitable for situations in which the signal or noise is stationary. Considering the figure 1.8:

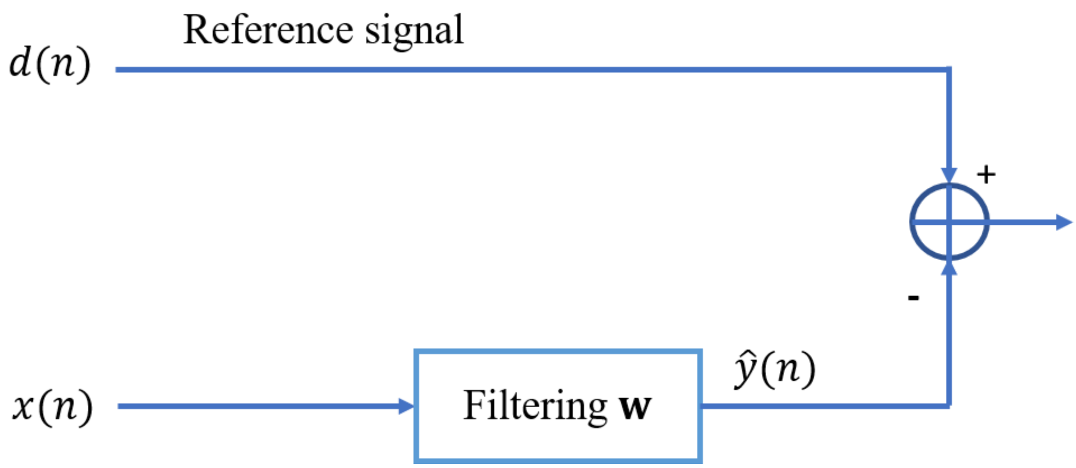

Figure 1. 8: Wiener filtering scheme.

The problem of the optimal filter is to find the best filter, i.e., the one who makes it possible to obtain at output a response $\hat{y}(n)$ as close as possible to a desired response $d(n)$, when the input is a certain sequence of $x(n)$ [7], [8].

Therefore, the problem consists in looking for the filter ensuring the smallest error $e(n)$, in the sense of an optimization criterion:

$$\mathbf{w}_{\text{opt}} = \text{argmin}_{\text{w}} J[e(n)] \qquad (1.11).$$

Where the operator $\text{argmin}_{\text{w}} J[.]$ is minimization of the cost function with respect to **w**.

$$\mathbf{w}_{\text{opt}} = [w_{\text{opt},0} \quad w_{\text{opt},1} \quad \dots \quad w_{\text{opt,L}-1}]^T \qquad (1.12).$$

Where $\mathbf{w}_{\text{opt}}$ is the optimum vector, and $J(\mathbf{w}) = J[e(n)]$ represents the cost function shown in the equation (1.8).

We have:

$$e(n) = d(n) - \mathbf{w}^T\mathbf{x}(n) \qquad (1.13).$$

$$J[e(n)] = E[|e(n)|^2] \qquad (1.14).$$

$$J[e(n)] = E\{[d(n) - \mathbf{w}^T\mathbf{x}(n)]^2\} \qquad (1.15).$$

$$\nabla_{\mathbf{w}} J[e(n)] = 0 \qquad (1.16).$$

$$2\mathbf{R}_{L,xx}\mathbf{w} - 2\mathbf{p}_{L,xd} = 0 \tag{1.17}$$

Where $\mathbf{R}_{L,xx} = E[\mathbf{x}_L(n)\mathbf{x}_L^T(n)]$is the autocorrelation matrix of order L of the input signal $x(n)$, and $\mathbf{p}_{L,xd} = E[\mathbf{x}_L(n)d(n)]$ is the cross-correlation vector between the desired output $d(n)$ and the input $x(n)$.

Then, the optimum filter is given by the following expression:

$$\mathbf{w} = \mathbf{R}_{L,xx}^{-1}\mathbf{p}_{L,xd} \tag{1.18}$$

So, to achieve this objective, it is necessary to know the statistical properties of $\mathbf{R}_{L,xx}$ and $\mathbf{p}_{L,xd}$.

1.6. Principle of deterministic gradient algorithm

The modification of the adaptive filter coefficients in the direction of the steepest descent of error hypersurface, leads to the deterministic gradient algorithm, written as [2], [9], [10]:

$$\mathbf{w}(n) = \mathbf{w}(n-1) - \frac{1}{2}\alpha\, g(n) \tag{1.19}$$

Where n denotes the iteration, α is a step size, possibly variable, which controls the stability and convergence speed of the algorithm.

$$g(n) = \frac{\partial J[\mathbf{w}(n)]}{\partial \mathbf{w}(n)} \tag{1.20}$$

$g(n)$ is the gradient of the cost function, which can be written as [11]:

$$J[\mathbf{w}(n)] = E[e^2(n)].$$

From the iteration $n-1$ to the iteration n, the vector $\mathbf{w}(n)$ is updated as follows:

$$\Delta\mathbf{w}(n) = \mathbf{w}(n) - \mathbf{w}(n-1) \tag{1.21}$$

$$\Delta\mathbf{w}(n) = -\frac{1}{2}\alpha\, g(n) \tag{1.22}$$

By substitution of the Equation (1.13) in the Equation (1.20), we get:

$$g(n) = \frac{\partial E[e^2(n)]}{\partial \mathbf{w}(n)}$$

$g(n) = -2E[\mathbf{x}(n)e(n)]$

$g(n) = -2E[\mathbf{x}(n)d(n) - \mathbf{x}(n)\mathbf{x}^T(n)\mathbf{w}(n-1)]$

$g(n) = -2\mathbf{p}_{L,xd} + 2\mathbf{R}_{L,xx}\mathbf{w}(n-1)$ (1.23).

We can deduce the deterministic gradient algorithm for the Wiener filtering.

By substitution of the Equation (1.23) in the Equation (1.19), we get:

$\mathbf{w}(n) = \mathbf{w}(n-1) + \alpha\, E[\mathbf{x}(n)e(n)]$ (1.24).

It is the update equation of the deterministic gradient algorithm.

1.7. Stochastic gradient algorithm LMS

The stochastic gradient algorithm, or Least Mean Square (LMS) is an approximation of the deterministic gradient algorithm. This algorithm has been proposed by Bernard Widrow and Marcian Hoff in 1960 [12].

The idea of the algorithms of type LMS is to replace the statistical mean of the deterministic gradient algorithm of the Equation (1.24), by its instantaneous value. Then, we get the Equation (1.25) [2], [7]:

$\mathbf{w}(n) = \mathbf{w}(n-1) + \mu_{LMS}\, \mathbf{x}(n)e(n)$ (1.25).

This algorithm is the most widely used algorithm in adaptive filtering applications, due to its simplicity and its reduced complexity.

The parameter μ_{LMS} is the step size the LMS algorithm, which depends on the power of the signal $x(n)$. For the non-stationary signals (the power of the signal $x(n)$ changes over time), the LMS algorithm fails to work properly.

The necessary and sufficient condition of convergence and numerical stability of the LMS algorithm is $0 < \mu_{LMS} < \frac{2}{\lambda_{max}}$, where λ_{max} represents the greatest eigenvalue of the autocorrelation matrix $\mathbf{R}_{L,xx}$. In practice, to ensure the convergence to the quadratic mean, we choose $0 < \mu_{LMS} < \frac{2}{trace(\mathbf{R}_{L,xx})} = \frac{2}{L\sigma_x^2}$, where the operator $trace(.)$ denotes the sum of diagonal elements of a matrix, and σ_x^2 the variance of the input signal $x(n)$ assumed centered

The LMS algorithm is summarized in Table 1.1, where L is the filter length,

Table1.1. The LMS algorithm

Initialization parameters: μ_{LMS}, L

$\mathbf{w}(0) = [w_0(0)\ \ w_1(0), \dots\ w_{L-1}(0)]^T = 0_{L\times 1};$

For each instant of time $n = 1,2 \dots$

$d(n) = y(n) + v(n);$

$\mathbf{x}(n) = [x(n)\ x(n-1), \dots x(n-L+1)]^T$

Filtering Part:

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

$e(n) = d(n) - \mathbf{w}^T(n-1)\mathbf{x}(n)$

Adaptation Part:

$\mathbf{w}(n) = \mathbf{w}(n-1) + \mu_{LMS}\, e(n)\mathbf{x}(n)$

1.8. The Normalized LMS (NLMS) algorithm

One of the family of the LMS algorithm, called Normalized Least Mean Square (NLMS) algorithm [13], avoids the drawback of the LMS algorithm, where the adaptation gain is normalized by the power of the input signal $x(n)$. The major problem of the LMS algorithm is in the case of the power of the input signal changes over time, as a result, the step size between two adjacent coefficients changes also, this will affect the convergence speed. Therefore, the normalization is a solution to solve this problem. Then, to avoid division by zero in case of small values of the power of the input signal, we introduce a small parameter c_0, where $c_0 > 0$ is a regularization parameter. The update of the adaptive filter coefficients is done by the following equation [2], [14]:

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \frac{\mu_{NLMS}}{\mathbf{x}^T(n)\mathbf{x}(n) + c_0} e(n)\mathbf{x}(n) \qquad (1.26).$$

Where μ_{NLMS} represents the step size of the NLMS algorithm. The sufficient condition of convergence and numerical stability is $0 < \mu_{NLMS} < 2$. In order to have a computational complexity of about $2L$ for the NLMS algorithm, the denominator term of Equation (1.26) must be estimated recursively by [15]:

$$P_x(n) = \mathbf{x}^T(n)\mathbf{x}(n) = \beta P_x(n-1) + (1-\beta)Lx^2(n) \qquad (1.27).$$

Where β is a forgetting factor.

1.9. The Recursive Least Square (RLS) algorithm

The method of Recursive Least Square consists of minimizing with respect of the parameter $\mathbf{w}(n)$, at each instant n, a defined criterion on the weighted errors committed from the initial instant. This criterion is given by [2]:

$$J_{\mathbf{w}}(n) = \sum_{i=1}^{n} \lambda^{n-i}[d(i) - \mathbf{w}^T(n)\mathbf{x}(i)]^2 \qquad (1.28).$$

Where λ $(0 < \lambda \leq 1)$ is an exponential forgetting factor, which allows the algorithm to forget the too far away past and to track the non-stationarity present in signals.

We suppose that the signals $x(n)$ and $d(n)$ are zero before the initial instant $n = 0$. The solution that expresses the nullity of the functional gradient $J_{\mathbf{w}}(n)$:

$$\nabla_{\mathbf{w}} J_{\mathbf{w}}(n) = 0 \Leftrightarrow -2\left\{\sum_{i=1}^{n} \lambda^{n-i}\mathbf{x}(i)d(i) - \left[\sum_{i=1}^{n} \lambda^{n-i}\mathbf{x}(i)\mathbf{x}^T(i)\right]\mathbf{w}(n)\right\} = 0 \qquad (1.29).$$

is given by:

$$\mathbf{R}_{xx}(n)\mathbf{w}(n) = \mathbf{p}_{xd}(n) \qquad (1.30).$$

as a result, we have:

$$\mathbf{w}(n) = \mathbf{R}_{xx}^{-1}(n)\mathbf{p}_{xd}(n) \qquad (1.31).$$

Where the matrix $\mathbf{R}_{xx}(n)$ represents the short-term autocorrelation matrix, given by:

$$\mathbf{R}_{xx}(n) = \sum_{i=1}^{n} \lambda^{n-i}\mathbf{x}(i)\mathbf{x}^T(i) \qquad (1.32).$$

and the vector $\mathbf{p}_{xd}(n)$ represents a short-term cross-correlation vector, given by:

$\mathbf{p}_{xd}(n) = \sum_{i=1}^{n} \lambda^{n-i}\mathbf{x}(i)d(i)$ (1.33).

The solution (1.31) requires the inversion of a square matrix of order L whose the calculation cost is about L^3 arithmetic operations per iteration.

By developing equations (1.32) and (1.33), we deduce their recursive relations, respectively:

$\mathbf{R}_{xx}(n) = \lambda\mathbf{R}_{xx}(n-1) + \mathbf{x}(n)\mathbf{x}^T(n)$ (1.34).

$\mathbf{p}_{xd}(n) = \lambda\mathbf{p}_{xd}(n-1) + \mathbf{x}(n)d(n)$ (1.35).

Using equations (1.34) and (1.35), we get a solution equivalent to equation (1.31), which can be written as the following recursive form:

$\mathbf{w}(n) = \mathbf{w}(n-1) + \mathbf{c}(n)e(n)$ (1.36).

where $e(n)$ is the a priori error (calculated from the vector coefficients of the previous instant), and

$\mathbf{c}(n) = [\mathrm{c}_1(n), \mathrm{c}_2(n), \dots, \mathrm{c}_L(n)]^T = \mathbf{R}_{xx}^{-1}(n)\mathbf{x}(n)$ (1.37).

is called the Kalman gain.

In order to reduce the complexity cost from $O(L^3)$ to $O(L^2)$, a matrix inversion lemma must be applied.

The matrix inversion lemma is defined as follows [16]:

Given four matrices A, B, C and D of appropriate dimensions, we have:

$(\mathbf{A} + \mathbf{BCD})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{BC}(\mathbf{C}^{-1} + \mathbf{DA}^{-1}\mathbf{B})^{-1}\mathbf{DA}^{-1}$ (1.38).

By applying this lemma to the expression (1.34) and by taking:

$\mathbf{A} = \lambda\mathbf{R}_{xx}(n-1)$, $\mathbf{B} = \mathbf{x}(n)$, $\mathbf{C} = \mathbf{I}$ and $\mathbf{D} = \mathbf{x}^T(n)$

We get:

$$\mathbf{R}_{xx}^{-1}(n) = \lambda^{-1}\mathbf{R}_{xx}^{-1}(n-1) - \lambda^{-2}\frac{\mathbf{R}_{xx}^{-1}(n-1)\mathbf{x}(n)\mathbf{x}^{T}(n)\mathbf{R}_{xx}^{-1}(n-1)}{1+\lambda^{-1}\mathbf{x}^{T}(n)\mathbf{R}_{xx}^{-1}(n-1)\mathbf{x}(n)} \quad (1.39a).$$

Equivalent to:

$$\mathbf{R}_{xx}^{-1}(n) = \lambda^{-1}\left[\mathbf{R}_{xx}^{-1}(n-1) - \frac{\mathbf{R}_{xx}^{-1}(n-1)\mathbf{x}(n)\mathbf{x}^{T}(n)\mathbf{R}_{xx}^{-1}(n-1)}{\lambda+\mathbf{x}^{T}(n)\mathbf{R}_{xx}^{-1}(n-1)\mathbf{x}(n)}\right] \quad (1.39b).$$

The equation (1.39a) or (1.39b) calculates recursively the inversion of the matrix $\mathbf{R}_{xx}(n)$, knowing the inverse of $\mathbf{R}_{xx}(n-1)$. This procedure is less complex than inversing directly at each instant the matrix $\mathbf{R}_{xx}(n)$.

The RLS algorithm, first proposed by Godard [17], is summarized in Table 1.2:

Table1.2. The RLS algorithm with complexity of $O(L^2)$

Initialization parameters: $0 < \lambda < 1$, L

$\mathbf{R}_{xx}^{-1}(0) = \frac{1}{\mu^2}\mathbf{I}$; where $\mu < 1$ and **I** is the identity matrix of order of L

$\mathbf{w}(0) = \mathbf{c}(0) = 0_{L\times 1}$;

Available variables at instant n : $\mathbf{R}_{xx}^{-1}(n-1)$, $\mathbf{w}(n-1)$

New informations : $x(n)$, $d(n)$

Kalman gain : $\mathbf{c}(n) = \frac{\mathbf{R}_{xx}^{-1}(n-1)\mathbf{x}(n)}{\lambda+\mathbf{x}^{T}(n)\mathbf{R}_{xx}^{-1}(n-1)\mathbf{x}(n)}$

Filtering error: $e(n) = d(n) - \mathbf{w}^{T}(n-1)\mathbf{x}(n)$

Filter adaptation : $\mathbf{w}(n) = \mathbf{w}(n-1) + \mathbf{c}(n)\, e(n)$

Propagation of the inverse of $\mathbf{R}_{xx}(n)$: $\boldsymbol{R}_{xx}^{-1}(n) = \lambda^{-1}[\mathbf{R}_{xx}^{-1}(n-1) - \mathbf{c}(n)\mathbf{x}^{T}(n)\mathbf{R}_{xx}^{-1}(n-1)]$

1.10. Fast Recursive Least Square (FRLS) algorithms

The RLS algorithm is obtained using a recursion equation over the time in the inversion of the autocorrelation matrix. The FRLS algorithm uses a recursion equation over the time and on the order of the Kalman gain, simultaneously, which results to complexity proportional to L. Several versions of FRLS have been proposed in the literature, such as the Fast Kalman (FK) algorithm, first proposed in [18], the Fast a Posteriori Error Sequential Technique (FAEST) algorithm [19] and the Fast Transversal Filter (FTF) algorithm [20], that uses a transversal filter in the adaptation.

As we have seen in the previous subsection Equation (1.36), the a priori error is calculated from the vector coefficients of the previous instant. Using the same procedure, we can define the a posteriori error $\epsilon(n)$ (calculated after the update of the filter) that generates a new adaptation gain called dual Kalman gain $\tilde{\mathbf{c}}(n)$. So, we have:

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \tilde{\mathbf{c}}(n)\epsilon(n) \quad (1.40).$$

$$\epsilon(n) = d(n) - \mathbf{w}^T(n)\mathbf{x}(n) \quad (1.41).$$

$$\tilde{\mathbf{c}}(n) = \lambda^{-1}\mathbf{R}_{xx}^{-1}(n-1)\mathbf{x}(n) \quad (1.42).$$

One of the advantages of the FRLS algorithms, is the a posteriori errors can be calculated from the a priori errors before the adaptation process [21], [22].

For that reason, and in order to calculate recursively the Kalman gain, some properties can be used, such as, the invariance by shifting the input signal vector extended to the order $(L+1)$:

$$\mathbf{x}_{L+1}(n) = \begin{bmatrix} x(n) \\ \mathbf{x}(n-1) \end{bmatrix} \quad (1.43a).$$

$$\mathbf{x}_{L+1}(n) = \begin{bmatrix} \mathbf{x}(n) \\ x(n-L) \end{bmatrix} \quad (1.43b).$$

These two forms of the input vector define two partitioned matrices of the autocorrelation of order of $(L+1)$.

Using the vector of Equation (1.43a), the first matrix can be written as:

$$\mathbf{R}_{xx,L+1}(n) = \sum_{i=1}^{n} \lambda^{n-i}\, \mathbf{x}_{L+1}(i)\mathbf{x}_{L+1}^T(i) = \sum_{i=1}^{n} \lambda^{n-i} \begin{bmatrix} x(i) \\ \mathbf{x}(i-1) \end{bmatrix} \begin{bmatrix} x(i) & \mathbf{x}^T(i-1) \end{bmatrix} =$$

$$\begin{bmatrix} p^f(n) & (\mathbf{p}^f)^T(n) \\ \mathbf{p}^f(n) & \boldsymbol{R}_{xx}(n-1) \end{bmatrix} \quad (1.44a).$$

$$\text{Where :} \begin{cases} p^f(n) = \sum_{i=1}^{n} \lambda^{n-i} x^2(i) \\ \quad\quad and \\ \mathbf{p}^f(n) = \sum_{i=1}^{n} \lambda^{n-i} x(i)\mathbf{x}(i-1) = \lambda\, \mathbf{p}^f(n-1) + x(n)\mathbf{x}(n-1) \end{cases}$$

The superscript f represents the forward variable.

Now, using the vector of Equation (1.43b), the second matrix can be written as:

$$\mathbf{R}_{xx,L+1}(n) = \sum_{i=1}^{n} \lambda^{n-i}\, \mathbf{x}_{L+1}(i)\mathbf{x}_{L+1}^{T}(i) = \sum_{i=1}^{n} \lambda^{n-i} \begin{bmatrix} \mathbf{x}(i) \\ x(i-L) \end{bmatrix} [\mathbf{x}^{T}(i) \quad x(i-L)] =$$

$$\begin{bmatrix} \boldsymbol{R}_{xx}(n) & \mathbf{p}^{b}(n) \\ (\mathbf{p}^{b})^{T}(n) & p^{b}(n) \end{bmatrix} \tag{1.44b}$$

Where : $\begin{cases} p^{b}(n) = \sum_{i=1}^{n} \lambda^{n-i} x^{2}(i-L) \\ and \\ \mathbf{p}^{b}(n) = \sum_{i=1}^{n} \lambda^{n-i} x(i-L)\mathbf{x}(i) = \lambda\, \mathbf{p}^{b}(n-1) + x(n-L)\mathbf{x}(n) \end{cases}$

The superscript b represents the backward variable.

In order to solve this system (equations 1.44a and 1.44b), the inversion of the partitioned matrix lemma can be used [20], which results to:

$$\mathbf{R}_{xx,L+1}^{-1}(n) = \begin{bmatrix} 0 & 0 \\ 0 & \mathbf{R}_{xx}^{-1}(n-1) \end{bmatrix} + \begin{bmatrix} 1 \\ -\mathbf{R}_{xx}^{-1}(n-1)\mathbf{p}^{f}(n) \end{bmatrix} \left(p^{f}(n) - (\mathbf{p}^{f})^{T}(n)\mathbf{R}_{xx}^{-1}(n-1)\mathbf{p}^{f}(n) \right)^{-1} [1 \quad -(\mathbf{p}^{f})^{T}(n)\mathbf{R}_{xx}^{-1}(n-1)] \tag{1.45a}$$

$$\mathbf{R}_{xx,L+1}^{-1}(n) = \begin{bmatrix} \mathbf{R}_{xx}^{-1}(n) & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} -\mathbf{R}_{xx}^{-1}(n)\mathbf{p}^{b}(n) \\ 1 \end{bmatrix} \left(p^{b}(n) - (\mathbf{p}^{b})^{T}(n)\mathbf{R}_{xx}^{-1}(n)\mathbf{p}^{b}(n) \right)^{-1} [-(\mathbf{p}^{b})^{T}(n)\mathbf{R}_{xx}^{-1}(n) \quad 1] \tag{1.45b}$$

It can be seen that, the forms (1.45a) and (1.45b) provide, respectively, an optimum forward and backward predictors by least squares minimization.

The forward linear prediction is based on the cost function:

$$J_{\mathbf{a}}(n) = \sum_{i=1}^{n} \lambda^{n-i} [x(i) - \mathbf{a}^{T}(n)\mathbf{x}(i-1)]^{2} \tag{1.46}$$

where $\mathbf{a}(n)$ is the coefficient vector of the predictor whose optimum is given by:

$$\mathbf{a}(n) = \mathbf{R}_{xx}^{-1}(n-1)\mathbf{p}^f(n) \quad (1.47).$$

and the updates of the coefficients are obtained by the same way as the recursive versions of the transversal filter $\mathbf{w}(n)$ (equations (1.36) and (1.40)):

$$\mathbf{a}(n) = \mathbf{a}(n-1) + \mathbf{c}(n-1)e^f(n) \quad (1.48a).$$

$$\mathbf{a}(n) = \mathbf{a}(n-1) + \tilde{\mathbf{c}}(n-1)\epsilon^f(n) \quad (1.48b).$$

where $e^f(n)$ and $\epsilon^f(n)$ represent, respectively, the a priori and a posteriori forward prediction errors, given by:

$$e^f(n) = x(n) - \mathbf{a}^T(n-1)\mathbf{x}(n-1) \quad (1.49).$$

$$\epsilon^f(n) = x(n) - \mathbf{a}^T(n)\mathbf{x}(n-1) \quad (1.50).$$

The backward linear prediction is based on the cost function:

$$J_{\mathbf{b}}(n) = \sum_{i=1}^{n} \lambda^{n-i}[x(i-L) - \mathbf{b}^T(n)\mathbf{x}(i)]^2 \quad (1.51).$$

where $\mathbf{b}(n)$ is the coefficient vector of the predictor whose optimum is given by:

$$\mathbf{b}(n) = \mathbf{R}_{xx}^{-1}(n)\mathbf{p}^b(n) \quad (1.52).$$

by the same way, the updates of the coefficients are obtained as follows:

$$\mathbf{b}(n) = \mathbf{b}(n-1) + \mathbf{c}(n)e^b(n) \quad (1.53a).$$

$$\mathbf{b}(n) = \mathbf{b}(n-1) + \tilde{\mathbf{c}}(n)\epsilon^b(n) \quad (1.53b).$$

where $e^b(n)$ and $\epsilon^b(n)$ represent, respectively, the a priori and a posteriori backward prediction error, given by:

$$e^b(n) = x(n-L) - \mathbf{b}^T(n-1)\mathbf{x}(n) \quad (1.53).$$

$$\epsilon^b(n) = x(n-L) - \mathbf{b}^T(n)\mathbf{x}(n) \quad (1.54).$$

The terms $(p^f(n) - (\mathbf{p}^f)^T(n)\mathbf{R}_{xx}^{-1}(n-1)\mathbf{p}^f(n))$ and $(p^b(n) - (\mathbf{p}^b)^T(n)\mathbf{R}_{xx}^{-1}(n)\mathbf{p}^b(n))$ in equations (1.45a) and (1.45b) represent, respectively, forward and backward prediction error variances (obtained by minimizing the cost functions (1.46) and (1.51)). These terms are noted $\alpha(n)$ and $\beta(n)$, can be expressed recursively [23], as follows:

$$\alpha(n) = \lambda\alpha(n-1) + e^f(n)\epsilon^f(n) \tag{1.55}$$

$$\beta(n) = \lambda\beta(n-1) + e^b(n)\epsilon^b(n) \tag{1.56}$$

We consider $\mathbf{C}_{L+1}(n) = \mathbf{R}_{xx,L+1}^{-1}(n-1)\mathbf{x}_{L+1}(n)$, extended Kalman gains of order $(L+1)$, obtained using equations (1.45a) and (1.45b) are given by:

$$\mathbf{C}_{L+1}(n) = \begin{bmatrix} 0 \\ \mathbf{c}(n-1) \end{bmatrix} + \frac{\epsilon^f(n)}{\alpha(n)}\begin{bmatrix} 1 \\ -\mathbf{a}(n) \end{bmatrix} \tag{1.57a}$$

$$\mathbf{C}_{L+1}(n) = \begin{bmatrix} \mathbf{c}(n) \\ 0 \end{bmatrix} + \frac{\epsilon^b(n)}{\beta(n)}\begin{bmatrix} -\mathbf{b}(n) \\ 1 \end{bmatrix} \tag{1.57b}$$

Extended dual Kalman gains of order $(L+1)$, $\tilde{\mathbf{C}}_{L+1}(n) = \lambda^{-1}\mathbf{R}_{xx,L+1}^{-1}(n-1)\mathbf{x}_{L+1}(n)$, can be written as:

$$\tilde{\mathbf{C}}_{L+1}(n) = \begin{bmatrix} 0 \\ \tilde{\mathbf{c}}(n-1) \end{bmatrix} + \frac{e^f(n)}{\lambda\alpha(n-1)}\begin{bmatrix} 1 \\ -\mathbf{a}(n-1) \end{bmatrix} \tag{1.58a}$$

$$\tilde{\mathbf{C}}_{L+1}(n) = \begin{bmatrix} \tilde{\mathbf{c}}(n) \\ 0 \end{bmatrix} + \frac{e^b(n)}{\lambda\beta(n-1)}\begin{bmatrix} -\mathbf{b}(n) \\ 1 \end{bmatrix} \tag{1.58b}$$

Using the 1st component or the $(L+1)^{th}$ component of the extended Kalman gain or the extended dual Kalman gain, we can deduce:

$$\epsilon^f(n) = \alpha(n)C_{L+1}^1(n) \tag{1.59}$$

$$e^f(n) = \lambda\alpha(n-1)\tilde{C}_{L+1}^1(n) \tag{1.60}$$

$$\epsilon^b(n) = \beta(n)C_{L+1}^{L+1}(n) \tag{1.61}$$

$$e^b(n) = \lambda\beta(n-1)\tilde{C}_{L+1}^{L+1}(n) \tag{1.62}$$

In order to reduce the complexity of the algorithms, a relationship between the a priori filtering error and the a posteriori filtering error must be calculated, by substituting $\mathbf{w}(n)$ by its formula from Equation (1.41) to (1.13), which results to:

$$\epsilon(n) = e(n)[1 - \mathbf{c}^T(n)\mathbf{x}(n)] \tag{1.63}$$

We can put:

$$\gamma(n) = 1 - \mathbf{c}^T(n)\mathbf{x}(n) = 1 - \mathbf{x}^T(n)\mathbf{R}_{xx}^{-1}(n)\mathbf{x}(n) \tag{1.64}$$

where $\gamma(n)$ is called the likelihood variable, and $0 < \gamma(n) < 1$.

By the same way, we can get relationships between the a priori and the a posteriori prediction errors, given by:

$$\epsilon^f(n) = \gamma(n)e^f(n) \tag{1.65}$$

$$\epsilon^b(n) = \gamma(n)e^b(n) \tag{1.66}$$

In other hand, the update equations of the vectors $\mathbf{w}(n)$, $\mathbf{a}(n)$ and $\mathbf{b}(n)$, using either a priori errors or a posteriori errors, are equivalents.

This conduct, by substitutions, to a relationship between the Kalman gain and the dual Kalman gain:

$$\mathbf{c}(n) = \gamma(n)\tilde{\mathbf{c}}(n) \tag{1.67}$$

By substitution this latter into (1.64), we obtain the following equation:

$$\gamma(n) = \frac{1}{1+\tilde{\mathbf{c}}^T(n)\mathbf{x}(n)} \tag{1.68}$$

In order to calculate recursively the likelihood variable, we define the extended likelihood variable to order $(L+1)$, $\gamma_{L+1}(n) = 1 - \mathbf{x}_{L+1}^T(n)\mathbf{C}_{L+1}(n)$, using (1.57a) and (1.57b), two new expressions can be found:

$$\gamma_{L+1}(n) = \frac{\lambda\alpha(n-1)}{\alpha(n)}\gamma(n-1) \tag{1.69a}$$

$$\gamma_{L+1}(n) = \frac{\lambda\beta(n-1)}{\beta(n)}\gamma(n) \tag{1.69b}$$

By putting:

$$\omega(n) = \frac{\lambda\beta(n-1)}{\beta(n)} \quad (1.70).$$

and by substitution in Equations (1.56), (1.61) and (1.62), we get:

$$\omega(n) = 1 - e^{b}(n)C_{L+1}^{L+1}(n) = 1 - e^{b}(n)\gamma_{L+1}(n)\tilde{C}_{L+1}^{L+1}(n) \quad (1.71).$$

The likelihood variable of order L, can be also calculated by the following formula:

$$\gamma(n) = \frac{\gamma_{L+1}(n)}{\omega(n)} \quad (1.72).$$

The fastest algorithms of the FRLS algorithm are characterized by the dual Kalman gain, and its a posteriori errors are calculated from the a priori errors. Its complexity is of order of $7L$ multiplications per sample.

The FTF algorithm is summarized in Table 1.3.

Table1.3. The FRLS algorithm with complexity of $(7L)$

Initialization: $\mathbf{w}(0) = \mathbf{a}(0) = \mathbf{b}(0) = \tilde{\mathbf{c}}(0) = 0_L$; $\gamma(0) = 1$; $\alpha(0) = \lambda^{L}E_0$; $\beta(0) = E_0$ where $E_0 \geq \sigma_x^2 \frac{L}{100}$.

Available variables at instant n : $\mathbf{a}(n-1)$; $\mathbf{b}(n-1)$; $\tilde{\mathbf{c}}(n-1)$; $\gamma(n-1)$; $\alpha(n-1)$; $\beta(n-1)$; $\mathbf{w}(n-1)$

New informations : $x(n)$, $d(n)$

Prediction part :

$e^{f}(n) = x(n) - \mathbf{a}^{T}(n-1)\mathbf{x}(n-1)$;

$\mathbf{C}_{L+1}(n) = \begin{bmatrix} 0 \\ \tilde{\mathbf{c}}(n-1) \end{bmatrix} + \frac{e^{f}(n)}{\lambda\alpha(n-1)} \begin{bmatrix} 1 \\ -\mathbf{a}(n-1) \end{bmatrix}$;

$\mathbf{a}(n) = \mathbf{a}(n-1) + e^{f}(n)\gamma(n-1)\tilde{\mathbf{c}}(n-1)$;

$\alpha(n) = \lambda\alpha(n-1) + \gamma(n-1){e^{f}}^{2}(n)$;

$e^{b}(n) = \lambda\beta(n-1)\tilde{C}_{L+1}^{L+1}(n)$;

$\begin{bmatrix} \tilde{\mathbf{c}}(n) \\ 0 \end{bmatrix} = \tilde{\mathbf{C}}_{L+1}(n) + \tilde{C}_{L+1}^{L+1} \begin{bmatrix} \mathbf{b}(n-1) \\ -1 \end{bmatrix}$;

$\gamma_{L+1}(n) = \frac{\lambda\alpha(n-1)}{\alpha(n)}\gamma(n-1)$;

$\gamma(n) = \frac{\gamma_{L+1}(n)}{1-\gamma_{L+1}(n)e^{b}(n)\tilde{c}_{L+1}^{L+1}(n)}$;

$\mathbf{b}(n) = \mathbf{b}(n-1) + e^{b}(n)\gamma(n)\tilde{\mathbf{c}}(n)$;

Filtering part:

$e(n) = d(n) - \mathbf{w}^{T}(n-1)\mathbf{x}(n)$;

$\mathbf{w}(n) = \mathbf{w}(n-1) + e(n)\gamma(n)\tilde{\mathbf{c}}(n)$.

---

where E_0 is an initialization parameter, and σ_x^2 is the power of the input signal.

Several algorithms can be derived from the FRLS algorithm, which have a complexity proportional to L, such as the Simplified FTF (SMFTF) algorithm [24], the authors have observed that impulse responses to be identified are decreasing with the order of the filter, especially in AEC applications, and by discarding completely the backward predictor from the FTF algorithm, which can affect the last small components of the adaptive filter $\mathbf{w}(n)$, the most significant components can be obtained from forward predictor, due to the down-shift property used in the calculation of the dual Kalman gain. More complexity reduction can be obtained using this algorithm, when use a reduced size predictor on the forward variables, which results to complexity of $(2L + 5P)$, where P is the order of the predictor and $(P \ll L)$. A recent FRLS based algorithms, obtained from the SMFTF algorithm are proposed in [25], the authors propose to calculate the likelihood variable $\gamma(n)$ recursively, which results to a complexity of $6L$ when used without reduced size predictor, and a complexity of $(2L + 4P)$ using a reduced size predictor.

1.11. The Fast NLMS algorithm

Recently, a new low complexity and fast convergence adaptive filtering algorithm has been proposed in [26], [27], obtained also from FRLS algorithm, noted Fast NLMS (FNLMS) algorithm, because it has the property and the simplicity in its adaptation as the basic NLMS structure (complexity of $O(L)$ and step-size) with fast convergence rate as RLS algorithms. In order to get additional complexity reductions compared to FRLS algorithms previously described, authors in [26] proposed to discard completely the forward and the backward predictors from the FTF algorithm, and to use only a forward error prediction $\varepsilon(n)$ of the input signal to calculate the adaptation gain, instead of using $e^f(n)$.

Remember, from the previous subsection, that, by eliminating $\tilde{\mathbf{C}}_{L+1}$ from the two expressions (1.58a) and (1.58b), a recursive form for the dual Kalman gain can be obtained, as:

$$\begin{bmatrix} \tilde{\mathbf{c}}(n) \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \tilde{\mathbf{c}}(n-1) \end{bmatrix} + \frac{e^f(n)}{\lambda\alpha(n-1)} \begin{bmatrix} 1 \\ -\mathbf{a}(n-1) \end{bmatrix} - \frac{e^b(n)}{\lambda\beta(n-1)} \begin{bmatrix} -\mathbf{b}(n) \\ 1 \end{bmatrix} \tag{1.73}$$

If $\mathbf{a}(n) = \mathbf{b}(n) = 0$, the dual Kalman gain becomes:

$$\begin{bmatrix} \tilde{\mathbf{c}}(n) \\ c(n) \end{bmatrix} = \begin{bmatrix} \frac{e^f(n)}{\lambda\alpha(n-1)} \\ \tilde{\mathbf{c}}(n-1) \end{bmatrix} \tag{1.74}$$

where the variable $c(n)$ denotes the quantity $\frac{e^b(n)}{\lambda\beta(n-1)}$. Then, as suggested by the authors in [26], the dual Kalman gain becomes:

$$\begin{bmatrix} \tilde{\mathbf{c}}(n) \\ c(n) \end{bmatrix} = \begin{bmatrix} \frac{\varepsilon(n)}{\lambda\alpha(n-1)+c_0} \\ \tilde{\mathbf{c}}(n-1) \end{bmatrix} \tag{1.75}$$

where c_0 is a small positive constant, introduced to avoid division by zero, in case of the absence of the input signal. Thus, the forward prediction error variance is expressed as:

$$\alpha(n) = \lambda\alpha(n-1) + \varepsilon^2(n) \tag{1.76}$$

In addition, and in order to improve the convergence speed, the prediction error $\varepsilon(n)$ is calculated using first-order prediction model (Whitening or decorrelating the input signal), as follows:

$$\varepsilon(n) = x(n) - ax(n-1) \tag{1.77}$$

where a is a prediction parameter. Since the input statistics are unknown or variable over time, the parameter a must be estimated from the input signal, the choice of this parameter is obtained by the minimization of the cost function $E[\varepsilon^2(n)]$ with respect to a, as follows:

$$\frac{\partial E[\varepsilon^2(n)]}{\partial a} = \frac{\partial E[x^2(n) - 2ax(n)x(n-1) + a^2x^2(n-1)]}{\partial a} = -2E[x(n)x(n-1)] + 2aE[x^2(n-1)]$$

$$\frac{\partial E[\varepsilon^2(n)]}{\partial a} = 0 \quad \overset{equivalent\ to}{\Longleftrightarrow} \quad -2E[x(n)x(n-1)] + 2aE[x^2(n-1)] = 0$$

Thus:

$$a = \frac{E[x(n)x(n-1)]}{[x^2(n-1)]} = \frac{r_1}{r_0} \tag{1.78}$$

where r_0 and r_1 represent, respectively, the power of the input signal and the first lag of the autocorrelation function for a stationary input signal. The estimation of the prediction parameter can be obtained by recursive estimations of r_0 and r_1, written as:

$$r_1(n) = \lambda_a r_1(n-1) + x(n)x(n-1) \tag{1.79a}$$

$$r_0(n) = \lambda_a r_0(n-1) + x^2(n) \tag{1.79b}$$

where λ_a is an exponential forgetting factor.

Now the prediction parameter is written as:

$$a(n) = \frac{r_1(n)}{r_0(n)+c_a} \tag{1.80}$$

where c_a is a small positive constant.

The recursive expression of the likelihood variable $\gamma(n)$ can be obtained, using time shift invariance properties of the input signal vector extended to $L+1$, defined as follows:

$$\mathbf{x}_{L+1}^T(n) = [\mathbf{x}^T(n) \quad x(n-L)] \tag{1.81a}$$

$$\mathbf{x}_{L+1}^T(n) = [x(n) \quad \mathbf{x}^T(n-1)] \tag{1.81b}$$

and by multiplying the left and the right sides of the expression (1.75) by equations (1.81a) and (1.81b), respectively, which results to:

$$\mathbf{x}^T(n)\tilde{\mathbf{c}}(n) + c(n)x(n-L) = \mathbf{x}^T(n-1)\tilde{\mathbf{c}}(n-1) + \frac{x(n)\varepsilon(n)}{\lambda\alpha(n-1)+c_0} \tag{1.82}$$

By substitution $\mathbf{x}^T(n)\tilde{\mathbf{c}}(n) = \gamma^{-1}(n) - 1$ into equation (1.82), we find:

$$\gamma^{-1}(n) = \gamma^{-1}(n-1) - c(n)x(n-L) + \frac{x(n)\varepsilon(n)}{\lambda\alpha(n-1)+c_0} \tag{1.83}$$

Or:

$$\gamma(n) = \frac{\gamma(n-1)}{1+\gamma(n-1)\delta(n)} \tag{1.84}$$

where:

$$\delta(n) = -c(n)x(n-L) + \frac{x(n)\varepsilon(n)}{\lambda\alpha(n-1)+c_0} \tag{1.85}$$

In order to get more controlling to the adaptation gain a constant step size μ_{FNLMS}, can be added to the update equation, as follows:

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \mu_{FNLMS}e(n)\gamma(n)\tilde{\mathbf{c}}(n) \tag{1.86}$$

The total computational complexity in term of multiplications is $2L$, the FNLMS algorithm is given in the following table:

Table1.4. The FNLMS algorithm with complexity of $(2L)$

Initialization parameters :

$\mathbf{w}(0) = \tilde{\mathbf{c}}(0) = 0, \gamma(0) = 1, r_1(0) = 0,$

$\alpha(0) = r_0(0) = E_0 = 1$, where E_0 is an initialization

$d(n) = y(n) + v(n);$

For each instant of time n=1,2...

Prediction error:

$r_1(n) = \lambda_a r_1(n-1) + x(n)x(n-1)$

$r_0(n) = \lambda_a r_0(n-1) + x^2(n)$

$$a(n) = \frac{r_1(n)}{r_0(n) + c_a}$$

$\varepsilon(n) = x(n) - a(n)x(n-1)$

$\alpha(n) = \lambda\alpha(n-1) + \varepsilon^2(n)$

Adaptation gain:

$$\begin{bmatrix}\tilde{\mathbf{c}}(n)\\ c(n)\end{bmatrix} = \begin{bmatrix}\dfrac{\varepsilon(n)}{\lambda\alpha(n-1)+c_0}\\ \tilde{\mathbf{c}}(n-1)\end{bmatrix}$$

$$\delta(n) = -c(n)x(n-L) + \frac{x(n)\varepsilon(n)}{\lambda\alpha(n-1) + c_0}$$

$$\gamma(n) = \frac{\gamma(n-1)}{1 + \gamma(n-1)\delta(n)}$$

Filtering Part:

$$e(n) = d(n) - \mathbf{w}^T(n-1)\mathbf{x}(n)$$

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \mu_{FNLMS}e(n)\gamma(n)\tilde{\mathbf{c}}(n)$$

1.12 Conclusion

During the first chapter, we have presented the AEC system and the adaptive filtering from basic definitions to some applications, then we have summarized the adaptive filtering algorithms from Weiner filtering and deterministic gradient algorithm to stochastic gradient algorithms and its variants (LMS and NLMS). The LMS algorithms suffer from low convergence rate compared to RLS algorithms that have a drawback of high complexity, for that reason FRLS versions have been proposed in the literature that have a high convergence speed with lower computational complexity. Several techniques have been introduced to these algorithms, to reduce once more the computational complexity, this is what we will present in the next chapter.

Chapter 2:

Set-Membership Adaptive Filtering Algorithms

2.1 Introduction

In many applications where the number of filter coefficients is high (e.g acoustic echo cancellation), there is a need for reducing the complexity of the algorithms. Several techniques of reducing the computational complexity have been proposed, such as Partial Update (PU) [28]–[31] that is based on updating only part of filter coefficient vector at each time by maintaining the performances of the original algorithms, another technique based on Set-Membership Filtering (SMF) approach [32], [33], where the filter weights are not updated when the estimation error is lower than a chosen constant, which results to a reduction in overall computational complexity and a faster convergence speed compared to original algorithms.

In this chapter, we present some adaptive filtering algorithms based on SMF approach, after that we will expand in our first contribution noted Set-Membership FNLMS (SM-FNLMS) algorithm, then as a second contribution, we will show how a novel SM with fast convergence and a good tracking capability, has been derived to address the problem of acoustic echo.

2.2 Set-Membership NLMS (SM-NLMS) algorithm

The SMF is defined as a technique that allows the update of the adaptive filter coefficients below prescribed upper bound, the key idea of this technique is to find a feasibility set such that the bounded error specification is met for any member of this set [34]. As a result, the SM based adaptive filtering algorithms have a reduced computational complexity due to data selective updates [28]. The SMF concept is an approach that can be applied to adaptive filtering problems that are linear in parameters. The first proposed SM algorithm in [35], called Set-Membership NLMS (SM-NLMS) algorithm, it has a form similar to the conventional NLMS algorithm, the basic idea of the SM-NLMS algorithm is to perform a test to verify if the previous estimate $\mathbf{w}(n-1)$ lies outside the constraint set $\mathcal{H}(n)$, which is defined as the set containing all vectors $\mathbf{w}(n)$ such that the associated output error at time instant n is upper bounded in magnitude by ζ. That is:

$$\mathcal{H}(n) = \{\mathbf{w} \in \mathbb{R}^{N+1} : |d(n) - \mathbf{w}^T(n-1)\mathbf{x}(n)| \leq \zeta\} \quad (2.1).$$

where ζ is a parameter that gives a threshold error. If the modulus of the error signal is greater than the specified bound, the new estimate $\mathbf{w}(n)$ will be updated to the closest boundary of $\mathcal{H}(n)$ at a minimum distance, which means that the SM-NLMS minimizes $||\mathbf{w}(n) - \mathbf{w}(n-1)||^2$ subjected to $\mathbf{w}(n) \in \mathcal{H}(n)$ [36]. The updating is performed by an orthogonal projection of the previous estimate onto the closest boundary of $\mathcal{H}(n)$. Figure 2.1 illustrates the updating procedure of the SM-NLMS algorithm.

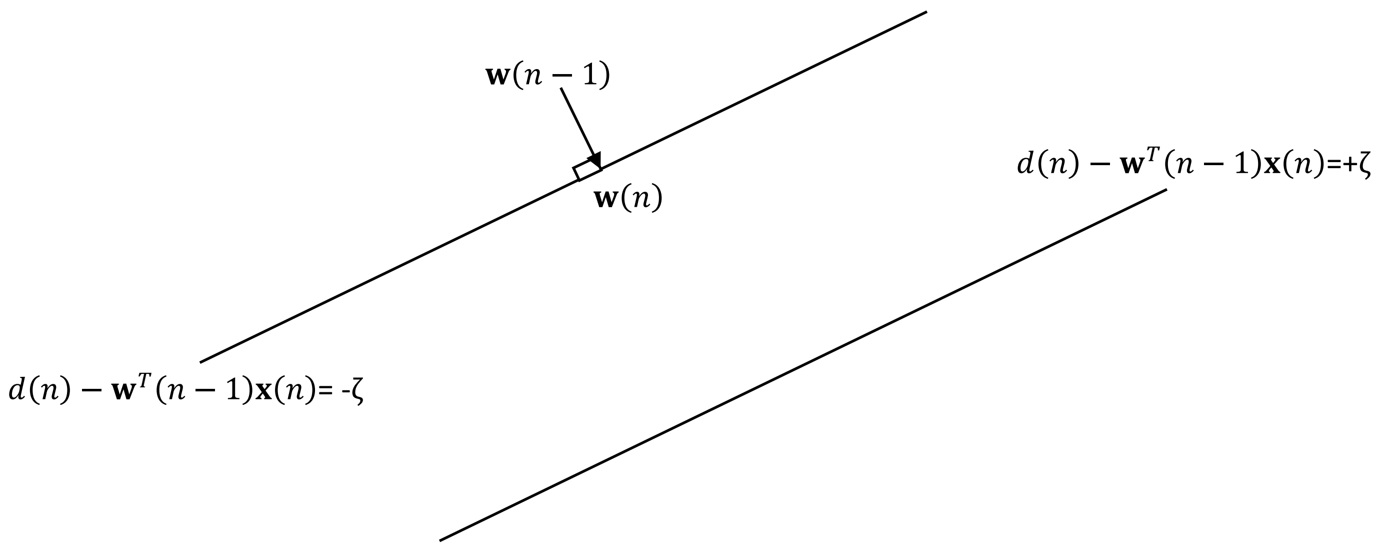

Figure 2.1: Coefficient vector updating for the SM-NLMS algorithm.

In order to derive the update equations, first we consider the a priori error $e(n)$ defined in equation (1.13), then the equation (1.26) of the NLMS algorithm can be written as:

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \frac{\mu(n)}{\mathbf{x}^T(n)\mathbf{x}(n)+c_0} e(n)\mathbf{x}(n) \quad (2.2).$$

where $\mu(n)$ is the variable step size, introduced in order to satisfy the following set-membership condition:

$$e(n) = d(n) - \mathbf{w}^T(n-1)\mathbf{x}(n) > \zeta \quad (2.3).$$

or

$$e(n) = d(n) - \mathbf{w}^T(n-1)\mathbf{x}(n) < -\zeta \quad (2.4).$$

Since the coefficients are updated to the closest boundary of $\mathcal{H}(n)$, the a posteriori error $\epsilon(n)$ defined in equation (1.41) should be equal to $\pm\zeta$, thus:

$$\epsilon(n) = d(n) - \mathbf{w}^T(n)\mathbf{x}(n) = \pm\zeta \quad (2.5).$$

Using equation (2.2), $\epsilon(n)$ can be written as follows:

$$\epsilon(n) = d(n) - \mathbf{w}^T(n-1)\mathbf{x}(n) - \frac{\mu(n)}{\mathbf{x}^T(n)\mathbf{x}(n)+c_0} e(n)\mathbf{x}(n)\mathbf{x}^T(n) \quad (2.6).$$

$$\epsilon(n) = e(n) - \frac{\mu(n)}{\mathbf{x}^T(n)\mathbf{x}(n)+c_0} e(n)\mathbf{x}^T(n)\mathbf{x}(n) \quad (2.7).$$

$$\epsilon(n) = e(n)[1-\mu(n)] = \pm\zeta \quad (2.8).$$

It leads to (for $c_0 = 0$):

$$1-\mu(n) = \pm\frac{\zeta}{e(n)} \quad (2.9).$$

Therefore, the variable step size $\mu(n)$, is given by:

$$\mu(n) = \begin{cases} 1-\frac{\zeta}{|e(n)|} & if\ |e(n)| > \zeta \\ 0 & otherwise \end{cases} \quad (2.10).$$

The SM-NLMS algorithm is outlined in Table 2.1:

Table 2.1. The SM-NLMS algorithm

Initialization parameters: L, error bound ζ= small constant;

$\mathbf{w}(0) = [w_0(0)\ \ w_1(0), \dots\ w_{L-1}(0)]^T = 0_{L\times 1}$;

For each instant of time $n = 1,2 \dots$

$d(n) = y(n) + v(n)$;

$\mathbf{x}(n) = [x(n)\ x(n-1), \dots x(n-L+1)]^T$

Filtering Part:

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

$$e(n) = d(n) - \mathbf{w}^T(n-1)\mathbf{x}(n)$$

$$\mu(n) = \begin{cases} 1 - \frac{\zeta}{|e(n)|} & if\ |e(n)| > \zeta \\ 0 & otherwise \end{cases}$$

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \frac{\mu(n)}{\mathbf{x}^T(n)\mathbf{x}(n)+c_0} e(n)\mathbf{x}(n)$$

The value of ζ is chosen around $\zeta = \sqrt{5\sigma_v^2}$, where σ_v^2 is the variance of the additive noise [35], [37]

By choosing the error bound $\zeta = 0$, it can be seen that $\mu(n) = 1$ in the SM-NLMS whenever $w(n-1) \notin \mathcal{H}(n)$, we perform a valid update since the hyperplane with zero a posteriori error is in $\mathcal{H}(n)$. In this case, the resulting algorithm does not minimize the Euclidean distance $||\mathbf{w}(n) - \mathbf{w}(n-1)||^2$ since the a posteriori error is zero and less than ζ.

2.3.The SM-NLMS with Adaptive Error Bound (SMAEB-NLMS) algorithm

In this subsection, we present a SM based algorithm that has a convergence speed and a complexity similar to SM-NLMS algorithm, the aim of this algorithm is to handle the over-bounding and the under-bounding problem simultaneously, by incorporating a variable error bound (adaptive error bound). The authors in [38] propose to use a variable error bound calculated recursively by NLMS update instead of using the error bound calculated from the statistics of the additive noise, and by maintaining the condition of SMF defined previously.

The update equation (2.2) can be written in terms of the weight error vector $\Delta\mathbf{w}(n) = \mathbf{w}_{opt}(n) - \mathbf{w}(n)$, as:

$$\Delta\mathbf{w}(n) = \Delta\mathbf{w}(n-1) - \mathbf{x}(n)\frac{1}{\mathbf{x}^T(n)\mathbf{x}(n)+c_0}[e(n) - \zeta] \quad (2.11).$$

Doing the square of the difference of the weight error vectors of $n-1$ and n, we get:

$$||\Delta\mathbf{w}(n) - \Delta\mathbf{w}(n-1)||^2 = \left\|\mathbf{x}(n)\frac{1}{\mathbf{x}^T(n)\mathbf{x}(n)+c_0}[e(n) - \zeta]\right\|^2 \quad (2.12).$$

By taking partial derivative of the right-hand term of Equation (2.12) with respect to ζ and by eliminating the constant terms, we find:

$$\nabla_{\zeta} = \frac{1}{\mathbf{x}^T(n)\mathbf{x}(n)+c_0}[e(n)-\zeta] \qquad (2.13).$$

So, the updating formula for the error bound becomes:

$$\zeta(n) = \begin{cases} \zeta(n-1) + \mu_g \frac{1}{\mathbf{x}^T(n)\mathbf{x}(n)+c_0}[|e(n)|-\zeta(n-1)] & if \ \ |e(n)| > \zeta(n-1) \\ \zeta(n-1) & otherwise \end{cases} \qquad (2.14).$$

where μ_g is the step size. The Set Membership NLMS with Adaptive Error Bound algorithm denoted SMAEB-NLMS is given in Table 2.2:

Table 2.2. The SMAEB-NLMS algorithm

Initialization parameters: L, error bound $\zeta(0) = \sqrt{5\sigma_v^2}$;

$\mathbf{w}(0) = [w_0(0) \ \ w_1(0), \dots \ w_{L-1}(0)]^T = 0_{L\times 1}$;

$d(n) = y(n) + v(n)$;

For each instant of time $n = 1,2 \dots$

$\mathbf{x}(n) = [x(n) \ x(n-1), \dots x(n-L+1)]^T$

Filtering Part:

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

$e(n) = d(n) - \mathbf{w}^T(n-1)\mathbf{x}(n)$

if $|e(n)| > \zeta(n-1)$

$$\mu(n) = 1 - \frac{\zeta(n-1)}{|e(n)|}$$

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \frac{\mu(n)}{\mathbf{x}^T(n)\mathbf{x}(n)+c_0}e(n)\mathbf{x}(n)$$

$$\zeta(n) = \zeta(n-1) + \mu_g \frac{1}{\mathbf{x}^T(n)\mathbf{x}(n)+c_0}[|e(n)|-\zeta(n-1)]$$

else

$$\mathbf{w}(n) = \mathbf{w}(n-1)$$

$$\zeta(n) = \zeta(n-1)$$

The advantage of this algorithm compared to other SM algorithms is that, the update weight error and the final MSE can be easily controlled by adjusting the step size in the error bound update formula. The complexity and robustness of SMAEB-NLMS algorithm are similar to the SM-NLMS algorithm.

2.4. The SM-NLMS with Robust Error Bound (SMREB-NLMS) algorithm

In this subsection, another algorithm based on the SM framework is presented, the authors of [39], have derived a new robust error bound to improve the robustness against impulsive noise and to get a lower Update probability (UP) compared to other SM based algorithms. The robust error bound (REB) can be expressed using two conditions, the first one is introduced in order to get a lower SS-MSE, thus the REB is chosen as:

$$\zeta(n) = |e(n)|\frac{|e(n)|}{\nu\theta(n)+|e(n)|} \quad (2.15).$$

where ν is a control parameter, and $0 < \nu < 1$ to obtain a lower steady state MSE.

$\theta(n)$ is the impulsive free estimation of $E[|e(n)|]$, estimated as:

$$\theta(n) = \hat{\sigma}(n) = \beta\hat{\sigma}(n-1) + (1-\beta)median(\psi(n)) \quad (2.16).$$

where $\psi(n) = [|e(n)|, |e(n-1)|, \dots, |e(n-L+1)|\ ]^T$.

The second condition is introduced to reduce the UP at steady state, if $\zeta(n) < \frac{\sqrt{5\sigma_v^2}}{\nu+1}$, we choose:

$$\zeta(n) = \frac{\sqrt{5\sigma_v^2}}{\nu+1} \quad (2.17).$$

By combining the two conditions, the REB formula can be written as:

$$\zeta(n) = \begin{cases} \frac{\sqrt{5\sigma_v^2}}{\nu+1}, & if \quad \frac{|e(n)|^2}{\nu\theta(n)+|e(n)|} < \frac{\sqrt{5\sigma_v^2}}{\nu+1} \\ \frac{|e(n)|^2}{\nu\theta(n)+|e(n)|}, & otherwise \end{cases} \quad (2.18).$$

Thus, the SM-NLMS with Robust Error Bound algorithm, denoted SMREB-NLMS, is demonstrated in Table 2.3.

Table 2.3. The SMREB-NLMS algorithm

Initialization parameters: L, error bound $\zeta(0) = \sqrt{5\sigma_v^2}$;

$\mathbf{w}(0) = [w_0(0)\;\; w_1(0), \dots\; w_{L-1}(0)]^T = 0_{L\times 1}$;

For each instant of time $n = 1,2 \dots$

$d(n) = y(n) + v(n)$;

$\mathbf{x}(n) = [x(n)\; x(n-1), \dots x(n-L+1)]^T$

Filtering Part:

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

$e(n) = d(n) - \mathbf{w}^T(n-1)\mathbf{x}(n)$

$\psi(n) = [|e(n)|, |e(n-1)|, \dots, |e(n-L+1)|\,]^T$

$\theta(n) = \hat{\sigma}(n) = \beta\hat{\sigma}(n-1) + (1-\beta) median(\psi(n))$

$$\zeta(n) = \begin{cases} \dfrac{\sqrt{5\sigma_v^2}}{\nu+1}, & if \quad \dfrac{|e(n)|^2}{\nu\theta(n)+|e(n)|} < \dfrac{\sqrt{5\sigma_v^2}}{\nu+1} \\ \dfrac{|e(n)|^2}{\nu\theta(n)+|e(n)|}, & otherwise \end{cases}$$

If $|e(n)| > \zeta(n)$

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \frac{\mu}{\mathbf{x}^T(n)\mathbf{x}(n)+c_0} e(n)\mathbf{x}(n)$$

else

$$\mathbf{w}(n) = \mathbf{w}(n-1)$$

This algorithm behaves much better than other SM based algorithms presented previously, in terms of steady state MSE, and complexity (lower UP), especially in impulsive noise environments.

2.5. The proposed SM Fast NLMS (SM-FNLMS) algorithm

We have proposed a new SM based algorithm called Set Membership Fast NLMS (SM-FNLMS) algorithm [40], which is an improvement of the FNLMS algorithm [26], presented in the subsection **1.11** in terms of convergence speed and update probability (UP). Since the FNLMS algorithm has a step size μ_{FNLMS} similar to the conventional NLMS algorithm, we proposed to incorporate the SMF into the FNLMS algorithm, by the same way as SM-NLMS algorithm presented in **Subsection 2.2**, a variable step size is introduced in the update part of the new proposed algorithm. The proposed algorithm is demonstrated in Table 2.4.

Table 2.4. The proposed SM-FNLMS algorithm

Initialization parameters :

$\mathbf{w}(0) = \tilde{\mathbf{c}}(0) = 0, \gamma(0) = 1, r_1(0) = 0,$

$\alpha(0) = r_0(0) = E_0 = 1$, where E_0 is an initialization

$d(n) = y(n) + v(n);$

For each instant of time n=1,2...

Prediction error:

$$r_1(n) = \lambda_a r_1(n-1) + x(n)x(n-1)$$

$$r_0(n) = \lambda_a r_0(n-1) + x^2(n)$$

$$a(n) = \frac{r_1(n)}{r_0(n) + c_a}$$

$$\varepsilon(n) = x(n) - a(n)x(n-1)$$

$$\alpha(n) = \lambda\alpha(n-1) + \varepsilon^2(n)$$

Adaptation gain:

$$\begin{bmatrix} \tilde{\mathbf{c}}(n) \\ c(n) \end{bmatrix} = \begin{bmatrix} \dfrac{\varepsilon(n)}{\lambda\alpha(n-1) + c_0} \\ \tilde{\mathbf{c}}(n-1) \end{bmatrix}$$

$$\delta(n) = -c(n)x(n-L) + \frac{x(n)\varepsilon(n)}{\lambda\alpha(n-1) + c_0}$$

$$\gamma(n) = \frac{\gamma(n-1)}{1 + \gamma(n-1)\delta(n)}$$

Filtering Part:

$$e(n) = d(n) - \mathbf{w}^T(n-1)\mathbf{x}(n)$$

$$\mu(n) = \begin{cases} 1 - \dfrac{\zeta}{|e(n)|} & if\ |e(n)| > \zeta \\ 0 & otherwise \end{cases}$$

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \mu_{FNLMS}(n)e(n)\gamma(n)\tilde{\mathbf{c}}(n)$$

2.5.1 Performance evaluation of the proposed SM-FNLMS algorithm

In the part of performance evaluation of all proposed algorithms in this thesis, we use three different types of far-end input signal $x(n)$. All these signals are sampled at a rate of 16 kHz, with 16 bits resolution on amplitude and normalized to one in absolute value. The first one (figure.2.2-a), noted WGN-AR20 is a stationary White Gaussian Noise filtered by an autoregressive model of order 20. In order to show the convergence, re-convergence speed and the tracking ability, we use the well-known, in the field of acoustic echo, the stationary USASI signal (United State of America Standards Institute now ANSI) (figure.2.2-b). This latter has a spectrum close to the average spectrum of the speech with a spectral range of 32 dB.

The third input signal used in our simulation is a real speech signal, that is obtained by concatenation of two male and female sentences of 6.75s duration (figure.2.2-c). It is a non-stationary signal of an average power equals to 0.16.

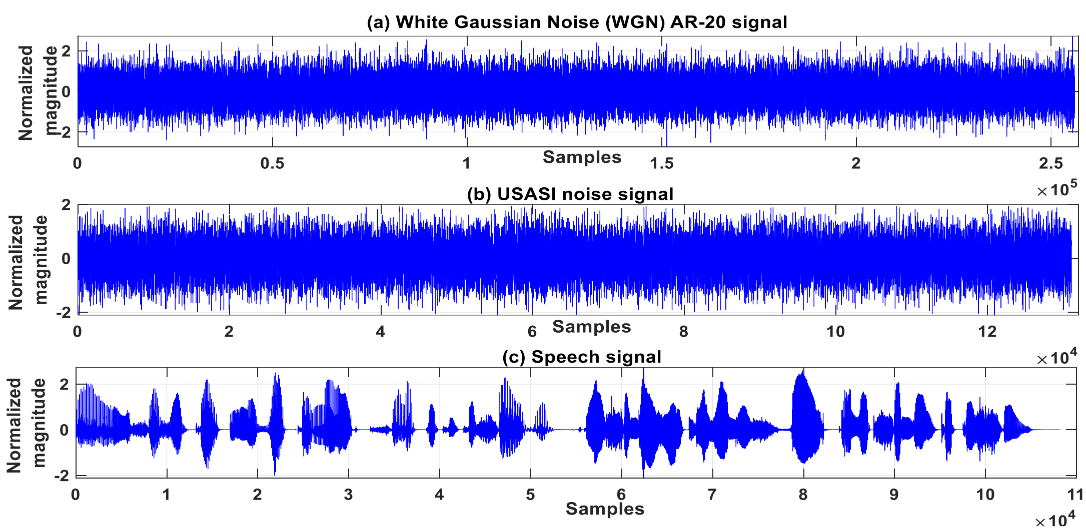

Figure.2.2. The used normalized input signals. (a) WGN AR-20 signal with a length of 256000 samples, (b) USASI noise signal with a length of 131072 samples, (c) Speech signal with a length of 108208 samples.

Figure 2.3 represents the impulse response used for the simulation of the proposed SM-FNLMS algorithm, which is a long impulse response of length $L = 1024$ measured in car cabin.

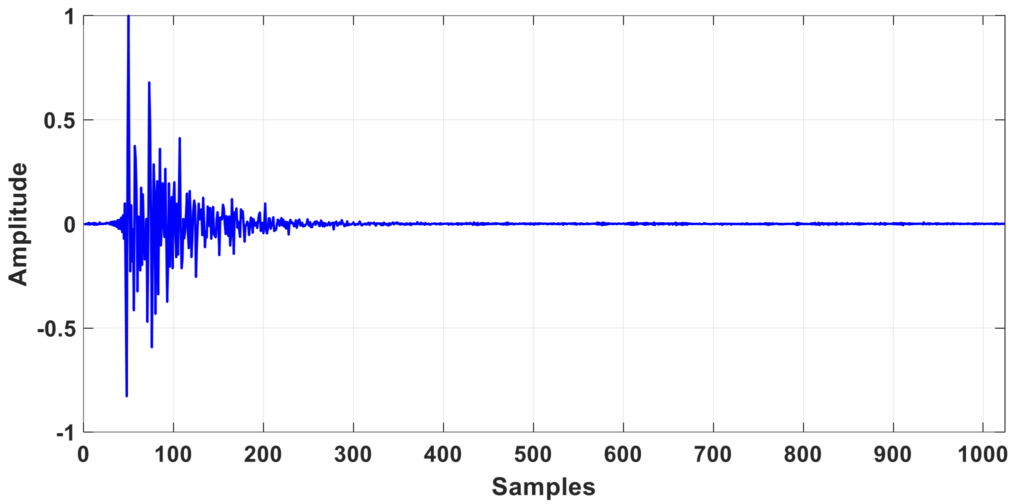

Figure.2.3. The used car impulse response of $L = 1024$.

The performance analysis is evaluated using the echo loss in terms of the mean square error (MSE). For the stationary input signals, we used the normalized mean square error (NMSE) given by [26]:

$$NMSE(dB) = 10\log_{10}\left(\frac{<e^2(n)>}{<d^2(n)>}\right) \quad (2.19).$$

where < . > denotes a time average over 256, 512 or 1024 samples according to the clarity of the depicted curves. For the speech and noisy signals, we evaluate the performance using the estimated MSE, defined by:

$$MSE(dB) = 10\log_{10}(< e^2(n) >) \quad (2.20).$$

Figures 2.4 and 2.5 show the NMSE curves of the four algorithms: NLMS, SM-NLMS, Fast NLMS and SM- FNLMS, using, respectively, the two input signals described previously in figure 2.2 (a and b). Figure 2.6 shows the MSE curves when the input signal is the speech signal represented in figure 2.2 (c). The step-size is taken μ= 0.6 for the NLMS and the FNLMS algorithm, the parameters λ and λ_a are set, respectively to 0.99 and 0.9985 for the FNLMS and the SM-FNLMS. For the SM-NLMS and the SM-FNLMS. In the case of the presence of additive noise, the error bound ζ is calculated according to [35] by the expression $\zeta = \sqrt{5\sigma_v^2}$.

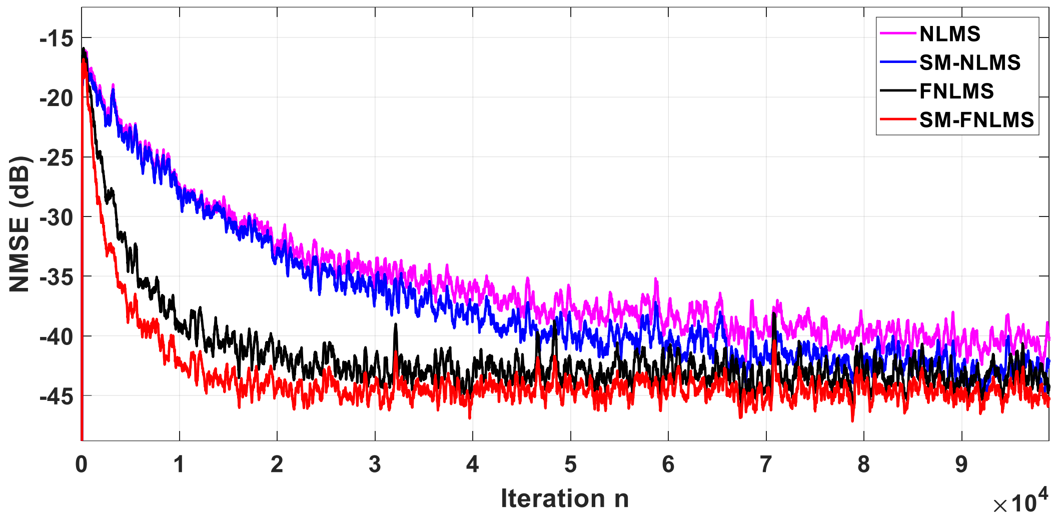

Figure.2.4. The Comparison of the initial convergence and the steady-state NMSE. Input: WGN-AR(20) signal with Car system of length $L = 1024$. SNR=50 dB.

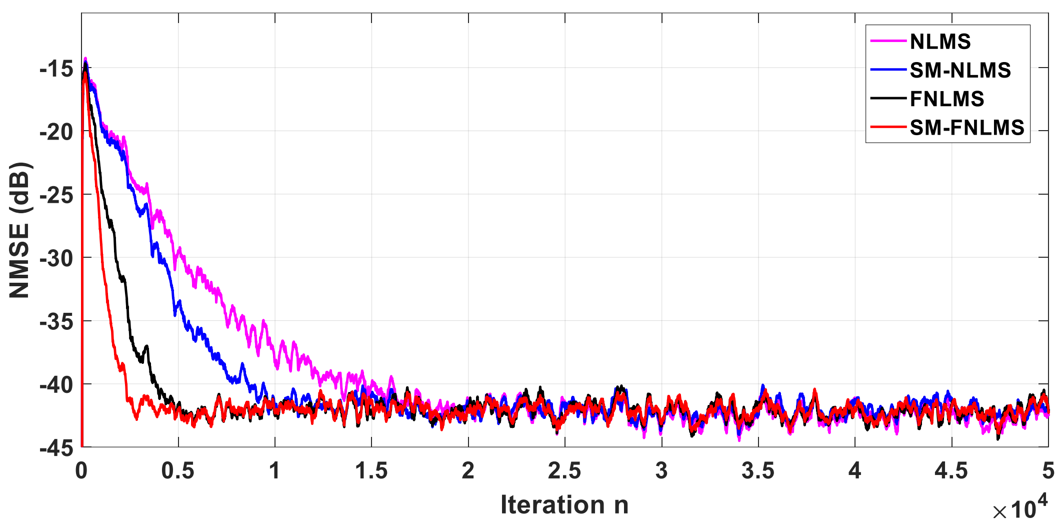

Figure.2.5. The Comparison of the initial convergence and the steady-state NMSE. Input: USASI signal with Car system of length $L = 1024$. SNR=50 dB.

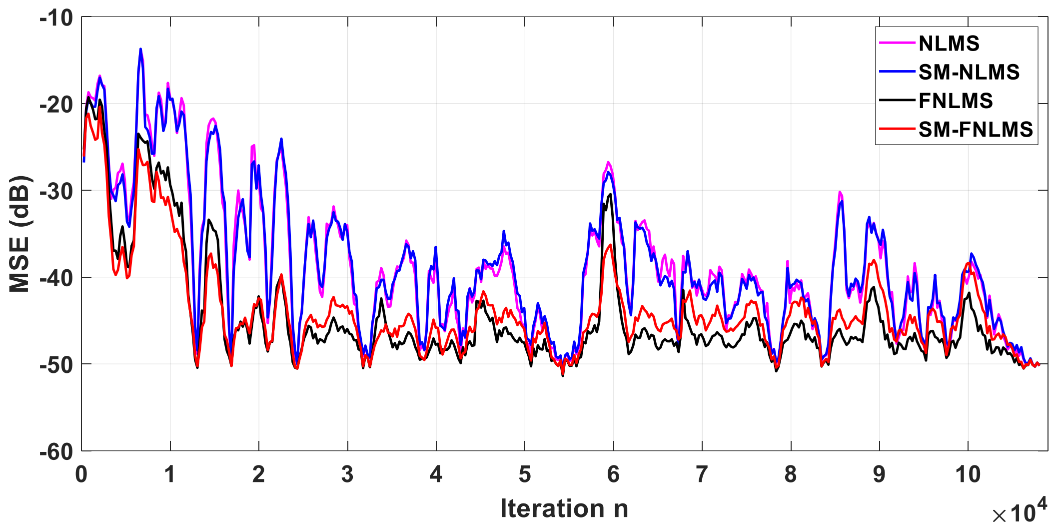

Figure.2.6. The Comparison of the initial convergence and the steady-state MSE. Input: speech signal with Car system of length $L = 1024$. SNR=50 dB.

As we can see, from figures 2.4, 2.5 and 2.6, the proposed SM-FNLMS algorithm has improved the convergence speed compared to the FNLMS, SM-NLMS and NLMS algorithms. Also using speech signal as input, the SM-FNLMS algorithm shows a superiority in the initial convergence, compared to other algorithms, so it is a good candidate in the AEC applications.

2.5.2 The effect of the error bound on the SM-FNLMS algorithm

We have studied by simulation the effect of the error bound ζ on the behavior of the SM-FNLMS algorithm. We have fixed the parameters as follows: $\lambda = 0.99$ and $\lambda_a = 0.9985$, figure 2.7 shows simulation results for four different decreasing values of the error bound 0.1, 0.05, 0.01 and 0.001, with USASI noise as input signal.

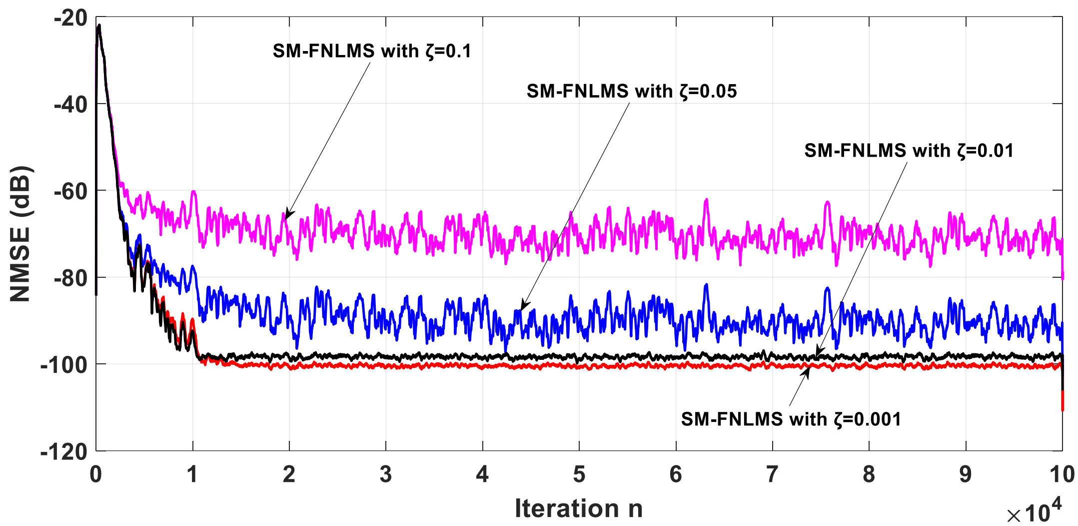

Figure.2.7. Learning curves of the SM-FNLMS, error bound ζ effect.

We find that when ζ decreases from 0.1 to 0.01, the steady state NMSE is reduced. However, the steady state NMSE stops decreasing when $\zeta = 0.01$ because the performances of $\zeta = 0.01$ and $\zeta = 0.001$ are almost the same.

2.6 The proposed Improved SM-FNLMS (ISM-FNLMS) algorithm

In this subsection, we present another proposed algorithm called ISM-FNLMS [41], this novel algorithm is obtained following two steps, summarized as follows:

- Extension of the SMF concept to the FNLMS algorithm, as done in the SM-FNLMS, to obtain a superior performance with a reduction in the overall computational complexity compared with the original FNLMS algorithm.
- A recursive estimation of the absolute output error is proposed instead of using the absolute output error to compute the step size in the SM-FNLMS algorithm proposed in [40], which results to a good tracking capability in case of acoustic channel variations.

2.6.1 The derivation of the proposed ISM-FNLMS algorithm

We assume that the input signal is a white Gaussian stationary signal and consider that all recursive variables of the FNLMS algorithm have reached their true asymptotic values. In particular [42], we replace the following slowing quantities by their asymptotic values:

$$\alpha(n) \approx \frac{\sigma_x^2}{1-\lambda} \tag{2.21.a}$$

$$\tilde{\mathbf{c}}(n) \approx \frac{\mathbf{x}(n)}{\frac{\lambda}{1-\lambda}\sigma_x^2 + c_0} \tag{2.21.b}$$

$$\gamma(n) = \frac{1}{1+\tilde{\mathbf{c}}^T(n)\mathbf{x}(n)} \approx \frac{1}{1+\frac{L\sigma_x^2}{\frac{\lambda\sigma_x^2}{1-\lambda}+c_0}} \tag{2.21.c}$$

where $\sigma_x^2 = E[x^2(n)]$. Then, the adaptation gain for the FNLMS algorithm, using approximations, will be:

$$\mathbf{g}(n) = \gamma(n)\tilde{\mathbf{c}}(n) \approx \frac{1}{L\sigma_x^2(1+\frac{\lambda}{(1-\lambda)L}+\frac{c_0}{L\sigma_x^2})}\mathbf{x}(n) \qquad (2.22).$$

The a posteriori error $\epsilon(n)$ can be calculated, as follows:

$$\epsilon(n) = e(n) + \mu(n)e(n)\gamma(n)\tilde{\mathbf{c}}^T(n)\mathbf{x}(n) \qquad (2.23).$$

According to equation 2.22, the term $\gamma(n)\tilde{\mathbf{c}}^T(n)\mathbf{x}(n)$ will be approximately:

$$\gamma(n)\tilde{\mathbf{c}}^T(n)\mathbf{x}(n) \approx \frac{1}{L\sigma_x^2\left(1+\frac{\lambda}{(1-\lambda)L}+\frac{c_0}{L\sigma_x^2}\right)}\mathbf{x}^T(n)\mathbf{x}(n) \approx 1 \qquad (2.24).$$

Using set-membership update conditions equations 2.3, 2.4 and 2.8, the variable step size for the ISM-FNLMS, becomes similar to that of the SM-NLMS algorithm of equation 2.10. In addition, and in order to get an efficient tracking capability, we suggest to replace $|e(n)|$ in the relation equation 2.10 with their own estimates, according to the following recursive formula:

$$\sigma_e(n) = \beta\sigma_e(n-1) + (1-\beta)|e(n)| \qquad (2.25).$$

where β is a forgetting factor, and $\sigma_e(n)$ is initialized with a value closed to σ_x. Therefore, the variable step size $\mu(n)$ is changed as:

$$\mu(n) = \begin{cases} 1 - \frac{\zeta}{\sigma_e(n)} & if\ |e(n)| > \zeta \\ 0 & otherwise \end{cases} \qquad (2.26).$$

Then, the update equation of the proposed algorithm can be written as:

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \mu_{ISM}(n)\gamma(n)\tilde{\mathbf{c}}(n)e(n) \tag{2.27}$$

The proposed ISM-FNLMS algorithm is demonstrated in Table 2.5:

Table 2.5. The proposed ISM-FNLMS algorithm

Initialization parameters :

$\mathbf{w}(0) = \tilde{\mathbf{c}}(0) = 0, \gamma(0) = 1, r_1(0) = 0$, error bound ζ: small constant

$\alpha(0) = r_0(0) = E_0 = 1$, where E_0 is an initialization

$d(n) = y(n) + v(n)$;

For each instant of time n=1,2...

Prediction error:

$$r_1(n) = \lambda_a r_1(n-1) + x(n)x(n-1)$$

$$r_0(n) = \lambda_a r_0(n-1) + x^2(n)$$

$$a(n) = \frac{r_1(n)}{r_0(n) + c_a}$$

$$\varepsilon(n) = x(n) - a(n)x(n-1)$$

$$\alpha(n) = \lambda\alpha(n-1) + \varepsilon^2(n)$$

Adaptation gain:

$$\begin{bmatrix} \tilde{\mathbf{c}}(n) \\ c(n) \end{bmatrix} = \begin{bmatrix} \dfrac{\varepsilon(n)}{\lambda\alpha(n-1) + c_0} \\ \tilde{\mathbf{c}}(n-1) \end{bmatrix}$$

$$\delta(n) = -c(n)x(n-L) + \frac{x(n)\varepsilon(n)}{\lambda\alpha(n-1) + c_0}$$

$$\gamma(n) = \frac{\gamma(n-1)}{1 + \gamma(n-1)\delta(n)}$$

Filtering Part:

$$e(n) = d(n) - \mathbf{w}^T(n-1)\mathbf{x}(n)$$

$$\sigma_e(n) = \beta\sigma_e(n-1) + (1-\beta)|e(n)|$$

$$\mu_{ISM}(n) = \begin{cases} 1 - \dfrac{\zeta}{\sigma_e(n)} & if\ |e(n)| > \zeta \\ 0 & otherwise \end{cases}$$

$\mathbf{w}(n) = \mathbf{w}(n-1) + \mu_{ISM}(n)e(n)\gamma(n)\tilde{\mathbf{c}}(n)$

The maximal computational complexity of the ISM-FNLMS algorithm is about $2L$. According to the filtering part and the error bound condition of this algorithm, we can well notice that the computational complexity in the overall adaptation is reduced compared to FNLMS algorithm since the filter coefficients are updated only when the filtering error is higher than a pre-determined threshold ζ. This complexity reduction depends on the value of the error bound threshold, which itself must be chosen according to the noise level in the signal.

2.6.2 Comparative performances for stationary input signal

As a first step, we evaluate the performances of the proposed ISM-FNLMS algorithm compared with the NLMS, SM-NLMS and FNLMS algorithms, using the two stationary input signals, WGN-AR20 and USASI noise, presented previously in figure 2.2-a and 2.2-b, respectively. Two real acoustic impulse responses, the first one is a short impulse response measured in car and truncated to $L = 256$ (figure 2.8-a) and the second is a long impulse response measured in real video-conference room (denoted VCN) and truncated to $L = 1024$ (figure 2.8-b) are used as echo path systems.

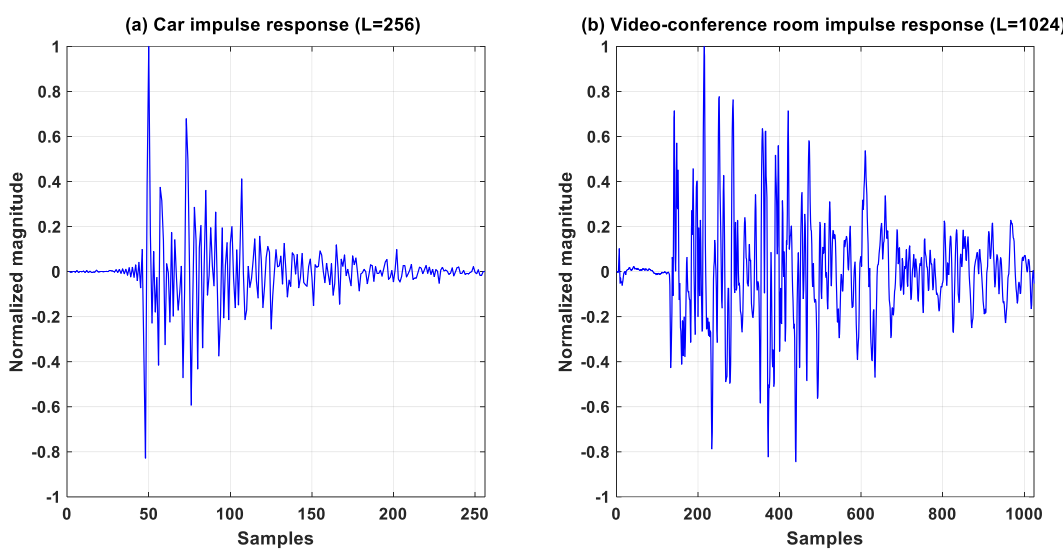

Figure.2.8. The used real acoustic impulse responses.

The constants c_0 and c_a are regularization constants. The initial variances $r_0(0)$ and $\alpha(0)$ are close, in the steady-state, to the values $\sigma_x^2/(1-\lambda)$ or $\sigma_x^2/(1-\lambda_a)$, where forgetting factors have values close to one. For this reason, all these constants are initialized to one. The initial value of $\sigma_e(0)$ is set to $\sigma_x/10$, in order to avoid initial numerical problems.

The parameters λ_a and β are forgetting factors used to track the non-stationarity of the input signal $x(n)$. They are evaluated over a period of 25ms (i.e a window of length of 400 samples) where the speech signal can be considered as stationary: $\lambda_a = \beta = 1 - 1/400 = 0.9975$. The forgetting factor λ is used to track the variations of the unknown system, as shown in [24], [26], a wide range of values are possible for this parameter. We set $\lambda = 0.99$ in order to have good performances in terms of convergence speed and tracking ability for the proposed algorithm. Fixed step-sizes μ and μ_{FNLMS} are set to 0.6 for all simulations. This choice is motivated in order to have approximately the same steady-state error for all tested algorithms in the stationary case.

The error bound value ζ is an important parameter for overall performances and for the computational complexity reduction of SM based algorithms. The variance of additive noise σ_v^2 is a good indicator for the choice of the error bound. We have set ζ to values close to σ_v when the noise is present, and we take $\zeta = 2 \times 10^{-5}$ when there is no output additive noise.

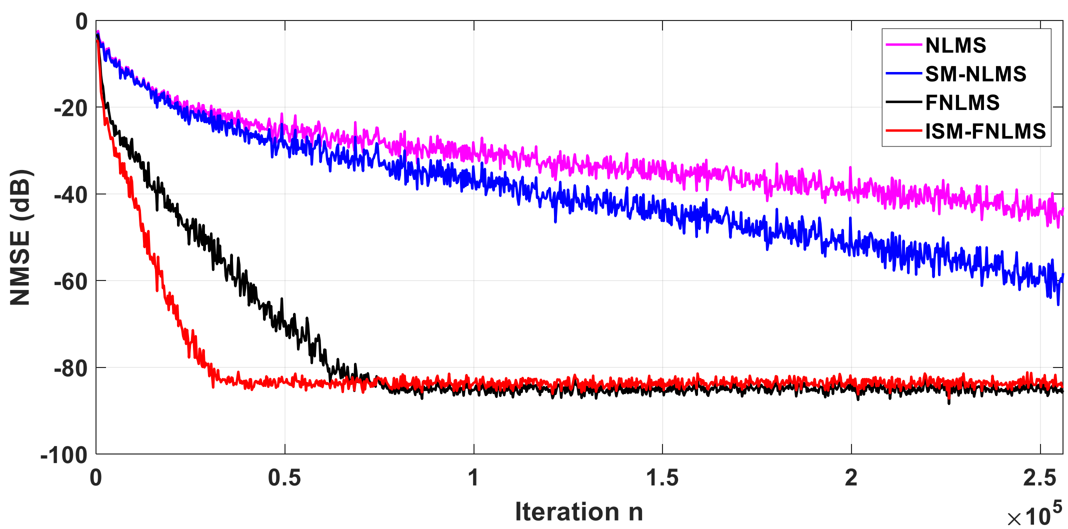

Figure.2.9. Comparison of the initial convergence and the SS-NMSE. Input: WGN-AR(20) signal with Car system of length $L = 256$. No output noise.

Figures 2.9 and 2.10 show the results for two cases $L = 256$ and $L = 1024$, respectively. According to these results, we note that the proposed ISM-FNLMS algorithm has a better convergence speed, especially for the filter size $L = 256$. For the case $L = 1024$, the convergence of the ISM-FNLMS algorithm is slightly better than that of the FNLMS, this latter performance is obtained with a reduced computational complexity as it will be shown later. The ISM-FNLMS and FNLMS algorithms have almost identical SS-NMSE. The NLMS algorithm and its SM version have a very slow convergence rate and cannot reach the steady-state for about 256,000 iterations. This behavior is due essentially to the nature of the input signal WGN-AR20 which has a high spectral range.

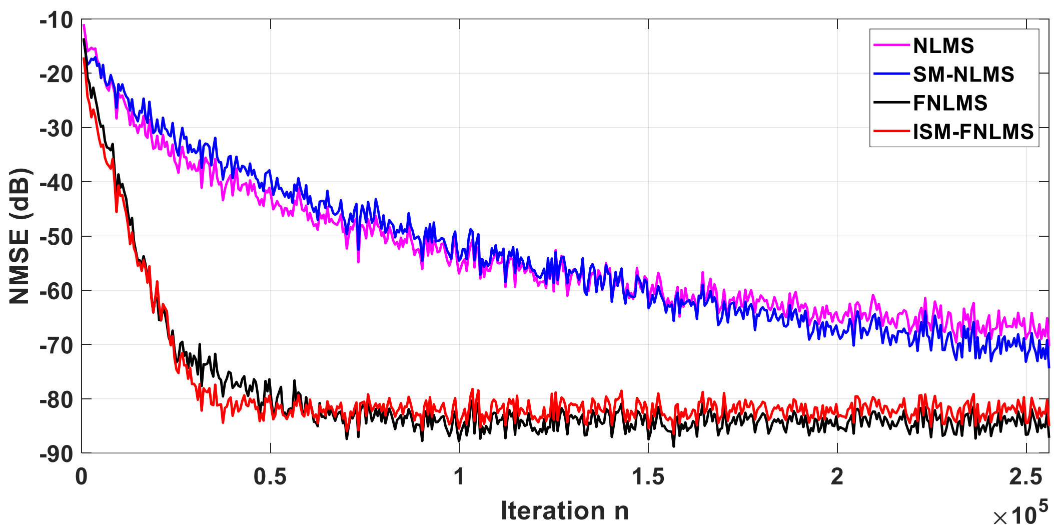

Figure.2.10. Comparison of the initial convergence and the SS-NMSE. Input: WGN-AR(20) signal with VCN system of length $L = 1024$. No output noise.

Figure.2.11 shows the results obtained with USASI noise filtered with 256-taps car impulse response. The USASI noise has a lower spectral range than WGN-AR20 noise, so we can see that all the tested algorithms reach the same steady-state NMSE. We also observe that the proposed algorithm has a better convergence speed.

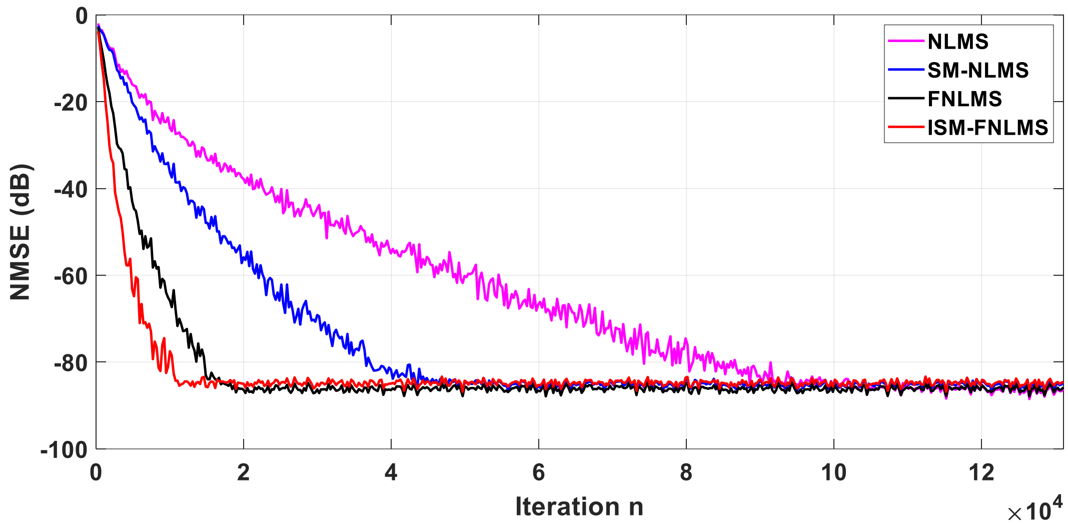

Figure.2.11. Comparison of the initial convergence and the SS-NMSE. Input: USASI signal with Car system of length $L = 256$. No output noise.

2.6.3 Comparative performances with noisy signals

Figures 2.12, 2.13 and 2.14 compare the simulation results for noisy signals with three different Signal to Noise Ratios (SNR) of 15 dB, 30 dB and 50 dB, respectively. It is clear that the most of the degradation in the NMSE is due to the level of the additive noise present on the desired signal. These results clearly show the robustness of the proposed algorithm. We can say that it is important to choose the error bound threshold depending on the level of noise in the signal. The standard deviation of the noise variance is a good indicator for the choice of ζ.

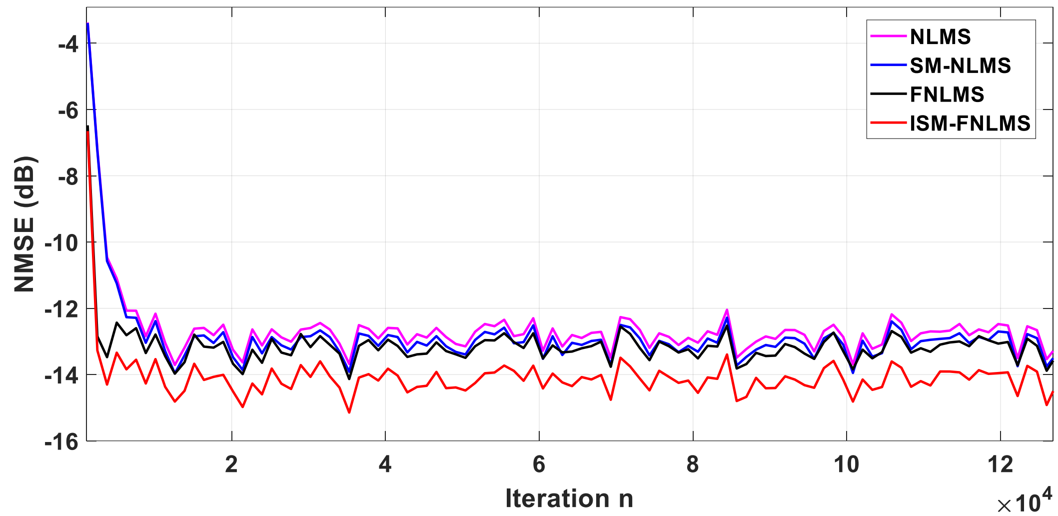

Figure.2.12. Comparison with noisy input: USASI signal with car system of length $L = 256$. Output noise: SNR=15 dB. Threshold $\zeta = 0.1$. $\sigma_y^2 = 0.38$.

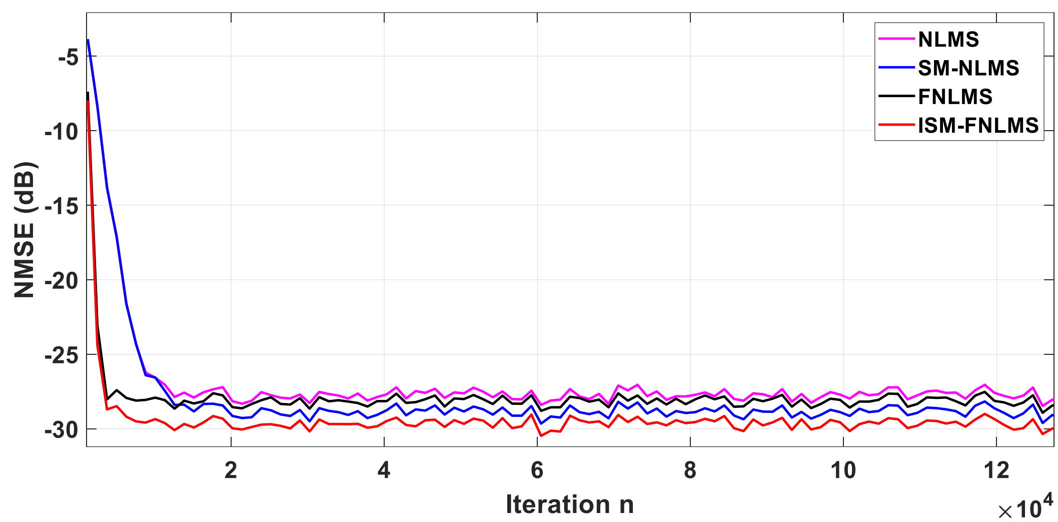

Figure.2.13. Comparison with noisy input: USASI signal with car system of length $L = 256$. Output noise: SNR=30 dB. Threshold $\zeta = 0.025$. $\sigma_y^2 = 0.38$.

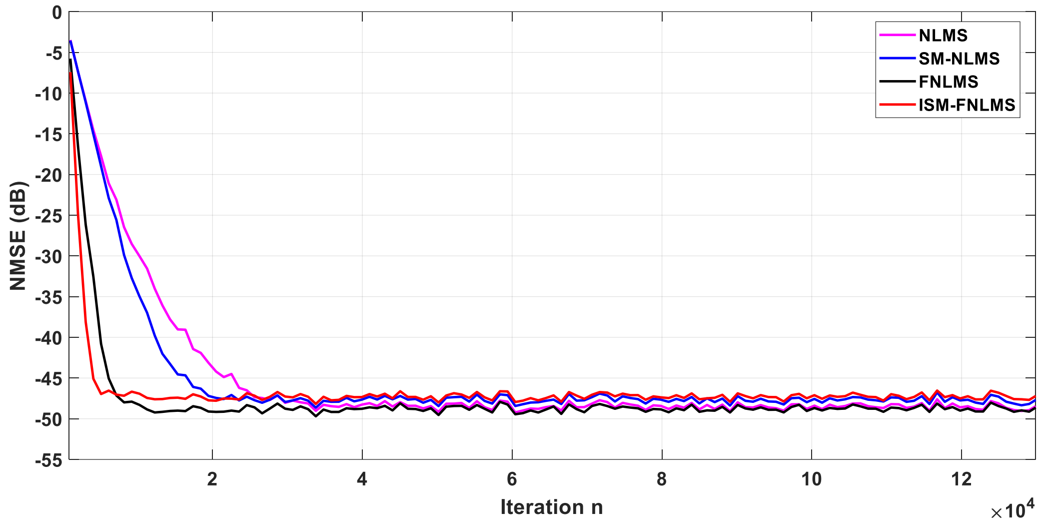

Figure.2.14. Comparison with noisy input: USASI signal with car system of length $L = 256$. Output noise: SNR=50 dB. Threshold $\zeta = 0.0032$. $\sigma_y^2 = 0.38$.

2.6.4 Comparative performances for non-stationary systems

We discuss in this paragraph one of the most important performance of an adaptive algorithm: its ability to track variations of the unknown system. First, we have tested the effect of a sudden change in the system. This abrupt change is obtained by multiplying the echo signal $y(n)$ by a constant factor equals to 1.75 in the steady-state at the middle. The results obtained in figures 2.15 and 2.16 show that the proposed algorithm has better initial convergence speed and re-convergence speed than the three other tested algorithms in both short and long real impulse responses.

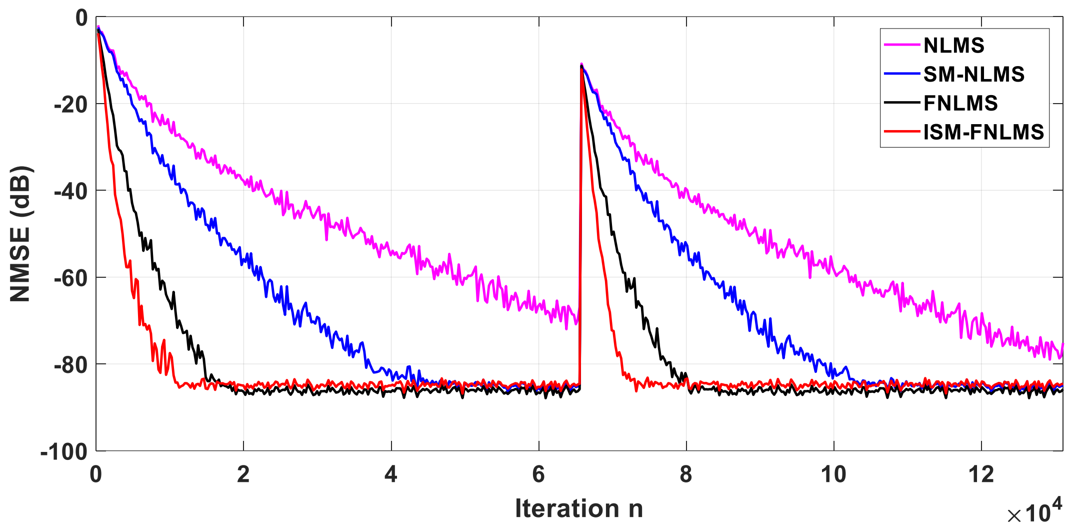

Figure.2.15. Comparison of the initial convergence and the re-convergence after an abrupt change in the impulse response of car system of length $L = 256$. Input: USASI noise. No output noise.

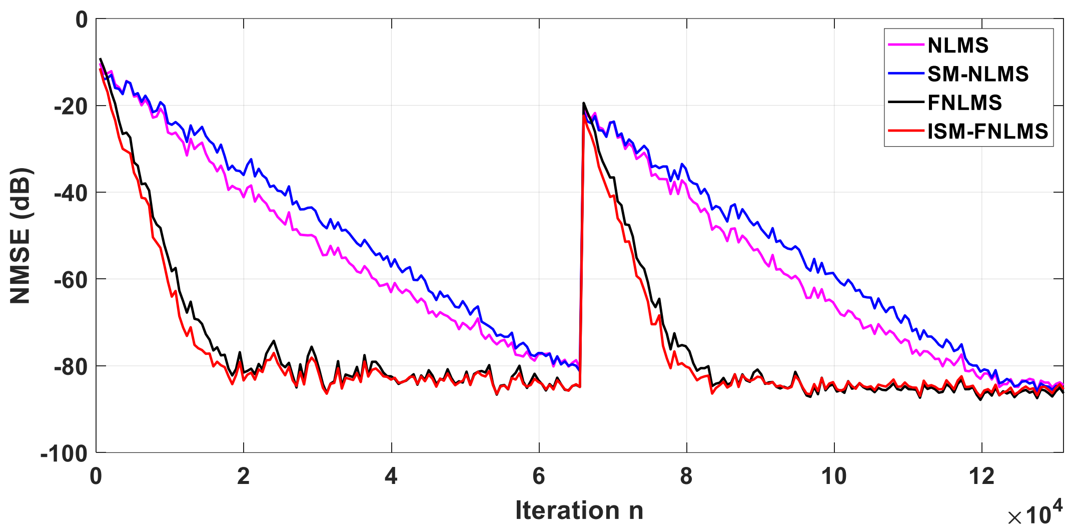

Figure.2.16. Comparison of the initial convergence and the re-convergence after an abrupt change in the impulse response of VCN system of length $L = 1024$. Input: USASI noise. No output noise.

In the second time, we considered a continuous variation of the system during a finite period of time duration between 51200 and 71680 samples of the total number of iterations. This artificial variation is obtained by multiplying the desired signal by a linear gain of amplitude between 1 and 2 represented in figure 2.17. Simulations carried out from figures 2.18 and 2.19, show that the proposed algorithm has better tracking capability performance than NLMS, SM-NLMS and FNLMS algorithms. During echo path variation, we note that there are gains in dB, for the proposed algorithm, of around 12 dB compared to SM-NLMS and 4 dB compared to FNLMS, when there is no output noise for the car system of length 256 (figure 1.18), also there are gains of around 7 dB compared to SM-NLMS and 4 dB compared to FNLMS for the long VCN system of length 1024 (figure 1.19).

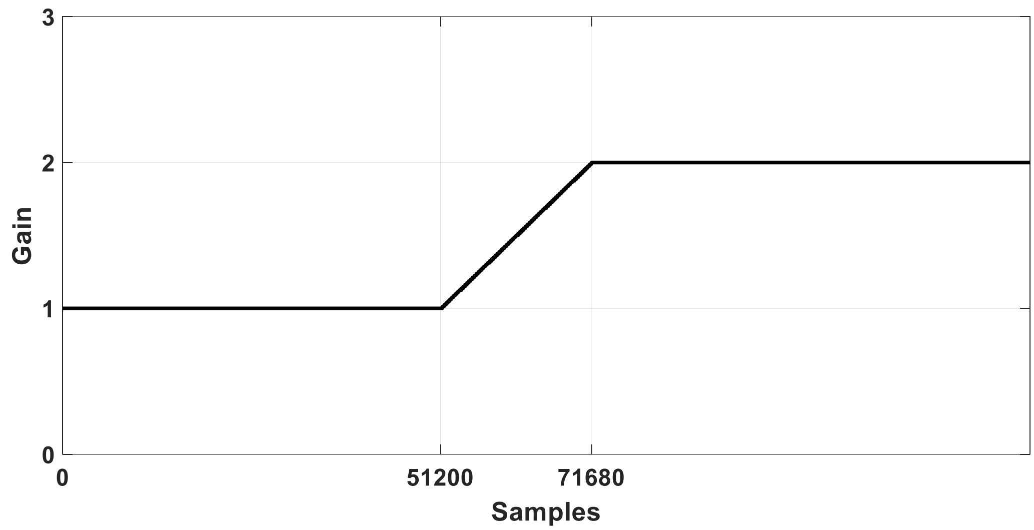

Figure.2.17. Variable artificial gain used for the tracking experiment. The variation is made between $n = 51200$ and $n = 71680$.

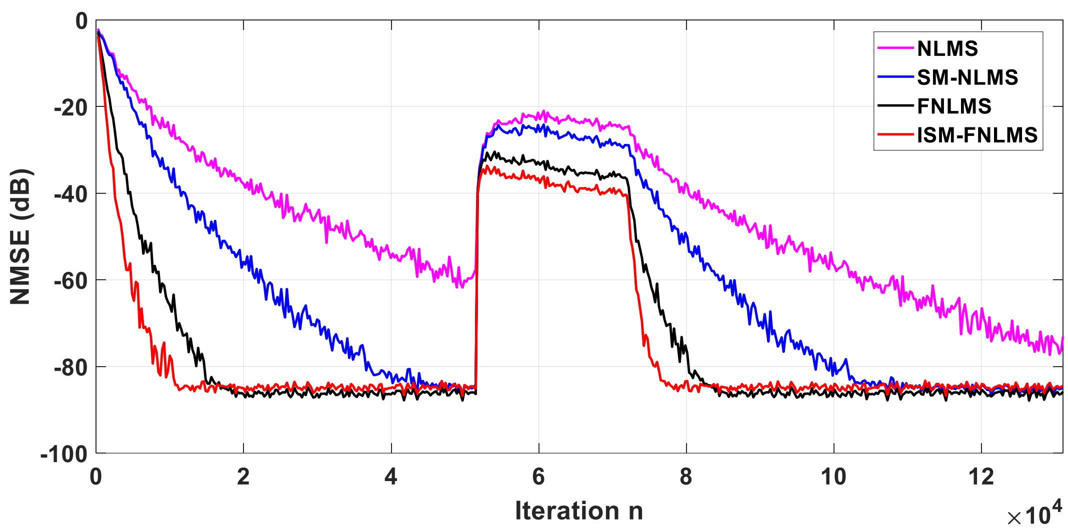

Figure.2.18. Comparison of tracking time-varying system of length $L = 256$. Input: USASI noise. No output noise.

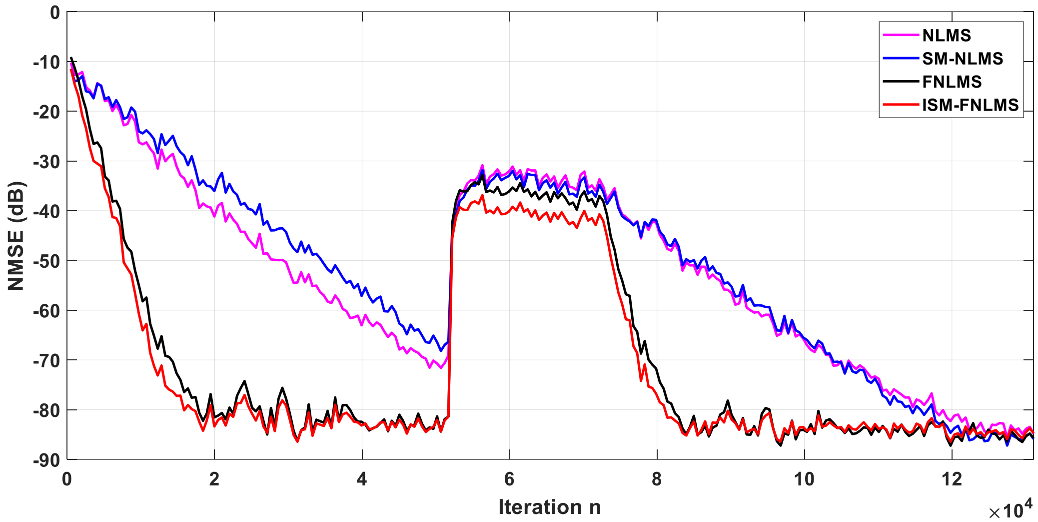

Figure.2.19. Comparison of tracking time-varying system of length $L = 1024$. Input: USASI noise. No output noise.

2.6.5 Comparative performances with speech inputs

In this subsection, two comparison experiments of the adaptive AEC based on the proposed ISM-FNLMS algorithm are carried out, using the speech signal presented in figure 2.2-c as input, filtered by the two acoustic impulse responses given in figure 2.8. For both experiments, we added a noise component with an SNR of $50\ dB$. Figures 2.20 and 2.21 show the results obtained for the car system of length $L = 256$ and the VCN system of length $L = 1024$, respectively. We have noted that the proposed ISM-FNLMS algorithm has a good AEC performance in transient phase and the steady-state regimes in comparison with the FNLMS, SM-NLMS and NLMS algorithms.

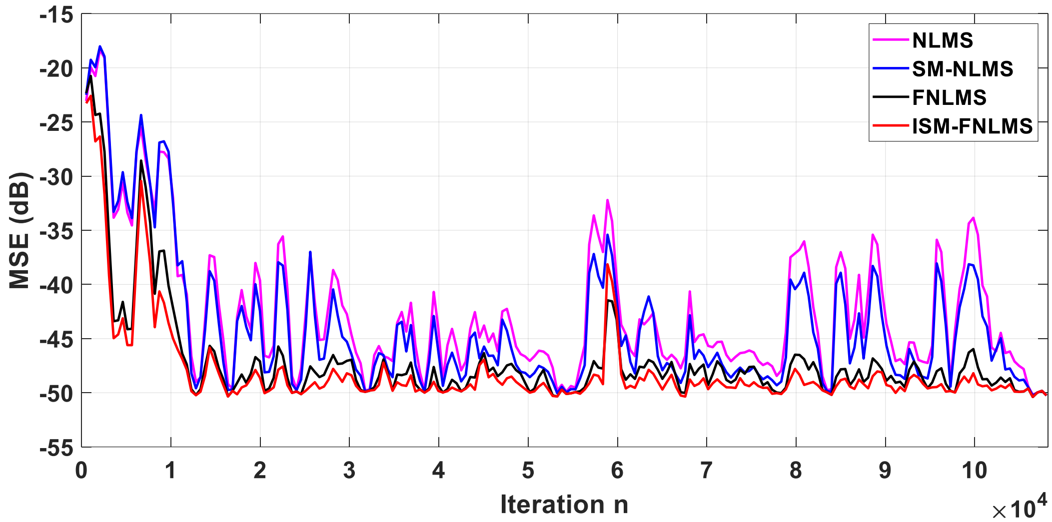

Figure.2.20. MSE evaluation Input: Speech signal with car system of length $L = 256$. Output noise: SNR=50dB.

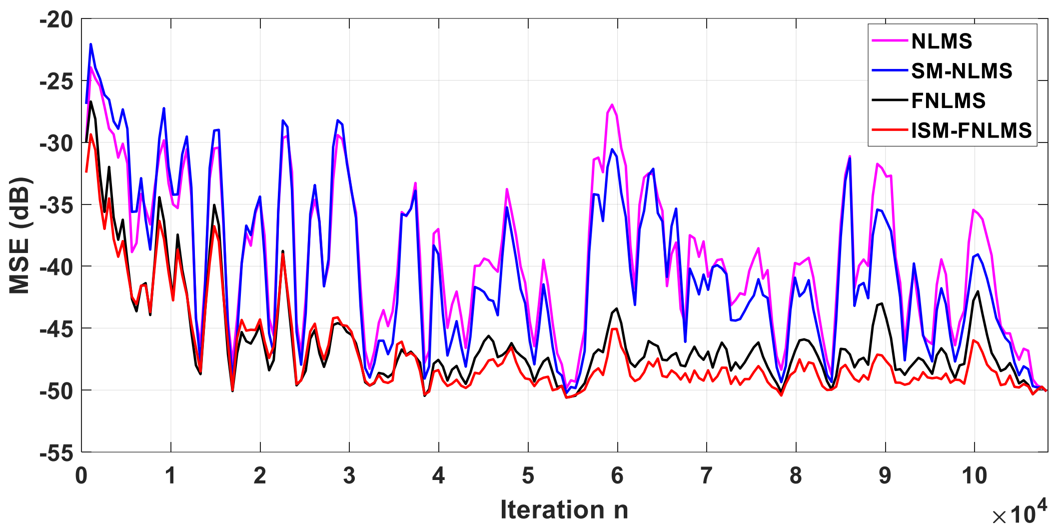

Figure.2.21. MSE evaluation Input: Speech signal with VCN system of length $L = 1024$. Output noise: SNR=50dB.

2.6.6 Computational complexity evaluation

In this subsection, we have evaluated the computational complexity of the proposed algorithm and we compare it with three versions of set-membership algorithms: SM-NLMS [35], SM-FNLMS

[40] and the SM-NLMS with Robust Error Bound (SMREB-NLMS) [39] algorithms. Parameters of SMREB-NLMS algorithm are set according to [39], as follows: $\mu_{REB} = 1, \upsilon_{REB} = 0.5, \beta_{REB} = 0.9985, \hat{\sigma}_0 = 5$. The convergence performance is evaluated with three different SNRs and using USASI noise as input filtered by acoustic impulse responses of figure 2.8. The results for SNR=30 dB are given in figure 2.22 and 2.23.

All the algorithms considered in this paper have a complexity of calculations (in number of multiplications only) of about $2L$. The four SM versions considered here have L multiplications for the filtering error plus a percentage of L for the adaptation part of the filter. This latter complexity is calculated by the ratio of the number of times the adaptive filter is updated on the total number of iterations. Thus, complexity reduction is evaluated experimentally. It is important to know that this performance depends strongly on the value of the error bound threshold which itself depends on the noise level in the used signal. A large value of the threshold leads to a significant reduction in complexity but introduces degradations on the convergence speed and the SS-MSE. The complexity results are summarized in tables 2.6 and 2.7 for filter sizes 256 and 1024 respectively. There is also an evaluation of the average SS-MSE (noted ST-MSE).

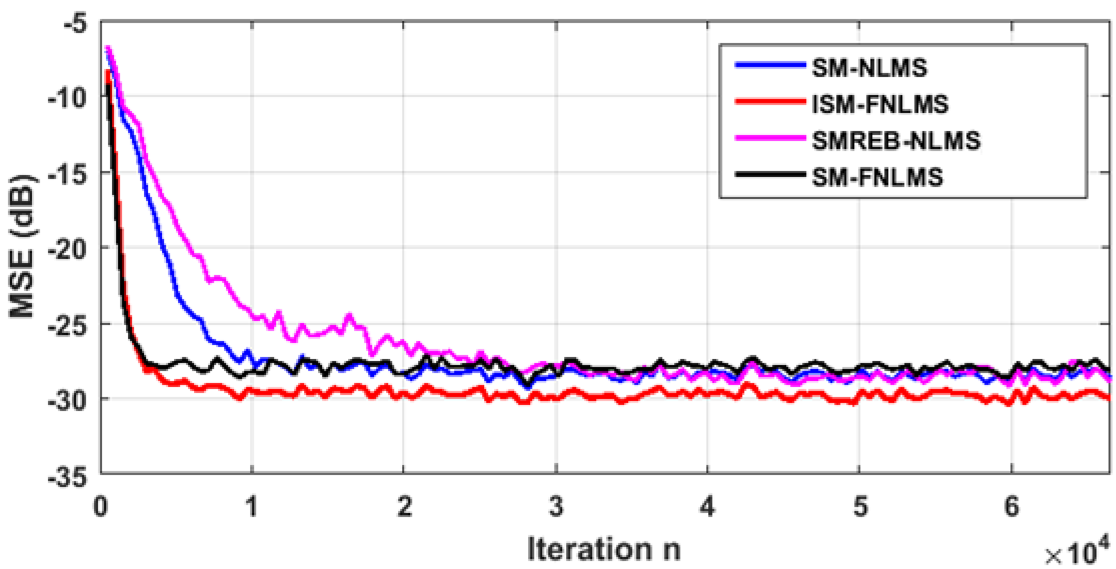

Figure.2.22. Comparison of SM algorithms with car system of length $L = 256$.Input USASI noise. Output noise: SNR=30dB. Threshold $\zeta = 0.025$.

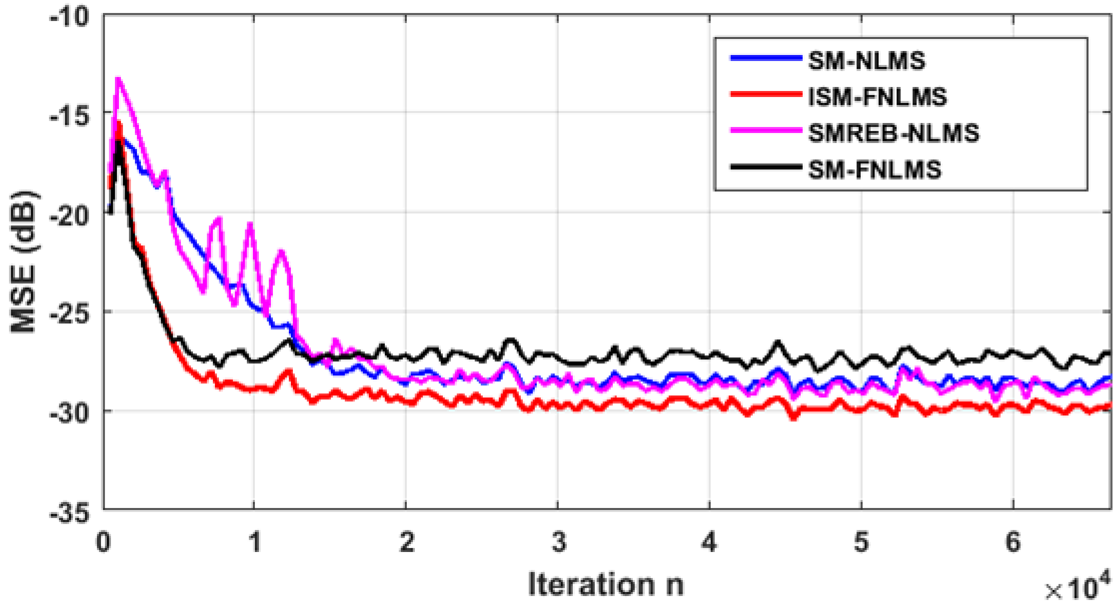

Figure.2.23. Comparison of SM algorithms with VCN system of length $L = 1024$.Input USASI noise. Output noise: SNR=30dB. Threshold $\zeta = 0.025$.

As we can see from these results, compared to the SM-FNLMS and SM-NLMS algorithms, the proposed algorithm gives the lowest complexity for the three considered SNRs. The complexity reduction is between 34% and 56% of L. The ST-MSE values remain acceptable depending on the noise level. Based on the results of tables 2.6 and 2.7, among the four studied SM versions, the SMREB-NLMS algorithm presents the best reduction in computational complexity, it reaches 76% of L for an SNR = 15 dB, but this interesting performance is paid by a degradation of the convergence speed and the ST- MSE (figures 2.22 and 2.23).

Table 2.6 : Computational complexity evaluation USASI Noise, CAR AIR, N=256, ($\sigma_y^2 = 0.38$, with no noise)

SNR (dB)	Error Bound ζ	ISM-FNLMS		SM-FNLMS		SM-NLMS		SMREB-NLMS	
		Complexity	ST-MSE (dB)	Complexity	ST-MSE (dB)	Complexity	ST-MSE (dB)	Complexity	ST-MSE (dB)
15	0.1	$L + 0.63L$	-13.8	$L + 0.66L$	-12.7	$L + 0.65L$	-13.2	$L + 0.28L$	-11.9
30	0.025	$L + 0.45L$	-29.80	$L + 0.53L$	-28.00	$L + 0.53L$	-28.50	$L + 0.28L$	-27.10
50	0.0032	$L + 0.44L$	-47.90	$L + 0.42L$	-48.10	$L + 0.45L$	-48.60	$L + 0.32L$	-46.8

Table 2.7: Computational complexity evaluation USASI Noise, VCN AIR, N=1024, ($\sigma_y^2 = 0.43$, with no noise)

SNR (dB)	Error Bound ζ	ISM-FNLMS		SM-FNLMS		SM-NLMS		SMREB-NLMS	
		Complexity	ST-MSE (dB)	Complexity	ST-MSE (dB)	Complexity	ST-MSE (dB)	Complexity	ST-MSE (dB)
15	0.098	$L + 0.66L$	-13.10	$L + 0.70L$	-11.70	$L + 0.66L$	-13.10	$L + 0.24L$	-13.8
30	0.03	$L + 0.44L$	-28.5	$L + 0.48L$	-27.5	$L + 0.45L$	-28.7	$L + 0.25L$	-27.8
50	0.0025	$L + 0.48L$	-49.70	$L + 0.58L$	-47.20	$L + 0.59L$	-48.04	$L + 0.31L$	-48.70

Results of figures 2.22 and 2.23 confirm the superiority in convergence speed and in steady-state NMSE of the proposed algorithm over the considered SM versions in this chapter.

Finally, we note that we can further reduce the computational complexity by choosing the right threshold and tolerating certain acceptable levels of performance degradation; there is a trade-off between good performances and reduced complexity.

2.7 Conclusion

Chapter 2 describes one of the most popular technique to reduce the complexity which is SM filtering, and how it was applied to NLMS algorithms with some modifications in the error bound, such as SMAEB-NLMS and SMREB-NLMS algorithms to increase the convergence rate with a lower computational cost. We took advantage of the SM technique on the FNLMS algorithm to increase the convergence speed with lower computational complexity, then we have proposed a novel SM based algorithm (ISM-FNLMS), this algorithm is robust for AEC applications in terms of convergence speed, computational complexity and tracking ability, as has been demonstrated during this chapter.

The robustness of these algorithms has been proven in the AEC application during single-talk, but in the presence of the near-end signal (Double-talk), there is a risk of divergence of the adaptive filter and distorting the near-end. To overcome this problem, a DTD should be added in the AEC system to control the adaptation of the filter. The next chapter is dedicated to DTD methods and its implementation on our proposed algorithms.

Chapter 3:

Acoustic Echo Cancellation with Double-talk detection

3.1. Introduction

In some systems, it is ovious that both near-end and far-end talk simultaneously, this situation is known as Doule Talk (DT). The presence of DT in the AEC system may cause the adaptive filter to diverge, because the near-end speech can be interpreted by the adaptive filter as a large level uncorrelated noise. The soution to this problem is to add a Double-Talk Detector (DTD) that be able to freeze or slow down the adaptation process when the near-end is present. In this chapter, we present some DTD methods proposed in the literature, then we propose to use one of the existing DTD method on the proposed FNLMS and ISM-FNLMS algorithms.

3.2. Double-Talk Detection

3.2.1. General structure of DTD

The major problem of the AEC is that the performance of the adaptive filter can be degraded on periods of DT, because the system estimation process fail to update the coefficients and produce extremely erroneous results. The task of DTD is to halt the adaptation process in case of near-end speech present to avoid the divergence of the adaptive algorithm [43]. Figure 3.1 shows a general structure of the AEC system controled by DTD [44].

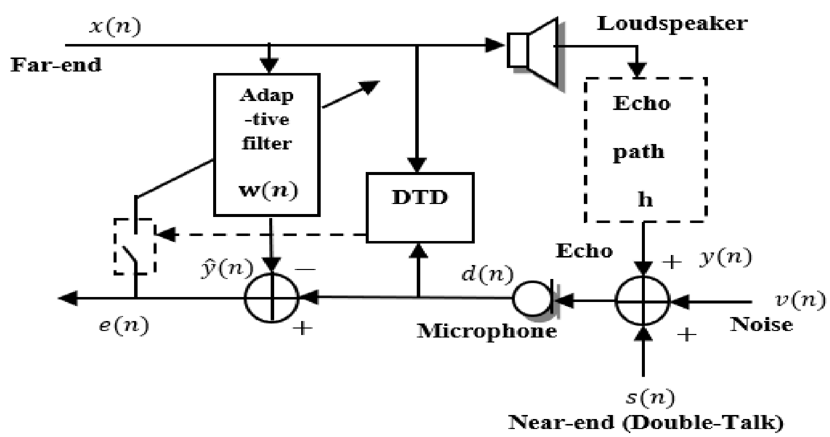

Figure.3.1. General structure of the AEC with DTD.

A robust AEC system should include a detector, that can differentiate between DT and single talk periods. In the case of DT, the near-end speech signal is considered by the acoustic echo canceller as a noise, which results to the fault of the adaptive filter, for that reason the only solution to this issue is to prevent the adaptive filter from divergence by freezing its parameters.

Several DTD methods have been proposed to overcome this problem, they can be divided into time domaine based DTD such as: Geigel algorithm [45] that compares the amplitude between the far-end and near-end speech signals, methods of signal envelope and its fast version [46], [47], method of coherence function [48], Cross-Correlation (CC) and Normalized Cross-Correlation (NCC) methods [49]–[52], in the other hand frequency-domain based DTD have been investigated such as: method based on Gaussian Mixture Model (GMM) [53], spectral analysis [54], [55]. A method based on the Holder inequality is proposed in [56], DTD based on singular value decomposition of the far-end signal [57], speech features extraction [58] and DTD based on joint energy and cross-correlation estimation [59].

Typically, the DTD calculates a variable decision $\xi(n)$, and DT is declared if it is lower than a given threshold value T [60], the optimum variable decision $\xi(n)$ for DTD will behaves as follows:

- ✓ If $s(n) = 0$ (Double-talk is not present), $\xi(n) \geq T$.
- ✓ If $s(n) \neq 0$ (Double-talk is present), $\xi(n) < T$.

The control of the adaptive filter by DTD is defined as:

$$Decision = \begin{cases} \xi(n) \geq T\ , DTD = 0, & update\ filter\ coefficients \\ \xi(n) < T\ , DTD = 1, & freeze\ adaptation \end{cases}$$

3.2.2. The Geigel algorithm

It is a very primitive algorithm used for DTD invented by Geigel in 1977 [61], in this algorithm the amplitude of near-end signal and the maximal amplitude of far-end signal (i.e L recent samples of $x(n)$) are compared to obtain the variable decision, calculated by the following mathematical expression:

$$\xi_G(n) = \frac{\max\{|x(n)|, \ldots, |x(n-L+1)|\}}{|d(n)|} \qquad (3.1).$$

The determination of DT is done by comparing the decision variable with a positive predefined threshold T_G, if $\xi_G(n) < T_G$, DT is present, otherwise there is no DT.

The Geigel DTD is a low complexity algorithm, however it is unstable and not suitable in the case of acoustic channel variation.

3.2.3. The cross-correlation algorithm

The original idea comes from Ye and Wu in [62] by using cross-correlation vector between the far-end signal $\mathbf{x}(n)$ and the error signal $e(n)$ for DTD which is given as:

$$\mathbf{p}_{ex}(n) = E\{e(n)\mathbf{x}^T(n)\} \quad (3.2).$$

where $\mathbf{p}_{ex}(n)$ is the cross-correlation vector between far-end and error signal.

But Benesty in [49] worked on this with different approach and he claimed that the above approach does not work well for DTD. He mentioned that both near-end speech $s(n)$ and the far-end speech signal $x(n)$ are independent and assume that all the signals are zero mean.

According to him, the cross-correlation $\mathbf{p}_{xd}(n)$ between far-end signal and microphone signal will be used to calculate the decision statistic.

$$\mathbf{p}_{xd}(n) = E\{\mathbf{x}(n)d^T(n)\} = E\left\{\mathbf{x}(n)\big(y(n) + s(n)\big)^T\right\} = E\{\mathbf{x}(n)\big(\mathbf{h}^T\mathbf{x}(n)\big)^T\} = \mathbf{R}_{xx}(n)\mathbf{h} \quad (3.3).$$

where $\mathbf{R}_{xx}(n)$ is the autocorrelation matrix of the far-end signal.

The variance of the microphone signal $\sigma_d^2(n)$ is:

$$\begin{aligned}
\sigma_d^2(n) = \ E\{d(n)d^T(n)\} &= E\{[y(n) + s(n)][y(n) + s(n)]^T\} \\
&= E\{y(n)y^T(n)\} + E\{s(n)s^T(n)\} \\
&= E\{[\mathbf{h}^T\mathbf{x}(n)][\mathbf{h}^T\mathbf{x}(n)]^T\} + \sigma_s^2(n) \\
&= \mathbf{h}^T\mathbf{R}_{xx}(n)\mathbf{h} + \sigma_s^2(n) = \mathbf{p}_{xd}{}^T(n)\mathbf{R}_{xx}{}^{-1}(n)\mathbf{p}_{xd}(n) + \sigma_s^2(n) \quad (3.4).
\end{aligned}$$

where $\sigma_s^2(n)$ is the variance of the near-end speech.

and for $s(n) = 0$, we have:

$$\sigma_d^2(n) = \mathbf{p}_{xd}{}^T(n)\mathbf{R}_{xx}{}^{-1}(n)\mathbf{p}_{xd}(n) \quad (3.5).$$

Dividing 3.5 by $\sigma_d^2(n)$ and taking the square root, then using 3.3 and 3.4, the decision variable of CC can be written as [49]:

$$\xi_{CC}(n) = \sqrt{\mathbf{p}_{xd}{}^{T}(\sigma_d^2(n)\mathbf{R}_{xx}(n))^{-1}\mathbf{p}_{xd}(n)} = \frac{\sqrt{\mathbf{h}^T\mathbf{R}_{xx}(n)\mathbf{h}}}{\sqrt{\mathbf{h}^T\mathbf{R}_{xx}(n)\mathbf{h}+\sigma_s^2(n)}} \qquad (3.6).$$

Therefore, from the Equation 3.6, it can be easily seen that:

- ✓ If near-end speech is not present ($s(n) = 0$), then $\xi_{CC}(n) \approx 1$
- ✓ If near-end speech is present ($s(n) \neq 0$), then $\xi_{CC}(n) < 1$

Thus, finally we get the CC based DTD as:

$$Decision = \begin{cases} \xi_{CC}(n) < T & DT\ is\ declared \\ \xi_{CC}(n) \geq T & no\ DT\ is\ declared \end{cases}$$

where T is threshold with a value approximately equal to 1.

3.2.4 The Normalized Cross-Correlation (NCC) algorithm

The NCC algorithm computes the decision variable depending on the relations of microphone signal $d(n)$ and error signal $e(n)$, known also as Microphone Error Cross-Correlation (MECC) and normalized by the variance of the microphone signal [49], [50], figure 3.2 represents a basic structure of the AEC with NCC-DTD.

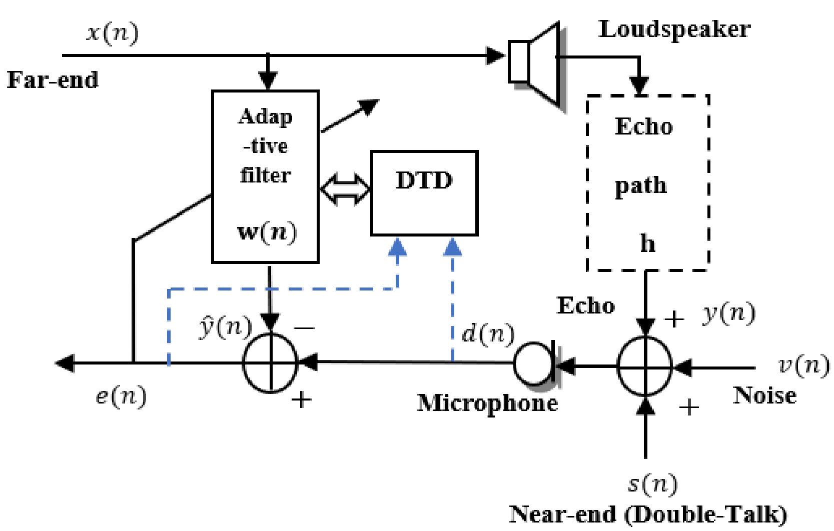

Figure.3.2. Basic structure of the AEC with DTD based on NCC.

The decision variable of NCC based DTD is defined by:

$$\xi_{NCC}(n) = 1 - \frac{p_{de}(n)}{\sigma_d^2(n)} \tag{3.7}$$

where $p_{de}(n)$ is the cross-correlation between $d(n)$ and $e(n)$ and $\sigma_d^2(n)$ is the variance of $d(n)$.

Since the values of $p_{de}(n)$ and $\sigma_d^2(n)$ are not available in practice, the authors in [50] suggest to replace these values by its estimated values calculated using the exponential recursive weighting formula, as follows:

$$\hat{p}_{ed}(n) = \lambda_{NCC}\hat{p}_{ed}(n-1) + (1-\lambda_{NCC})e(n)d(n) \tag{3.8}$$

$$\hat{\sigma}_d^2(n) = \lambda_{NCC}\hat{\sigma}_d^2(n-1) + (1-\lambda_{NCC})d(n)d(n) \tag{3.9}$$

where λ_{NCC} is the exponential weighting factor ($\lambda_{NCC} < 1$ and $\lambda_{NCC} \approx 1$).

Then the decision variable of Equation (3.7) becomes:

$$\xi_{NCC}(n) = 1 - \frac{\hat{p}_{ed}(n)}{\hat{\sigma}_d^2(n)} \tag{3.10}$$

So, the DT is declared if $\xi_{NCC}(n) < T$, where T is a positive threshold.

3.2.5. DTD based on the signal envelope

This method is based on a comparison of the microphone signal energy with the energy of far-end speech, similarly to the Geigel algorithm. However, the Geigel algorithm is based on a comparison of the absolute sample values, which results in a detection function that changes rapidly. More accurate results may be obtained by using the signal energy instead of the absolute sample values. The approach proposed by the authors of [46] is based on the calculation of the signal envelope as a measure of signal energy. This method is less computationally complex than the direct energy calculation. Figure 3.3 shows the global structure of the AEC with the DTD based on the signal envelope.

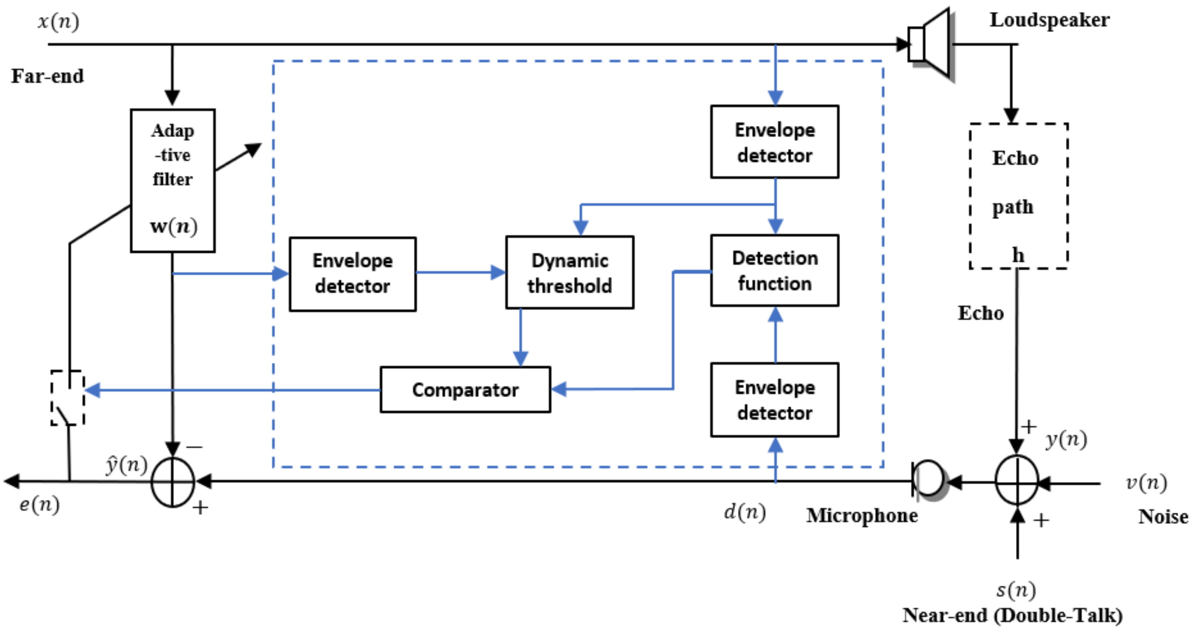

Figure.3.3. Bloc diagram of the AEC with DTD based on the signal envelope [46].

Various methods of the envelope detection may be used; one example is a low-pass filtering of the signal. In the signal envelope-based DTD algorithm, the envelope $v_x(n)$ is calculated from the absolute values of $x(n)$ samples, using the formula:

$$v_x(n) = \alpha v_x(n-1) + (1-\alpha)|x(n)| \qquad (3.11).$$

where α is a forgetting factor that defines how quickly the envelope detector reacts to rapid changes in the signal amplitude. In order to obtain an accurate estimation of the signal energy, the value of α should be slightly less than one ($\alpha = 0.99$).

In order to form the decision function, two envelope detectors are needed: $v_x(n)$ for the far-end speech signal $x(n)$ and $v_d(n)$ for the microphone signal $d(n)$. The detection function is given by the formula:

$$\xi_{env}(n) = \frac{v_d(n)}{v_x(n)+\gamma} \qquad (3.12).$$

The parameter γ is used in order to limit the values of detection function during parts of the signal containing only the noise, when values of both envelopes are low, after several experiments, the authors have fixed the value of γ to 0.05 to obtain the highest detection accuracy [46]. Moreover, to avoid the dependency of the DTD accuracy on the threshold selection, the authors proposed a method of dynamic threshold setting $T(n)$, defined as:

$$T(n) = \frac{v_y(n)}{v_x(n)+\gamma} + \beta \tag{3.12}$$

where $v_y(n)$ is the envelope of $y(n)$ and β is a small positive value used in order to leave some margin for the detection error (the suitable value is $\beta = 0.02$).

The envelope of the echo signal $y(n)$ in $T(n)$ is compared with the envelope of the microphone $d(n)$ in $\xi_{env}(n)$, if $v_d(n)$ is higher than $v_y(n)$, it indicates the presence of DT, than the DT is declared if:

$$\xi_{env}(n) > T, \text{ where } T = \begin{cases} T(n) \ , & T_{min} < T(n) < T_{max} \\ T_{min} \ , & T(n) < T_{min} \\ T_{max} \ , & T(n) > T_{max} \end{cases} \tag{3.13}$$

3.2.6. Fast DTD based on signal envelope:

Several improvements to the previous method are proposed in [47], by introducing an efficient envelope detection method, at first stage the authors propose to calculate all envelope signals using the following formula :

$$v_x(n) = \begin{cases} \lambda v_x(n-1) + (1-\lambda)|x(n)| & , \ |x(n)| \le v_x(n-1) \\ (1-\lambda)v_x(n-1) + \lambda|x(n)| & , \ |x(n)| > v_x(n-1) \end{cases} \tag{3.14}$$

where λ is a constant, practical values of λ is $0.85 \le \lambda \le 0.99$

During DT period:

$$v_d(n) > v_y(n) + v_v(n) \tag{3.15}$$

During single-talk period:

$$v_d(n) \approx v_y(n) + v_v(n) \tag{3.16}$$

Where $v_d(n)$, $v_y(n)$ and $v_v(n)$, are, respectively, envelopes of the desired, echo and additive noise signals.

At a second stage, the authors suggest to use in comparison the envelope of the estimated echo $\hat{y}(n)$ instead of the real echo $y(n)$, and by adding an approximate measure proportional to the far-end signal:

$$v_d(n) > v_{\hat{y}}(n) + \phi(n)v_x(n) + \varepsilon \tag{3.17}$$

where ε is a threshold, and $\phi(n) \geq 0$ is measure of the adaptive filter divergence, defined as:

$$\phi(n+1) = \alpha\sqrt{\frac{\Delta\mathbf{w}^T(n)\Delta\mathbf{w}(n)}{\mathbf{w}^T(n)\mathbf{w}(n)}} \quad (3.18).$$

α is a proportionality parameter and $\Delta\mathbf{w}(n)$ is the changes in the adaptive filter coefficients.

The expression of the decision variable of this method is:

$$\xi_{f\_env}(n) = \frac{v_d(n)}{v_{\hat{y}}(n)+\phi(n)v_x(n)+\varepsilon} \quad (3.19).$$

and the DT is declared when $\xi_{f\_env}(n) > T$, where $T \approx 1$.

3.2.7. DTD based on the holder inequality

The authors in [56] proposed a method of DTD based on the Holder inequality to calculate a set of decision variables.

We consider the vector of real values $\mathbf{a}$ of length L

$$\mathbf{a} = [a_0 \;\; a_1 \;\; \dots \;\; a_{L-1}]^T \quad (3.20).$$

The norms l_1, l_2 and l_∞ (maximum) of the vector $\mathbf{a}$ are defined as follows:

$$\|\mathbf{a}\|_1 = \sum_{l=0}^{L-1}|a_l| \quad (3.21).$$

$$\|\mathbf{a}\|_2 = \sqrt{\sum_{l=0}^{L-1}|{a_l}^2|} = \sqrt{\mathbf{a}^T\mathbf{a}} \quad (3.22).$$

$$\|\mathbf{a}\|_\infty = \max_{0\leq l\leq L-1}|a_l| \quad (3.23).$$

It can be demonstrated that:

$$1 \leq \frac{\|\mathbf{a}\|_1}{\|\mathbf{a}\|_2} \leq \sqrt{L} \quad (3.24).$$

$$1 \leq \frac{\|\mathbf{a}\|_1}{\|\mathbf{a}\|_\infty} \leq L \quad (3.25).$$

$$1 \leq \frac{\|\mathbf{a}\|_2}{\|\mathbf{a}\|_\infty} \leq \sqrt{L} \quad (3.26).$$

Theses inequalities are very important, since the ratios of different vector norms are bounded by values that are independent of the characteristic of the vector.

By applying the holder inequality on the two vectors **a** and **b** of length L, we get:

$$|\mathbf{a}^T\mathbf{b}| \leq \|\mathbf{a}\|_p\|\mathbf{b}\|_q, \frac{1}{p} + \frac{1}{q} = 1 \qquad (3.27).$$

In particular:

$$|\mathbf{a}^T\mathbf{b}| \leq \|\mathbf{a}\|_\infty\|\mathbf{b}\|_1 \qquad (3.28).$$

$$|\mathbf{a}^T\mathbf{b}| \leq \|\mathbf{a}\|_2\|\mathbf{b}\|_2 \qquad (3.29).$$

In the situation of single talk, the microphone signal consists of echo signal $y(n)$ and additive noise $v(n)$:

$$d(n) = \mathbf{x}^T(n)\mathbf{h} + v(n) \qquad (3.30).$$

Using Equation (3.28) and (3.30), we get:

$$\begin{aligned} |d(n)| &\leq |\mathbf{x}^T(n)\mathbf{h}| + |v(n)| \\ &\leq \|\mathbf{h}\|_\infty\|\mathbf{x}^T(n)\|_1 + |v(n)| \end{aligned} \qquad (3.31).$$

Then from Equation (3.31), we can deduce the first decision

$$\xi_1(n) = T_\infty\|\mathbf{x}^T(n)\|_1 + \sigma_v \qquad (3.32).$$

where T_∞ is a threshold depends on $\|\mathbf{h}\|_\infty$.

So, the DT is declared if $\xi_1(n) < |d(n)|$.

We can use Equation (3.28) in other manner to get:

$$\begin{aligned} |d(n)| &\leq |\mathbf{x}^T(n)\mathbf{h}| + |v(n)| \\ &\leq \|\mathbf{h}\|_1\|\mathbf{x}^T(n)\|_\infty + |v(n)| \end{aligned} \qquad (3.33).$$

Then, the Equation (3.33) leads to a second decision, written as:

$$\xi_2(n) = T_1\|\mathbf{x}^T(n)\|_\infty + \sigma_v \qquad (3.34).$$

where T_1 is an approximation of $\|\mathbf{h}\|_1$.

If $\xi_2(n) < |d(n)|$, the DT is declared.

This algorithm can be considered as a generalization of the Geigel algorithm previously presented, since the noise is taken into account. We can define the decision statistic of the Geigel algorithm in the following general form:

$$\xi_G(n) = T_G\|\mathbf{x}^T(n)\|_\infty \tag{3.35}$$

The DT is present if $\xi_G(n) < |d(n)|$, in this decision the noise is not taken into account, then the Geigel algorithm can not work properly when the level of the noise is high and may consider this noise as a presence of DT.

Finally, using Equation (3.29) we get:

$$\begin{aligned} |d(n)| &\leq |\mathbf{x}^T(n)\mathbf{h}| + |v(n)| \\ &\leq \|\mathbf{h}\|_2\|\mathbf{x}^T(n)\|_2 + |v(n)| \end{aligned} \tag{3.36}$$

So, the third decision is defined by:

$$\xi_3(n) = T_2\|\mathbf{x}^T(n)\|_2 + \sigma_v \tag{3.37}$$

where T_2 is an approximation of $\|\mathbf{h}\|_2$. DT is declared If $\xi_3(n) < |d(n)|$.

The choice of the thresholds in this method can be done as follows:

The threshold T_1 is similar to the threshold of the Geigel algorithm T_G, T_2 and T_∞ can be calculated using the properties of Equations (3.24) and (3.25) that give:

$$\|\mathbf{h}\|_1 \leq \sqrt{L}\|\mathbf{h}\|_2 \tag{3.38}$$

$$\|\mathbf{h}\|_1 \leq L\|\mathbf{h}\|_\infty \tag{3.39}$$

Then, Equations (3.38) and (3.39) conduct to:

$$T_2 = \frac{T_1}{\sqrt{L}} \tag{3.40}$$

$$T_\infty = \frac{T_1}{\sqrt{L}} \tag{3.41}$$

3.2.8. DTD using the singular value decomposition

The authors in [57] proposed a DTD based on the use of the singular value decomposition (SVD) calculated for each M-length history of the far-end signal $x(n)$, defined as:

$$\mathbf{x}_M(n) = [x(n)\ \ x(n-1)\ \ \ldots\ \ x(n-M+1)]^T \tag{3.42}$$

Where $M < L$.

Firstly, converting the vector $\mathbf{x}_M(n)$ to a matrix $\mathbf{X}_R(n)$ using a reshaping matrix of $P \times P$ dimension.

$$\mathbf{X}_R(n) = \begin{bmatrix} x(n) & x(n-1) & \cdots & x(n-P+1) \\ x(n-P) & x(n-P-1) & & x(n-2P+1) \\ \vdots & & \ddots & \vdots \\ x(n-(P-1)P) & & \cdots & x(n-M+1) \end{bmatrix} \tag{3.43}$$

where P is a positive integer and $P \times P = M$.

Secondly, applying SVD to $\mathbf{X}_R(n)$ gives the product of the three matrices; (two unitary matrices $\mathbf{U}_X$ and $\mathbf{V}_X^T$) and a diagonal matrix $\mathbf{\Sigma}_X$ [63]given by:

$$\mathbf{\Sigma}_X = \begin{bmatrix} S_{X_1} & 0 & \cdots & 0 \\ 0 & S_{X_2} & & \\ \vdots & & \ddots & \vdots \\ 0 & & \cdots & S_{X_P} \end{bmatrix} \tag{3.44}$$

where $S_{X_1} > S_{X_2} > \cdots > S_{X_P} > 0$.

The maximum of the singular values ϕ is defined by:

$$\begin{aligned} \phi &= \max\{diag\ \Sigma_X\} \\ &= \max\{S_{X_1}, S_{X_2}, \ldots, S_{X_P}\} \\ &= S_{X_1} \end{aligned} \tag{3.45}$$

Then, the decision variable of this method is defined as:

$$\xi_{SVD_X}(n) = \frac{\phi}{|\hat{p}_{ed}(n)|}$$

Where $\hat{p}_{ed}(n)$ is the estimated cross-correlation between the error and the microphone signals defined previously in Equation (3.8) of NCC based DTD.

Finally, DT is declared if $\xi_{SVD_X}(n) < T_{SVD_X}$, where $T_{SVD_X} \gg 1$.

3.2.9. DTD based on joint signal energy and cross-correlation estimation

This algorithm is based on the calculations of the envelopes of the energies $E_d(n)$ of microphone signal and $E_x(n)$ of far-end signal, using the following formulas [59]:

$$E_d(n) = \alpha E_d(n-1) + (1-\alpha)|d(n)| \quad (3.46).$$

$$E_x(n) = \alpha E_x(n-1) + (1-\alpha)|x(n)| \quad (3.47).$$

where α is a forgetting factor and $0 \ll \alpha < 1$.

Then, calculating the mean of $E_d(n)$ and $E_x(n)$, as follows:

$$m_d(n) = \frac{1}{L}\sum_{n=0}^{L-1} E_d(n) \quad (3.48).$$

$$m_x(n) = \frac{1}{L}\sum_{n=0}^{L-1} E_x(n) \quad (3.49).$$

Where L is the length of the adaptive filter.

The mean $m_d(n)$ will be compared with dynamic threshold $T_x(n)$, if $m_d(n)$ is greater than $T_x(n)$, the DT is declared.

The threshold $T_x(n)$ can be defined as:

$$T_x(n) = \begin{cases} T_{min} & , T_{min} > m_x(n) \\ m_x(n) & , otherwise \end{cases} \quad (3.50).$$

where T_{min} is a constant lower bound introduced in order to avoid detection errors, after several tests the authors have fixed the value of T_{min} to 0.24 [59].

In the case of the energy $E_d(n)$ falls below the input energy $E_x(n)$ the decision function $m_d(n)$ makes some wrong decisions during DT periods and it can cause filter divergence. To overcome this problem, a modified cross-correlation method between $d(n)$ and $x(n)$ is used in the DTD system. The modified decision parameter of the cross-correlation is:

$$\delta_{xd}(n) = (1-\gamma)\rho(n) + \gamma\delta_{xd}(n-1) \quad (3.51).$$

where $\gamma = 0.992$ and $\delta_{xd}(n)$ is the decision parameter of correlation coefficients, and $\rho(n)$ is the maximum correlation value calculated for different delays of 'l', by the following expression:

$$\rho(n) = max_l\{\frac{|\sum_{i=0}^{M-1} x(n-i-l)y(n-i)|}{|\sum x(n-i-l)y(n-i)|}\} \quad (3.52).$$

where M is the total number of samples and i is the sample length.

Finally, The DTD decision will be generated as follows:

$$\xi_{E_{dx}}(n) = \begin{cases} 1 \; if \;\; m_d(n) > T_x(n) \;\; or \;\; m_d(n) < T_x(n) \text{ and } \delta_{xd}(n) < K \\ 0 \, , \, otherwise \end{cases} \quad (3.53).$$

where K is a static threshold ($K = 0.339$ according to [59]).

This algorithm can be summarized in the flowchart of figure 3.4.

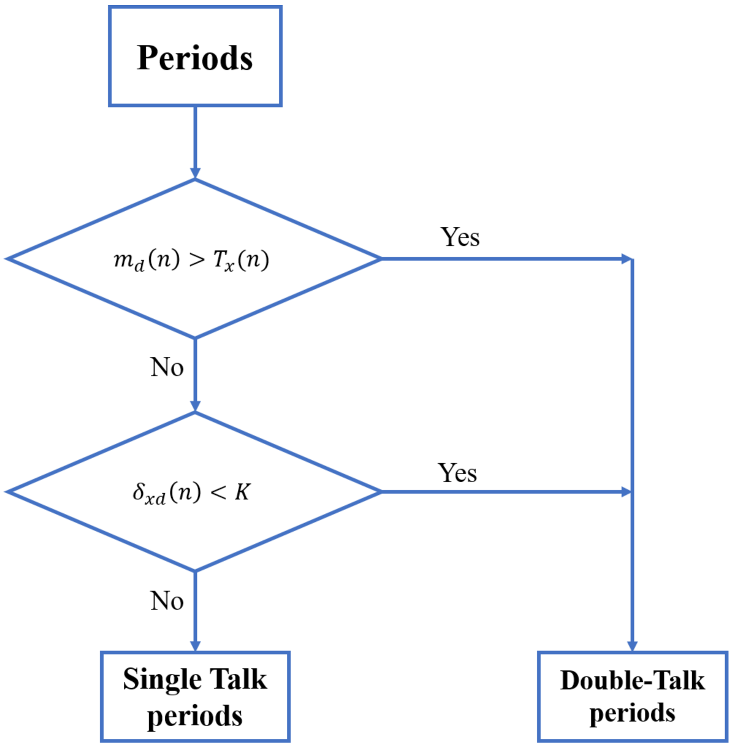

Figure.3.4. The flowchart of the DTD based on joint signal energy and cross-correlation estimation [59].

3.3. The AEC with DTD

The implementation of the DTD methods and their performances evaluation are generally achieved using the basic NLMS algorithm, since this latter is the simplest and the less complex algorithm, but the NLMS algorithm suffers from the lower convergence speed compared to the FRLS algorithms recently proposed. For that reason, we have proposed in [64] to implement a NCC based DTD presented in subsection 3.2.4 on the FNLMS algorithm presented in subsection 1.11. The step-size μ_{FNLMS} in the FNLMS algorithm is the key parameter for controlling the adaptation when the DT is declared by the decision variable $\xi_{NCC}(n)$ similar as the simple NLMS algorithm, a summary of the AEC with DTD based on NCC is given in Table 3.1.

Table 3.1. FNLMS algorithm with DTD based on NCC

Initialization parameters:

$\mathbf{w}(0) = \tilde{\mathbf{c}}(0) = 0, \gamma(0) = 1, r_1(0) = 0,$

$\alpha(0) = r_0(0) = E_0 = 1$, where E_0 is an initialization

$T = 0.92;$ Set the threshold

$\lambda_{NCC} = 0.95;$ to calculate decision statistic of DTD

$d(n) = y(n) + s(n) + v(n);$

For each instant of time n=1,2...

Prediction error:

$$r_1(n) = \lambda_a r_1(n-1) + x(n)x(n-1)$$

$$r_0(n) = \lambda_a r_0(n-1) + x^2(n)$$

$$a(n) = \frac{r_1(n)}{r_0(n) + c_a}$$

$$\varepsilon(n) = x(n) - a(n)x(n-1)$$

$$\alpha(n) = \lambda\alpha(n-1) + \varepsilon^2(n)$$

Adaptation gain:

$$\begin{bmatrix} \tilde{\mathbf{c}}(n) \\ c(n) \end{bmatrix} = \begin{bmatrix} \dfrac{\varepsilon(n)}{\lambda\alpha(n-1) + c_0} \\ \tilde{\mathbf{c}}(n-1) \end{bmatrix}$$

$$\delta(n) = -c(n)x(n-L) + \frac{x(n)\varepsilon(n)}{\lambda\alpha(n-1) + c_0}$$

$$\gamma(n) = \frac{\gamma(n-1)}{1 + \gamma(n-1)\delta(n)}$$

Filtering Part:

$$e(n) = d(n) - \mathbf{w}^T(n-1)\mathbf{x}(n)$$

----------------------------------NCC based DTD--------------------------------------

$$\hat{p}_{ed}(n) = \lambda_{NCC}\hat{p}_{ed}(n-1) + (1 - \lambda_{NCC})e(n)d(n)$$

$$\hat{\sigma}_d^2(n) = \lambda_{NCC}\hat{\sigma}_d^2(n-1) + (1 - \lambda_{NCC})d(n)d(n)$$

$$\xi_{NCC}(n) = 1 - \frac{\hat{p}_{ed}(n)}{\hat{\sigma}_d^2(n)}$$

if $\xi_{NCC}(n) > T$

$\mu_{FNLMS} = 1$, Update filter coefficients.

else

$\mu_{FNLMS} = 0$; Freeze adaptation

End

$\mathbf{w}(n) = \mathbf{w}(n-1) + \mu_{FNLMS} e(n)\gamma(n)\tilde{\mathbf{c}}(n)$

In addition, and since the novel ISM-FNLMS algorithm presented in subsection 2.6 is derived from the FNLMS algorithm, we have proposed to incorporate the DTD on this algorithm to obtain a faster convergence in case of DT is present, then the ISM-FNLMS algorithm with DTD based on NCC, is given in Table 3.2.

Table 3.2. The ISM-FNLMS algorithm with DTD based on NCC

Initialization parameters:

$\mathbf{w}(0) = \tilde{\mathbf{c}}(0) = 0, \gamma(0) = 1, r_1(0) = 0,$ error bound ζ: small constant

$\alpha(0) = r_0(0) = E_0 = 1,$ where E_0 is an initialization

$T = 0.92;$ Set the threshold

$\lambda_{NCC} = 0.95;$ to calculate decision statistic of DTD

$d(n) = y(n) + s(n) + v(n);$

For each instant of time n=1,2...

Prediction error:

$r_1(n) = \lambda_a r_1(n-1) + x(n)x(n-1)$

$r_0(n) = \lambda_a r_0(n-1) + x^2(n)$

$a(n) = \dfrac{r_1(n)}{r_0(n) + c_a}$

$\varepsilon(n) = x(n) - a(n)x(n-1)$

$\alpha(n) = \lambda\alpha(n-1) + \varepsilon^2(n)$

Adaptation gain:

$$\begin{bmatrix} \tilde{\mathbf{c}}(n) \\ c(n) \end{bmatrix} = \begin{bmatrix} \dfrac{\varepsilon(n)}{\lambda\alpha(n-1) + c_0} \\ \tilde{\mathbf{c}}(n-1) \end{bmatrix}$$

$\delta(n) = -c(n)x(n-L) + \frac{x(n)\varepsilon(n)}{\lambda\alpha(n-1)+c_0}$

$\gamma(n) = \frac{\gamma(n-1)}{1+\gamma(n-1)\delta(n)}$

Filtering Part:

$e(n) = d(n) - \mathbf{w}^T(n-1)\mathbf{x}(n)$

$\sigma_e(n) = \beta\sigma_e(n-1) + (1-\beta)|e(n)|$

---------------------------------NCC based DTD-------------------------------------

$\hat{p}_{ed}(n) = \lambda_{NCC}\hat{p}_{ed}(n-1) + (1-\lambda_{NCC})e(n)d(n)$

$\hat{\sigma}_d^2(n) = \lambda_{NCC}\hat{\sigma}_d^2(n-1) + (1-\lambda_{NCC})d(n)d(n)$

$\xi_{NCC}(n) = 1 - \frac{\hat{p}_{ed}(n)}{\hat{\sigma}_d^2(n)}$

if $\xi_{NCC}(n) > T$

$$\mu(n) = \begin{cases} 1 - \frac{\zeta}{\sigma_e(n)} & if\ |e(n)| > \zeta \\ 0 & otherwise \end{cases}$$

else

$\mu(n) = 0$; Freeze adaptation

End

$\mathbf{w}(n) = \mathbf{w}(n-1) + \mu(n)e(n)\gamma(n)\tilde{\mathbf{c}}(n)$

We can well see that, the decision variable $\xi_{NCC}(n)$ is calculated using NCC method under SM condition proposed in [41].

3.4. Experiments on AEC during DT

Experiments on the AEC of the proposed ISM-FNLMS, FNLMS and the traditional NLMS algorithms when DT exists were conducted to compare the performance of these algorithms. The used input far-end signal is the speech signal presented in subsection 2.5.1 and on figure 2.2.(c), the near-end signal is also a speech signal different of the far-end and taken on a period between the samples 55000 and 70000 as shown in figure 3.5

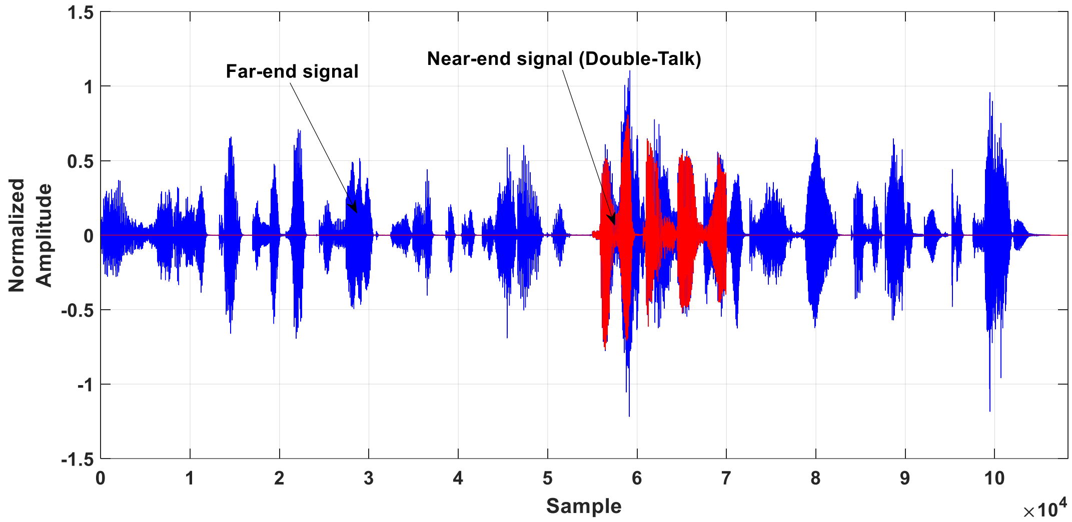

Figure.3.5. The used signals, $x(n)$: Far-end signal (in Blue), $s(n)$:Near-end signal DT (in red).

The echo paths for these experiments are the two real acoustic impulse responses presented in subsection 2.6.2 and on figure 2.8. The criteria used for performance evaluation is the Normalized Misalignment (NM) [65], [66], this criterion is robust it is calculated using the Euclidian distance between the coefficients of the used impulse response (real echo path) and the estimated coefficients of the adaptive filter, it is defined as [67]:

$$NM(dB) = 10\, log_{10} \left[\frac{\|\mathbf{w}(n)-\mathbf{h}\|^2}{\|\mathbf{h}\|^2}\right] \quad (3.54).$$

where $\|\mathbf{w}(n) - \mathbf{h}\|$ is the Euclidian distance between the vector of adaptive filter coefficients and the vector of the used impulse response.

All the parameters of the algorithms are fixed as in subsection 2.6.2 and according to Tables 3.1 and 3.2.

Figure 3.6 illustrates the evaluation of NM for the AEC system with DTD, in car impulse response of length 256, the objective of AEC is to minimize the NM, we can see an improvement by employing the DTD in the ISM-FNLMS algorithm compared to the FNLMS and NLMS algorithms in terms of NM minimization during DT period.

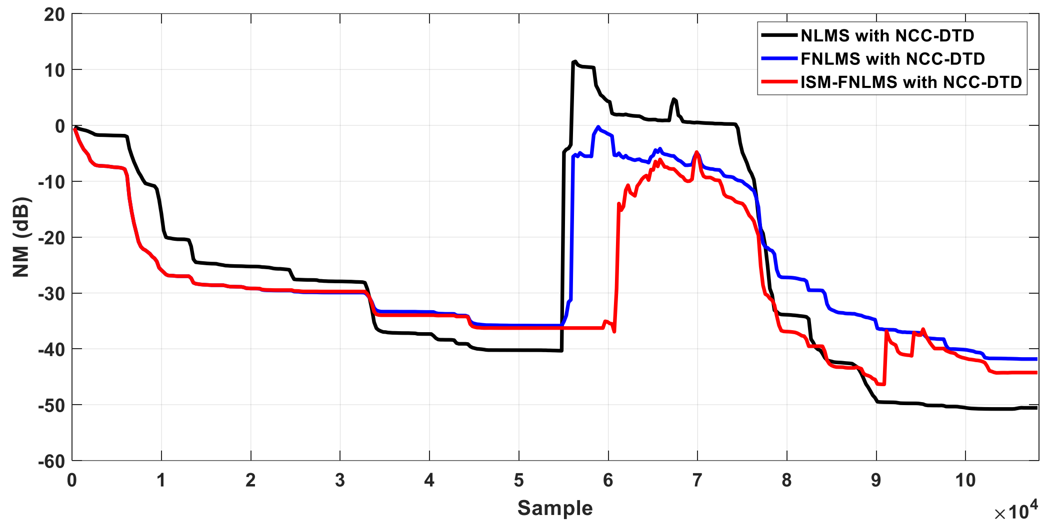

Figure.3.6. NM of ISM-FNLMS, FNLMS and NLMS with DTD in car system of $L = 256$.

Figure 3.7 is the comparison of the output error waveforms for the three algorithms, on the DT segment, the output of the NLMS and FNLMS algorithms with DTD have an obvious distortion. In the other hand, the output of the ISM-FNLMS algorithm with DTD achieved the lowest distortion.

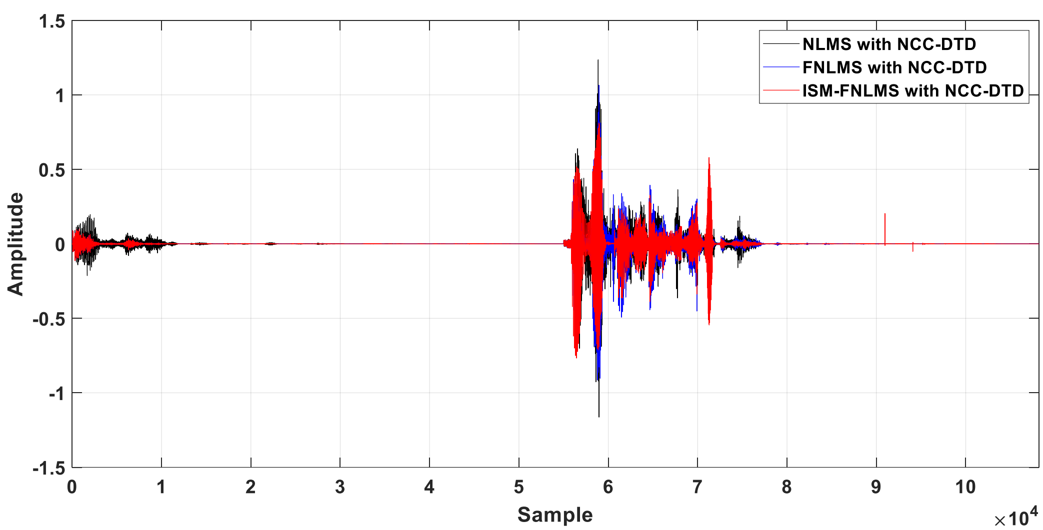

Figure.3.7. Output waveforms of ISM-FNLMS, FNLMS and NLMS with DTD in car system of $L = 256$.

In figures 3.8 and 3.9 we have performed an experiment in a long real acoustic impulse response (VCN) with length of 1024, the FNLMS and ISM-FNLMS algorithms with DTD achieved a faster convergence than the NLMS algorithm with DTD, during DT period, with lower computational complexity of the ISM-FNLMS algorithm compared to the FNLMS algorithm.

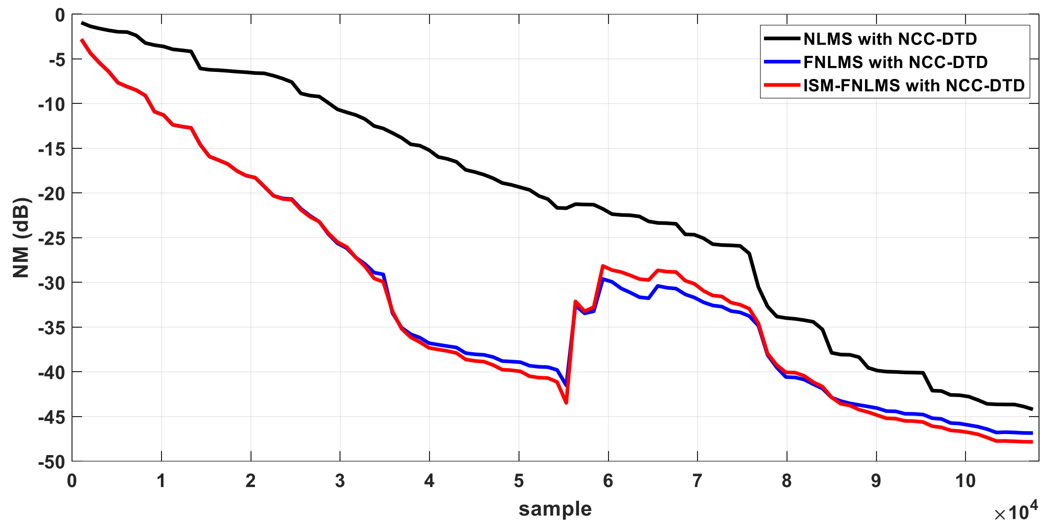

Figure.3.8. NM of ISM-FNLMS, FNLMS and NLMS with DTD in VCN system of $L = 1024$.

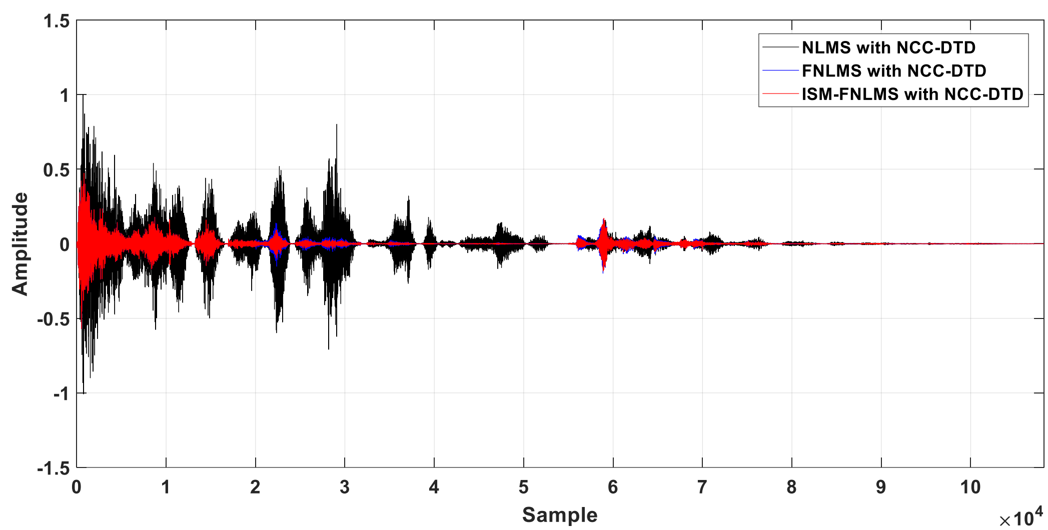

Figure.3.9. Output waveforms of ISM-FNLMS, FNLMS and NLMS with DTD in VCN system of $L = 1024$.

In addition, the ISM-FNLMS algorithm with DTD provides a fast decision of the DT by the minimization of NM, also from figures 3.7 and 3.9, the output signals were played and the subject assessed that the output error of the ISM-FNLMS algorithm with DTD provided the clearest near-end speech during the DT period.

3.5 Conclusion

In a full-duplex conversation, the presence of another signal in addition to the echo signal disturbs the capability of the adaptive algorithm to model the echo path. Therefor, the near-end signal is a source of disturbance in filter adaptation. To control this adaptation, methods of DTD should be integrated to freeze or slow-down the adaptation process during DT periods, in order to avoid the divergence of the filter ceofficients.

In this chapter, we have presented the AEC system with DTD, by employing fast adaptive algorithms, the proposed algorithms use a DTD based on NCC to ensure good performances in the presence of DT, these performances have been validated objectively using NM and subjectively using lestening tests.

General conclusion and perspectives

The problem of acoustic echo appears due to the coupling between the microphone and the speaker in a teleconferencing system. The acoustic echo cancellation is a typical application of the adaptive filtering which is based on the identification of the echo path. The echo path can be variable with physical changes of any equipement, object or people. Several adaptive filtering algorithms have been proposed in the literature, to address the problem of acoustic echo, but some of these algorithms have the issue of either low convergence rate or high computational complexity, consequently, they are not suitable for recent acoustic echo cancellation systems. Nowaday, information and telecommunication services look for systems that include robust algorithms in order to enhance the quality in any conversation.

An efficient adaptive filtering algorithm should remove the acoustic echo under different effects such as: non-stationary acoustic channel, presence of noise and presence of the double-talk, with good performances and a low complexity.

In this context, we have proposed two adaptive filtering algorithms, the first one is derived from the exisiting FNLMS algorithm, by using the set-membership technique to upper-bound the output error which results to a lower computational complexity and better convergence speed compared to the FNLMS algorithm. The second proposed algorithm denoted ISM-FNLMS, use the estimates of the output error in addition to the set-membership identification, to get higher convergence speed and tracking capability compared to the first proposed (SM-FNLMS) and the FNLMS algorithms. The main drawback of these proposed algorithms is the numerical instability if their parameters are not well examined.

Simulation results carried out from chapter two confirm good performances of the two proposed algorithms compared to the original one FNLMS, the SM-NLMS and the NLMS algorithms, in terms of convergence rate, tracking capability and computational complexity.

Also we have given a state of art of different proposed double-talk detector methods, that have been implemented using the NLMS algorithm, then we have proposed to integrate a double-talk detector based on normalized cross-correlation to the two proposed algorithms to treat its behaviours on the acoustic echo cancellation system in case of the presence of double-talk. The obtained results confirm the superiority of the proposed algorithms in the convergence to the steady-state and in the detection of double-talk, also the clarity of the outputs, compared to that when implementing the double-talk detector on the NLMS algorithm.

As a perspective, we work on the development of a new adaptive filtering algorithm with new double-talk detector and echo path change detector, that is able to perform these detections simultaneously, and as a practical project, we work on the real time implementation of all these algorithms on the Field Programmable Gate Array (FPGA) environment.

Bibliography

[1] C. B. Pop and D. Cabrera, “Auditory room size perception for real rooms,” *Proc Acoust*, vol. 2005, pp. 115–121, 2005.

[2] S. S. Haykin, *Adaptive Filter Theory*, 5th ed. Pearson Education India, 2014.

[3] T. Alexander, *Adaptive signal processing: theory and applications*. Springer Science & Business Media, 2012.

[4] A. H. Sayed, *Fundamentals of adaptive filtering*. John Wiley & Sons, 2003.

[5] K. Ozeki and T. Umeda, “An adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties,” *Electronics and Communications in Japan (Part I: Communications)*, vol. 67, no. 5, pp. 19–27, 1984.

[6] K. Shi and X. Ma, “A frequency domain step-size control method for LMS algorithms,” *IEEE Signal Processing Letters*, vol. 17, no. 2, pp. 125–128, 2009.

[7] A. Uncini, *Fundamentals of adaptive signal processing*. Springer, 2015.

[8] C. F. N. Cowan, “Introduction to adaptive filters,” *IEE Colloquium (Digest)*, no. 48. CRC Press LLC, 1996.

[9] A. Sayed, *Adaptive filters*. John Wiley & Sons, 2011.

[10] V. H. Nascimento and M. T. M. Silva, “Adaptive filters,” in *Academic Press Library in Signal Processing*, vol. 1, Elsevier, 2014, pp. 619–761.

[11] M. Hamidia, “Amélioration de l’annulation d’écho acoustique avec détection de double-parole dans les communications mobiles,” *Thèse de doctorat, Université des sciences et de la technologie Houari Boumediene, Alger*, 2017.

[12] B.Widrow; S.D.Stearns, *Adaptive signal processing*, 1st ed. Englewood Cliffs, NJ, Prentice Hall. Inc, 1985.

[13] Arthur E Albert; Leland A. Gardner, “Stochastic approximation and nonlinear regression,” *MIT press*, 1967.

[14] I. Hassani, A. Kedjar, M. A. Ramdane, M. Arezki, and A. Benallal, “Fast sparse adaptive filtering algorithms for acoustic echo cancellation,” in *2018 International Conference on Communications and Electrical Engineering (ICCEE)*, 2018, pp. 1–5.

[15] A. Gilloire, E. Moulines, D. Slock, and P. Duhamel, “State of the Art in Acoustic Echo Cancellation,” in *Digital Signal Processing in Telecommunications*, Springer London, 1996, pp. 45–91.

[16] M. E. Khan, “Matrix inversion lemma and information filter,” *Honeywell Techonology Solutions Lab, Bangalore, India*, 2005.

Bibliography

[17] D. Godard, “Channel equalization using a Kalman filter for fast data transmission,” *IBM Journal of Research and Development*, vol. 18, no. 3, pp. 267–273, 1974.

[18] D. D. Falconer and L. Ljung, “Application of Fast Kalman Estimation to Adaptive Equalization,” *IEEE Transactions on Communications*, vol. 26, no. 10, pp. 1439–1446, 1978.

[19] G. Carayannis, D. G. Manolakis, and N. Kalouptsidis, “A Fast Sequential Algorithm for Least-Squares Filtering and Prediction,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 31, no. 6, pp. 1394–1402, 1983.

[20] J. Cioffi and T. Kailath, “Fast, recursive-least-squares transversal filters for adaptive filtering,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 2, pp. 304–337, Apr. 1984.

[21] M.Arezki, “Algorithmes adaptatifs évolués de filtrage,” *Thèse de doctorat, Ecole nationale polytechnique, Alger*, 2011.

[22] M.Djendi, “Amélioration du comportement des algorithmes des moindres carrées transversaux rapides. Application à l’identification de réponses impulsionnelles acoustiques longues,” *Thèse de doctorat, Ecole nationale polytechnique, Alger*, 2006.

[23] G. Carayannis, D. Manolakis, and N. Kalouptsidis, “A unified view of parametric processing algorithms for prewindowed signals,” *Signal Processing*, vol. 10, no. 4, pp. 335–368, Jun. 1986.

[24] A. Benallal and A. Benkrid, “A simplified FTF-type algorithm for adaptive filtering,” *Signal Processing*, vol. 87, no. 5, pp. 904–917, May 2007.

[25] M. Arezki, A. Benallal, P. Meyrueis, and D. Berkani, “A new algorithm with low complexity for adaptive filtering,” *Engineering Letters*, vol. 18, no. 3, p. 205, 2010.

[26] A. Benallal and M. Arezki, “A fast convergence normalized least-mean-square type algorithm for adaptive filtering,” *International Journal of Adaptive Control and Signal Processing*, vol. 28, no. 10, pp. 1073–1080, Oct. 2014.

[27] M. Djendi, “New efficient adaptive fast transversal filtering (FTF)-type algorithms for mono and stereophonic acoustic echo cancelation,” *International Journal of Adaptive Control and Signal Processing*, vol. 29, no. 3, pp. 273–301, Mar. 2015.

[28] S. Werner, M. L. R. deCampos, and P. S. R. Diniz, “Partial-Update NLMS Algorithms with Data-Selective Updating,” *IEEE Transactions on Signal Processing*, vol. 52, no. 4, pp. 938–949, Apr. 2004.

[29] S. C. Douglas, “Adaptive filters employing partial updates,” *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 44, no. 3, pp. 209–216, 1997.

[30] K. Mayyas, “A variable step-size selective partial update LMS algorithm,” *Digital Signal Processing*, vol. 23, no. 1, pp. 75–85, Jan. 2013.

[31] M. Godavarti and A. O. Hero, “Partial update LMS algorithms,” *IEEE Transactions on signal processing*, vol. 53, no. 7, pp. 2382–2399, 2005.

Bibliography

[32] J. R. Deller and Y. F. Huang, "Set-membership identification and filtering for signal processing applications," *Circuits, Systems, and Signal Processing*, vol. 21, no. 1, pp. 69–82, Jan. 2002.

[33] P. S. R. Diniz and S. Werner, "Set-Membership Binormalized Data-Reusing LMS Algorithms," *IEEE Transactions on Signal Processing*, vol. 51, no. 1, 2003.

[34] E. Fogel and Y.-F. Huang, "On the value of information in system identification—bounded noise case," *Automatica*, vol. 18, no. 2, pp. 229–238, 1982.

[35] S. Gollamudi, S. Nagaraj, S. Kapoor, and Yih-Fang Huang, "Set-membership filtering and a set-membership normalized LMS algorithm with an adaptive step size," *IEEE Signal Processing Letters*, vol. 5, no. 5, pp. 111–114, May 1998.

[36] A. Antoniou and W.-S. Lu, *Practical optimization: algorithms and engineering applications*. Springer Science & Business Media, 2007.

[37] L. Guo and Y.-F. Huang, "Set-membership adaptive filtering with parameter-dependent error bound tuning," in *Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, 2005, vol. 4, pp. iv--369.

[38] Md. Z. A. Bhotto and A. Antoniou, "A set-membership nlms algorithm with adaptive error bound," in *2008 Canadian Conference on Electrical and Computer Engineering*, May 2008, pp. 002007–002010.

[39] S. Zhang and J. Zhang, "Set-Membership NLMS Algorithm With Robust Error Bound," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 61, no. 7, pp. 536–540, Jul. 2014.

[40] I. HASSANI, M. AREZKI, and A. BENALLAL, "Set-Membership Fast-NLMS Algorithm for Acoustic Echo Cancellation," in *2018 International Conference on Signal, Image, Vision and their Applications (SIVA)*, 2018, pp. 1–5.

[41] I. Hassani, M. Arezki, and A. Benallal, "A novel set membership fast NLMS algorithm for acoustic echo cancellation," *Applied Acoustics*, vol. 163, p. 107210, Jun. 2020.

[42] M. Arezki, A. Namane, and A. Benallal, "Performance Analysis of the Fast-NLMS type Algorithm," in *Proceedings of the World Congress on Engineering*, 2017, vol. 1.

[43] J. Gunther, "Learning echo paths during continuous double-talk using semi-blind source separation," *IEEE transactions on audio, speech, and language processing*, vol. 20, no. 2, pp. 646–660, 2011.

[44] M. A. Iqbal, J. W. Stokes, and S. L. Grant, "Normalized double-talk detection based on microphone and AEC error cross-correlation," in *2007 IEEE International Conference on Multimedia and Expo*, 2007, pp. 360–363.

[45] D. Duttweiler, "A twelve-channel digital echo canceler," *IEEE Transactions on Communications*, vol. 26, no. 5, pp. 647–653, 1978.

[46] G. Szwoch, A. Czyzewski, and M. Kulesza, "A low complexity double-talk detector based on the signal envelope," *Signal Processing*, vol. 88, no. 11, pp. 2856–2862, Nov. 2008.

[47] M. S. Hussain, M. A. Hasan, M. F. Bari, and A. B. M. Harun-Ur-Rashid, "A fast double-talk detection algorithm based on signal envelopes for implementation of acoustic echo cancellation in embedded systems," in *4th International Conference on Advances in Electrical Engineering, ICAEE 2017*, Jul. 2017, vol. 2018-January, pp. 199–204.

[48] T. Gänsler, M. Hansson, C. J. Ivarsson, and G. Salomonsson, "A double-talk detector based on coherence," *IEEE Transactions on Communications*, vol. 44, no. 11, pp. 1421–1427, 1996.

[49] J. Benesty, D. R. Morgan, and J. H. Cho, "A new class of doubletalk detectors based on cross-correlation," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 2, pp. 168–172, 2000.

[50] T. Gänsler and J. Benesty, "A frequency-domain double-talk detector based on a normalized cross-correlation vector," *Signal Processing*, vol. 81, no. 8, pp. 1783–1787, Aug. 2001.

[51] J. Benesty and T. Gänsler, "A multichannel acoustic echo canceler double-talk detector based on a normalized cross-correlation matrix," *European Transactions on Telecommunications*, vol. 13, no. 2, pp. 95–101, Mar. 2002.

[52] T. Gänsler and J. Benesty, "The fast normalized cross-correlation double-talk detector," *Signal Processing*, vol. 86, no. 6, pp. 1124–1139, Jun. 2006.

[53] K. H. Lee, J. H. Chang, N. S. Kim, S. Kang, and Y. Kim, "Frequency-domain double-talk detection based on the gaussian mixture model," *IEEE Signal Processing Letters*, vol. 17, no. 5, pp. 453–456, 2010.

[54] G. C. Sih, "Doubletalk detection by means of spectral content." Google Patents, 1998.

[55] K. Rahbar, "Double talk detection method based on spectral acoustic properties." Google Patents, 2012.

[56] C. Paleologu, J. Benesty, T. Gaensler, and S. Ciochină, "Class of double-talk detectors based on the holder inequality," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2011, pp. 425–428.

[57] M. Hamidia and A. Amrouche, "Double-talk detection using the singular value decomposition for acoustic echo cancellation," in *IEEE International Conference on Communications*, 2013, pp. 4745–4749.

[58] M. Hamidia and A. Amrouche, "Double-talk detector based on speech feature extraction for acoustic echo cancellation," in *2014 22nd International Conference on Software, Telecommunications and Computer Networks, SoftCOM 2014*, Feb. 2011, pp. 393–397.

[59] A. A. M. Muzahid, K. M. R. Ingrid, S. I. M. M. R. Mondol, and Y. Zhou, "Advanced double-talk detection algorithm based on joint signal energy and cross-correlation estimation," in *Proceedings of 2016 8th IEEE International Conference on Communication Software and Networks, ICCSN 2016*, Oct. 2016, pp. 303–306.

Bibliography

[60] C. Schuldt, F. Lindstrom, and I. Claesson, “A delay-based double-talk detector,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 20, no. 6, pp. 1725–1733, 2012.

[61] A. Geigel, 912 RE Lamarche - US Patent 4, 029, and U. 1977, “Common control digital echo suppressor,” *Google Patents*.

[62] H. Ye and B. X. Wu, “A New Double-Talk Detection Algorithm Based on the Orthogonality Theorem,” *IEEE Transactions on Communications*, vol. 39, no. 11, pp. 1542–1545, 1991.

[63] V. C. Klema and A. J. Laub, “The Singular Value Decomposition: Its Computation and Some Applications,” *IEEE Transactions on Automatic Control*, vol. 25, no. 2, pp. 164–176, 1980.

[64] I. Hassani, M. Arezki, and A. Benallal, “The Fast-NLMS Algorithm for Acoustic Echo Cancellation with Double-Talk Detection,” *Lecture Notes in Electrical Engineering*, vol. 682, pp. 1271–1280, 2021.

[65] M. Hamidia and A. Amrouche, “A new structure for acoustic echo cancellation in double-talk scenario using auxiliary filter,” in *2014 14th International Workshop on Acoustic Signal Enhancement, IWAENC 2014*, Nov. 2014, pp. 253–257.

[66] M. Z. Ikram, “Double-talk detection in acoustic echo cancellers using zero-crossings rate,” in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, Aug. 2015, vol. 2015-August, pp. 1121–1125.

[67] A. Jain, S. Goel, K. Nathwani, and R. M. Hegde, “Robust acoustic echo cancellation using Kalman filter in double talk scenario,” *Speech Communication*, vol. 70, pp. 65–75, Jun. 2015.

Author contribution

[1] I. HASSANI, M. AREZKI, and A. BENALLAL, “Set-Membership Fast-NLMS Algorithm for Acoustic Echo Cancellation,” in *2018 International Conference on Signal, Image, Vision and their Applications (SIVA)*, 2018, pp. 1–5.

[2] I. Hassani, A. Kedjar, M. A. Ramdane, M. Arezki, and A. Benallal, “Fast sparse adaptive filtering algorithms for acoustic echo cancellation,” in *2018 International Conference on Communications and Electrical Engineering (ICCEE)*, 2018, pp. 1–5.

[3] I. Hassani, M. Arezki, and A. Benallal, “A novel set membership fast NLMS algorithm for acoustic echo cancellation,” *Applied Acoustics*, vol. 163, p. 107210, Jun. 2020, doi: 10.1016/j.apacoust.2020.107210.

[4] I. Hassani, M. Arezki, and A. Benallal, “The Fast-NLMS Algorithm for Acoustic Echo Cancellation with Double-Talk Detection,” *Lecture Notes in Electrical Engineering*, vol. 682, pp. 1271–1280, 2021, doi: 10.1007/978-981-15-6403-1_90.