

UNIVERSITE BLIDA 1

Faculté de Technologie
Département d'Électronique

THESE DE DOCTORAT

En Electronique

Annulation stéréophonique de l'écho acoustique par des techniques avancées de traitement du signal et implémentation sur FPGA

Par

Merouane MESSINI

Devant le jury composé de :

Z.BENSELAMA

Professeur, U.Blida 1

Président

A.MERAGHNI

Professeur, ENS, Kouba, Alger

Examineur

M.MAAMOUN

MCA, U.Blida 1

Examineur

M.BOUHEDDA

Professeur, U.Médéa

Examineur

M.DJENDI

Professeur, U.Blida 1

Rapporteur

A.GUESSOUM

Professeur, U.Blida 1

Co-apporteur

Blida, Janvier 2024

DEDICACE

Je dédie ce modeste travail à mes parents qui m'a soutenu et encouragé durant ces années d'études.

Je dédie également à mes frères et sœurs qui m'ont soutenu le long de ma vie et quine ces sentent à le faire.

À tous mes amis

MESSINI Merouane

REMERCIEMENT

Le plus grand merci revient à '**ALLAH**' le tout puissant qui lui seul m'a guidé dans le bon chemin durant ma vie et qui m'a aidé à réaliser ce modeste travail, ainsi que les personnes qui ont contribué de près ou de loin dans la réalisation de ce travail, qu'ils trouvent ici le témoignage de nos gratitude.

Nous tenons à remercier vivement Professeur **Mohamed Djendi**, qui a accepté de diriger ce travail de recherche. Qu'il trouve ici l'expression de nos plus grandes gratitude pour son soutien bien veillant et la confiance qu'il nous a manifesté dans les moments les plus difficiles. Qu'il soit assuré de nos profondes gratitude, de nos reconnaissances et de nos plus grandes sympathies.

Nous remercions également le Co-promoteur le professeur **Abderrezak Guessoum**.

Enfin, nos remerciements vont également à tous nos amis et collègues de l'université **SAAD DAHLEB** de **Blida1**, tous **nos enseignants** et tout le personnel du département d'Electronique.

Résumé

Dans cette thèse, nous étudions l'annulation d'écho acoustique stéréophonique par le filtrage adaptatif. Nous proposons cinq nouveaux algorithmes stéréophoniques basés sur les techniques et algorithmes efficaces existants. Le premier algorithme proposé est une version étendue de l'algorithme FNLMS, adapté aux applications stéréophonique avec une amélioration significatives de la partie de prédiction, assurant la stabilité dans le contexte stéréophonique. Les quatre autres algorithmes proposés dérivés d'une combinaison de deux techniques : la technique de sélection des données et la technique de mise à jour partielle (PU). Ces derniers sont appliqués à un algorithme adaptatif à deux canaux, Stereophonic Normalized Least Mean Squares (SNLMS), et prennent en compte uniquement les données d'entrée qui apportant de nouvelles informations au système. Par conséquent, la complexité du calcul est réduite.

A la fin de cette thèse, l'algorithme Stereophonic Fast Normalized Least Mean Squares (SFNLMS) proposé est implémenté en code VHDL pour une application FPGA.

Les résultats expérimentaux obtenus ont montré que les algorithmes proposés sont des choix intéressants pour l'AEAS en sélectionnant l'algorithme approprié faisant un compromis entre la complexité, le niveau de l'EQM, le niveau de l'ERLE et le niveau du Désalignement.

Mots clés : Annulation d'Echo Acoustique Stéréophonique (AEAS) ; filtre adaptatif ; mise à jour partielle (PU) ; la technique de sélection de données ; complexité du calcul; VHDL ; FPGA.

Abstract

In this thesis, we study the stereophonic acoustic echo cancellation by adaptive filter. We propose five new stereophonic algorithms based on the existing efficient technics and algorithm. The first proposed algorithm is an extended version of the FNLMS algorithm, adapted for stereophonic applications with significant improvement in the prediction part, insuring the stability in the stereophonic context. The other four proposed algorithms derived from a combination of two techniques: the data selection technique and the partial update technique (PU). These are applied to two-channel adaptive algorithm, Stereophonic Normalized Least Mean Squares (SNLMS), and consider only input data that contributes new information to the system. This approach effectively reduces computational complexity.

At the end of this these, the proposed Stereophonic Fast Normalized Least Mean Squares (SFNLMS) algorithm is implemented in VHDL code for FPGA application.

The experiment results given by our analysis confirm the given theoretical expressions. We have shown that the proposed algorithms are interesting choices for the AEAS by selecting the algorithm appropriate to the need, taking into account the trade-off between the complexity, the level of the EQM, the level of the ERLE and the level of the Misalignment.

Keywords: stereophonic acoustic echo cancellation ; adaptive filter ; partial update; data selection; computational complexity; VHDL ; FPGA.

ملخص

في هاته الأطروحة، درسنا إلغاء الصدى الصوتي المجسم بواسطة المرشح التكيفي. حيث اقترحنا خمس خوارزميات مجسمة جديدة بناءً على التقنيات والخوارزمية الفعالة الحالية. أول خوارزمية مقترحة هي نسخة موسعة من خوارزمية FNLMS معدة للتطبيقات المتعددة القنوات مع تحسين هام لجزء التنبؤ مما يضمن الاستقرار في الحالة المتعددة القنوات. الخوارزميات الأربعة الأخرى المقترحة المشتقة بناءً على مزيج من تقنيتين: تقنية اختيار البيانات وتقنية التحديث الجزئي (PU). هذه التقنيات مطبقة على خوارزمية تكيفية ثنائية القناة SNLMS، بالأخذ بالاعتبار فقط بيانات الإدخال التي تقدم معلومات جديدة للنظام. هذا النهج يقلل بفعالية تعقيد الحساب. في نهاية هذه الأطروحة، قمنا بتنفيذ الخوارزمية SFNLMS المقترحة والمناقشة في كود VHDL للتطبيق على FPGA. تأكد النتائج التجريبية الصيغ النظرية المقدمة. ولقد تم اثبات أن الخوارزميات المقترحة هي اختيارات مثيرة للاهتمام لانظمة إلغاء الصدى التكيفية ثنائية القنوات من خلال اختيار الخوارزمية المناسبة للحاجة، مع الأخذ في الاعتبار المفاضلة بين التعقيد ومستوى الخطأ التريبيعي المتوسط EQM ومستوى تحسين فقد الصوت المرتجع ERLE ومستوى عدم التوافق.

TABLE DES MATIERES

REMERCIEMENT	I
Résumé	II
Abstract	II
ملخص	II
LISTE DES FIGURES	VI
LISTE DES TABLEAUX	VIII
LISTE DES ABREVIATIONS	IX
INTRODUCTION GENERALE	1
CHAPITRE 1: GENERALITES SUR LE FILTRAGE ADAPTATIF	ET
L'ANNULATION DE L'ÉCHO ACOUSTIQUE	4
1.1. Introduction	4
1.2. L'Écho.....	4
1.3. L'Annulation d'Écho Acoustique	4
1.4. Les filtres adaptatifs	5
1.4.1. Approches d'adaptation.....	5
A. Approche basée sur la théorie du filtre de Wiener	5
B. Méthode des moindres carrés	6
1.4.2. Les structures de filtre adaptatif	6
1.5. Les algorithmes adaptatifs.....	8
1.5.1. Algorithme du gradient stochastique LMS.....	8
1.5.2. L'algorithme du gradient stochastique normalisé NLMS	10
1.5.3. L'algorithme des moindres carrés récurrents RLS	10
1.6. Conclusion.....	13
CHAPITRE 2: ÉTUDE DE L'ANNULATION STEREOPHONIQUE D'ÉCHO	
ACOUSTIQUE	14
2.1. Introduction	14
2.2. La configuration de l'annulation de l'écho acoustique stéréophonique.....	14
2.3. Le problème de non-unicité de solution.....	16
2.4. Technique de décorrélation des signaux	17
2.5. Conclusion.....	20
CHAPITRE 3: L'Algorithme de Gradient Stochastique Normalisé Rapide (FNLMS)	
pour l'Annulation Stéréophonique d'Écho Acoustique	21

3.1. Introduction	21
3.2. Description de l'algorithme adaptatif FNLMS à un canal	21
3.3. Proposition de l'algorithme FNLMS stéréophoniques (SFNLMS)	25
3.4. Conclusion.....	28
CHAPITRE 4: UN NOUVEL ALGORITHME ADAPTATIF SELECTIF DES DONNEES POUR L'AEA STEREOPHONIQUE.....	29
4.1. Introduction	29
4.2. Description de l'algorithme adaptatif de l'appartenance à un ensemble à deux canaux (stéréophonique) ES-SM-NLMS	29
4.3. Algorithme NLMS non paramétrique à pas variable stéréophonique (SNP-VSS-N)	33
4.4. Description de l'algorithme adaptatif XMNL-NLMS avec la technique de l'appartenance à un ensemble étendu au cas stéréophonique (SM-PU-SNLMS).....	36
4.5. Description de l'algorithme adaptatif XMNL-NLMS avec pas variable non paramétrique étendu au cas stéréophonique (SNP-VSS-PU-N)	38
4.6. Conclusion.....	40
CHAPITRE 5: Étude Comparative Des Performances et Résultats de Simulation des Algorithmes Adaptatifs Monophoniques et Stéréophoniques	41
5.1. Introduction	41
5.2. Signaux utilisés en simulation.....	41
5.3. Simulation de réponses impulsionnelles	43
5.4. Critères objectifs de qualité.....	44
5.4.1. Critère de l'erreur quadratique moyenne « EQM ».....	45
5.4.2. Critère de désajustement « Misalignement ».....	45
5.4.3. Critère d'amélioration de perte de retour d'écho « ERLE »	45
5.5. Influence des paramètres des algorithmes adaptatifs monophoniques	46
5.5.1. Effet de la taille du filtre des algorithmes NLMS et FNLMS	46
5.5.2. Effet du pas d'adaptation μ sur les algorithmes NLMS et FNLMS	48
5.6. Comparaison des performances des algorithmes stéréophoniques	50
5.6.1. Algorithme FNLMS stéréophonique (SFNLMS).....	50
5.6.2. Algorithmes sélectifs des données pour l'AEAS	55
5.6.3. Complexité de calcul	62
5.6.4. Synthèse de l'évaluation des performances des algorithmes proposés.....	62
5.7. Conclusion.....	62

CHAPITRE 6: Implémentation sur FPGA « Description VHDL » d'un Algorithme d'Annulation d'Écho Acoustique Stéréophonique	64
6.1. Introduction	64
6.2. FPGAs	65
6.3. Les langages de description matériel	66
6.4. Structure d'une description VHDL	66
6.5. Conception sur FPGA et mise en œuvre	68
6.5.1. Technique de codage de données	68
6.5.2. Organigramme de l'AEAS par l'algorithme SNLMS.....	69
6.5.3. Organigramme de l'AEAS par l'algorithme SFNLMS	70
6.5.4. Astuces de conception VHDL et pièces de déclaration.....	72
6.6. Résultats de simulation	74
6.7. Conclusion.....	82
CONCLUSION GENERALE.....	83
RÉFÉRENCES.....	85

LISTE DES FIGURES

Figure 1.1: La structure de base de l'AEA	4
Figure 1.2: configuration générale du filtre adaptatif	5
Figure 1.3 : La structure d'un filtre FIR	6
Figure 1.4: Combinateur linéaire adaptatif	7
Figure 1.5: La structure d'un filtre IIR	8
Figure 2.1: La configuration générale de l'annulation d'écho acoustique stéréophonique	14
Figure 2.2: La configuration du système stéréo avec prétraitement.	18
Figure 4.1: la procédure de mise à jour SMF	30
Figure 5.1 : Bruit USASI.	41
Figure 5.2 : Bruit Blanc Gaussien.	42
Figure 5.3 : Signal parole	42
Figure 5.4 : Exemple d'une réponse impulsionnelle simulé de taille L=256	43
Figure 5.5 : Exemple d'une réponse impulsionnelle simulé de taille L=1024	44
Figure 5.6 : EQM convergence de NLMS et FNLMS algorithmes pour L=256 et L=1024 avec signal d'entrée bruit USASI	47
Figure 5.7 : Misalignement convergence de NLMS et FNLMS algorithmes pour L=256 et L=1024 avec signal d'entrée bruit USASI	47
Figure 5.8 : Effet du pas d'adaptation sur la convergence de l'EQM pour l'algorithme NLMS.	48
Figure 5.9 : Effet du pas d'adaptation sur la convergence de Misalignement pour l'algorithme NLMS.	48
Figure 5.10 : Effet du pas d'adaptation sur la convergence de l'EQM pour l'algorithme FNLMS	49
Figure 5.11 : Effet du pas d'adaptation sur la convergence de Misalignement pour l'algorithme FNLMS	49
Figure 5.12: l'évaluation de l'EQM pour tester la stabilité numérique de l'algorithme SFNLMS par rapport l'algorithme SNLMS	50
Figure 5.13: l'évaluation de l'ERLE pour tester la stabilité numérique de l'algorithme SFNLMS par rapport l'algorithme SNLMS	51
Figure 5.14: l'évaluation de l'EQM pour tester la vitesse de convergence de l'algorithme SFNLMS par rapport l'algorithme SNLMS.	52
Figure 5.15: l'évaluation de l'ERLE pour tester la vitesse de convergence de l'algorithme SFNLMS par rapport l'algorithme SNLMS.	53
Figure 5.16: l'évaluation de l'EQM pour tester AEAS de l'algorithme SFNLMS par rapport l'algorithme SNLMS en présence de signal parole	54
Figure 5.17: l'évaluation de l'ERLE pour tester AEAS de l'algorithme SFNLMS par rapport l'algorithme SNLMS en présence de signal parole	54
Figure 5.18: l'évaluation de l'EQM pour tester AEAS de l'algorithme SFNLMS par rapport l'algorithme SNLMS avec RI longue de taille L=1024	55
Figure 5.19: l'évaluation de l'EQM et le Misalignement pour étudier l'effet de la bande d'erreur γ =gamma de l'algorithme SM-PU-SNLMS	56
Figure 5.20: nombre d'itérations nécessaires pour atteindre le régime permanent	56
Figure 5.21: l'EQM, le Misalignement et le nombre d'itérations nécessaires pour atteindre l'état d'équilibre en utilisant le bruit USASI comme signal d'entrée	57
Figure 5.22: l'EQM, le Misalignement et le nombre d'itérations nécessaires pour atteindre l'état d'équilibre en utilisant le signal parole en entrée et des réponses impulsionnelles de taille L=256	58
Figure 5.23: l'EQM, le Misalignement et le nombre d'itérations nécessaires pour atteindre l'état d'équilibre en utilisant le signal parole en entrée et des réponses impulsionnelles de taille L=1024	59
Figure 5.24: l'évaluation de Misalignement en utilisant le bruit USASI en entrée et des réponses impulsionnelles de taille L=256	60
Figure 5.25: l'évaluation de l'EQM en utilisant le bruit USASI en entrée et des réponses impulsionnelles de taille L=256	60

Figure 5.26: l'évaluation de Misalignement en utilisant le signal parole en entrée et des réponses impulsives de taille $L=256$	61
Figure 5.27: l'évaluation de l'EQM en utilisant le signal parole en entrée et des réponses impulsives de taille $L=256$	61
Figure 6.1: La structure interne de la carte FPGA	66
Figure 6.2 : Les flux des données de MATLAB vers ModelSim	69
Figure 6.3: Organigramme de l'algorithme adaptatif SNLMS dans l'application AEAS	70
Figure 6.4: Organigramme de l'algorithme adaptatif SFNLMS dans l'application AEAS	71
Figure 6.5: Interface externe de l'algorithme SNLMS/ SFNLMS en VHDL	72
Figure 6.6: Les étages en cascade du processus de traitement	73
Figure 6.7: La simulation de l'algorithme SNLMS pour AEAS en ModelSim avec WGN en entrée	75
Figure 6.8: La simulation de l'algorithme SFNLMS pour AEAS en ModelSim avec WGN en entrée	75
Figure 6.9: Évaluation de EQM obtenu par les deux algorithmes SNLMS et SFNLMS sur MATLAB et ModelSim avec bruit blanc en entrée	76
Figure 6.10: Évaluation de l'ERLE obtenu par les deux algorithmes SNLMS et SFNLMS sur MATLAB et ModelSim avec bruit blanc en entrée	76
Figure 6.11: La simulation de l'algorithme SNLMS pour AEAS en ModelSim avec le bruit USASI en entrée	77
Figure 6.12: La simulation de l'algorithme SFNLMS pour AEAS en ModelSim avec le bruit USASI en entrée	78
Figure 6.13: Évaluation de EQM obtenu par les deux algorithmes SNLMS et SFNLMS sur MATLAB et ModelSim avec le bruit USASI en entrée	78
Figure 6.14: Évaluation de l'ERLE obtenu par les deux algorithmes SNLMS et SFNLMS sur MATLAB et ModelSim avec le bruit USASI en entrée	79
Figure 6.15: La simulation de l'algorithme SNLMS pour AEAS en ModelSim avec le signal parole en entrée	80
Figure 6.16: La simulation de l'algorithme SFNLMS pour AEAS en ModelSim avec le signal parole en entrée	80
Figure 6.17: Évaluation de EQM obtenu par les deux algorithmes SNLMS et SFNLMS sur MATLAB et ModelSim avec le signal parole en entrée	81
Figure 6.18: Évaluation de l'ERLE obtenu par les deux algorithmes SNLMS et SFNLMS sur MATLAB et ModelSim avec le signal parole en entrée	81

LISTE DES TABLEAUX

Tableau 1.1: Peseudo code de l'algorithme du gradient stochastique LMS	9
Tableau 1.2: Peseudo code de l'algorithme du gradient stochastique normalisé NLMS	10
Tableau 1.3: Peseudo code de l'algorithme des moindres carrés récursifs RLS	12
Tableau 2.1: Peseudo code de l'algorithme du gradient stochastique normalisé stéréophonique SNLMS [34]	16
Tableau 2.2 : Peseudo code de l'algorithme de gradient stochastique normalisé non linéaire avec maximum exclusif (XMNL-NLMS) [37].	19
Tableau 3.1 : Peseudo code de l'algorithme du gradient stochastique normalisé rapide FNLMS [38]	24
Tableau 3.2 : Peseudo code de l'algorithme du gradient stochastique normalisé rapide stéréophonique SFNLMS [33].	27
Tableau 4.1: Peseudo code de l'algorithme NLMS avec la technique de l'appartenance à un ensemble étendu au cas stéréophonique (ES-SM-NLMS) [48].	32
Tableau 4.2: Peseudo code de l'algorithme NLMS avec à pas variable non paramétrique étendu au cas stéréophonique SNP-VSS-N [49].	35
Tableau 4.3: Peseudo code de l'algorithme XMNL-NLMS avec la technique de l'appartenance à un ensemble étendu au cas stéréophonique (SM-PU-SNLMS) [48].	37
Tableau 4.4: Peseudo code de l'algorithme XMNL-NLMS avec pas variable non paramétrique étendu au cas stéréophonique (SNP-VSS-PU-N) [50].	39
Tableau 5.1 : Complexité de calcul des algorithmes proposés	62
Tableau 5.2: Synthèse de l'évaluation des performances des algorithmes proposés	62
Tableau 6.1 : Modification requise pour les algorithmes complexes	68
Tableau 6.2: Les paramètres de réglage SFNLMS	74

LISTE DES ABREVIATIONS

- AEA** Acoustic Echo Cancellation.
- CLB** blocs logiques configurables.
- EQM** Erreur quadratique moyenne.
- ERLE** amélioration de perte de retour d'écho.
- ES-SM-NLMS** l'algorithme NLMS avec la technique de l'appartenance à un ensemble étendu au cas stéréophonique.
- FIR** Finite Impulse Response.
- FNLMS** Fast Normalized Least Mean Square.
- FPGA** réseau de portes programmables (Field Programmable GateArray).
- FTF** Fast Transversal Filter.
- HDL** Hardware Description Language.
- IIR** Infinite Impulse Response.
- LMS** Least Mean Square.
- NLMS** Normalized Least Mean Square.
- NP-VSS** un pas variable non paramétrique en anglais (non-parametric variable step size)
- PU** Partial Update.
- RLS** Recursive Least Square.
- RSB** rapport signal sur bruit.
- SAEA** Stereophonic Acoustic Echo Cancellation Stereophonic.
- SFNLMS** L'algorithme du gradient stochastique normalisé rapide stéréophonique.
- SNLMS** Stereophonic NLMS Normalized Least Mean Square.
- SNP-VSS-N** l'algorithme NLMS à pas variable non paramétrique étendu au cas stéréophonique.

SNP-VSS-PU-N l'algorithme XMNL-NLMS à pas variable non paramétrique étendu au cas stéréophonique.

SM-PU-SNLMS l'algorithme XMNL-NLMS avec la technique de l'appartenance à un ensemble étendu au cas stéréophonique.

SMF Set-Membership Filtering.

TNL Traitement non linéaire

USASI United State of America Standard Institut.

VHDL Very High Description Language

VHSIC Very High Speed Integrated Circuit

WGN White Gaussian Noise.

XMNL-NLMS Gradient stochastique normalisat non linéaire avec maximum exclusif.

INTRODUCTION GENERALE

Le phénomène d'annulation d'écho, tant dans les systèmes monophonique que stéréophonique, n'est pas une nouveauté [2-4]. Malgré ses racine historiques, ce domaine demeure un terrain vaste et productif pour des recherches avancées pour objectif d'amélioration. L'amélioration continue des techniques et des algorithmes du traitement numérique du signal confirme des perfectionnements progressifs dans la qualité du signal, jour après jour [1-2].

L'écho acoustique est un phénomène lié à la réflexion du son qui diminue la qualité de communication. Si des réflexions multiples arrivent à un auditeur à partir d'un son émis et qu'il est incapable de faire la distinction entre les deux, le signal original sera perdu. Donc, il est nécessaire d'utiliser des dispositifs d'annulation d'écho avec notre system de communication. En plus, l'annulation d'écho acoustique stéréophonique (AEAS) n'est pas aussi simple que dans le cas monophonique à cause de l'apparition des nouveaux problèmes.

Les systèmes de téléconférence stéréophoniques offrent une présence plus réaliste que les systèmes monophoniques. La nécessité de disposer de plusieurs canaux pour le son stéréo crée un problème d'intercorrélation entre deux canaux différents [5]. Dans la suite de notre travail, nous considérons le cas d'un système de téléconférence stéréophonique à deux canaux utilisant deux haut-parleurs et deux microphones du côté de la réception.

Le problème fondamental d'AEAS et la non-unicité des coefficients de filtrage, qui ne converge pas vers des vraies estimations des réponses de trajet d'écho. Cela pose le problème de l'unicité de solution à l'identification du chemin d'écho [6-8]. Ce problème est dû à la corrélation entre les signaux d'entrées. Ainsi, les techniques adaptatives utilisées n'identifient pas les réponses correctes du trajet d'écho. Cependant, ce problème peut être contourné en utilisant des techniques de décorrélation pour décorréler les signaux stéréo à l'entrée des haut-parleurs afin d'augmenter l'intelligibilité de la parole sans affecter la perception stéréo.

Les solutions les plus anciennes à ce problème ont été proposées dans [3-5]. Dans la référence [3], il a été souligné que dans une situation réelle ; les filtres adaptatifs convergent vers le vrai trajet d'écho en raison des caractéristiques de l'environnement acoustique dans la salle de transmission. Une petite non-linéarité dans chaque canal a été introduite dans le papier [4] pour réduire la cohérence entre les canaux afin de ne pas affecter la qualité du signal de la parole. Dans [5], les auteurs décrivent le problème de non-unicité des solutions dans l'AEAS, ils expliquent les points faibles de certaines solutions proposées et prouvent que le filtre combiné constitue un meilleur choix pour réduire la cohérence inter-canaux et obtenir la solution unique et appropriée.

Les premiers modèles de mise en œuvre de l'AEAS ont été illustrés dans les références [6-8]. Ces solutions ont été apportées avant que la théorie AEAS ne soit pleinement comprise.

Récemment, de nombreuses approches pour résoudre l'AEAS ont été proposées. Dans [9], afin d'améliorer la solution existée, les auteurs proposent une solution hybride utilisant les décalages de fréquence lors de l'ajout d'un correcteur demi-onde. Dans [2,10,11], les auteurs utilisent les algorithmes efficaces existants pour améliorer les performances des résultats du système d'AEAS. D'autres techniques ont été proposées dans [12-14] en utilisant un modèle largement linéaire. De plus, les auteurs de [15-17] choisissent de travailler dans un domaine différent (domaine fréquentiel, domaine temps-fréquence, domaine temps-échelle) au lieu d'un domaine temporaire pour accéder aux informations cachées et obtenir plus de contrôle sur le système AEAS.

Le filtre adaptatif est la partie critique d'AEAS qui évalue le trajet d'écho de la salle pour obtenir une réplique du signal d'écho [9]. Il faut une mise à jour adaptative pour s'adapter aux changements environnementaux, par exemple des salles de conférences où de nombreuses personnes parlent. Un problème important du filtre adaptatif est la vitesse de convergence, qui mesure la rapidité avec laquelle le filtre converge vers la meilleure estimation du chemin acoustique de la salle.

Dans cette thèse, des techniques et algorithmes efficaces sont présentés. Le premier algorithme proposé est une version étendue de l'algorithme FNLMS pour le cas stéréophonique avec une amélioration notable de la partie de prédiction qui devient stable lorsque l'on traite le cas stéréophonique. De plus, quatre nouveaux algorithmes sont introduits, dérivés combinaison de deux techniques : la technique de sélection des données et la technique de mise à jour partielle (PU). Ces derniers sont appliqués à un algorithme adaptatif à deux canaux SNLMS, et prennent en compte uniquement les données d'entrée qui apportant de nouvelles informations au système. Par conséquent. Cette approche réduit efficacement la complexité du calcul. A la fin de ce travail, l'algorithmes SNLMS et SFNLMS sont implémentés en code VHDL pour une application FPGA.

Le présent manuscrit est présenté selon la répartition suivante :

- ❖ Dans le premier chapitre, nous présentons en premier lieu, le modèle de l'annulation d'écho acoustique par le filtrage adaptatif, ainsi que les approches et les structures d'adaptation. En deuxième lieu, nous présentons les grandes familles d'algorithmes adaptatifs.

- ❖ Dans le deuxième chapitre, nous abordons le problème de l'annulation d'écho acoustique stéréophonique. En premier lieu, nous présentons le modèle tel que la configuration et les équations en utilisant l'algorithme classique NLMS stéréophonique. En deuxième lieu, nous présentons les défis rencontrés et ses solutions disponibles.

- ❖ Dans le troisième chapitre, nous proposons une nouvelle version stéréophonique de l'algorithme FNLMS. L'algorithme proposé SFNLMS est une version étendue de l'algorithme FNLMS monophonique au cas stéréophonique

avec une amélioration importante à la partie prédiction qui maintient la stabilité dans le cas stéréophonique.

❖ Dans le quatrième chapitre, nous présentons quatre nouvelles versions d'algorithmes adaptatifs pour l'annulation d'écho acoustique stéréophonique. Ces algorithmes sont essentiellement basés sur la combinaison des deux techniques ; la technique de la sélection des données et la technique de mise à la jour partielle des coefficients des filtres, par ces techniques la complexité est réduite et les performances d'annulation sont améliorées.

❖ Le cinquième chapitre est réservé pour l'étude comparative des performances et la validation expérimentale des différents résultats théoriques.

❖ Dans le sixième chapitre, nous proposons une implémentation efficace et simple des algorithmes SNLMS et SFNLMS en VHDL pour la FPGA, puis nous confirmons la conception proposée en utilisant des outils offerts par Matlab et ModelSim.

❖ Enfin, nous terminerons notre thèse par une conclusion générale.

CHAPITRE 1: GENERALITES SUR LE FILTRAGE ADAPTATIF ET L'ANNULATION DE L'ÉCHO ACOUSTIQUE

1.1. Introduction

Dans les systèmes de télécommunication, l'écho peut entraîner une erreur de transmission de données volumineuses, et peut rendre les conversations impossibles. Ainsi, l'écho a été classé comme un gros problème dans les réseaux de communication [18]. Cette situation devient plus problématique dans le cas de système de communications stéréophoniques.

1.2. L'Écho

Il existe deux types d'écho dans les réseaux de communication :

Écho électrique : ce type d'écho est dû au décalage d'impédance en divers points du support de transmission. Cet écho se retrouve dans les réseaux de téléphonie publiques [19, 20].

Écho acoustique : ce type est lié ou systèmes de communication mains libres. L'onde sonore se propage d'un haut-parleur au microphone via les vibrations du circuit ou l'écho généré à l'air libre. Des exemples de tels systèmes sont les téléphones portables, les appels VOIP en utilisant, par exemple Skype, la téléconférence, etc [21].

1.3. L'Annulation d'Écho Acoustique

L'écho acoustique se produit lorsqu'un signal audio est réverbéré dans un environnement fermé tel que les salles de conférences. Le signal d'écho est la combinaison du signal source atténué et des images retardées de celui-ci.

Les filtres adaptatifs sont des filtres efficaces qui modifient de manière itérative leurs coefficients de filtre afin d'obtenir une sortie optimale. La fonction d'erreur qui est la différence entre le signal désiré $d(n)$ et la sortie filtrée $y(n)$ est minimisée algorithmiquement par le filtre adaptatif en modifiant les coefficients. Cette fonction est également appelée fonction coût du filtre adaptatif. La figure 1.1 illustre le schéma synoptique de la structure de base de l'annulation d'écho acoustique. Le filtre adaptatif est utilisé dans le chemin de retour désigné par $w(n)$. Le rôle du filtre adaptatif est de minimiser l'erreur entre le signal souhaité (c'est-à-dire le signal réverbéré dans l'environnement acoustique) et la sortie du filtre $y(n)$. Le signal d'erreur $e(n) = d(n) - y(n)$ sert à orienter les coefficients de filtrage pour qu'ils convergent rapidement vers la valeur optimale qui dépend des signaux d'entrée [22].

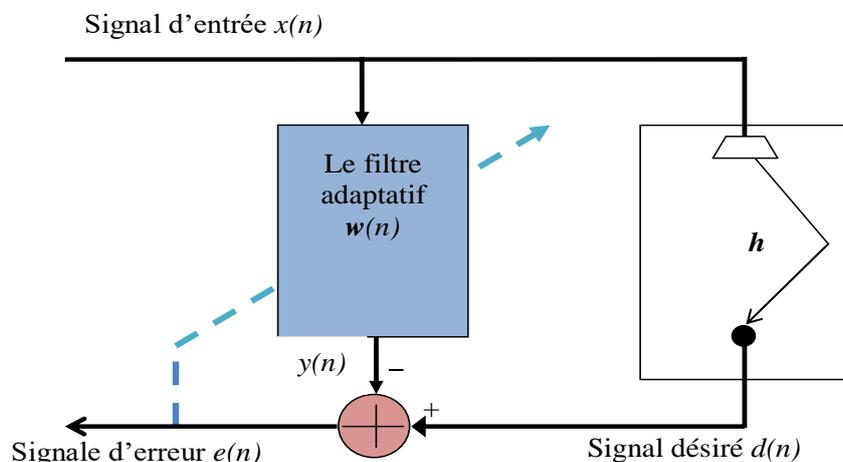


Figure 1.1: La structure de base de l'AEA

L'objectif principal du filtre adaptatif est donc d'estimer les réponses impulsionnelles de l'environnement acoustique par le calcul de la différence $e(n)$ entre le signal désiré et la sortie du filtre adaptatif. Ce signal d'erreur est renvoyé au filtre avec lequel ses coefficients sont convergents selon une équation de mise à jour afin de minimiser cette erreur. Dans le cas de l'AEA, la valeur optimale de la sortie du filtre adaptatif $y(n)$ est égale au signal désiré $d(n)$, par conséquent, le signal d'erreur devient nul. Dans cette situation particulière, nous voulons que l'utilisateur en émission ne reçoive pas la parole d'origine.

1.4. Les filtres adaptatifs

Dans le cas de systèmes variant dans le temps et pour lesquels les spécifications ne sont pas disponibles, la solution consiste à utiliser un filtre numérique à coefficients adaptatifs appelé filtre adaptatif. Puisque aucune spécification n'est disponible, l'algorithme adaptatif qui détermine la mise à jour des coefficients de filtre nécessite des informations supplémentaires sous la forme d'un signal appelé signal de référence ou signal désiré $d(n)$. La configuration générale d'un environnement de filtrage adaptatif est illustrée à la figure 1.2 ci-dessous. $x(n)$ désigne le signal d'entrée, $y(n)$ est le signal de sortie du filtre adaptatif et $d(n)$ définit le signal désiré ou le signal de référence. Le signal d'erreur $e(n)$ est calculé par $d(n) - y(n)$. Il est utilisé pour adapter les coefficients de filtrage, ce qui implique que le filtre adaptatif transmette des signaux correspondant au signal désiré dans un certain sens [23].

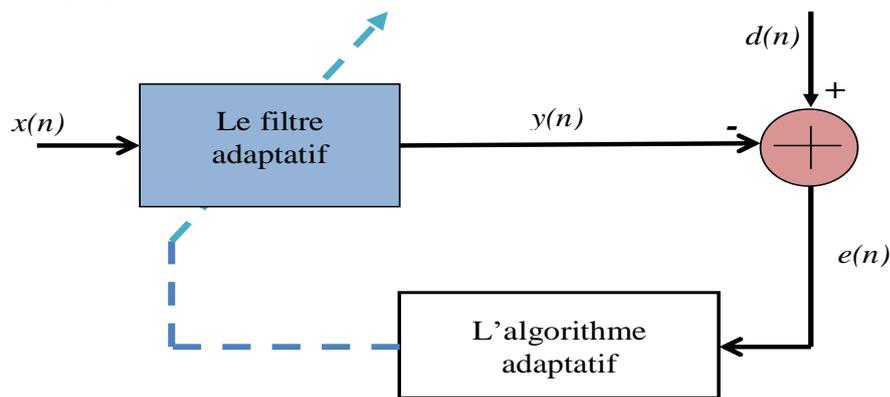


Figure 1.2. Configuration générale du filtre adaptatif

Le type d'application du filtre adaptatif est défini par le choix des signaux acquis de l'environnement comme signaux d'entrée et de sortie désirée. Certains exemples sont l'annulation d'écho, l'égalisation des canaux dispersifs, identification du système, l'amélioration du signal, l'annulation du bruit et contrôle [24-27].

1.4.1. Approches d'adaptation

Il existe deux approches stochastique et déterministe [28] distinctes et qui sont largement utilisées dans le développement de divers algorithmes d'adaptation. Ces deux approches ont de nombreuses variantes dans leurs implémentations, ce qui conduit à une grande variété d'algorithmes ; dont chacun offre ses propres caractéristiques souhaitables.

A. Approche basée sur la théorie du filtre de Wiener

Le filtre de Wiener [29] est dérivé du critère de l'erreur quadratique moyenne (EQM), et les coefficients optimaux d'un filtre linéaire sont obtenus par minimisation de son EQM. Plusieurs algorithmes adaptatifs proposés sont à base de ce filtre, tel que l'algorithme du gradient stochastique (least-mean square «LMS») et ses variantes, ces algorithmes sont

largement utilisés dans diverses applications de filtrage adaptatif. Cette famille d'algorithmes utilise la valeur instantanée du carré du signal d'erreur comme estimation d'EQM.

B. Méthode des moindres carrés

La méthode des moindres carrés permet de résoudre le problème de l'optimisation des filtres d'un point de vue déterministe [28]. L'indice de performance est la somme des carrés d'erreur pondérés par un facteur d'oubli. Par contre, comme mentionné précédemment, dans l'approche de filtre de Wiener, le filtre est obtenu en minimisant l'EQM, c'est-à-dire une quantité statistique. Dans le contexte des filtres adaptatifs, on préfère les formulations récursives de la méthode de moindres carrés qui mettent à jour les coefficients de filtre à l'arrivée de chaque échantillon d'entrée, donc on parle de l'algorithme de moindres carrés récursifs (Recursive Least Squares : RLS).

1.4.2. Les structures de filtre adaptatif

La structure la plus utilisée dans la mise en œuvre des filtres adaptatifs est la structure transversale, illustrée à la figure 1.3. On se réfère également à un filtre à réponse impulsionnelle finie (RIF) car sa réponse impulsionnelle est d'une durée finie [28]. Il a une seule entrée $x(n)$, et une sortie $y(n)$. La séquence $d(n)$ est le signal désiré. La sortie $y(n)$ est générée par la combinaison linéaire des échantillons retardés de la séquence d'entrée $x(n)$, donnée par l'équation (1.1)

$$y(n) = \sum_{i=0}^{L-1} w_i(n)x(n-i) \quad (1.1)$$

Où $w_i(n)$ sont les coefficients du filtre et L la longueur du filtre.

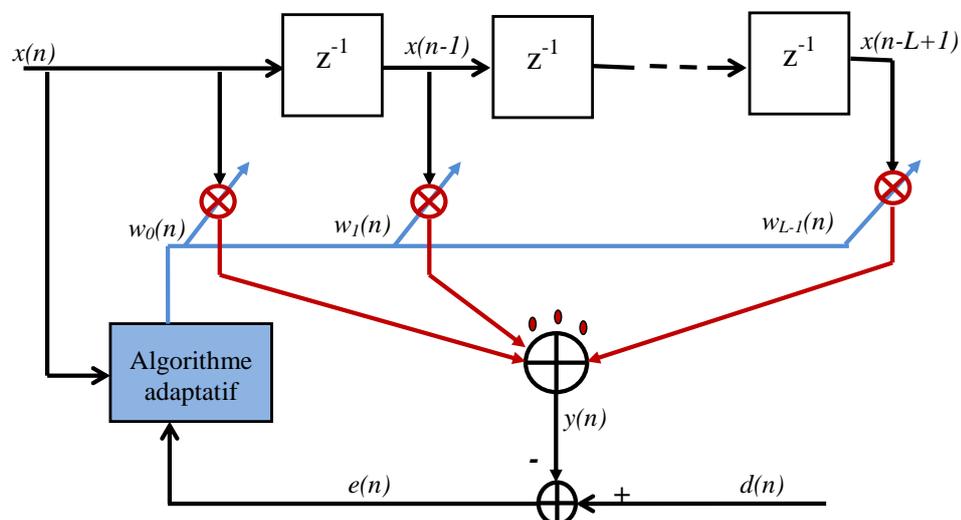


Figure 1.3. Structure d'un filtre FIR.

Les coefficients du filtre $w_i(n)$ sont contrôlés par l'algorithme d'adaptation.

Une autre structure peut se trouver dans certaines applications s'appeler le combinatoire linéaire. Les entrées de filtre ne sont pas les échantillons retardés d'une seule entrée. En effet, la sortie $y(n)$ du filtre est une combinaison linéaire des différents signaux reçus à son entrée [28]. La forme est illustrée à la figure 1.4 et donnée par l'équation (1.2).

$$y(n) = \sum_{i=0}^{L-1} w_i(n)x_i(n) \quad (1.2)$$

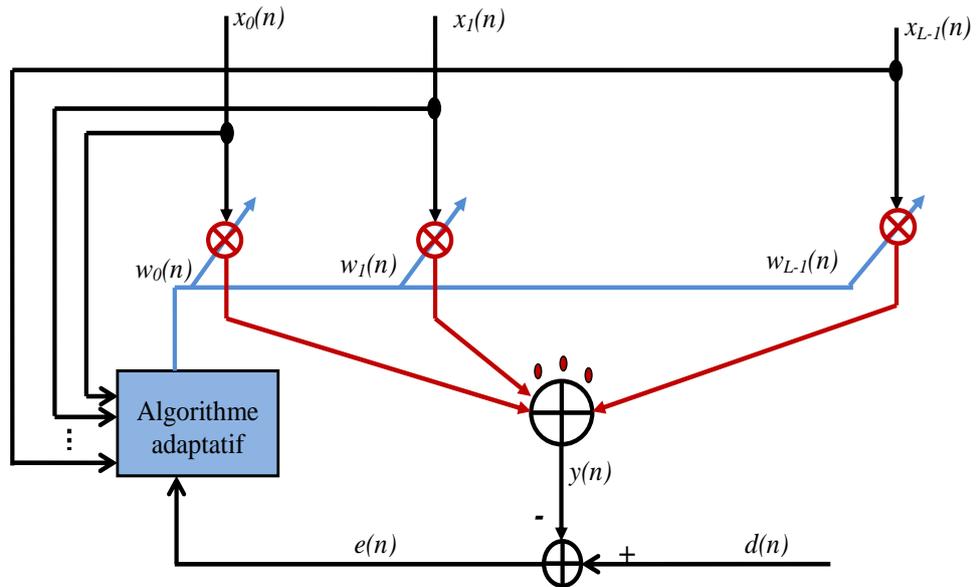


Figure 1.4. Combinateur linéaire adaptatif.

Dans ces deux structures (filtre FIR et le combineur linéaire), le calcul de la sortie du filtre n'entraîne aucun mécanisme de retour, c'est-à-dire une forme non récursive. Une autre structure appelée filtre à réponse impulsionnelle infinie (IIR) dans laquelle les zéros et les pôles du filtre peuvent être adaptés [28]. Il est régi par des équations récursives telles que :

$$y(n) = \sum_{i=0}^{L-1} a_i(n)x(n-i) + \sum_{j=1}^{L-1} b_j(n)y(n-j) \quad (1.3)$$

Sa forme est illustrée à la figure 1.5. où $a_i(n)$ et $b_i(n)$ sont les pondérations d'avant et de retour, respectivement. Le filtre IIR a été utilisé dans de nombreuses applications mais son utilisation est assez limitée dans le domaine du filtrage adaptatif [28], à cause de nombreuses difficultés liées à l'adaptation de filtre IIR.

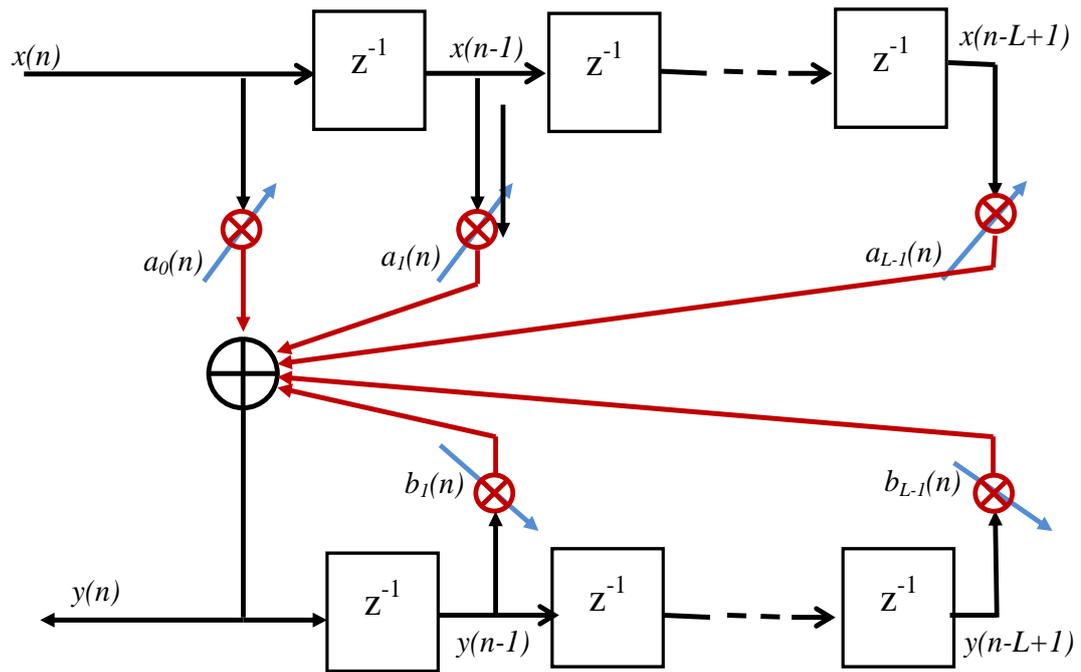


Figure 1.5. Sstructure d'un filtre IIR.

1.5. Les algorithmes adaptatifs

Les algorithmes adaptatifs permettent de calculer les coefficients du filtre $\mathbf{w}(n)$ de façon récursive et ceci est obtenu par la minimisation de la différence $e(n)$ entre le signal désiré $d(n)$ et la sortie du filtre $y(n)$ au sens d'un critère statistique préalablement défini.

Le choix du signal d'erreur $e(n)$ pour la définition d'un algorithme joue un rôle important sur les performances des algorithmes telles que la vitesse de convergence et la complexité de calcul.

Dans cette section, nous allons présenter certains algorithmes les plus utilisés dans le domaine du filtrage adaptatif, nous allons aussi présenter, en détails, la formulation de l'algorithme qui allons utiliser pour proposer et développer un nouvel algorithme dans le contexte d'annulation d'écho acoustique stéréophonique (AEAS).

1.5.1. Algorithme du gradient stochastique LMS

L'algorithme du gradient stochastique LMS a été proposé en 1960 par Widrow et Hoff [69]. Il est l'algorithme de filtrage adaptatif le plus utilisé en pratique, à cause de sa simplicité. Cet algorithme utilise une structure transversale du filtre comme illustrée à la figure 1.3.

Comme mentionné précédemment, l'algorithme LMS utilise la valeur instantanée du carré d'erreur $e^2(n)$ comme estimation de l'EQM. Ce dernier est appelé aussi la fonction coût dénommée $J = E[e^2(n)]$, donc son estimation est $\hat{J} = e^2(n)$. Le signal d'erreur en notation vectorielle est donné par :

$$e(n) = d(n) - y(n) = d(n) - \mathbf{w}^T(n-1)\mathbf{X}(n) \quad (1.4)$$

où $\mathbf{X}(n) = [x(n), x(n-1), \dots, x(n-L+1)]^T$ est un vecteur contenant les L derniers échantillons de signal d'entrée $x(n)$. Le paramètre $\mathbf{w}(n) = [w_1(n), w_2(n), \dots, w_L(n)]^T$ est le vecteur du coefficient estimé du filtre. L'exposant T désigne l'opérateur de transposition. La méthode de gradient consiste à [30] :

- Choisir un vecteur initial $\mathbf{w}(0)$.
- Obtenir, à partir d'un vecteur $\mathbf{w}(n-1)$ précédant, le vecteur $\mathbf{w}(n)$ par incrémentation de $\mathbf{w}(n-1)$, dans la direction opposée du gradient de la fonction coût estimée \hat{J} .

$$\mathbf{w}(n) = \mathbf{w}(n-1) - \mu \nabla e^2(n) \quad (1.5)$$

Où μ est le pas d'adaptation (constante positive), et ∇ est l'opérateur du gradient, c'est-à-dire $\nabla = \frac{\partial}{\partial \mathbf{w}}$, donc on peut calculer \hat{J} par l'expression suivante :

$$\hat{J} = \frac{\partial e^2(n)}{\partial \mathbf{w}} = 2e(n) \frac{\partial e(n)}{\partial \mathbf{w}} = -2e(n) \frac{\partial y(n)}{\partial \mathbf{w}} = -2e(n) \mathbf{X}(n) \quad (1.6)$$

D'où l'équation de mise à jour (1.4) se réécrit sous la forme suivante :

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \mu e(n) \mathbf{X}(n) \quad (1.7)$$

La condition nécessaire et suffisante de convergence de l'algorithme LMS est [31] :

$$0 < \mu \leq \frac{1}{L\sigma_x^2} \quad (1.8)$$

Où σ_x^2 désigne l'énergie de signal d'entrée $\mathbf{x}(n)$. L'algorithme LMS représente une méthode simple pour calculer les coefficients de filtre d'une manière adaptative. Mais il souffre d'une vitesse de convergence lente qui n'est pas cherchée dans certaines applications. L'algorithme LMS est résumé et illustré dans le tableau 1.1.

Tableau 1.1. Pseudo-code de l'algorithme du gradient stochastique LMS.

Entrées
<ul style="list-style-type: none"> • Signal d'entrée $\mathbf{X}(n)$. • Signal désiré $d(n)$.
Sorties :
<ul style="list-style-type: none"> • Les coefficients de filtre $\mathbf{w}(n)$. • La sortie de filtre $y(n)$. • L'erreur de filtrage $e(n)$.
Filtrage : $y(n) = \mathbf{w}^T(n-1)\mathbf{X}(n)$
Estimation d'erreur : $e(n) = d(n) - y(n)$
Adaptation : $\mathbf{w}(n) = \mathbf{w}(n-1) + \mu e(n)\mathbf{X}(n)$

1.5.2. L'algorithme du gradient stochastique normalisé NLMS

L'algorithme du gradient stochastique normalisé NLMS est une variante du LMS dont le gain d'adaptation est normalisé par l'énergie du signal d'entrée $\mathbf{X}(n)$; donc il suffit de remplacer le gain d'adaptation constant μ dans l'algorithme LMS par le gain variable $\mu(n)$, tel que :

$$\mu(n) = \frac{\alpha}{\mathbf{X}^T(n)\mathbf{X}(n)} \quad (1.9)$$

Où α représente le pas d'adaptation de l'algorithme NLMS, la condition suffisante pour choisir α est [31] :

$$1 < \alpha < 2$$

L'algorithme NLMS est résumé et illustré dans le Tableau 1.2.

Tableau 1.2. Pseudo-code de l'algorithme du gradient stochastique normalisé NLMS

Entrées	
<ul style="list-style-type: none"> • Signal d'entrée $\mathbf{X}(n)$. • Signal désiré $d(n)$. 	
Sorties	
<ul style="list-style-type: none"> • Les coefficients de filtre $\mathbf{w}(n)$. • La sortie de filtre $y(n)$. • L'erreur de filtrage $e(n)$. 	
<hr/>	
Filtrage : $y(n) = \mathbf{w}^T(n-1)\mathbf{X}(n)$ Estimation d'erreur : $e(n) = d(n) - y(n)$ Adaptation : $\mathbf{w}(n) = \mathbf{w}(n-1) + \frac{\alpha}{\mathbf{X}^T(n)\mathbf{X}(n)} e(n)\mathbf{X}(n)$	
<hr/>	

L'algorithme NLMS est aussi simple comme l'algorithme LMS. Il est numériquement stable à cause du gain normalisé. Mais il peut poser des problèmes quand le signal d'entrée est un signal de parole. En effet, le transitoire dépend de la dispersion spectrale du signal d'entrée.

1.5.3. L'algorithme des moindres carrés récursifs RLS

Dans la méthode des moindres carrés, les paramètres du filtre sont optimisés en utilisant toutes les observations jusqu'à l'instant n . D'où, on vise à minimiser un critère défini sur l'erreurs commise depuis l'instant initial, par rapport au vecteur du paramètre $\mathbf{w}(n)$. Ce critère est donné par [28] :

$$J(n) = \sum_{k=1}^n \lambda^{n-k} e_n^2(k) \quad (1.10)$$

Où k est l'instant de début de fonctionnement de l'algorithme, et n l'instant actuel. Le paramètre λ ($0 < \lambda < 1$) est un facteur d'oubli exponentiel qui permet à l'algorithme d'oublier le passé trop lointain. Ce facteur donne plus de poids aux échantillons récents des estimations d'erreur. On réécrit $J(n)$ par sa forme vectorielle.

$$J(n) = \mathbf{e}^T(n)\mathbf{\Lambda}(n)\mathbf{e}(n) \quad (1.11)$$

Où $\mathbf{e}(n) = [e(1), e(2), \dots, e(n)]^T$ est un vecteur contenant les n échantillons d'erreur, et $\mathbf{\Lambda}(n)$ est une matrice diagonale contenant les facteurs de poids tel que :

$$\mathbf{\Lambda}(n) = \begin{bmatrix} \lambda^{n-1} & 0 & 0 & \dots & 0 \\ 0 & \lambda^{n-2} & 0 & \dots & 0 \\ 0 & 0 & \lambda^{n-3} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix} \quad (1.12)$$

On développe $J(n)$, utilisant (1.4), on trouve :

$$J(n) = [\mathbf{d}(n) - \mathbf{y}(n)]^T \mathbf{\Lambda}(n) [\mathbf{d}(n) - \mathbf{y}(n)] \quad (1.13)$$

$$J(n) = \mathbf{d}(n)^T \mathbf{\Lambda}(n) \mathbf{d}(n) - 2\mathbf{d}(n)^T \mathbf{\Lambda}(n) \mathbf{y}(n) + \mathbf{y}(n)^T \mathbf{\Lambda}(n) \mathbf{y}(n) \quad (1.14)$$

Où $\mathbf{d}(n) = [d(1), d(2), \dots, d(n)]^T$ est le signal désiré et $\mathbf{y}(n) = [y(1), y(2), \dots, y(n)]^T$ est la sortie du filtre. On pose, $\mathbf{U}(n) = [\mathbf{X}(1), \mathbf{X}(2), \dots, \mathbf{X}(n)]$, la matrice d'observation du signal d'entrée, donc on peut récrire (1.14) par :

$$J(n) = \mathbf{d}(n)^T \mathbf{\Lambda}(n) \mathbf{d}(n) - 2\mathbf{d}(n)^T \mathbf{\Lambda}(n) \mathbf{U}^T(n) \mathbf{w}(n) \mathbf{w}(n)^T \mathbf{U}(n) \mathbf{\Lambda}(n) \mathbf{U}(n)^T \mathbf{w}(n) \quad (1.15)$$

$$J(n) = \mathbf{d}(n)^T \mathbf{\Lambda}(n) \mathbf{d}(n) - 2\boldsymbol{\theta}_\lambda^T(n) \mathbf{w}(n) + \mathbf{w}(n)^T \boldsymbol{\Psi}_\lambda(n) \mathbf{w}(n) \quad (1.16)$$

Où $\boldsymbol{\theta}_\lambda(n) = \mathbf{d}(n) \mathbf{\Lambda}(n) \mathbf{U}(n)$ et $\boldsymbol{\Psi}_\lambda(n) = \mathbf{U}(n) \mathbf{\Lambda}(n) \mathbf{U}(n)^T$.

En établissant le gradient de $J(n)$ par rapport $\mathbf{w}(n)$ égale a zéro, on obtient [28] :

$$\mathbf{w}(n) = \boldsymbol{\Psi}_\lambda^{-1}(n) \boldsymbol{\theta}_\lambda(n) \quad (1.17)$$

En écrivant $\boldsymbol{\Psi}_\lambda(n)$ et $\boldsymbol{\theta}_\lambda(n)$ en fonction de $\mathbf{X}(n)$ par :

$$\boldsymbol{\Psi}_\lambda(n) = \mathbf{X}(n) \mathbf{X}^T(n) + \lambda \mathbf{X}(n-1) \mathbf{X}^T(n-1) + \lambda^2 \mathbf{X}(n-2) \mathbf{X}^T(n-2) + \dots \quad (1.18)$$

$$\boldsymbol{\theta}_\lambda(n) = \mathbf{X}(n) d(n) + \lambda \mathbf{X}(n-1) d(n-1) + \lambda^2 \mathbf{X}(n-2) d(n-2) + \dots \quad (1.19)$$

Utilisant (1.18) et (1.19) $\boldsymbol{\Psi}_\lambda(n)$ et $\boldsymbol{\theta}_\lambda(n)$ peut se calculer d'une manière récursive :

$$\boldsymbol{\Psi}_\lambda(n) = \lambda \boldsymbol{\Psi}_\lambda(n-1) + \mathbf{X}(n) \mathbf{X}^T(n) \quad (1.20)$$

$$\boldsymbol{\theta}_\lambda(n) = \lambda \boldsymbol{\theta}_\lambda(n-1) + \mathbf{X}(n) d(n) \quad (1.21)$$

En appliquant à la matrice $\boldsymbol{\Psi}_\lambda(n)$ le lemme d'inversion matricielle suivante [32] :

$$(\mathbf{A} + \alpha \mathbf{u} \mathbf{u}^T)^{-1} = \mathbf{A}^{-1} - \frac{\alpha \mathbf{A}^{-1} \mathbf{u} \mathbf{u}^T \mathbf{A}^{-1}}{1 + \alpha \mathbf{u}^T \mathbf{A}^{-1} \mathbf{u}} \quad (1.22)$$

Tel que \mathbf{A} est une matrice carrée $L \times L$ arbitraire non singulière, \mathbf{u} un vecteur de $L \times 1$ et α un scalaire. En posant $\mathbf{A}(n) = \lambda \boldsymbol{\Psi}_\lambda(n)$, $\mathbf{u}(n) = \mathbf{X}(n)$ et $\alpha = 1$, Il vient :

$$\Psi_{\lambda}^{-1}(n) = \lambda^{-1}\Psi_{\lambda}^{-1}(n-1) - \frac{\lambda^{-2}\Psi_{\lambda}^{-1}(n-1)\mathbf{X}(n)\mathbf{X}^T(n)\Psi_{\lambda}^{-1}(n-1)}{1 + \lambda^{-1}\mathbf{X}^T(n)\Psi_{\lambda}^{-1}(n-1)\mathbf{X}(n)} \quad (1.23)$$

Pour simplifier, on définit le vecteur $\mathbf{k}(n)$ qui appelé le gain de Kalman par :

$$\mathbf{k}(n) = \frac{\lambda^{-1}\Psi_{\lambda}^{-1}(n-1)\mathbf{X}(n)}{1 + \lambda^{-1}\mathbf{X}^T(n)\Psi_{\lambda}^{-1}(n-1)\mathbf{X}(n)} \quad (1.24)$$

L'équation (1.23) devient après l'utilisation du gain de Kalman :

$$\Psi_{\lambda}^{-1}(n) = \lambda^{-1}[\Psi_{\lambda}^{-1}(n-1) - \mathbf{k}(n)\mathbf{X}^T(n)\Psi_{\lambda}^{-1}(n-1)] \quad (1.25)$$

En utilisant l'équation (1.24) et l'équation (1.25), on obtient :

$$\mathbf{k}(n) = \Psi_{\lambda}^{-1}(n)\mathbf{X}(n) \quad (1.26)$$

En substituant l'équation (1.21) dans (1.17) et on utilise (1.26), on obtient :

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \mathbf{k}(n)[d(n) - \mathbf{w}^T(n-1)\mathbf{X}(n)] \quad (1.27)$$

On définit l'erreur *a priori* suivante :

$$\varepsilon(n) = [d(n) - \mathbf{w}^T(n-1)\mathbf{X}(n)] \quad (1.28)$$

D'où l'équation (1.27) devient :

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \mathbf{k}(n)\varepsilon(n) \quad (1.27)$$

C'est la récursivité utilisée par l'algorithme RLS pour la mise à jour de $\mathbf{w}(n)$.

L'algorithme RLS est résumé dans le Tableau 1.3.

Tableau 1.3. Pseudo-code de l'algorithme des moindres carrés récursifs RLS

Entrées
<ul style="list-style-type: none"> • Signal d'entrée $\mathbf{X}(n)$. • Signal désiré $d(n)$.
Sorties :
<ul style="list-style-type: none"> • Coefficients de filtre $\mathbf{w}(n)$. • Sortie de filtre $y(n)$. • Erreur de filtrage $\varepsilon(n)$.
Calcul du vecteur de gain : $\mathbf{u}(n) = \Psi_{\lambda}^{-1}(n-1)\mathbf{X}(n)$, $\mathbf{k}(n) = \frac{1}{\lambda + \mathbf{X}^T(n)\mathbf{u}(n)} \mathbf{u}(n)$
Filtrage : $y(n) = \mathbf{w}^T(n-1)\mathbf{X}(n)$
Estimation d'erreur : $\varepsilon(n) = d(n) - y(n)$
Adaptation : $\mathbf{w}(n) = \mathbf{w}(n-1) + \mathbf{k}(n)\varepsilon(n)$
Mise à jour de $\Psi_{\lambda}^{-1}(n)$: $\Psi_{\lambda}^{-1}(n) = \lambda^{-1}[\Psi_{\lambda}^{-1}(n-1) - \mathbf{k}(n)\mathbf{X}^T(n)\Psi_{\lambda}^{-1}(n-1)]$

L'algorithme des moindres carrés récursifs (RLS) présente un coût de calcul élevé, à cause de la nécessité de l'inversion d'une matrice carrée d'ordre L , mais il possède une vitesse de convergence élevée par rapport aux algorithmes de type gradient stochastique.

1.6. Conclusion

Dans ce chapitre, nous avons présenté le modèle de l'annulation d'écho acoustique par les algorithmes du filtrage adaptatif. Les approches et les structures d'adaptation sont aussi présentées. Nous avons présenté les algorithmes adaptatifs de type gradient stochastique tels que LMS et sa version normalisée NLMS, et du type moindres carrés récursifs (RLS). Dans le prochain chapitre, nous allons présenter le modèle de l'annulation stéréophonique de l'écho acoustique et les équations de ces algorithmes dans ce nouveau contexte.

CHAPITRE 2: ÉTUDE DE L'ANNULATION STERÉOPHONIQUE D'ÉCHO ACOUSTIQUE

2.1. Introduction

Dans les systèmes de téléconférence les plus utilisés, la nécessité de disposer de plusieurs canaux pour le son stéréophonique en utilisant plusieurs haut-parleurs et microphones crée un problème de génération d'échos par diaphonie entre deux canaux différents [12]. Pour cela, on a besoin des annuleurs stéréophoniques de l'écho acoustique (AEAS). Contrairement à l'annulation de l'écho monophonique dans ce système de communication, nous devons trouver quatre chemins d'écho entre deux haut-parleurs et deux microphones, c'est-à-dire deux chemins directs et deux chemins de diaphonie [12]. Étant donné que l'utilisation du son stéréophonique offre une meilleure qualité sonore, la personne dans la salle de conférences peut facilement identifier et distinguer son locuteur.

2.2. La configuration de l'annulation de l'écho acoustique stéréophonique

L'annulation d'écho stéréophonique n'est rien d'autre qu'une généralisation du système d'annulation d'écho monophonique mais avec des problèmes qui sont fondamentalement différents des ceux de l'AEA à canal unique. La configuration générale de l'AEAS est donnée ci-dessous par la figure 2.1 [10,11,33]. Pour simplicité, un seul canal est présenté et une analyse similaire sera appliquée à un autre canal.

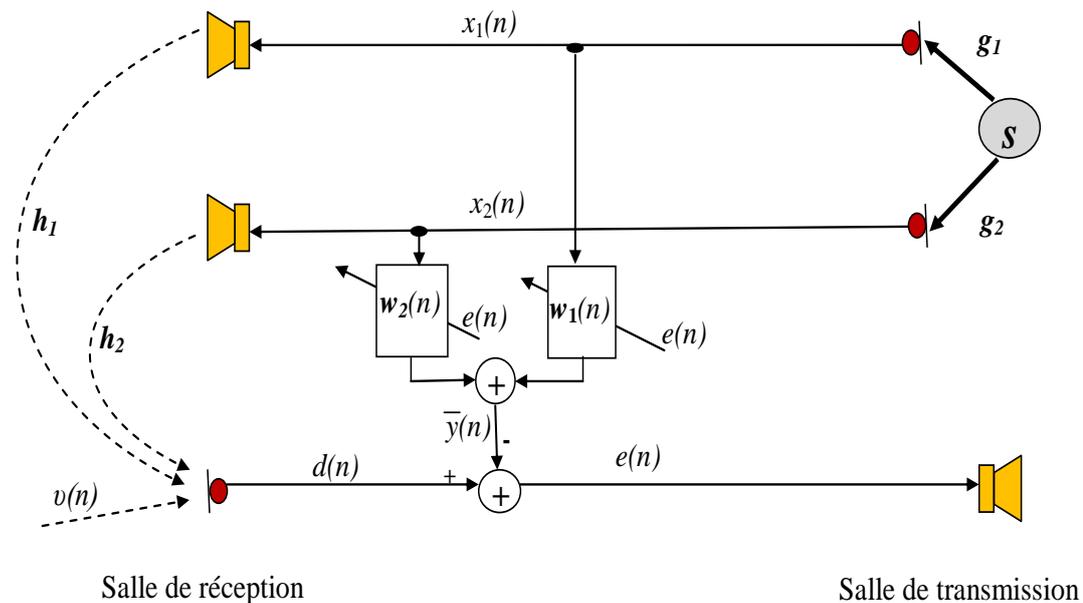


Figure 2.1. La configuration générale de l'annulation d'écho acoustique stéréophonique.

Selon la figure 2.1, la salle de transmission est à droite et la salle de réception est à gauche. Le couplage entre haut-parleurs et microphones dans la salle de réception conduit à la génération du signal d'écho. Dans cette configuration, les chemins d'écho acoustique h_1 et h_2 dans la salle de réception sont modélisés par les filtres FIR adaptatifs $w_1(n)$ et $w_2(n)$, respectivement. Cela produit un écho estimé $\bar{y}(n)$ du vrai écho $y(n)$, qui est soustrait du signal souhaité $d(n)$. En pratique, les vraies réponses impulsionnelles h_1 et h_2 ont une longueur infinie, cependant, les réponses impulsionnelles $w_1(n)$ et $w_2(n)$ ne le sont pas, bien que l'on suppose qu'elles sont suffisamment longues. De cette manière, nous ne

considérons que les premières parties de \mathbf{h}_1 et \mathbf{h}_2 qui contient le plus d'énergie et supposer être de la même taille que les filtre adaptatifs $\mathbf{w}_1(n)$ et $\mathbf{w}_2(n)$ de longueur L . Les paramètres \mathbf{g}_1 et \mathbf{g}_2 sont les réponses impulsionnelles de la salle d'émission, ils sont inclus dans le signal de référence $x_1(n)$ et $x_2(n)$. Le signal souhaité $d(n)$ contient le signal d'écho $y(n)$ et le signal du bruit de fond $v(n)$ est donné par :

$$d(n) = y(n) + v(n) \quad (2.1)$$

Le signal d'écho acoustique $y(n)$ est la convolution linéaire entre la sortie des haut-parleurs dans la salle de réception avec les chemins d'écho acoustique \mathbf{h}_1 et \mathbf{h}_2 , il est donné par:

$$y(n) = \mathbf{X}_1^T(n)\mathbf{h}_1 + \mathbf{X}_2^T(n)\mathbf{h}_2 \quad (2.2)$$

où $\mathbf{X}_i(n) = [x_i(n)x_i(n-1) \dots x_i(n-L+1)]^T$ sont les L dernières observations des signaux d'entrée avec $i = 1,2$. Les deux vecteurs $\mathbf{h}_i = [h_i(1) h_i(2) \dots h_i(L)]^T$ sont les deux réponses impulsionnelles des chemins d'écho de la salle de réception. L'écho estimé $\bar{y}(n)$ généré par les filtres adaptatifs est donné par :

$$\bar{y}(n) = \mathbf{X}_1^T(n)\mathbf{w}_1(n-1) + \mathbf{X}_2^T(n)\mathbf{w}_2(n-1) \quad (2.3)$$

où $\mathbf{w}_i(n) = [w_i(1,n)w_i(2,n) \dots w_i(L,n)]^T$ sont les coefficients des filtres adaptatifs. Le signal d'écho résiduel représente l'erreur linéaire *a priori* est donnée par :

$$e(n) = d(n) - \bar{y}(n) \quad (2.4)$$

Nous avons également défini l'erreur *a posteriori* $\varepsilon(n)$ comme suit :

$$\varepsilon(n) = d(n) - \mathbf{X}_1^T(n)\mathbf{w}_1(n) + \mathbf{X}_2^T(n)\mathbf{w}_2(n) \quad (2.5)$$

Les coefficients des filtres adaptatifs sont mis à jour, pour chaque nouvelle donnée entrante, par le retour du signal d'écho résiduel proportionnel aux gains d'adaptation notés $\mathbf{k}_1(n)$ et $\mathbf{k}_2(n)$ pour les canaux 1 et 2, respectivement.

$$\mathbf{w}_i(n) = \mathbf{w}_i(n-1) + \mathbf{k}_i(n)e(n) \quad (2.6)$$

Les gains d'adaptation $\mathbf{k}_1(n)$ et $\mathbf{k}_2(n)$ sont mis à jour en utilisant les différents algorithmes adaptatifs conçus pour cette tâche. Ils sont généralement définis en fonction des signaux d'entrée comme suit :

$$\mathbf{k}_i(n) = \mu(n)\mathbf{X}_i(n) \quad (2.7)$$

Où $\mu(n)$ est un scalaire positif introduit pour contrôler la variation le long de la direction choisie, il diffère d'un algorithme à un autre. Par exemple, dans l'algorithme LMS a deux canaux, la variable $\mu(n) = \alpha$, où α est un constant positif représente le pas d'adaptation et choisie entre $0 < \alpha < 2$. Dans l'algorithme NLMS a deux canaux (aussi appelé stéréophonique NLMS) (SNLMS), ce paramètre qui dépend du signal d'entrée est donné par [34] :

$$\mu(n) = \frac{\alpha}{\mathbf{X}_1^T(n)\mathbf{X}_1(n) + \mathbf{X}_2^T(n)\mathbf{X}_2(n) + c_0} \quad (2.8)$$

où c_0 est une constante positive, elle est introduite pour éviter la division par zéro. L'algorithme NLMS stéréophonique est résumé dans le Tableau 2.1.

Tableau 2.1. Pseudo-code de l'algorithme du gradient stochastique normalisé stéréophonique SNLMS [34]

Entrées
<ul style="list-style-type: none"> • Signal d'entrée $\mathbf{X}_1(n)$ et $\mathbf{X}_2(n)$ • Signal désiré $d(n)$.
Sorties :
<ul style="list-style-type: none"> • Les coefficients de filtre $\mathbf{w}_1(n)$ et $\mathbf{w}_2(n)$. • La sortie de filtre $\bar{y}(n)$. • L'erreur de filtrage $e(n)$.
Filtrage : $\bar{y}(n) = \mathbf{X}_1^T(n)\mathbf{w}_1(n-1) + \mathbf{X}_2^T(n)\mathbf{w}_2(n-1)$
Estimation d'erreur : $e(n) = d(n) - \mathbf{X}_1^T(n)\mathbf{w}_1(n-1) - \mathbf{X}_2^T(n)\mathbf{w}_2(n-1)$
Adaptation :
$\mu(n) = \frac{\alpha}{\mathbf{X}_1^T(n)\mathbf{X}_1(n) + \mathbf{X}_2^T(n)\mathbf{X}_2(n) + c_0}$
$\mathbf{k}_i(n) = \mu(n)\mathbf{X}_i(n)$
$\mathbf{w}_i(n) = \mathbf{w}_i(n-1) + \mathbf{k}_i(n)e(n)$

2.3. Le problème de non-unicité de solution

Dans l'AEAS, le problème de non-unicité est dû au fait que les signaux de référence hautement corrélés sont filtrés linéairement à partir de la même source [5]. Par conséquent, il y a quatre chemins indépendants à estimer, à cause de ceci, il est difficile d'estimer les réponses impulsionnelle \mathbf{h}_1 et \mathbf{h}_2 . Dans ce scénario et selon la figure 2.1, nous avons la relation suivante [4,35] :

$$x_i(n) = \mathbf{g}_i(n) * s(n) \quad (2.9)$$

où $i = 1,2$ et $\mathbf{g}_i(n)$ sont les chemins d'écho de salle de transmission de longueur M . La solution de la fonction coût (1.10) est obtenue via le système d'équations linéaires [34] suivante :

$$\mathbf{R}_{xx,N}(n) \begin{bmatrix} \mathbf{w}_1(n) \\ \mathbf{w}_2(n) \end{bmatrix} = \mathbf{r}_{yx}(n) \quad (2.10)$$

où $\mathbf{r}_{yx}(n) = \sum_{k=1}^n \lambda^{n-k} y(k) \begin{bmatrix} X_1(n) \\ X_2(n) \end{bmatrix}$ est le vecteur d'inter-corrélation et $\mathbf{R}_{xx}(n)$ est la matrice de corrélation (matrice carré de taille N)

$$\mathbf{R}_{xx,N}(n) = \sum_{k=1}^n \lambda^{n-k} \begin{bmatrix} \mathbf{X}_1(k) \\ \mathbf{X}_2(k) \end{bmatrix} [\mathbf{X}_1^T(k) \quad \mathbf{X}_2^T(k)] \quad (2.11)$$

Le problème fondamental de l'annulation d'écho acoustique stéréophonique est lié au conditionnement de la matrice de corrélation. Si nous définissons l'erreur de désajustement (Misalignment) comme étant :

$$Mis(n) = \frac{\|\mathbf{h} - \mathbf{w}(n)\|^2}{\|\mathbf{h}\|^2} \quad (2.12)$$

avec $\mathbf{h} = [\mathbf{h}_1^T \quad \mathbf{h}_2^T]^T$ Et $\mathbf{w} = [\mathbf{w}_1^T(n) \quad \mathbf{h}_2^T(n)]^T$. Même si l'erreur de désajustement est importante, il est parfois possible d'obtenir une bonne annulation d'écho. Pourtant si les signaux d'entrée changent, cela ne sera plus possible. Dans le cas monophonique, ceci peut être évité par la longueur appropriée du filtre adaptatif et de la réponse impulsionnelle. En stéréophonie, cela devient bien pire en raison de la forte corrélation entre les signaux d'entrée [36]. Il peut être montré que [34] :

$$\begin{aligned} L \geq M &\rightarrow \mathbf{R}_{xx,N}(n), \text{ est singulière } \forall n \\ L < M &\rightarrow \mathbf{R}_{xx,N}(n), \text{ est mal conditionnée} \\ L \geq N &\rightarrow Mis(n) = 0, n \geq N \\ L < N &\rightarrow Mis(n) \neq 0, \forall n \end{aligned} \quad (2.13)$$

L'équation (2.13) est valide dans la situation où aucun bruit n'est ajouté ou signal du microphone $y(n)$. Les deux dernières conditions de (2.13) exigent que $\mathbf{R}_{xx,N}(n)$ soit non singulière.

Les longueurs des réponses impulsionnelles dans les salles de transmission et de réception jouent un rôle important au problème de non-unicité. Théoriquement, M et N sont infiniment longs. Nous supposons que $L < N$ et $L < M$ pour le cas réel pourra être traité. Il a été montré dans [34] qu'il y a deux moyens de diminuer l'erreur de désajustement. La première méthode consiste à utiliser des filtres adaptatifs plus longs, mais cela introduit de façon proportionnelle une dégradation de la vitesse de convergence et une augmentation de complexité arithmétique et d'espace mémoire. Une seconde approche pratique, est de décorréler partiellement au complètement les signaux d'entrée, mais cela doit être traité sans dégrader la qualité du signal.

2.4. Technique de décorrélation des signaux

Les signaux stéréophoniques sont linéairement liés et il existe une forte corrélation entre ces signaux. Afin de réduire ceci, différentes méthodes ont été proposées pour la réduction cette cohérence. L'une de ces méthodes est basée sur le prétraitement des signaux d'entrée par l'introduction d'une fonction non linéaire. Cette technique doit être effectuée avec soin pour ne pas affecter la qualité du son. Une première méthode simple, qui donne des bonnes performances, consiste à utiliser un redresseur à demi-onde [5,10-12,33-35]. Ceci est nommé méthode 1 de traitement non linéaire (TNL1) et la relation non linéaire est donnée par :

$$x_i(n) = s_i(n) + \alpha_r \frac{s_i(n) + |s_i(n)|}{2}, \quad i = 1,2 \quad (2.14)$$

où $s_i(n)$ sont les signaux d'entrée avant la transformation. $x_i(n)$ sont les signaux d'entrée après la transformation et α_r est une constante pour contrôler la non-linéarité, on

recommande de la choisir entre 0.3 et 0.5. La configuration générale du système d'AEAS après l'introduction d'une phase de prétraitement est montrée à la figure 2.2 donnée ci-dessous :

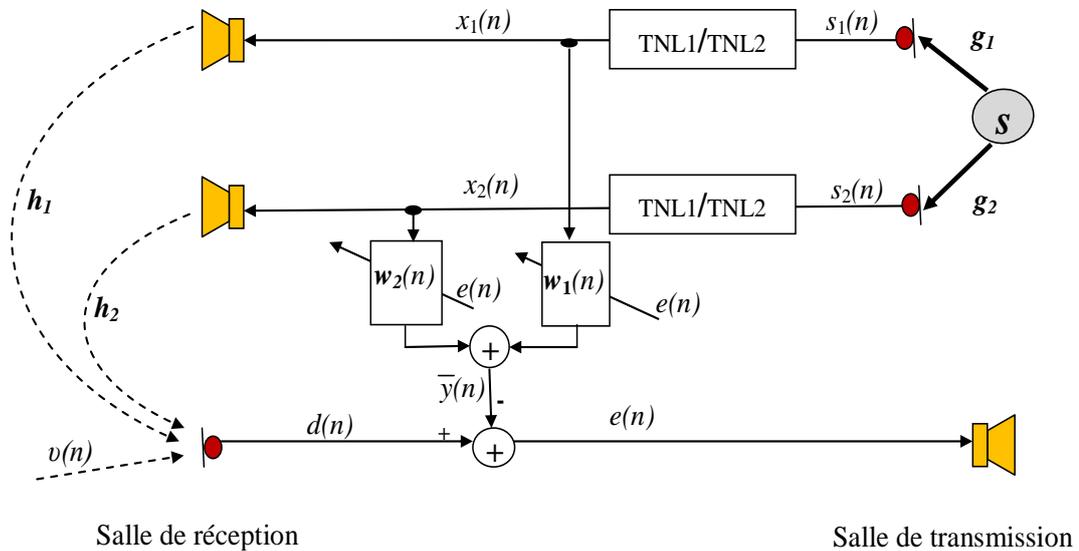


Figure 2.2. Configuration d'un système stéréophonique avec prétraitement.

Dans cette structure, la relation linéaire peut ne pas être complètement annulée. Par exemple si :

$$\begin{aligned} \forall n \quad & s_1(n) \geq 0 \text{ et } s_2(n) \geq 0 \\ & \text{ou} \\ & \alpha_r s_1(n - \tau_1) = s_2(n - \tau_2) \text{ avec } \alpha_r > 0 \end{aligned} \quad (2.15)$$

En pratique, la condition (2.15) ne se produit jamais car :

- Les signaux d'entrée sont des signaux à moyennes nulles.
- Les signaux d'entrée ne sont jamais liés par un simple retard.

Pour perfectionner et éviter ce cas, nous utilisons une version améliorée de cette technique [11]. En utilisant des redresseurs demi-onde positif et négatif un pour chaque canal, ou lieu d'utiliser uniquement le positif dans les deux transformations. Cette deuxième méthode avec la modification est nommée traitement non linéaire 2 (TNL2), et la relation est donnée par :

$$\left\{ \begin{array}{l} x_j(n) = s_j(n) + \alpha_r \frac{s_j(n) + |s_j(n)|}{2} \\ j = 1, 2 \end{array} \right\} \quad (2.16)$$

De cette façon, la relation linéaire est bien réduite selon la valeur du paramètre α_r .

Une troisième méthode pour la réduction de la cohérence entre les signaux d'entrée consiste à utiliser le système TNL2 avec une mise à jour partielle des coefficients des

filtres adaptatifs. L'un des algorithmes proposés pour cette tâche est l'algorithme du gradient stochastique normalisé non linéaire avec maximum exclusif (XMNL-NLMS) [37]. L'algorithme XMNL-NLMS peut être décrit par un traitement non linéaire 2 (TNL2) appliqué aux vecteurs d'entrée $s_1(n)$ et $s_2(n)$ afin de générer les signaux transmis $x_1(n)$ et $x_2(n)$, respectivement. Comme bien mentionné dans l'équation (2.16), la mise à jour partielle est réalisée par l'introduction d'un nouveau paramètre agissant sur les gains d'adaptation, l'équation (2.7) peut être réécrite comme suit :

$$\mathbf{k}_i(n) = \mu(n)\mathbf{Q}_i(n)\mathbf{X}_i(n) \quad (2.17)$$

Où $\mathbf{Q}_i(n) = \text{diag}\{\mathbf{q}_i(n)\}$ est une matrice de sélection de prises de $L \times L$. Où les éléments de $\mathbf{q}_i(n)$ sont donnés par :

$$q_{1,u}(n) = \begin{cases} 1, & p_u \in \{Z \text{ maxima de } \mathbf{p}(n)\} \\ 0, & \text{autrement} \end{cases} \quad (2.18)$$

$$q_{2,v}(n) = \begin{cases} 1, & p_v \in \{Z \text{ minima de } \mathbf{p}(n)\} \\ 0, & \text{autrement} \end{cases} \quad (2.19)$$

$$\mathbf{p}(n) = |\mathbf{x}_1(n)| - |\mathbf{x}_2(n)| \quad (2.20)$$

Les paramètres $u, v = 1, 2, \dots, L$ représentent l'élément u de q_1 et l'élément v de q_2 et le paramètre $0 < Z < L$, un bon taux de convergence est obtenu en choisissant $Z = 0.5 L$. L'algorithme XMNL-NLMS est résumé et illustré dans le Tableau 2.2.

Tableau 2.2. Pseudo-code de l'algorithme XMNL-NLMS [37].

Entrées

- Signal d'entrée $\mathbf{X}_1(n)$ et $\mathbf{X}_2(n)$
- Signal désiré $d(n)$.

Sorties :

- Les coefficients de filtre $\mathbf{w}_1(n)$ et $\mathbf{w}_2(n)$
 - La sortie de filtre $\bar{y}(n)$.
 - L'erreur de filtrage $e(n)$.
-

Filtrage : $\bar{y}(n) = \mathbf{X}_1^T(n)\mathbf{w}_1(n-1) + \mathbf{X}_2^T(n)\mathbf{w}_2(n-1)$

Estimation d'erreur : $e(n) = d(n) - \bar{y}(n)$

Adaptation :

$$\mathbf{p}(n) = |\mathbf{X}_1(n)| - |\mathbf{X}_2(n)|$$

$$q_{1,u}(n) = \begin{cases} 1, & p_u \in \{Z \text{ maxima de } \mathbf{p}(n)\} \\ 0, & \text{autrement} \end{cases}, \quad q_{2,v}(n) = \begin{cases} 1, & p_v \in \{Z \text{ minima de } \mathbf{p}(n)\} \\ 0, & \text{autrement} \end{cases}$$

$$\mathbf{Q}_i(n) = \text{diag}\{\mathbf{q}_i(n)\}, \quad \mu(n) = \frac{\alpha}{\mathbf{X}_1^T(n)\mathbf{X}_1(n) + \mathbf{X}_2^T(n)\mathbf{X}_2(n) + c_0}, \quad \mathbf{k}_i(n) = \mu(n)\mathbf{Q}_i(n)\mathbf{X}_i(n)$$

Mise à jour : $\mathbf{w}_i(n) = \mathbf{w}_i(n-1) + \mathbf{k}_i(n)e(n)$

La technique de sélection partielles avec maximum exclusif a été appliquée à l'algorithme SNLMS classique pour la réduction de cohérence. Cette technique est valable pour tous les algorithmes adaptatifs, et par ceci, on peut améliorer la décorrélation, faite préalablement et uniquement par l'étape du prétraitement seul.

2.5. Conclusion

Dans ce chapitre, nous avons présenté le modèle de l'annulation d'écho acoustique stéréophonique via le filtrage adaptatif en utilisant l'algorithme SNLMS comme exemple. Nous avons remarqué que l'extension du cas monophonique au cas stéréophonique fait apparaître le problème de non-unicité de solution de filtrage et cela est dû au couplage acoustique entre les canaux stéréophoniques. Ce qui nous a amené à utiliser les techniques de décorrélation pour résoudre ce problème, tel que, la technique de prétraitement non linéaire et la technique de sélection partielle avec maximum exclusif. Cette technique a bien boosté les performances des algorithmes adaptatifs dans l'identification et la convergence vers la solution optimale et unique.

CHAPITRE 3: L'Algorithme de Gradient Stochastique Normalisé Rapide (FNLMS) pour l'Annulation Stéréophonique d'Écho Acoustique

3.1. Introduction

Dans le domaine de l'annulation d'écho acoustique, plusieurs algorithmes adaptatifs ont été proposés récemment, pour améliorer les performances d'annulation, tel que, la vitesse, la stabilité de convergence, et la complexité de calcul. Parmi ces algorithmes nous intéressons à l'algorithme de gradient stochastique normalisé rapide (FNLMS) [38]. La principale motivation de ce choix est les bonnes performances de l'algorithme FNLMS en matière de vitesse de convergence et de faible complexité de calcul.

Dans ce chapitre, nous proposons une nouvelle version stéréophonique de l'algorithme FNLMS (SFNLMS) [33]. L'algorithme proposé est une version étendue de l'algorithme FNLMS au cas stéréophonique avec une amélioration importante de la partie prédiction qui maintient la stabilité dans le cas stéréophonique. L'idée de base de l'algorithme proposé SFNLMS est la normalisation du paramètre de prédiction par la variance des entrées pour chaque canal.

3.2. Description de l'algorithme adaptatif FNLMS à un canal

L'algorithme FNLMS est dérivé de l'algorithme de filtre transversal rapide (FTF) [38]. Il est basé sur l'erreur de prédiction d'avant (forward) du signal d'entrée au lieu d'utiliser les prédicteurs d'avant (forward) et d'arrière (backward). L'erreur de filtrage adaptatif est donnée par les relations suivantes :

$$e(n) = d(n) - \mathbf{w}^T(n-1)\mathbf{X}(n) \quad (3.1)$$

$$d(n) = \mathbf{h}^T\mathbf{X}(n) \quad (3.2)$$

Où $\mathbf{h}^T = [h_0 h_1 \dots h_{L-1}]^T$ est la réponse impulsionnelle réelle, et $\mathbf{w}^T(n) = [w_0 w_1 \dots w_{L-1}]^T$ est la réponse impulsionnelle estimée "visait à être égale à la réponse réelle" et L est la longueur du filtre adaptatif.

L'équation de mise à jour de la réponse impulsionnelle estimée est donnée par :

$$\mathbf{w}(n) = \mathbf{w}(n-1) - \mathbf{k}(n)e(n) \quad (3.3)$$

Où $\mathbf{k}(n)$ est le vecteur de gain d'adaptation défini comme suit:

$$\mathbf{k}(n) = \gamma(n)\tilde{\mathbf{k}}(n) \quad (3.4)$$

Le paramètre $\gamma(n)$ est la variable de vraisemblance et $\tilde{\mathbf{k}}(n)$ est le gain dual de Kalman, ce dernier est mis à jour selon $\mathbf{a}(n)$ et $\mathbf{b}(n)$ respectivement appelés prédicteurs d'avant et prédicteur d'arrière qui sont utilisés et définis pour la première fois dans les algorithmes FTF d'origine [39-41].

$$\begin{bmatrix} \tilde{\mathbf{k}}(n) \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \tilde{\mathbf{k}}(n-1) \end{bmatrix} + \frac{\varepsilon(n)}{\lambda\alpha(n-1)} \begin{bmatrix} 1 \\ -\mathbf{a}(n-1) \end{bmatrix} + \frac{r(n)}{\lambda\beta(n-1)} \begin{bmatrix} -\mathbf{b}(n-1) \\ 1 \end{bmatrix} \quad (3.5)$$

Dans l'équation (3.5), $\varepsilon(n)$ et $r(n)$ sont les erreurs de prédiction d'avant et d'arrière, respectivement. Les paramètres $\alpha(n)$ et $\beta(n)$ sont les variances d'erreur de prédiction d'avant et d'arrière, respectivement.

L'algorithme FNLMS utilise uniquement l'erreur de prédiction d'avant du signal d'entrée et élimine complètement les prédicteurs d'avant et d'arrière. La relation (3.5) peut réduire à :

$$\begin{bmatrix} \tilde{\mathbf{k}}(n) \\ k(n) \end{bmatrix} = \begin{bmatrix} \frac{\varepsilon(n)}{\lambda\alpha(n-1)+c_0} \\ \tilde{\mathbf{k}}(n-1) \end{bmatrix} \quad (3.6)$$

Dans cette relation, c_0 est un petit nombre positif pour éviter la division par zéro, et la variance de l'erreur de prédiction d'avant peut être écrite comme suit :

$$\alpha(n) = \lambda\alpha(n-1) + \varepsilon^2(n) \quad (3.7)$$

Par conséquent, $\varepsilon(n)$ peut être calculé en utilisant n'importe quel algorithme de prédiction adaptative indépendamment du gain de Kalman, l'opération de prédiction produit $\varepsilon(n)$ à partir d'une séquence de $x(n)$ (signal d'entrée). Cette technique est connue sous le nom de filtre de blanchiment ou de décorrélation [42]. Le modèle de prédiction de premier ordre est utilisé pour obtenir un algorithme efficace en calcul. $\varepsilon(n)$ est donc donné par :

$$\varepsilon(n) = x(n) - ax(n-1) \quad (3.8)$$

La variable a est un paramètre de prédiction, et elle peut être constante si nous avons suffisamment d'informations a priori sur le signal d'entrée $x(n)$. Par exemple, $a = 0$ pour un bruit blanc et pour un bruit USASI (USA Standard Institut), le paramètre de prédiction est d'environ $a = 0.925$. Cependant, dans la plupart des cas, les statistiques du signal d'entrée sont inconnues. Par conséquent, le paramètre de prédiction peut être estimé à partir du signal d'entrée, ce qui minimise la fonction de coût $E(\varepsilon^2(n))$ en le fixant à zéro, l'équation (3.8) donne :

$$a = \frac{E[x(n)x(n-1)]}{E[x^2(n-1)]} = \frac{r_1}{r_0} \quad (3.9)$$

Où r_0 est la puissance du signal d'entrée et r_1 est le premier retard de la fonction d'autocorrélation. Pour estimer le paramètre de prédiction $a(n)$ qui peut être effectué par :

$$a(n) = \frac{r_1(n)}{r_0(n)+c_a} \quad (3.10)$$

Où c_a est une constante positive comparable à la puissance du signal d'entrée, $r_1(n)$ et $r_0(n)$ peuvent être estimés de manière récursive comme suit :

$$r_1(n) = \lambda_a r_1(n-1) + x(n)x(n-1) \quad (3.11)$$

$$r_0(n) = \lambda_a r_0(n-1) + x^2(n) \quad (3.12)$$

Où λ_a est un facteur d'oubli exponentiel. Dans ce développement, la variable de vraisemblance $\gamma(n)$ est définie par :

$$\gamma(n) = \frac{1}{1-\tilde{\mathbf{k}}(n)^T \mathbf{X}(n)} \quad (3.13)$$

Et en utilisant les propriétés d'invariance de décalage temporel du signal d'entrée étendu à $L + 1$, on a :

$$\mathbf{X}_*^T(n) = [\mathbf{X}^T(n)x(n-L)] \quad (3.14)$$

$$\mathbf{X}_*^T(n) = [x(n)\mathbf{X}^T(n-1)] \quad (3.15)$$

Où * indique que le vecteur est de taille $L + 1$. En multipliant le côté gauche et le côté droit de l'équation (3.6) par le côté gauche des équations (3.14) et (3.15) respectivement, nous obtenons l'équation suivante :

$$\mathbf{X}^T(n)\tilde{\mathbf{k}}(n) + k(n)x(n-L) = \mathbf{X}^T(n-1)\tilde{\mathbf{k}}(n-1) - \frac{x(n)\varepsilon(n)}{\lambda.\alpha(n-1)+c_0} \quad (3.16)$$

De (3.13), on a $\tilde{\mathbf{k}}(n)^T\mathbf{X}(n) = 1 - \gamma^{-1}(n)$ et en substituant ce dernier à l'équation (3.16), nous obtenons l'équation de mise à jour récursive suivante de $\gamma(n)$:

$$\gamma^{-1}(n) = \gamma^{-1}(n-1) + k(n)x(n-L) + \frac{x(n)\varepsilon(n)}{\lambda.\alpha(n-1)+c_0} \quad (3.17)$$

La relation (3.17) peut également réécrite sous la forme récursive suivante :

$$\gamma(n) = \frac{\gamma(n-1)}{1+\gamma(n-1)\delta(n)} \quad (3.18)$$

Où

$$\delta(n) = k(n)x(n-L) + \frac{x(n)\varepsilon(n)}{\lambda.\alpha(n-1)+c_0} \quad (3.19)$$

L'algorithme NLMS rapide désigné par (FNLMS) est résumé et illustré dans le Tableau 3.1.

Tableau 3.1 : Pseudo-code de l'algorithme du gradient stochastique normalisé rapide
FNLMS [38]

Les entrées

- Signal d'entrée $\mathbf{X}(n)$.
- Signal désiré $d(n)$.

Les sorties :

- Les coefficients de filtre $\mathbf{w}(n)$.
- La sortie de filtre $\bar{y}(n)$.

L'erreur de filtrage $e(n)$.

Initialisations:
 $\mathbf{k}(0) = \mathbf{0}, \mathbf{w}(0) = \mathbf{0}, \gamma(0) = 1, r_1(0) = 0, \alpha(0) = 1, r_0(0) = 1.$

Partie de prédiction :

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

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

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

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

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

Calcul du gain d'adaptation :

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

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

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

$$\mathbf{k}(n) = \gamma(n)\tilde{\mathbf{k}}(n)$$

Filtrage

$$\bar{y}(n) = \mathbf{w}^T(n-1)\mathbf{X}(n)$$

Estimation d'erreur

$$e(n) = d(n) - \bar{y}(n)$$

Equation de mise à jour

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

On note que le pas d'adaptation μ a été introduit pour mieux contrôler le gain d'adaptation de la partie filtrante.

Dans cette première partie de ce chapitre, l'algorithme FNLMS appliqué à l'AEA à monovoie est présenté. Le gain d'adaptation pour l'algorithme FNLMS est obtenu par la simplification du calcul du gain d'adaptation de l'algorithme FTF. La quantité de calcul, soit $2L$ multiplications par échantillon, est la même pour l'algorithme FNLMS que pour l'algorithme NLMS, donc par la suite, on peut prendre ce dernier comme référence dans la simulation.

3.3. Proposition de l'algorithme FNLMS stéréophoniques (SFNLMS)

Dans cette section, nous développons l'extension stéréophonique de l'algorithme du gradient stochastique normalisé rapide (SFNLMS). Pour ce faire, la partie prédiction est générée deux fois, le premier traitement du signal d'entrée du premier et deuxième canal. En plus, la variable de vraisemblance est générée deux fois pour les deux canaux. Nous avons deux filtres adaptatifs et un seul vecteur d'erreur, voir la figure 2.2. L'erreur de filtrage est donnée par :

$$e(n) = d(n) - \bar{y}(n) \quad (3.20)$$

Où $d(n)$ et $\bar{y}(n)$ sont l'écho stéréophonique de la salle de réception et son estimé, respectivement. Selon la figure 2.2, l'écho de la salle de réception (en suppose que le bruit de fond $u(n)$ est nul) est donné par la relation suivante :

$$d(n) = \mathbf{h}_1^T \mathbf{X}_1(n) + \mathbf{h}_2^T \mathbf{X}_2(n) \quad (3.20)$$

Et la relation de l'écho estimé est donnée comme suite :

$$\bar{y}(n) = \mathbf{w}_1^T(n-1) \mathbf{X}_1(n) + \mathbf{w}_2^T(n-1) \mathbf{X}_2(n) \quad (3.21)$$

L'équation de mise à jour des filtres adaptatifs est donnée par :

$$\mathbf{w}_i^T(n) = \mathbf{w}_i^T(n-1) - \mu e(n) \mathbf{k}_i(n), \quad i = 1,2 \quad (3.22)$$

Où μ est le pas d'adaptation et les vecteurs gains d'adaptation sont définis comme suite :

$$\mathbf{k}_i(n) = \gamma_i(n) \tilde{\mathbf{k}}_i(n), \quad i = 1,2 \quad (3.23)$$

Où $\gamma_i(n)$ sont les variables de vraisemblance et $\tilde{\mathbf{k}}_i(n)$ sont les vecteurs de gains de Kalman. Ces paramètres sont fournis par les parties de prédiction suivantes :

$$\begin{bmatrix} \tilde{\mathbf{k}}_i(n) \\ k_i(n) \end{bmatrix} = \begin{bmatrix} \frac{\varepsilon_i(n)}{\lambda \alpha_i(n-1) + c_0} \\ \tilde{\mathbf{k}}_i(n-1) \end{bmatrix}, \quad i = 1,2 \quad (3.24)$$

L'opération de prédiction produit respectivement $\varepsilon_1(n)$ et $\varepsilon_2(n)$ à partir du signal d'entrée des canaux un et deux. Où, la variance des erreurs de prédiction d'avant peut être calculée comme suite :

$$\alpha_i(n) = \lambda \alpha_i(n-1) + \varepsilon_i^2(n), \quad i = 1,2 \quad (3.25)$$

Les paramètres d'erreur peuvent être calculés à l'aide du modèle de prédiction de premier ordre, afin d'obtenir un algorithme efficace en matière de calcul.

$$\varepsilon_i(n) = x_i(n) - a_i x_i(n-1), \quad i = 1,2 \quad (3.26)$$

Où a_i sont les paramètres de prédiction. Nous avons noté que la relation (3.10) n'est pas valable pour le cas stéréophonique de l'algorithme SFNLMS et pourrait être une source de divergence. Afin de surmonter ce problème de divergence, nous proposons de normaliser la relation (3.10) non seulement par la puissance d'un seul canal, mais avec les puissances de deux canaux telles que données par la relation suivante :

$$a_i(n) = \frac{r_{1,i}(n)}{r_{01}(n) + r_{02}(n) + c_a}, \quad i = 1,2 \quad (3.27)$$

Où c_a est une constante positive comparable à la puissance du signal d'entrée, $r_{1,i}(n)$ sont les premiers retards des matrices d'autocorrélation des canaux un et deux, respectivement. $r_{01}(n)$ et $r_{02}(n)$ sont les puissances des signaux d'entrée des canaux un et deux, respectivement. Cette modification proposée permet à l'algorithme SFNLMS de converger rapidement vers la solution optimale sans divergence.

Nous notons que non seulement la puissance du signal d'entrée du canal correspondant est suffisante pour estimer les paramètres de prédiction, bien que les puissances des signaux d'entrée des deux canaux soient utilisées pour estimer chaque paramètre de prédiction. Ces puissances et ces matrices d'autocorrélation peuvent être estimées de manière récursive comme ils sont mentionnés ci-dessous :

$$r_{1,i}(n) = \lambda_a r_{1,i}(n-1) + x_i(n)x_i(n-1), \quad i = 1,2 \quad (3.28)$$

$$r_{0i}(n) = \lambda_a r_{0i}(n-1) + x_i^2(n), \quad i = 1,2 \quad (3.29)$$

Où λ_a est un facteur d'oubli exponentiel.

En utilisant les équations (18) et (19), les deux variables de vraisemblance sont données par:

$$\gamma_i(n) = \frac{\gamma_i(n-1)}{1 + \gamma_i(n-1)\delta_i(n)}, \quad i = 1,2 \quad (3.30)$$

Où

$$\delta_i(n) = k_i(n)x_i(n-L) + \frac{x_i(n)\varepsilon_i(n)}{\lambda_a \alpha_i(n-1) + c_0}, \quad i = 1,2 \quad (3.31)$$

L'algorithme SFNLMS proposé est résumé et illustré dans le Tableau 3.2.

Tableau 3.2 : Pseudo-code de l'algorithme FNLMS stéréophonique (SFNLMS) [33].

Les entrées

- Signal d'entrée $\mathbf{X}_1(n)$ et $\mathbf{X}_2(n)$
- Signal désiré $d(n)$.

Les sorties :

- Les coefficients de filtre $\mathbf{w}_1(n)$ et $\mathbf{w}_2(n)$.
- La sortie de filtre $\bar{y}(n)$.

L'erreur de filtrage $e(n)$.

Initialisations :

$$\mathbf{k}_i(0) = \mathbf{0}, \mathbf{w}_i(0) = \mathbf{0}, \gamma_i(0) = 1, r_{1,i}(0) = 0, \alpha_i(0) = 1, r_{0i}(0) = 1,$$

Partie de prédiction :

$$i = 1, 2$$

$$r_{1,i}(n) = \lambda_a r_{1,i}(n-1) + x_i(n)x_i(n-1)$$

$$r_{0i}(n) = \lambda_a r_{0i}(n-1) + x_i^2(n)$$

$$a_i(n) = \frac{r_{1,i}(n)}{r_{01}(n) + r_{02}(n) + c_a}$$

$$\varepsilon_i(n) = x_i(n) - a_i x_i(n-1)$$

$$\alpha_i(n) = \lambda \alpha_i(n-1) + \varepsilon_i^2(n)$$

Calcul du gain d'adaptation :

$$\begin{bmatrix} \tilde{\mathbf{k}}_i(n) \\ k_i(n) \end{bmatrix} = \begin{bmatrix} \varepsilon_i(n) \\ \lambda \alpha_i(n-1) + c_0 \\ \tilde{\mathbf{k}}_i(n-1) \end{bmatrix}$$

$$\delta_i(n) = k_i(n)x_i(n-L) + \frac{x_i(n)\varepsilon_i(n)}{\lambda \alpha_i(n-1) + c_0}$$

$$\gamma_i(n) = \frac{\gamma_i(n-1)}{1 + \gamma_i(n-1)\delta_i(n)}$$

$$\mathbf{k}_i(n) = \gamma_i(n)\tilde{\mathbf{k}}_i(n)$$

Filtrage

$$\bar{y}(n) = \mathbf{X}_1^T(n)\mathbf{w}_1(n-1) + \mathbf{X}_2^T(n)\mathbf{w}_2(n-1)$$

Estimation d'erreur

$$e(n) = d(n) - \bar{y}(n)$$

Equation de mise à jour de $\mathbf{w}_i(n)$

$$\mathbf{w}_i(n) = \mathbf{w}_i(n-1) - \mu e(n)\mathbf{k}_i(n).$$

Dans cette deuxième partie de ce chapitre, l'algorithme SFNLMS adapté l'AEAS à deux canaux est proposé et présenté. La formulation de l'algorithme SFNLMS est obtenue en étendant ses équations au cas stéréophonique avec une modification nécessaire pour assurer la stabilité de l'algorithme SFNLMS dans sa nouvelle forme.

3.4. Conclusion

Dans ce chapitre, nous avons proposé un nouvel algorithme stéréophonique noté SFNLMS. Ce nouvel algorithme est dérivé de l'algorithme FNLMS proposé dans le cas de l'AEA monophonique. La complexité de calcul est la même pour l'algorithme proposé que pour l'algorithme SNLMS classique, dû à la simplification du calcul du gain d'adaptation de l'algorithme FTF. Dans la suite de ce document, et afin de confronter cette étude, nous présenterons dans le chapitre 5 les résultats de simulations obtenus par la comparaison de l'algorithme proposé avec l'algorithme SNLMS classique dans le cas d'AEAS.

CHAPITRE 4: UN NOUVEL ALGORITHME ADAPTATIF SELECTIF DES DONNEES POUR L'AEA STEREOPHONIQUE

4.1. Introduction

Récemment dans le domaine de l'AEA, il a été proposé plusieurs approches basées sur des structures de filtrage sélectif pour réduire la complexité de calcul et améliorer les performances d'adaptation. Par exemple, dans l'annulation d'écho acoustique monophonique, nous pouvons trouver de nombreuses variantes des algorithmes LMS et NLMS avec la technique de prise sélective «mise à jour partielle» en anglais «partial update» (PU) [43-45]. Une autre technique robuste permettant de réduire la complexité de calcul a été introduite en appliquant la technique de mises à jour sélectives des données en utilisant un pas adaptatif variable. Parmi ces techniques, nous intéressons au filtrage par appartenance à un ensemble en anglais «set-membership filtering» (SMF) [46] et au filtrage par un pas variable non paramétrique en anglais «non-parametric variable step size» (NP-VSS) [47].

Dans ce chapitre, nous proposons d'utiliser une nouvelle méthode basée sur la combinaison de deux techniques, la technique de sélection des données et la technique de mise à jour partielle (PU) appliquée à un algorithme adaptatif à deux canaux. Cela permet de prendre en compte uniquement les données d'entrée qui apportent de nouvelles informations au système. Par conséquent, la complexité de calcul est réduite. En se basant sur ces techniques, nous proposons quatre nouvelles versions d'algorithmes adaptatifs pour l'AEA stéréophonique. Ces algorithmes proposés sont : l'algorithme NLMS avec la technique de l'appartenance à un ensemble étendu au cas stéréophonique (ES-SM-NLMS) [48], l'algorithme NLMS non paramétrique à pas variable étendu au cas stéréophonique SNP-VSS-N [49], l'algorithme XMNL-NLMS avec la technique de l'appartenance à un ensemble étendu au cas stéréophonique (SM-PU-SNLMS) [48] et l'algorithme XMNL-NLMS à pas variable non paramétrique étendu au cas stéréophonique (SNP-VSS-PU-N) [50].

4.2. Description de l'algorithme adaptatif de l'appartenance à un ensemble à deux canaux (stéréophonique) ES-SM-NLMS

Dans cette section, nous allons proposer une version étendue de la fonction SMF adaptée au cas stéréophonique. En résumé, le concept de base du filtrage par l'ensemble d'appartenances (SMF), ce concept avait été proposé pour la première fois dans la

référence [46] et plus profondément dans la référence [51]. L'idée principale de la technique SMF est de vérifier si l'estimation précédente $\mathbf{w}(n-1)$ se situe en dehors de l'ensemble de contraintes H_{n-1} qui contient tous les vecteurs $\mathbf{w}(n)$ tel que le module de l'erreur de sortie associée à l'instant n est limité par γ , tel que:

$$H_{n-1} = \{\mathbf{w}(n) \in \mathbf{R}^{N+1} : |e(n)| = |d(n) - \mathbf{w}^T(n-1)\mathbf{X}(n)| \leq \gamma\} \quad (4.1)$$

Où $\mathbf{X}(n) = [x(n), x(n-1), \dots, x(n-L+1)]^T$ est un vecteur contenant les L dernières observations de signal d'entrée $\mathbf{x}(n)$. Dans le cas stéréophonique, l'ensemble H_{n-1} contient tous les couples possibles $(w_1(n), w_2(n))$ au lieu des vecteurs $\mathbf{w}(n)$, cette modification conduit à récrire (4.1) comme suit :

$$H_{n-1} = \{(w_1(n), w_2(n)) \in \mathbf{R}^{2N+1} : |e(n)| = |d(n) - \mathbf{w}_1^T(n-1)\mathbf{X}_1(n) - \mathbf{w}_2^T(n-1)\mathbf{X}_2(n)| \leq \gamma\} \quad (4.2)$$

Si le module de l'erreur de filtrage *a priori* est supérieur à une limite spécifiée γ , le nouveau couple d'estimation $(w_1(n), w_2(n))$ sera mis à jour à la limite de la plus proche valeur du critère H_{n-1} avec une distance minimale, ce qui signifie que la technique SMF minimise les deux distances euclidiennes $\|w_1(n) - w_1(n-1)\|^2$ et $\|w_2(n) - w_2(n-1)\|^2$ soumises à $(w_1(n), w_2(n)) \in H_{n-1}$. La procédure de mise à jour SMF est effectuée par une projection orthogonale des valeurs estimées antérieures à la limite la plus proche de H_{n-1} . La Figure 4.1 illustre la procédure de mise à jour SMF.

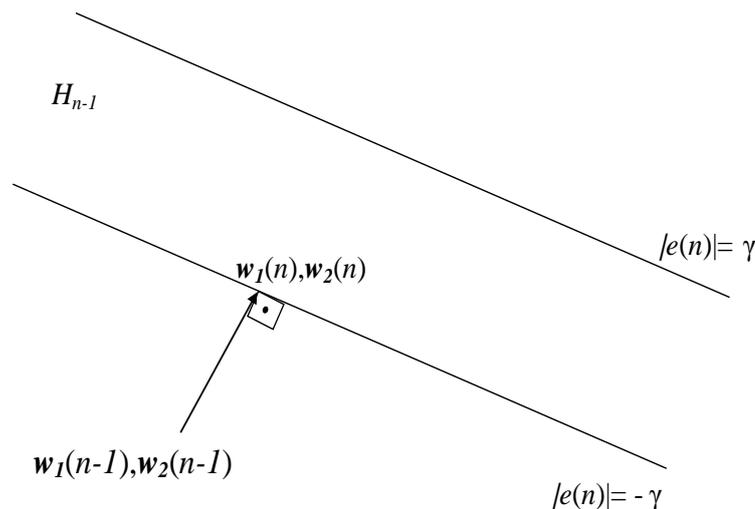


Figure 4.1: Procédure de mise à jour par la technique SMF.

donc, le pas d'adaptation peut s'écrire comme suit :

$$\alpha(n) = \begin{cases} 1 - \frac{\gamma}{|e(n)|}, & \text{si } [\mathbf{w}_1^T(n-1), \mathbf{w}_2^T(n-1)]^T \notin H_{n-1} \\ 0, & \text{autrement} \end{cases} \quad (4.3)$$

et le pas d'adaptation normalisé :

$$\mu(n) = \frac{\alpha(n)}{\mathbf{X}_1^T(n)\mathbf{X}_1(n) + \mathbf{X}_2^T(n)\mathbf{X}_2(n) + c_0} \quad (4.4)$$

Où c_0 est un petit nombre positif pour éviter la division par 0.

L'algorithme **ES-SM-NLMS** proposé est basé sur l'utilisation et la combinaison du system stéréophonique présenté dans le chapitre 2 avec la technique d'adaptation SMF proposée. L'algorithme ES-SM-NLMS résultant est résumé dans le Tableau 4.1.

Tableau 4.1: Pseudo-code de l'algorithme NLMS avec la technique de l'appartenance à un ensemble étendu au cas stéréophonique (ES-SM-NLMS) [48].

Entrées	
•	Signal d'entrée $\mathbf{X}_1(n)$ et $\mathbf{X}_2(n)$
•	Signal désiré $d(n)$.
Sorties :	
•	Les coefficients de filtre $\mathbf{w}_1(n)$ et $\mathbf{w}_2(n)$.
•	La sortie de filtre $\bar{y}(n)$.
•	L'erreur de filtrage $e(n)$.

Initialisation	
	$i = 1, 2; \mathbf{w}_i(0) = \mathbf{0}$
	$c_0 > 0$ un petit nombre pour éviter la division par 0.
Filtrage	
	$\bar{y}(n) = \mathbf{X}_1^T(n)\mathbf{w}_1(n-1) + \mathbf{X}_2^T(n)\mathbf{w}_2(n-1)$
Estimation d'erreur	
	$e(n) = d(n) - \mathbf{X}_1^T(n)\mathbf{w}_1(n-1) - \mathbf{X}_2^T(n)\mathbf{w}_2(n-1)$
Adaptation	
	$if e(n) > \gamma$
{	$\alpha(n) = 1 - \frac{\gamma}{ e(n) }$
	$\mu(n) = \frac{\alpha(n)}{\mathbf{X}_1^T(n)\mathbf{X}_1(n) + \mathbf{X}_2^T(n)\mathbf{X}_2(n) + c_0}$
	$\mathbf{k}_i(n) = \mu(n)\mathbf{X}_i(n)$
	$\mathbf{w}_i(n) = \mathbf{w}_i(n-1) + \mathbf{k}_i(n)e(n)$
}	
Else	
{	$\mathbf{w}_i(n) = \mathbf{w}_i(n-1)$
}	

L'introduction de la fonction SMF dans les algorithmes stéréophoniques proposés permet de réduire la complexité de calcul moyenne. En outre, la fonction SMF utilise un pas d'adaptation variable dépendant du module de l'erreur linéaire *a priori*, et qui fournit une convergence rapide ainsi qu'un faible désalignement en régime permanent.

4.3. Algorithme NLMS non paramétrique à pas variable stéréophonique (SNP-VSS-N)

Dans cette section, nous présentons une nouvelle version étendue de l'algorithme adaptatif NLMS avec pas variable non paramétrique adaptée au cas stéréophonique. Cette technique a été premièrement proposée dans le cas monophonique sous le nom « L'algorithme NLMS à pas variable non paramétrique » (NP-VSS-NLMS) [47]. Dans notre proposition, nous adaptons cette technique au système stéréophonique. Les avantages de l'algorithme stéréophonique NP_VSS_NLMS (SNP-VSS-N) par rapport à l'algorithme stéréophonique classique SNLMS sont la convergence rapide, une bonne poursuite et un désalignement faible.

Pour le nouvel algorithme proposé SNP-VSS-N, nous avons cherché à trouver $\mu(n)$, en présence du bruit et nous visons avoir :

$$\mathbf{X}_1^T(n)[\mathbf{h}_1 - \mathbf{w}_1(n-1)] + \mathbf{X}_2^T(n)[\mathbf{h}_2 - \mathbf{w}_2(n-1)] = 0 \quad \forall n \quad (4.5)$$

Cela implique que $\varepsilon(n) = v(n)$, où $\varepsilon(n)$ est l'erreur a posteriori donnée par (2.5) « voir Chapitre 2 » et $v(n)$ représente le bruit de fond. Par conséquent, dans le nouvel algorithme proposé, nous souhaitons trouver le pas d'adaptation en respectant :

$$E\{\varepsilon^2(n)\} = \sigma_v^2 \quad \forall n \quad (4.6)$$

Où $E\{\cdot\}$ est l'espérance mathématique et $\sigma_v^2 = E\{v^2\}$ est la puissance du système de bruit. En utilisant l'approximation du calcul de puissance, on obtient :

$$\mathbf{X}_1^T(n)\mathbf{X}_1(n) + \mathbf{X}_2^T(n)\mathbf{X}_2(n) = L(\sigma_{x_1}^2 + \sigma_{x_2}^2) = L(E\{\mathbf{x}_1^2\} + E\{\mathbf{x}_2^2\}) \quad (4.7)$$

Cette approximation n'est vraie que si les signaux d'entrées ne sont pas corrélés. En admettant que $\mu(n)$ soit de nature déterministe, en substituant (2.2) en (2.1) et (2.7) en (2.6), on obtient :

$$d(n) = \mathbf{X}_1^T(n)\mathbf{h}_1 + \mathbf{X}_2^T(n)\mathbf{h}_2 + v(n) \quad (4.8)$$

$$\mathbf{w}_i(n) = \mathbf{w}_i(n-1) + \mu(n)e(n)\mathbf{X}_i(n) \quad (4.9)$$

les nouvelles équations obtenues en substituant ces dernières en (2.5), et en utilisant (2.3) et (2.4) pour estimer $\mathbf{w}_i(n-1)$, on obtient:

$$E\{\varepsilon^2(n)\} = [1 - \mu(n)L(\sigma_{x_1}^2 + \sigma_{x_2}^2)]^2 \sigma_e^2(n) = \sigma_v^2 \quad (4.10)$$

Où σ_e^2 dénote la puissance du signal d'erreur *a priori*. En utilisant les deuxièmes et troisièmes termes de l'équation (4.10), on obtient l'équation de second ordre suivante :

$$\mu(n)^2 - \frac{2}{L(\sigma_{x_1}^2 + \sigma_{x_2}^2)} \mu(n) + \frac{1}{L^2(\sigma_{x_1}^2 + \sigma_{x_2}^2)^2} \left[1 - \frac{\sigma_v^2}{\sigma_e^2(n)} \right] = 0 \quad (4.11)$$

Pour laquelle la solution évidente est donnée par :

$$\mu(n) = \frac{\alpha(n)}{\mathbf{X}_1^T(n)\mathbf{X}_1(n) + \mathbf{X}_2^T(n)\mathbf{X}_2(n)} \quad (4.12)$$

Où $\alpha(n) = \left[1 - \frac{\sigma_v^2}{\sigma_e^2} \right]$ est le pas d'adaptation de l'algorithme SNP-VSS-N respectant la condition suivante $0 \leq \alpha(n) \leq 1$.

Nous remarquons, à partir de l'équation (4.12), qu'avant la convergence de l'algorithme, $\sigma_e(n)$ est grand en comparaison avec σ_v ; ainsi, $\mu(n)_{SNP-VSS-N} \simeq \mu(n)_{SNLMS}$. D'autre part, lorsque l'algorithme commence à converger vers la solution vraie, $\sigma_e(n) \simeq \sigma_v$ et $\mu(n)_{SNP-VSS-N} \simeq 0$. C'est exactement ce que nous souhaitons ; à la fois une bonne convergence et un faible désajustement.

La quantité $\sigma_e^2(n)$ est estimée comme suit:

$$\sigma_e^2(n) = \lambda \sigma_e^2(n-1) + (1-\lambda)e^2(n) \quad (4.13)$$

Où λ est un facteur exponentiel. Cette estimation pourrait donner une amplitude inférieure à σ_v^2 , ce que ferait $\mu(n)_{SNP-VSS-N}$ négatif. La solution simple de ce problème est d'initialiser $\sigma_e^2(n) = 1$ lorsque ce problème se produit.

En utilisant, le system stéréophonique présenté dans le Chapitre 2, et la technique à un pas variable non paramétrique étendu au cas stéréophonique proposé, l'algorithme SNP-VSS-N résultant est résumé et illustré dans le Tableau 4.2.

Tableau 4.2: Pseudo-code de l'algorithme NLMS non paramétrique à pas variable stéréophonique SNP-VSS-N [49].

Les entrées	<ul style="list-style-type: none"> • Signal d'entrée $\mathbf{X}_1(n)$ et $\mathbf{X}_2(n)$ • Signal désiré $d(n)$. • la puissance du système de bruit σ_v^2
Les sorties :	<ul style="list-style-type: none"> • Les coefficients de filtre $\mathbf{w}_1(n)$ et $\mathbf{w}_2(n)$. • La sortie de filtre $\bar{y}(n)$. • L'erreur de filtrage $e(n)$.

Initialisation	$i = 1, 2; \mathbf{w}_i(0) = \mathbf{0}, \sigma_e^2(n) = 1$
Paramètres	$c_0 > 0$ un petit nombre pour éviter la division par 0. $\lambda = 1 - \frac{1}{QL} \quad , Q \geq 2$
Filtrage	$\bar{y}(n) = \mathbf{X}_1^T(n)\mathbf{w}_1(n-1) + \mathbf{X}_2^T(n)\mathbf{w}_2(n-1)$
Estimation d'erreur	$e(n) = d(n) - \mathbf{X}_1^T(n)\mathbf{w}_1(n-1) - \mathbf{X}_2^T(n)\mathbf{w}_2(n-1)$
Adaptation	$\sigma_e^2(n) = \lambda\sigma_e^2(n-1) + (1-\lambda)e^2(n)$ $\left\{ \begin{array}{l} \text{if } \sigma_e(n) > \sigma_v \\ \alpha(n) = 1 - \frac{\sigma_v}{\sigma_e(n) + c_0} \\ \mu(n) = \frac{\alpha(n)}{\mathbf{X}_1^T(n)\mathbf{X}_1(n) + \mathbf{X}_2^T(n)\mathbf{X}_2(n) + c_0} \\ \mathbf{k}_i(n) = \mu(n)\mathbf{X}_i(n) \\ \mathbf{w}_i(n) = \mathbf{w}_i(n-1) + \mathbf{k}_i(n)e(n) \end{array} \right.$
Else	$\left\{ \begin{array}{l} \mathbf{w}_i(n) = \mathbf{w}_i(n-1) \end{array} \right.$

L'algorithme SNP-VSS-N ressemble au premier coup d'œil à l'algorithme ES-SM-NLMS, mais ils ne le sont pas. En effet, dans le calcul de $\alpha(n)$ pour l'algorithme ES-SM-NLMS il n'y a pas de moyenne sur $|e(n)|$. En outre, l'algorithme SNP-VSS-N a été dérivé presque sans hypothèse. Donc, il est clair que nous ne pouvons pas nous attendre à un faible désalignement pour l'algorithme ES-SM-NLMS comme pour SNP-VSS-N.

4.4. Description de l'algorithme adaptatif XMNL-NLMS avec la technique de l'appartenance à un ensemble étendu au cas stéréophonique (SM-PU-SNLMS)

Le nouvel algorithme proposé pour l'AEAS est basé sur la combinaison de deux techniques, la technique de l'appartenance à un ensemble étendu au cas stéréophonique (voir référence SMF présenté dans la Section 4.2) et la technique de sélection de prises avec maximum exclusif non linéaire XMNL (voir référence PUF présenté dans le Chapitre 2). L'algorithme résultant est noté par XMNL-NLMS avec la technique de l'appartenance à un ensemble étendu au cas stéréophonique (SM-PU-SNLMS).

Comme nous l'avons discuté dans le Chapitre 2, la technique XMNL peut être décrite par un traitement non linéaire (TNL2) sur les vecteurs d'entrée $s_1(n)$ et $s_2(n)$ afin de générer les signaux transmis $x_1(n)$ et $x_2(n)$, comme mentionné dans l'équation (2.16), avec la mise à jour partielle et l'introduction d'un nouveau paramètre dans le gain d'adaptation comme suite :

$$\mathbf{k}_i(n) = \mu(n)\mathbf{Q}_i(n)\mathbf{X}_i(n) \quad (4.14)$$

On rappelle que le paramètre $\mu(n)$ est contrôlé, dans ce cas, par le SMF donné par l'équation (4.4) suivi par l'équation (4.3).

En utilisant le system stéréophonique présenté dans le Chapitre 2 et la technique d'adaptation SMF combinée avec la technique XMNL, et en utilisant les équations de l'algorithme XMNL-NLMS discuté dans le Chapitre 2, l'algorithme SM-PU-SNLMS proposé est résumé et illustré dans le tableau 4.3.

On rappelle que l'utilisation de la technique XMNL n'est pas seulement dans la réduction de la cohérence entre les signaux d'entrés mais c'est aussi pour la réduction de complexité de calcul.

Tableau 4.3: Pseudo-code de l'algorithme XMNL-NLMS avec la technique de l'appartenance à un ensemble étendu au cas stéréophonique (SM-PU-SNLMS) [48].

Les entrées	<ul style="list-style-type: none"> • Signal d'entrée $\mathbf{X}_1(n)$ et $\mathbf{X}_2(n)$ • Signal désiré $d(n)$.
Les sorties :	<ul style="list-style-type: none"> • Les coefficients de filtre $\mathbf{w}_1(n)$ et $\mathbf{w}_2(n)$. • La sortie de filtre $\bar{y}(n)$. • L'erreur de filtrage $e(n)$.

Initialisation	$i = 1, 2; w_i(0) = 0$ $c_0 > 0 \text{ un petit nombre pour éviter la division par } 0.$
Filtrage	$\bar{y}(n) = \mathbf{X}_1^T(n)\mathbf{w}_1(n-1) + \mathbf{X}_2^T(n)\mathbf{w}_2(n-1)$
Estimation d'erreur	$e(n) = d(n) - \mathbf{X}_1^T(n)\mathbf{w}_1(n-1) - \mathbf{X}_2^T(n)\mathbf{w}_2(n-1)$
Adaptation	$\left\{ \begin{array}{l} \text{if } e(n) > \gamma \\ \alpha(n) = 1 - \frac{\gamma}{ e(n) } \\ p(n) = \mathbf{X}_1(n) - \mathbf{X}_2(n) \\ q_{1,u}(n) = \begin{cases} 1, & p_u \in \{Z \text{ maxima dep}(n)\} \\ 0, & \text{autrement} \end{cases} \\ q_{2,v}(n) = \begin{cases} 1, & p_v \in \{Z \text{ minima dep}(n)\} \\ 0, & \text{autrement} \end{cases} \\ \mathbf{Q}_i(n) = \text{diag}\{q_i(n)\} \\ \mu(n) = \frac{\alpha(n)}{\mathbf{X}_1^T(n)\mathbf{X}_1(n) + \mathbf{X}_2^T(n)\mathbf{X}_2(n) + c_0} \\ \mathbf{k}_i(n) = \mu(n)\mathbf{Q}_i(n)\mathbf{X}_i(n) \\ \mathbf{w}_i(n) = \mathbf{w}_i(n-1) + \mathbf{k}_i(n)e(n) \end{array} \right. \\ \text{else} \\ \left\{ \begin{array}{l} \mathbf{w}_i(n) = \mathbf{w}_i(n-1) \end{array} \right. \\ \end{array}$

4.5. Description de l'algorithme adaptatif XMNL-NLMS avec pas variable non paramétrique étendu au cas stéréophonique (SNP-VSS-PU-N)

Dans ce qui cette section, nous proposons un nouvel algorithme adaptatif pour l'AEAS en se basant sur la combinaison de deux techniques, la technique d'adaptation par un pas variable non paramétrique adapté au cas stéréophonique SNP-VSS présenté dans la Section 4.3, et la technique de sélection de prises avec maximum exclusif non linéaire XMNL (voire référence PUF présenté dans le Chapitre 2). L'algorithme résultant est noté par XMNL-NLMS avec pas variable non paramétrique étendu au cas stéréophonique (SNP-VSS-PU-N).

On procède de la même manière de développement qu'on a fait pour la combinaison des techniques SMF et XMN pour combiner la technique SNP-VSS et la technique XMNL. Le gain d'adaptation de cette version est le même que celui de l'équation (4.14) avec un paramètre $\mu(n)$ contrôlé par la technique SNP-VSS, donné par l'équation (4.12), avec un pas d'adaptation égale à l'expression suivante $\alpha(n) = \left[1 - \frac{\sigma_b^2}{\sigma_e^2}\right]$.

En utilisant, le system stéréophonique présenté dans le Chapitre 2 et la technique d'adaptation SNP-VSS combiné avec la technique XMNL, et en utilisant les équations de l'algorithme XMNL-NLMS discuté dans le Chapitre 2, l'algorithme SNP-VSS-PU-N proposé, est résumé et illustré dans le Tableau 4.4.

Tableau 4.4: Pseudo-code de l'algorithme XMNL-NLMS avec pas variable non paramétrique étendu au cas stéréophonique (SNP-VSS-PU-N) [50].

Les entrées	
Signal d'entrée $\mathbf{X}_1(n)$ et $\mathbf{X}_2(n)$	
Signal désiré $d(n)$.	
Les sorties :	
Les coefficients de filtre $\mathbf{w}_1(n)$ et $\mathbf{w}_2(n)$.	
La sortie de filtre $\bar{y}(n)$.	
L'erreur de filtrage $e(n)$.	

Initialisation	$i = 1, 2; \mathbf{w}_i(0) = \mathbf{0}, \sigma_e^2(n) = 1$
Paramètres	$c_0 > 0$: petit nombre pour éviter la division par 0.
	$\lambda = 1 - \frac{1}{QL}, Q \geq 2$
Filtrage	$\bar{y}(n) = \mathbf{X}_1^T(n)\mathbf{w}_1(n-1) + \mathbf{X}_2^T(n)\mathbf{w}_2(n-1)$
Estimation d'erreur	$e(n) = d(n) - \mathbf{X}_1^T(n)\mathbf{w}_1(n-1) - \mathbf{X}_2^T(n)\mathbf{w}_2(n-1)$
Adaptation	$\sigma_e^2(n) = \lambda\sigma_e^2(n-1) + (1-\lambda)e^2(n)$ <p style="text-align: center;">if $\sigma_e(n) > \sigma_v$</p> <p style="text-align: center;">{</p> $\alpha(n) = 1 - \frac{\sigma_v}{\sigma_e(n) + c_0}, \mathbf{p}(n) = \mathbf{X}_1(n) - \mathbf{X}_2(n) $ $q_{1,u}(n) = \begin{cases} 1, & p_u \in \{Z \text{ maxima de } \mathbf{p}(n)\} \\ 0, & \text{autrement} \end{cases},$ $q_{2,v}(n) = \begin{cases} 1, & p_v \in \{Z \text{ minima de } \mathbf{p}(n)\} \\ 0, & \text{autrement} \end{cases}$ $\mathbf{Q}_i(n) = \text{diag}\{\mathbf{q}_i(n)\}$ $\mu(n) = \frac{\alpha(n)}{\mathbf{X}_1^T(n)\mathbf{X}_1(n) + \mathbf{X}_2^T(n)\mathbf{X}_2(n) + c_0}, \quad \mathbf{k}_i(n) = \mu(n)\mathbf{Q}_i(n)\mathbf{X}_i(n)$ $\mathbf{w}_i(n) = \mathbf{w}_i(n-1) + \mathbf{k}_i(n)e(n)$ <p style="text-align: center;">}</p>
Else	
{	
	$\mathbf{w}_i(n) = \mathbf{w}_i(n-1)$
}	

4.6. Conclusion

Dans ce chapitre, nous avons présenté quatre nouvelles versions d'algorithmes adaptatifs pour l'annulation d'écho acoustique stéréophonique. Ces algorithmes sont basés essentiellement sur la technique de sélection des données et la technique de mise à jour partielle des coefficients du filtre adaptatif. Ses nouveaux permettent de traiter uniquement des données qui apportent des nouvelles informations au systèmes, donc la complexité de calcul est bien réduite et les performances d'annulation d'écho acoustique sont améliorées. Les résultats des simulations obtenues sont présentés dans le Chapitre 5 suivant.

CHAPITRE 5: Étude Comparative Des Performances et Résultats de Simulation des Algorithmes Adaptatifs Monophoniques et Stéréophoniques

5.1. Introduction

Nous présentons, dans ce chapitre, les résultats de simulation des algorithmes adaptatifs présentés dans les chapitres précédents et ceci dans le cas monophonique et stéréophonique de l'annulation d'écho acoustique. Les différents signaux sources (parole et bruit) utilisés seront précisés. Une évaluation des effets de certains paramètres des algorithmes testés seront également détaillées. Les performances auxquelles nous nous intéressons dans cette étude sont exprimées en matière de vitesse de convergence, stabilité et du niveau de l'erreur de filtrage.

5.2. Signaux utilisés en simulation

Dans l'ensemble des simulations présentées dans ce chapitre, trois types des signaux seront utilisés pour obtenir les résultats dans ce chapitre.

Un bruit stationnaire (bruit USASI « United State of America Standards Institut maintenant ANSI ») ce signal a un spectre similaire aux signaux de parole et il est échantillonné à 16 kHz. Grâce à ses propriétés, ce signal est souvent utilisé dans les applications d'annulation d'écho acoustique pour évaluer la vitesse de convergence des algorithmes adaptatifs utilisés ainsi que leur capacité de poursuite des non-stationnarités intervenant dans le chemin d'écho à identifier. Ce signal est présenté dans la figure 5.1.

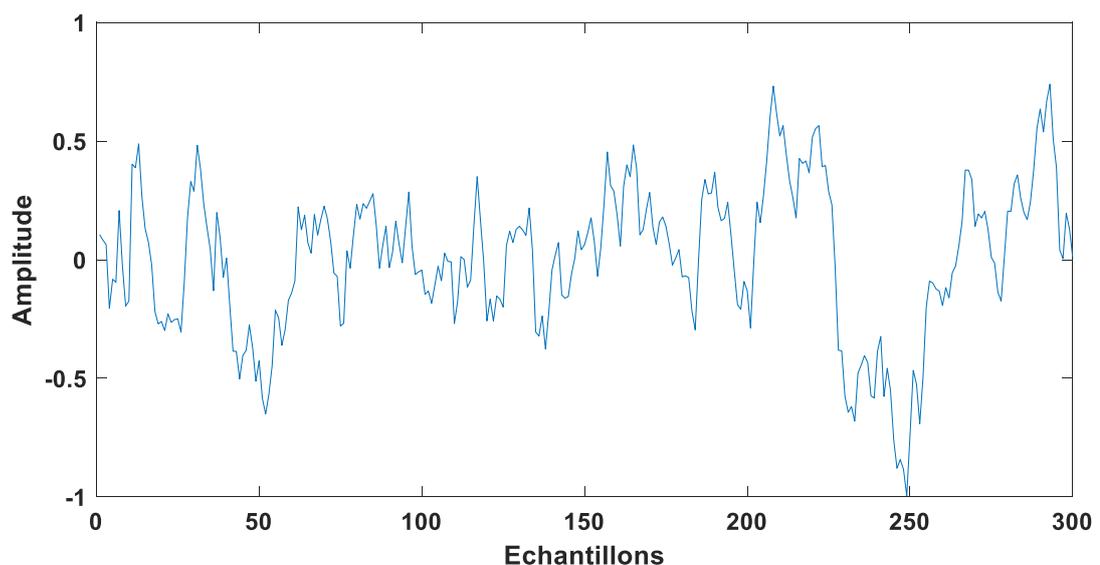


Figure 5.1. Bruit USASI.

En ce qui concerne la seconde source, celle-ci correspond à un bruit blanc Gaussien (WGN) qui est une réalisation d'un processus aléatoire dont la densité spectrale de puissance est la même pour toutes les fréquences (bruit stationnaire). Grâce à ses propriétés, ce signal est souvent utilisé pour vérifier la stabilité numérique des algorithmes adaptatifs utilisés. Ce signal est présenté dans la figure 5.2.

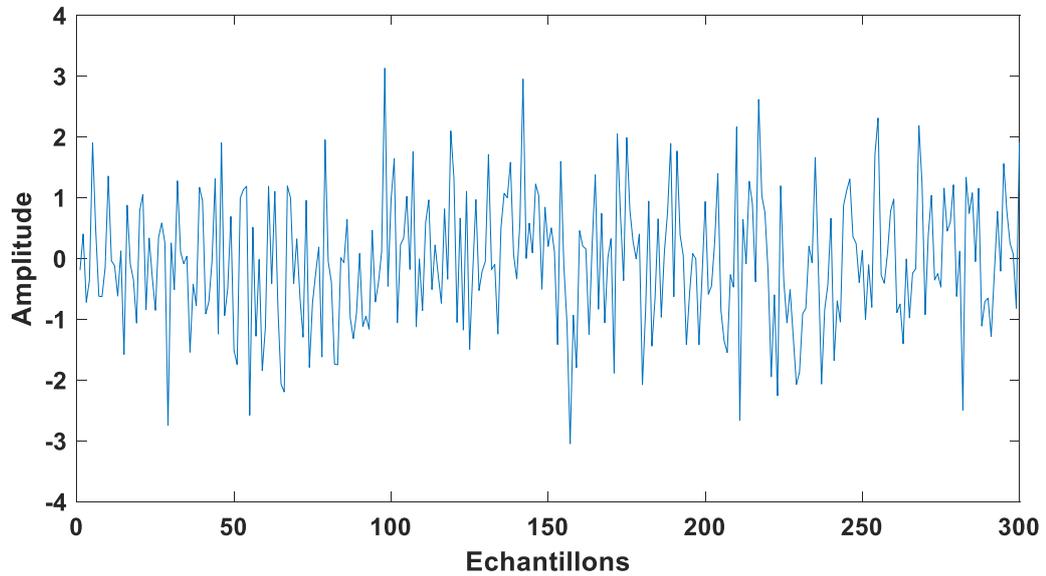


Figure 5.2. Bruit Blanc Gaussien.

La troisième source est de la parole constituée d'une phrase phonétiquement équilibrée prononcée par un locuteur masculin, il a de fréquence d'échantillonnage égale à 8 kHz. Ce signal de parole est présenté dans la figure 5.3.

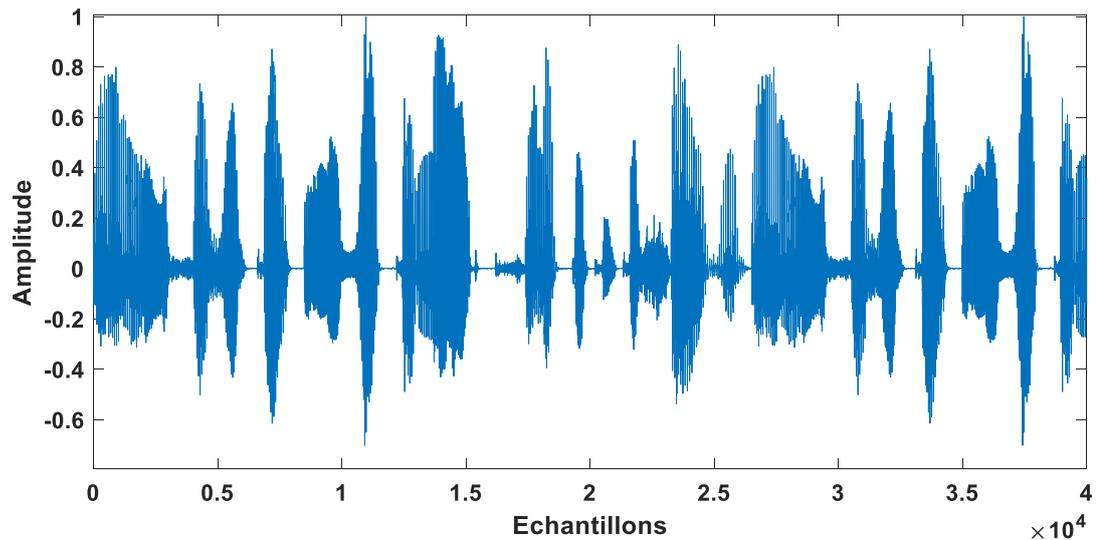


Figure 5.3. Signal parole

Le signal de parole est un phénomène essentiellement continu, il n'est pas Gaussien, ni ergodique, ni stationnaire. Il peut être classé de manière un peu sommaire, en trois catégories distincts, voisés, non voisés et silence.

5.3. Simulation de réponses impulsionnelles

Dans un premier temps, les réponses impulsionnelles sont générées par une séquence aléatoire (bruit blanc) dont l'enveloppe temporelle est contrôlée par la fonction de pondération suivante :

$$f(n) = Ae^{-Bn} \quad (5.1)$$

Où A est un facteur d'échelle (choisissons égal à 1), B est un facteur d'atténuation qui modélise l'absorption des ondes sonores sur les parois de la salle et qui est donc lié au temps de réverbération. Dans la figure 5.4, nous présentons un exemple de réponse impulsionnelle de taille $L = 256$ et de facteur d'atténuation égal à $B = 0.02$. Dans la figure 5.5, nous présentons un deuxième exemple de réponse impulsionnelle de taille $L = 1024$ et de facteur d'atténuation égal à $B = 0.015$.

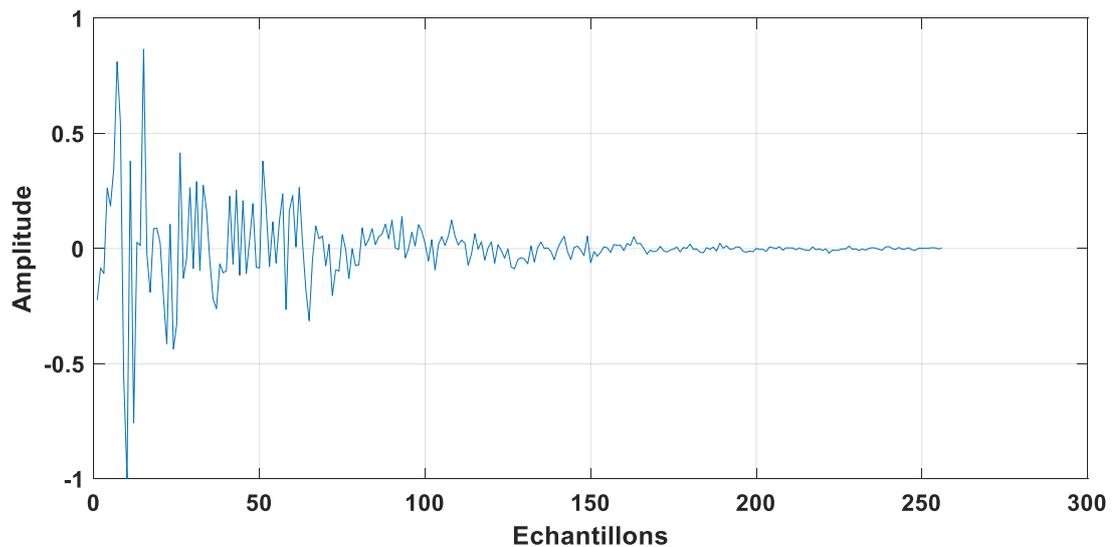


Figure 5.4. Exemple d'une réponse impulsionnelle simulé de taille $L=256$

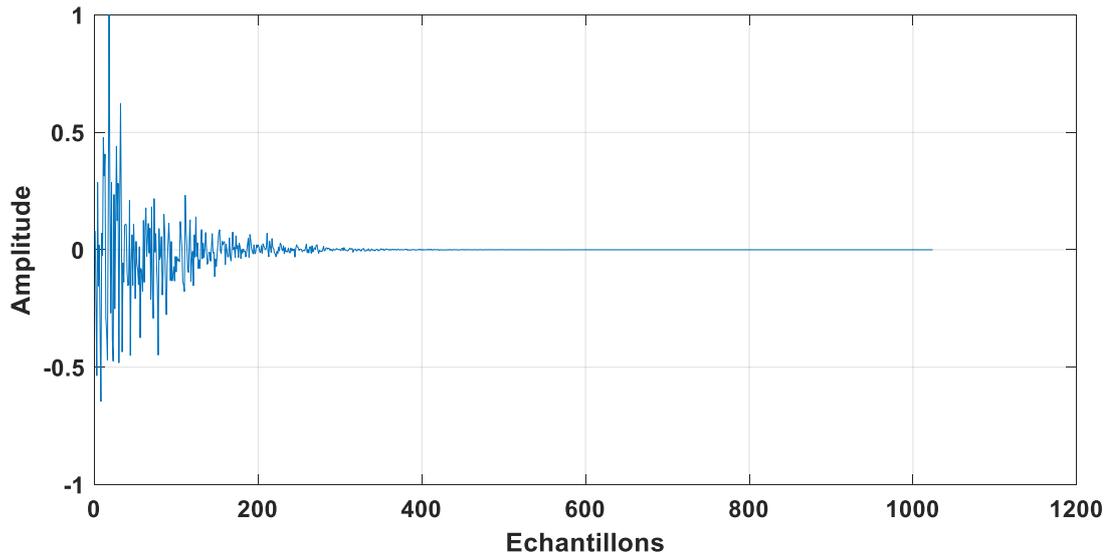


Figure 5.5. Exemple d'une réponse impulsionnelle simulé de taille $L=1024$.

5.4. Critères objectifs de qualité

Plusieurs critères ont été proposés pour évaluer la performance des algorithmes adaptatifs appliqués dans les systèmes d'annulations d'échos acoustiques monophoniques et stéréophoniques. Généralement il y a deux catégories de mesures, dites subjective et objective. Dans l'évaluation de la qualité subjective, ce sont des êtres humains qui sont amenés à juger la qualité de traitements. Elle est basée sur des tests d'écoutes et de comparaison avant et après l'annulation d'écho acoustique. Malheureusement, elle est coûteuse en matière de temps et de ressources. Dans l'autre côté, l'évaluation de la qualité objective est basée sur des mesures physiques et des calculs mathématiques entre les signaux, original et estimé. Cette méthode d'évaluation est beaucoup plus pratique et moins coûteuse.

Comme il existe une grande corrélation entre les deux types de mesures, nous allons utiliser l'évaluation la moins coûteuse « l'évaluation objective » pour valider la performance de tous les algorithmes classiques et proposés dans cette thèse. Et on s'intéresse aux critères de mesure suivants :

- Critère de l'erreur quadratique moyenne « EQM ».
- Critère de désajustement « Misalignement ».
- Critère d'amélioration de perte de retour d'écho « ERLE ».

5.4.1. Critère de l'erreur quadratique moyenne « EQM »

L'erreur quadratique moyenne (Mean Square Error : MSE) mesure la moyenne des carrés de l'évolution temporelle de l'erreur de filtrage *a priori* exprimée en décibel (dB). Ce critère est généralement utilisé pour évaluer le niveau de l'erreur, la stabilité du filtre et la vitesse de convergence. Le critère de L'EQM est donné par :

$$EQM_{dB} = 10 \log(\sigma_{e,N,n}^2) \quad (5.2)$$

Où $\sigma_{e,N,n}^2$ désigne une moyenne temporelle effectuée sur un certain nombre d'échantillons consécutifs N , et $\log(\cdot)$ représente le logarithme à base 10, avec :

$$\sigma_{e,N,n}^2 = \frac{1}{N} \sum_1^N e^2(n) \quad (5.3)$$

Où $e^2(n)$ représente les N éléments du vecteur d'erreur de filtrage *a priori*.

5.4.2. Critère de désajustement « Misalignement »

Le critère de désajustement (Misalignement) est calculé à partir de la distance euclidienne entre les coefficients des réponses impulsionnelles réelles et les coefficients des filtres estimés. Ce critère est également utilisé pour comparer la vitesse de convergence et évaluer la précision convergence des coefficients des filtres estimés vers les coefficients des réponses impulsionnelles réelles. Dans cette thèse, nous utilisons l'abréviation Mis (Misalignement) pour le critère de désajustement. Et il est donné par l'expression suivante :

$$Mis(n)_{dB} = 10 \log\left(\frac{\|\mathbf{h} - \mathbf{w}(n)\|^2}{\|\mathbf{h}\|^2}\right) \quad (5.4)$$

Où \mathbf{h} représente le vecteur des coefficients du filtre réels (coefficients d'une réponse impulsionnelle) dans le cas stéréophonique, $\mathbf{h} = [\mathbf{h}_1^T, \mathbf{h}_2^T]^T$. Et $\mathbf{w}(n)$ représente le vecteur des coefficients du filtre adaptatif estimé, dans le cas stéréophonique $\mathbf{w}(n) = [\mathbf{w}_1^T(n), \mathbf{w}_2^T(n)]^T$.

5.4.3. Critère d'amélioration de perte de retour d'écho « ERLE »

Le critère de l'amélioration de perte de retour d'écho (Echo Return Loss Enhancement « ERLE ») est un critère robuste pour l'évaluation des performances du système d'annulation d'écho acoustique. L'ERLE mesure la quantité de la perte introduite

par les filtres adaptatifs. Il s'agit du rapport entre la moyenne des carrés de l'écho et la moyenne des carrés de l'erreur résiduelle. L'ERLE exprimé en dB est donné par :

$$ERLE_{dB} = 10 \log \left(\frac{\sigma_{d,N,n}^2}{\sigma_{e,N,n}^2} \right) \quad (5.5)$$

Où $\sigma_{e,N,n}^2$ est une moyenne temporelle de l'erreur *a priori* effectuée sur un certain nombre d'échantillons consécutifs N donné par (5.3), et $\sigma_{d,N,n}^2$ une moyenne temporelle de l'écho réelle effectuée sur un le même nombre N donné par :

$$\sigma_{d,N,n}^2 = \frac{1}{N} \sum_1^N d^2(n) \quad (5.6)$$

5.5. Influence des paramètres dès les algorithmes adaptatifs monophoniques

Pour améliorer le comportement des algorithmes adaptatif NLMS et FNLMS dans le domaine d'annulation d'écho acoustique, nous avons réalisé plusieurs expériences pour voir l'influence de chacun des paramètres, afin d'en choisir les meilleurs pour atteindre les meilleures performances voulues.

5.5.1. Effet de la taille du filtre des algorithmes NLMS et FNLMS

Dans cette expérience nous évaluons le comportement des algorithmes NLMS et FNLMS pour des réponses impulsionnelles courtes et longues. Le signal d'excitation est un bruit USASI. Le pas d'adaptation est fixé à $\mu=1$, les facteurs d'oubli exponentiel sont choisis $\lambda_a = 0.9987$ et $\lambda = 0.98$. Les constants $c_a = 08$ et $c_0 = 01$. Le rapport signal à bruit est SNR=40dB. Les deux critères de mesure à savoir l'EQM, et le Mis sont utilisés. Les résultats de cette expérience sont présentés aux figures 5.6 et 5.7 :

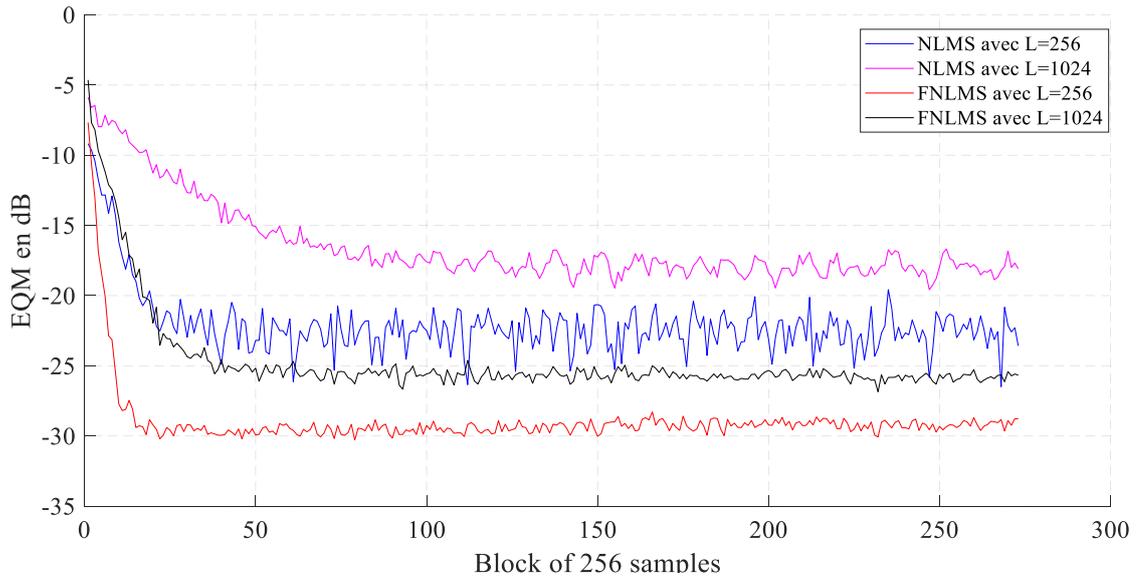


Figure 5.6. Evolution de l'EQM obtenues avec NLMS et FNLMS pour $L=256$ et $L=1024$ avec signal d'entrée bruit USASI.

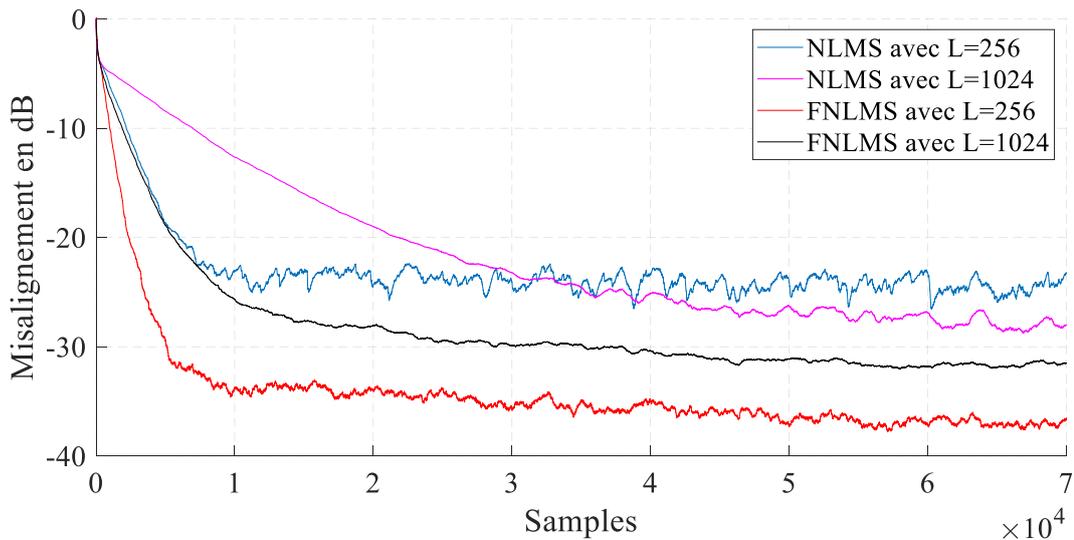


Figure 5.7. Evolution du Misalignement obtenus avec NLMS et FNLMS pour $L=256$ et $L=1024$ avec signal d'entrée bruit USASI.

Les deux figures (5.6 et 5.7) montrent qu'il y a une dégradation proportionnelle à la taille du filtre transverse pour les deux algorithmes NLMS ou FNLMS. On remarque aussi que l'algorithme FNLMS se comporte bien avec des filtres de tailles importantes par apport à l'algorithme NLMS en matière de vitesse de convergence et la valeur finale de l'EQM et du Misalignement. Pour avoir une bonne annulation d'écho, on doit choisir une taille du filtre supérieure ou égale à la taille de la réponse impulsionnelle de la salle.

5.5.2. Effet du pas d'adaptation μ sur les algorithmes NLMS et FNLMS

Dans cette expérience nous évaluons le comportement des algorithmes NLMS et FNLMS avec la variation du pas d'adaptation μ en prenant trois valeurs $\mu=1.4$, $\mu=1$ et $\mu=0.6$, la taille du filtre est $L=256$. Le signal d'excitation un bruit USASI. Le rapport signal à bruit est $SNR=40$ dB. Les deux critères de mesure « EQM, et le Misalignement » sont utilisés. Les résultats de cette expérience sont présentés dans les figures 5.8 et 5.9 pour l'algorithme NLMS et 5.10 et 5.11 pour l'algorithme FNLMS.

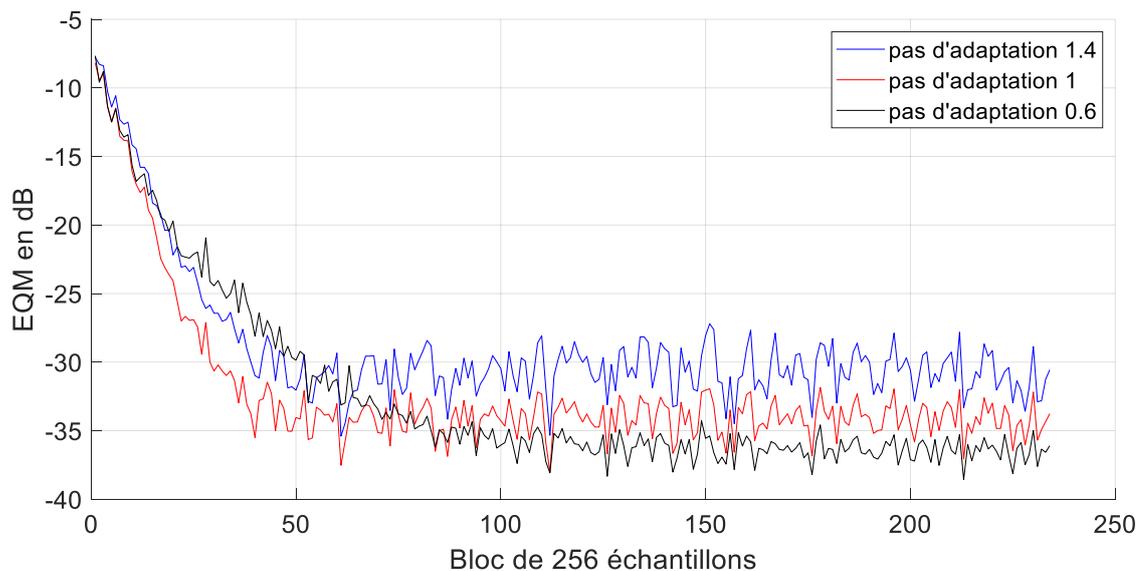


Figure 5.8. Evolution de l'EQM obtenues avec l'algorithme NLMS pour 3 valeurs de μ .

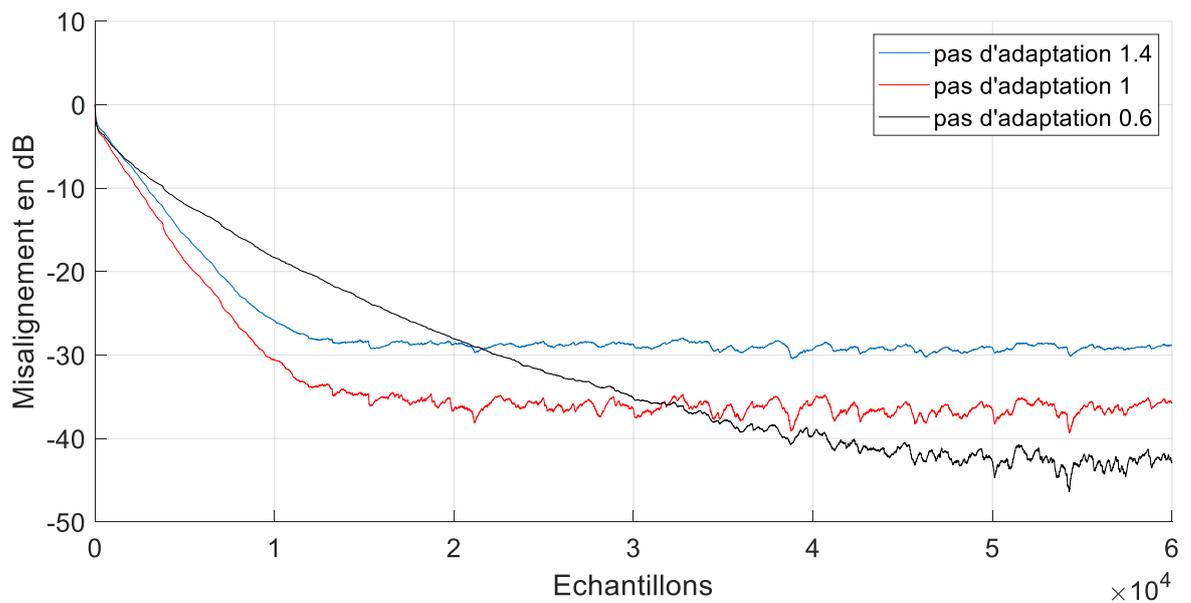


Figure 5.9. Evolution du Misalignement obtenues avec l'algorithme NLMS.

D'après ces deux figures, nous remarquons que la convergence de l'algorithme NLMS a un lien direct avec le pas d'adaptation. Le pas d'adaptation correspondant à 1 est la valeur critique avec laquelle il a donné le meilleur compromis entre la vitesse et le niveau de convergence. Pour l'algorithme FNLMS on a utilisé les mêmes paramètres utilisés pour l'algorithme NLMS avec les facteurs d'oubli exponentiel sont choisis $\lambda_a = 0.9987$ et $\lambda = 0.98$. les constants $c_a = 08$ et $c_0 = 01$.

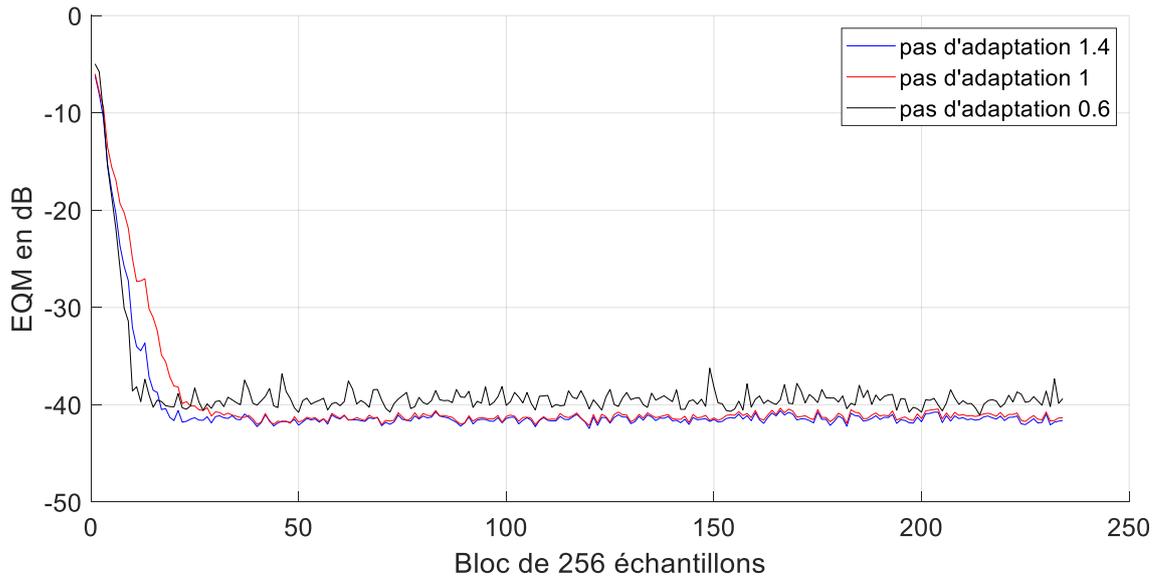


Figure 5.10. Evolution de EQM obtenues avec l'algorithme FNLMS pour 3 valeurs de μ

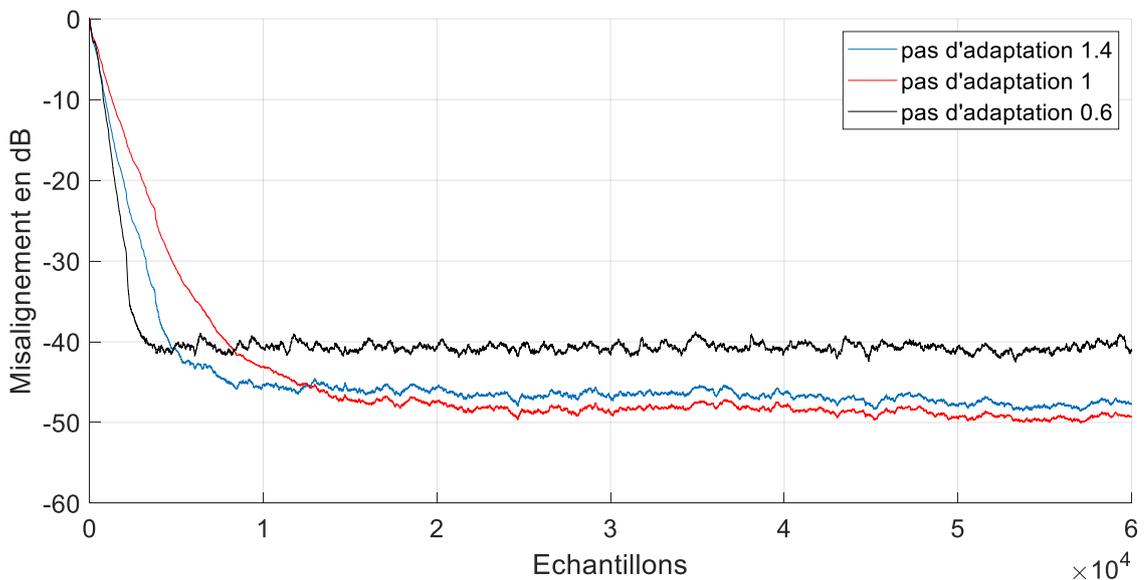


Figure 5.11. Evolution du Misalignement obtenues avec l'algorithme FNLMS.

D'après ces deux figures, nous remarquons que la valeur correspondante à 1 n'est pas une valeur critique tel que la valeur 1.4 qui a donné une vitesse mieux qu'avec le pas égale à 1. Mais il demeure proche de celle-ci. Donc pour éviter la dégradation due aux autres paramètres nous conseillons d'utiliser toujours le pas égal à 1.

5.6. Comparaison des performances des algorithmes stéréophoniques

5.6.1. Algorithme FNLMS stéréophonique (SFNLMS)

Dans les simulations suivantes, nous allons présenter les performances de l'algorithme SFNLMS développé et étudié précédemment (voir Chapitre 3) dans le cas stéréophonique. L'algorithme SNLMS a été utilisé comme référence et les trois signaux de sources (bruit blanc, bruit USASI et signal parole) sont utilisés dans cette simulation.

On a utilisé les deux critères objectifs de l'EQM et l'ERLE pour évaluer les performances de l'algorithme SFNLMS.

- *Test avec un bruit blanc à l'entrée*

Dans cette expérience, on utilise le bruit blanc (WGN) comme signal d'entrée. Ce signal est généralement utilisé pour tester la stabilité numérique de l'algorithme simulé. Pendant la simulation nous allons simuler un saut dans le milieu du signal. Nous allons choisir le $RSB=30$ dB, et un pas d'adaptation de 0.9, les facteurs d'oubli exponentiel sont choisis $\lambda_a = 0.9985$ et $\lambda = 0.98$, et les constants $c_a = 01$ et $c_0 = 01$. Les résultats obtenus pour les deux algorithmes SFNLMS et SNLMS sont présentés à la figure 5.12 pour l'EQM et la figure 5.13 pour l'ERLE :

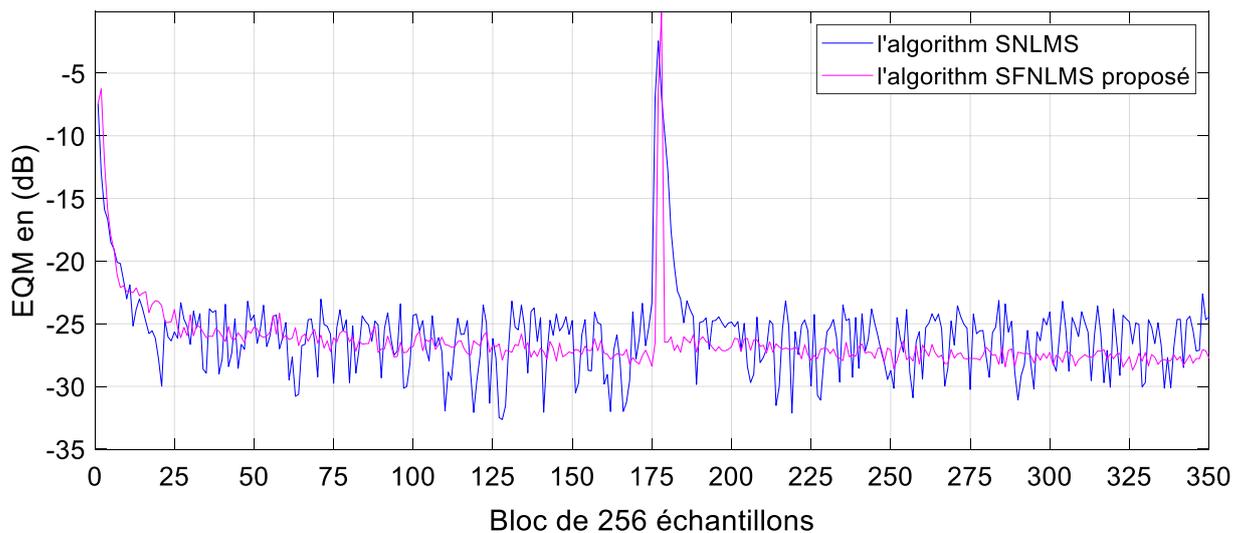


Figure 5.12. Evaluation de l'EQM obtenues avec les algorithmes SFNLMS et SNLMS.

avec un bruit blanc comme signal d'entrée.

Dans la figure 5.12, nous avons utilisé l'EQM comme critère d'évaluation et le bruit blanc comme signal d'excitation dans la configuration stéréophonique présentée précédemment dans le chapitre 2 (figure 2.1). Nous avons remarqué que l'algorithme proposé SFNLMS est stable et rapide en particulier dans le régime permanent. Nous avons aussi une valeur inférieure de l'EQM pour l'algorithme proposé en comparaison avec l'algorithme SNLMS. Avec les mêmes configurations nous avons évalué l'ERLE qui est présenté à la figure 5.13.

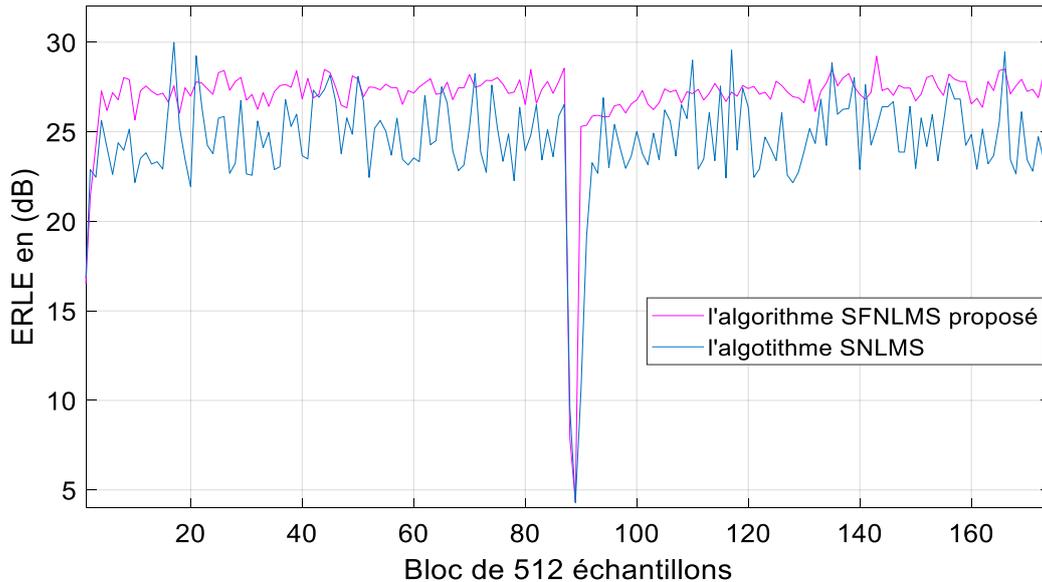


Figure 5.13. Evaluation de l'ERLE obtenues avec les algorithmes SFNLMS et SNLMS. avec un bruit blanc comme signal d'entrée.

Dans la figure 5.13, nous avons utilisé l'ERLE comme critère d'évaluation et le bruit blanc comme signal d'excitation. Nous avons confirmé la stabilité et la rapidité en régime permanent de l'algorithme proposé SFNLMS. Nous avons également montré une supériorité en valeur max de l'ERLE de l'algorithme SFNLMS proposé en comparaison avec l'algorithme SNLMS. D'après ces deux expériences, on peut dire que ces résultats confirment la théorie selon laquelle les algorithmes SNLMS et proposé SFNLMS ont les mêmes propriétés lorsque le signal d'excitation est un bruit blanc avec une légère supériorité du SFNLMS en régime permanent et avec les deux critères. Nous avons remarqué que la vitesse de convergence de l'algorithme proposé SFNLMS est proche de celle de SNLMS, et ceci dû au fait que le bruit blanc n'est pas un bon signal pour tester la vitesse de convergence de l'algorithme proposé SFNLMS. Pour cette raison, nous montrons dans la section suivante une expérience avec un bruit USASI qui est souvent utilisé pour tester la vitesse de convergence des filtres adaptatifs.

- *Test avec le bruit USASI*

Dans cette expérience, pour tester la vitesse de convergence de l’algorithme proposé, nous utilisons le bruit USASI comme signal d’excitation. Ce signal est généralement utilisé pour tester la vitesse de convergence des algorithmes simulés. Pendant la simulation nous allons générer un saut dans le milieu du signal. Nous allons choisir le $RSB=30$ dB, et un pas d’adaptation de 0.9, les facteurs d’oubli exponentiel sont choisis $\lambda_a = 0.9982$ et $\lambda = 0.99$, et les constants $c_a = 01$ et $c_0 = 01$. Les résultats obtenus avec les deux algorithmes SFNLMS et SNLMS sont présentés à la figure 5.14 pour l’EQM et la figure 5.15 pour l’ERLE.

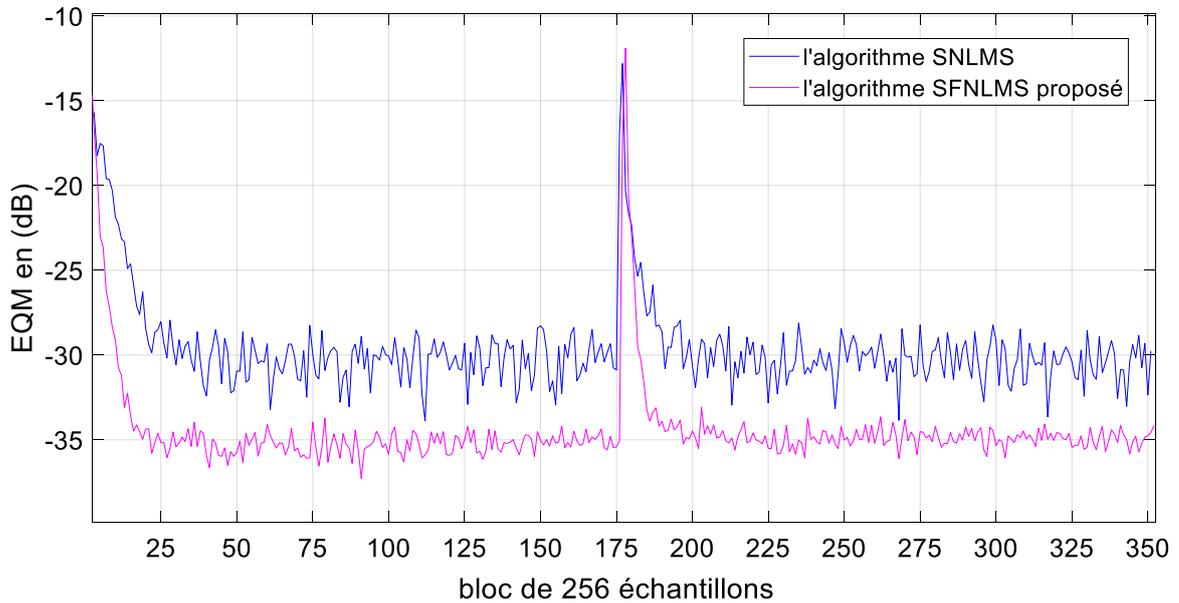


Figure 5.14. Evaluation de l’EQM obtenues avec les algorithmes SFNLMS et SNLMS avec le bruit USASI comme signal d’entrée.

Dans la figure 5.14, nous avons utilisé l’EQM comme critère d’évaluation et le bruit USASI comme signal d’excitation dans la configuration stéréophonique. Nous avons remarqué que l’algorithme proposé SFNLMS converge rapidement vers la solution optimale par rapport l’algorithme SNLMS avec les paramètres choisis. Le saut que nous avons généré au milieu du signal donne des informations de la vitesse de convergence pendant le régime permanent après la convergence de l’algorithme vers la solution optimale.

Avec les mêmes configurations d’avant, nous avons évalué l’ERLE et présenté à la figure 5.15. A partir de cette figure, nous avons confirmé la rapidité de convergence de l’algorithme proposé SFNLMS par rapport l’algorithme SNLMS. Nous avons remarqué

aussi que la perte introduit par le filtre adaptatif avec l'algorithme proposé est inférieur à celle introduite par l'algorithme SNLMS. Nous pouvons voir que l'algorithme proposé SFNLMS a toujours la meilleure performance en vitesse de convergence, des faibles valeurs de l'EQM et des plus hautes valeurs de l'ERLE.

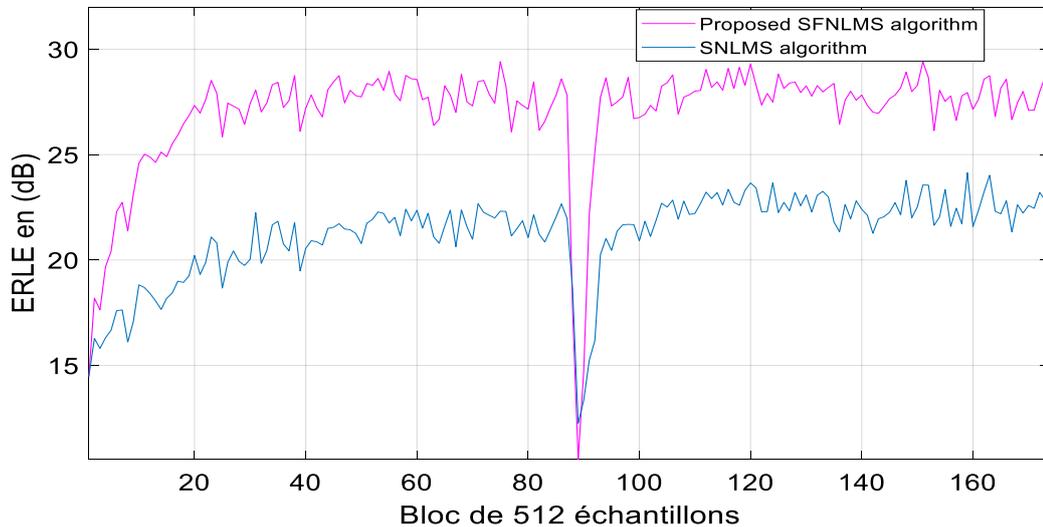


Figure 5.15. Evaluation de l'ERLE obtenus avec les algorithmes SFNLMS et SNLMS.

- **Test de des algorithmes SFNLMS et SNLMS dans l'AEAS**

Dans cette sous-section, nous réalisons deux expériences de comparaison. Dans la première expérience le signal d'entrée est un signal parole présenté à la figure 5.3. Dans cette expérience, les réponses impulsionnelles sont courtes de taille $L=256$. Le rapport signal à bruit est $RSB=30$ dB, le pas d'adaptation est choisi 0.9, les facteurs d'oubli exponentiel sont choisis $\lambda_a = 0.9985$ et $\lambda = 0.98$, et les constants $c_a = 01$ et $c_0 = 01$. Les résultats obtenus de cette expérience sont donnés à la figure 5.16 pour EQM et figure 5.17 pour l'ERLE.

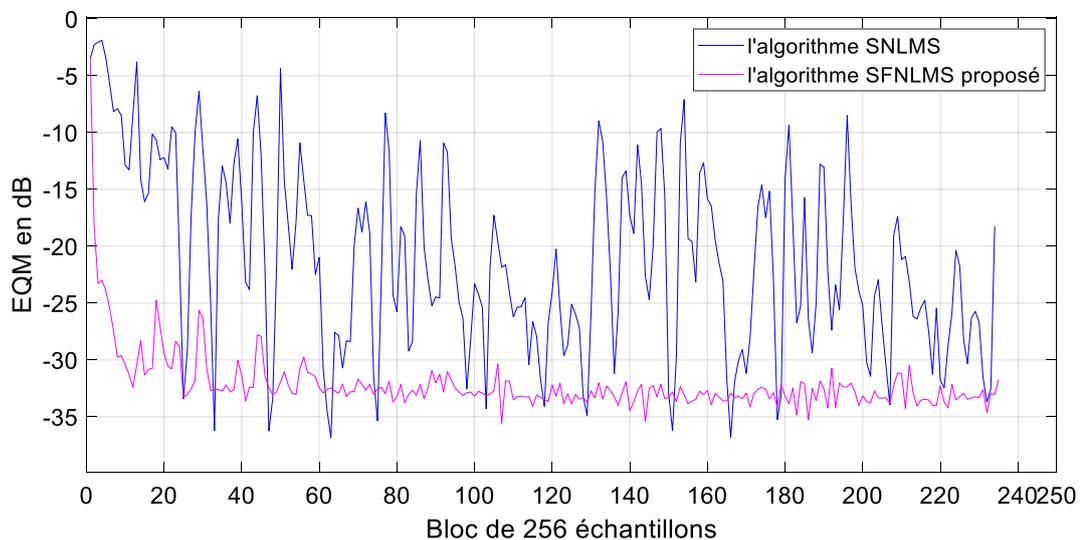


Figure 5.16. Evaluation de l'EQM obtenues avec les algorithmes SFNLMS et SNLMS dans un contexte d'AEAS.

D'après la figure 5.16, nous remarquons que l'algorithme proposé SFNLMS a des bonnes performances en AEAS par rapport l'algorithme SNLMS en deux phases transitoire et permanente. Ces meilleures propriétés sont dues à la modification proposée de la partie de prédiction, qui décorrèle les entrées et accélère la vitesse de convergence de l'algorithme SFNLMS même avec des signaux d'entrés hautement corrélé en cas d'échec de l'algorithme SNLMS. Avec les mêmes configurations, nous évaluons l'ERLE qui est présenté à la figure 5.17.

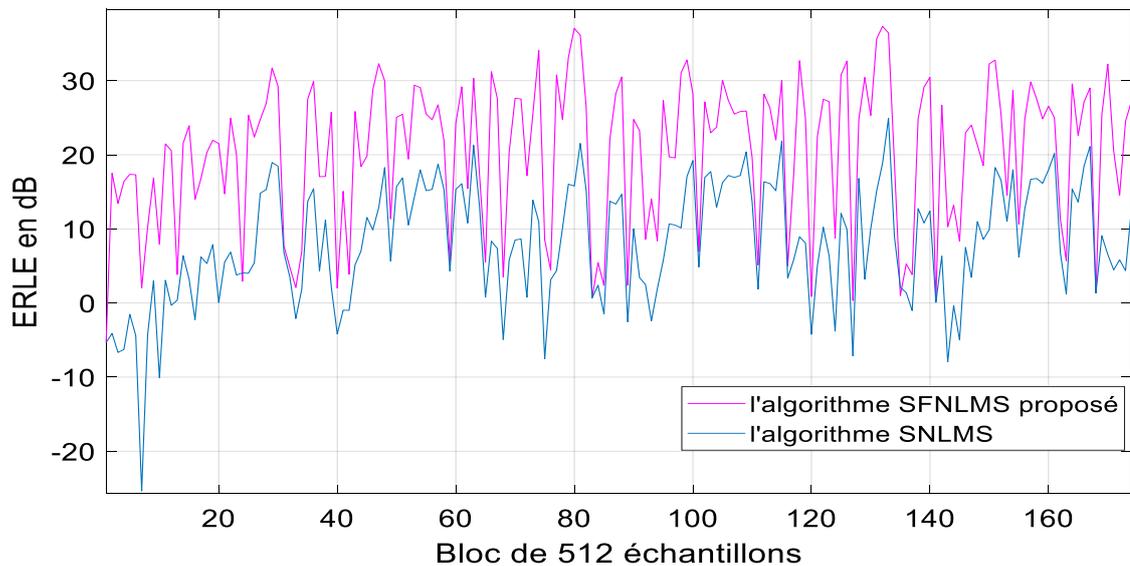


Figure 5.17. Evaluation de l'ERLE obtenus avec les algorithmes SFNLMS et SQLMS dans un contexte d'AEAS.

D'après la figure 5.17, nous remarquons que l'algorithme proposé SFNLMS garde les mêmes bonnes propriétés d'AEAS comme dans l'évaluation de l'EQM. D'après cette première expérience, nous confirmons les bonnes performances de l'algorithme proposé SFNLMS dans l'application d'AEAS.

Dans une deuxième expérience, nous gardons les mêmes configurations qua dans la première (cité ci-dessus) mais cette fois-ci nous utilisons des réponses impulsionnelles réelle longues de taille $L=1024$. Les résultats obtenus de cette expérience sont donnés à la figure 5.18.

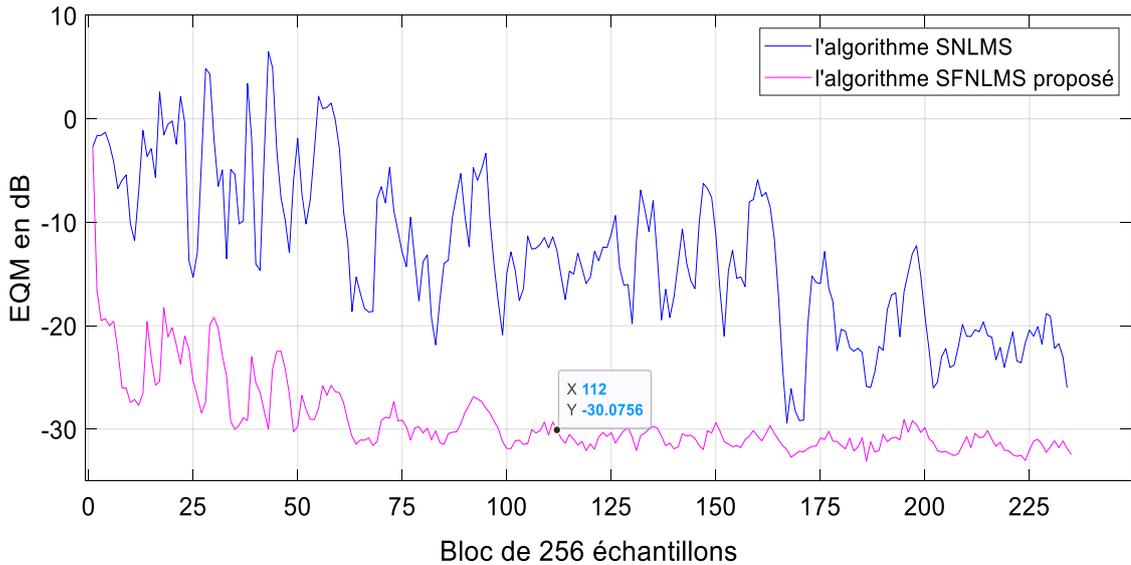


Figure 5.18. Evaluation de l'EQM obtenues avec SFNLMS et SNLMS dans un contexte d'AEAS avec une taille de RI longue $L=1024$.

D'après la figure 5.18, nous remarquons facilement que l'algorithme proposé SFNLMS garde sa supériorité en matière de EQM, vitesse de convergence, et stabilité en régime permanent, même avec des réponses impulsionnelles de filtre adaptatif longues.

5.6.2. Algorithmes sélectifs des données pour l'AEAS

Dans les simulations suivantes, nous allons présenter les performances des algorithmes proposés et étudiés précédemment (ES-SM-NLMS, SM-PU-SNLMS, SNP-VSS-N et SNP-VSS-PU-N) (voir chapitre 4) les techniques utilisées qui permettent de réduire la complexité de calcul et améliorer les performances d'annulation d'écho acoustique. Nous avons utilisé les deux critères objectifs de l'EQM et le Misalignement.

Pour des simplifications de lecture, nous divisons les simulations à deux parties. La première partie étudie les deux algorithmes qui utilisent le filtrage par appartenance à un ensemble SMF (ES-SM-NLMS et SM-PU-SNLMS). La deuxième partie étudie les deux algorithmes basés sur le filtrage par un pas variable non paramétrique NP-VSS (SNP-VSS-N et SNP-VSS-PU-N).

a. Filtrage par appartenance à un ensemble

Dans une première étape, nous testons l'influence du paramètre d'erreur γ de la technique SMF. Pour se faire, nous fixons tous les autres paramètres et nous varions γ . Le signal d'entrée est un bruit blanc Gaussien, les réponses impulsionnelles sont choisies courtes de

taille $L=64$, le rapport signal à bruit est $RSB=30$ dB et le facteur de non-linéarité $\alpha_r=0,5$. Les résultats obtenus sont donnés aux figures 5.19 et 5.20.

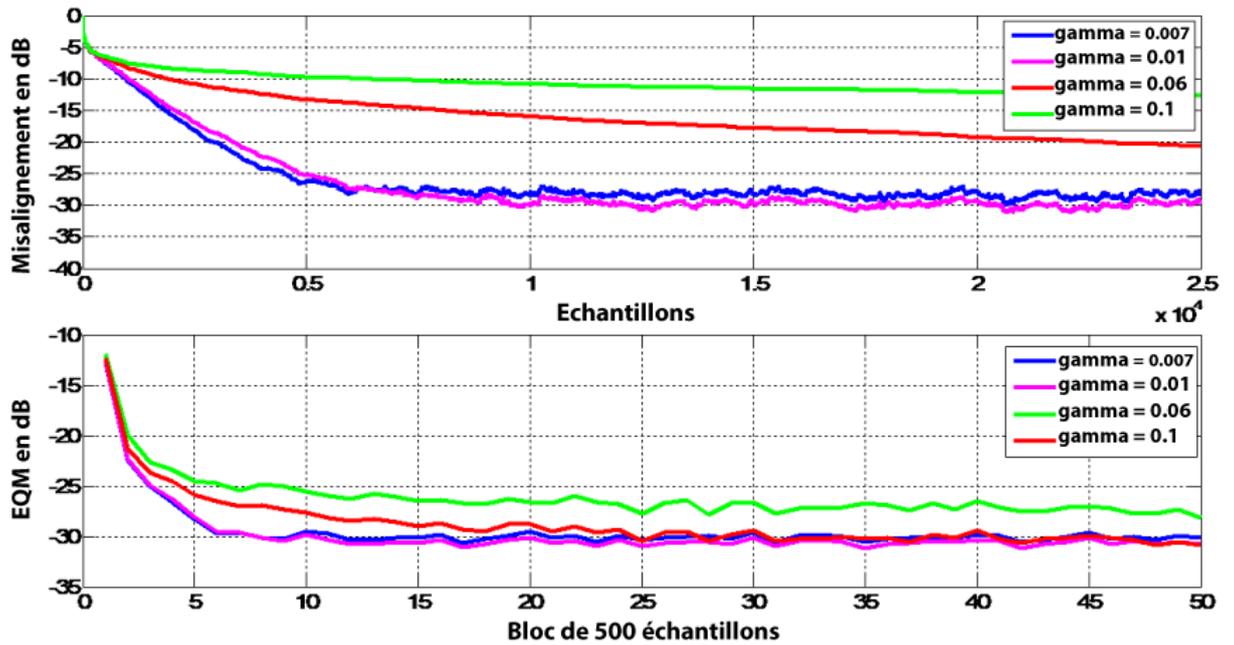


Figure 5.19. Evaluation de l'EQM avec l'algorithme SM-PU-SNLMS pour γ variables.

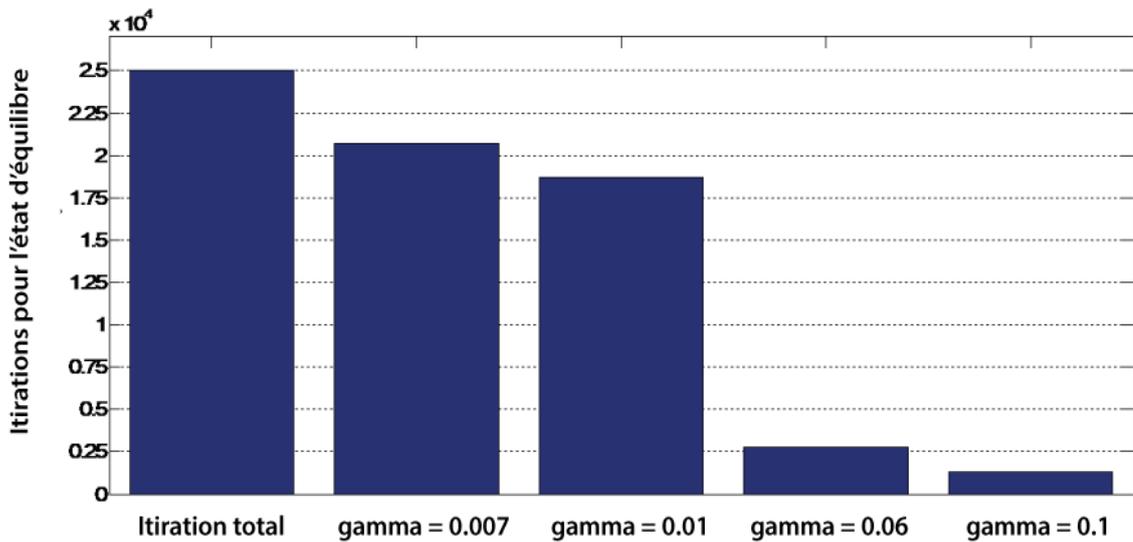


Figure 5.20. Nombre d'itérations nécessaires pour atteindre le régime permanent.

À partir de la figure 5.19, nous pouvons dire que lorsque γ décroît de 0,1 à 0,01, l'EQM et le Misalignement diminuent. Cependant, ils cessent de diminuer lorsque $\gamma < 0,01$, car les niveaux attendus de l'EQM et le Misalignement sont déjà atteints.

À partir de la figure 5.20, nous pouvons voir facilement que lorsque γ diminue, le nombre d'itérations augmente jusqu'à l'état d'équilibre.

D'après les figures 5.19 et 5.20, nous trouvons le meilleur compromis entre le niveau de l'EQM / Misalignement et le nombre d'itérations jusqu'à l'état d'équilibre. La valeur choisie de cette expérience pour les étapes suivantes est $\gamma = 0,01$.

- **Test avec bruit USASI**

Dans cette expérience, nous évaluons les performances des algorithmes proposés (ES-SM-NLMS et SM-PU-SNLMS), XMNL-NLMS avec l'algorithme classique SNLMS. Le signal d'excitation est un bruit USASI, les réponses impulsionnelles sont courtes de taille $L=256$, le rapport signal à bruit $RSB=30$ dB, le pas d'adaptation est choisi égale à 1, l'erreur $\gamma = 0,01$ de la technique SMF pour les deux algorithmes proposés et le facteur de non-linéarité est $\alpha_r=0,5$. Les résultats de cette expérience sont donnés à la figure 5.21 :

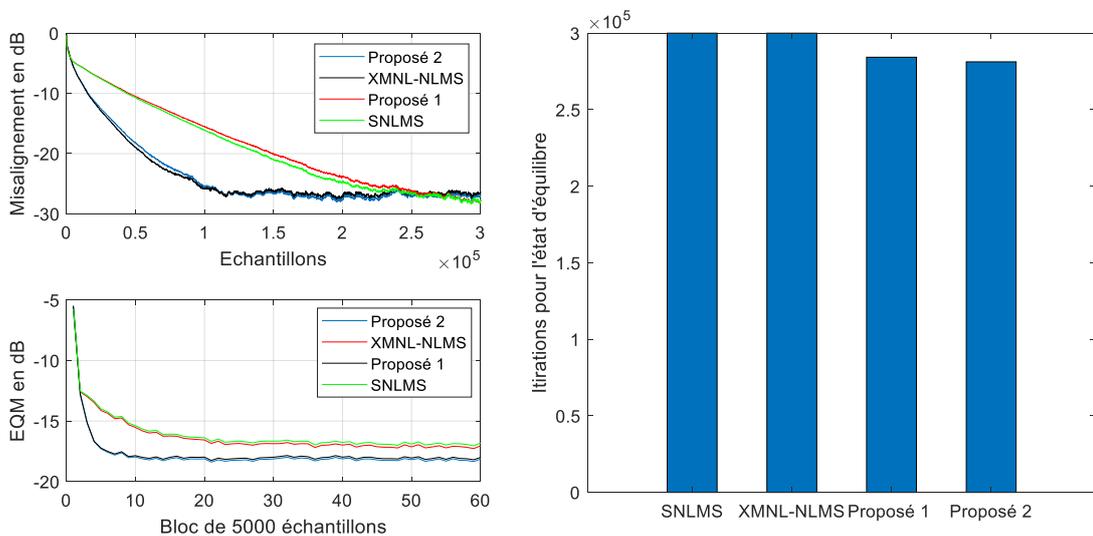


Figure 5.21. Evaluation de l'EQM, le Misalignement et le nombre d'itérations nécessaires pour atteindre l'état d'équilibre en utilisant le bruit USASI comme signal d'entrée avec les algorithmes proposés.

D'après la figure 5.21, nous notons que les algorithmes proposés ont conservé les mêmes performances en matière de l'EQM et le Misalignement lorsque la technique de filtrage SMF est ajoutée à la version d'origine. L'algorithme proposé ES-SM-NLMS a presque le même niveau de l'EQM et le Misalignement que l'algorithme SNLMS et l'algorithme proposé SM-PU-SNLMS est proches de l'algorithme XMNL-NLMS (chaque algorithme sa dérivée).

Nous notons également que les deux algorithmes proposés offrent l'avantage de la faible complexité en conservant le même nombre d'itérations lorsque l'état d'équilibre est atteint. En outre, l'algorithme proposé SM-PU-SNLMS, contrairement à d'autres algorithmes

présentés, permet un bon compromis entre une vitesse de convergence rapide, une faible complexité et un faible niveau de l'EQM et un bon Misalignement.

- **Test des algorithmes proposés dans un contexte d'AEAS**

Dans cette expérience, nous évaluons les performances de l'annulation d'écho acoustique stéréophonique des algorithmes proposés (ES-SM-NLMS et SM-PU-SNLMS), l'algorithme XMNL-NLMS et l'algorithme SNLMS. Le signal d'excitation est un signal parole présenté à la figure 5.3, les réponses impulsionnelles sont courtes de taille $L=256$ pour le premier test et longue de taille $L=1024$ pour le deuxième test, le rapport signal sur bruit $RSB=50$ dB, le pas d'adaptation est choisi égale à 1, l'erreur $\gamma = 0,01$ de la technique SMF pour les deux algorithmes proposés et le facteur de non-linéarité est $\alpha_r=0,5$. Les résultats de cette expérience sont donnés à la figure 5.22 pour des réponses impulsionnelles courtes et à la figure 5.23 pour des réponses impulsionnelles longues :

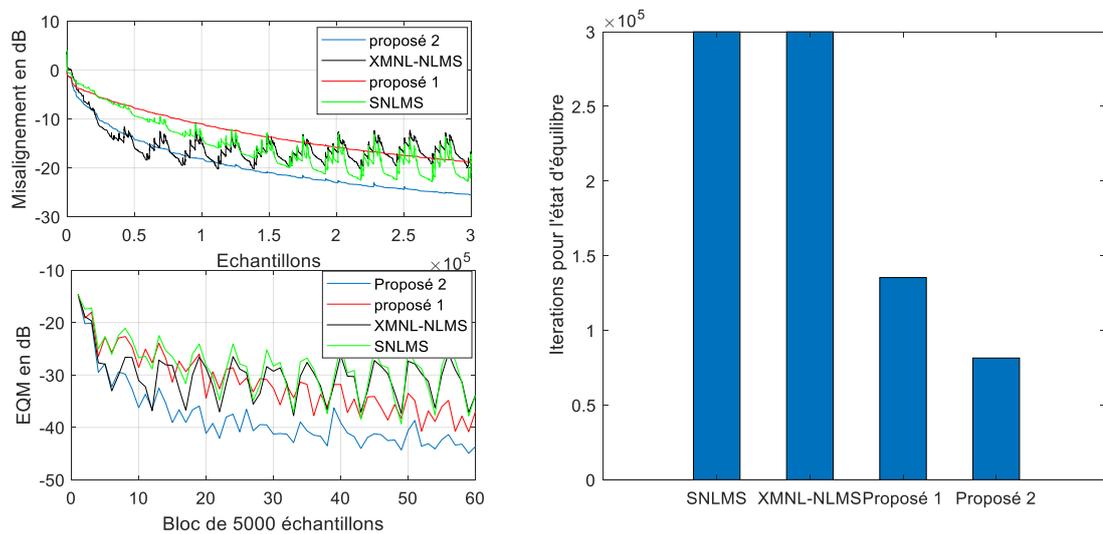


Figure 5.22. Evolution de l'EQM, le Misalignement et le nombre d'itérations nécessaires pour atteindre l'état d'équilibre avec les algorithmes proposés et une RI de taille $L=256$.

A partir de la figure 5.22, nous notons que l'algorithme proposé SM-PU-SNLMS présente de bonnes performances d'AEAS, que ce soit en EQM et Misalignement ou en nombre d'itérations nécessaires pour atteindre le régime permanent. Nous notons, également, que l'algorithme proposé ES-SM-NLMS présente des performances appropriées par rapport aux algorithmes SNLMS et XMNL-NLMS.

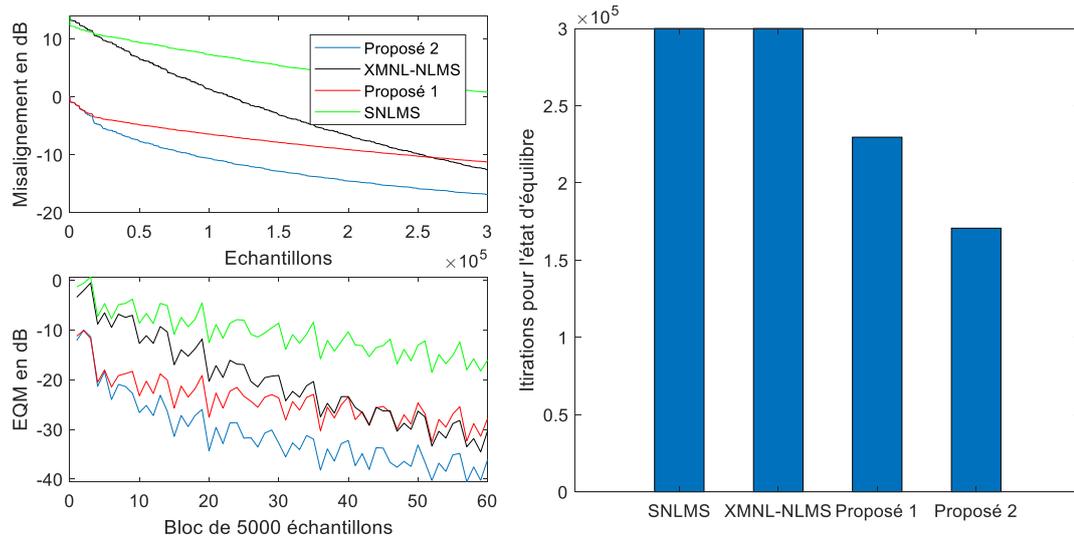


Figure 5.23. Evolution de l'EQM, le Misalignement et le nombre d'itérations nécessaires pour atteindre l'état d'équilibre avec les algorithmes proposés et une RI de taille $L=1024$.

D'après la figure 5.23, Nous notons que les algorithmes proposés ont conservé leur supériorité, en matière de EQM, de Misalignement et de nombre d'itérations jusqu'à l'état d'équilibre, même si les réponses impulsionnelles des filtres adaptatifs sont longues. A partir de ces expériences, nous remarquons que les algorithmes proposés (ES-SM-NLMS et SM-PU-SNLMS) ont une performance appropriée lorsque le signal d'entrée est un signal non stationnaire tel que le signal de parole présenté à la figure 5.3, avec des bons paramètres choisis.

b. Filtrage par un pas variable non paramétrique

Dans cette deuxième partie, nous étudions les deux algorithmes proposés qui utilisent le filtrage par un pas variable non paramétrique NP-VSS (SNP-VSS-N « proposé 3 » et SNP-VSS-PU-N « proposé 4 »), nous utilisons les mêmes signaux de test présentés précédemment.

- ***Test avec le bruit USASI***

Dans cette expérience, nous évaluons les performances des algorithmes proposés (SNP-VSS-N et SNP-VSS-PU-N) et l'algorithme SNLMS. Le signal d'excitation est un bruit USASI, les réponses impulsionnelles sont courtes de taille $L=256$, le rapport signal à bruit $RSB=50$ dB, le pas d'adaptation est choisi égale à 0.5, pour l'algorithme SNLMS, le facteur Q de la technique NP-VSS est choisi égale à 1000 pour les deux algorithmes proposés. Le facteur de non-linéarité est $\alpha_r=0,5$. Les résultats de cette expérience sont donnés aux figures 5.24 et 5.25.

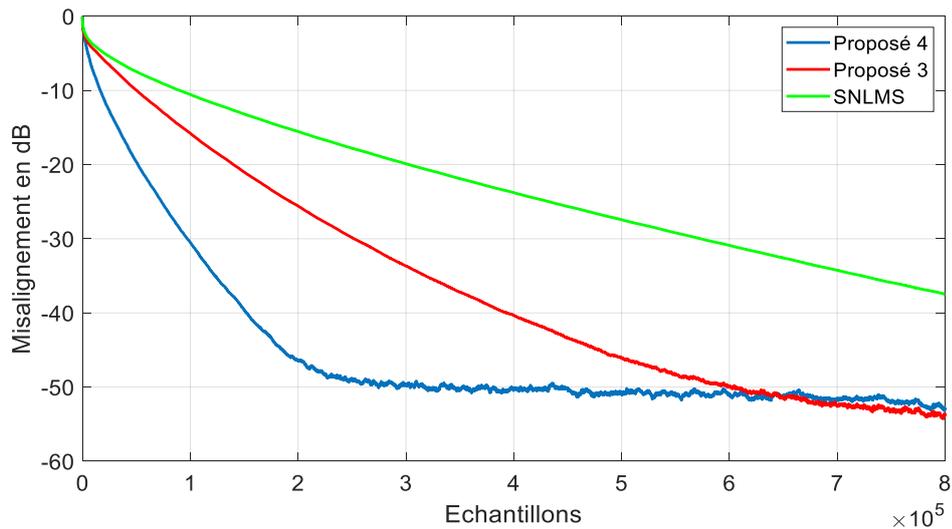


Figure 5.24. Evaluation du Misalignement obtenu avec les algorithmes proposés en utilisant le bruit USASI en entrée et une RI de taille $L=256$.

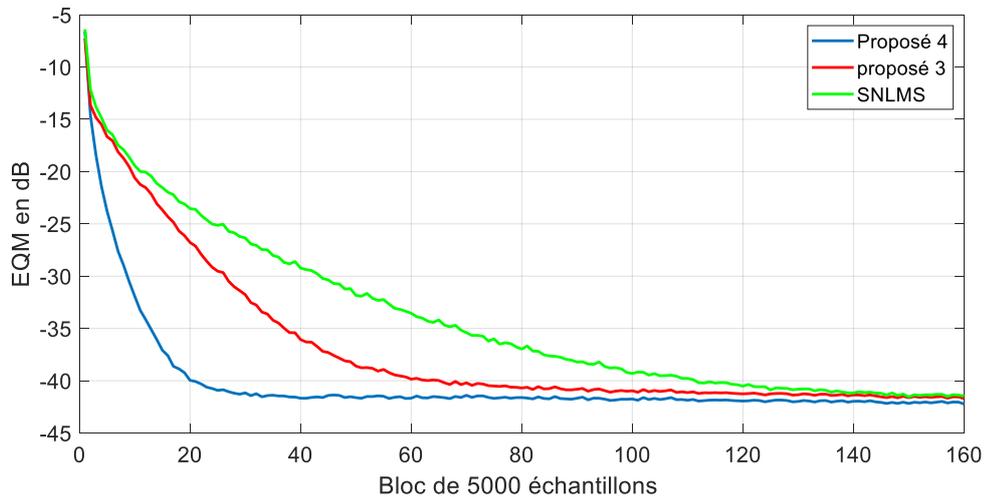


Figure 5.25. Evaluation du Misalignement obtenu avec les algorithmes proposés en utilisant le bruit USASI en entrée et une RI de taille $L=256$.

D'après les deux figures 5.24 et 5.25, nous notons que l'algorithme proposé SNP-VSS-PU-N donnent les meilleurs résultats en matière de vitesse de convergence et la valeur final de l'EQM et le Misalignement en état d'équilibre, entre autre, la réduction de la complexité de calcul. Cette supériorité est dû essentiellement aux combinaisons des deux techniques, la technique de mise à jour partielle qui accélère la vitesse et réduit la complexité et la technique de filtrage par un pas variable non paramétrique qui donne la bonne compromise entre la vitesse de convergence et le niveau final de l'état d'équilibre. L'algorithme proposé SNP-VSS-N donne aussi de résultat acceptable en comparaison avec l'algorithme classique SNLMS.

- *Test d'AEAS avec le signal parole*

Dans cette expérience, nous évaluons les performances de l'annulation d'écho acoustique stéréophonique des algorithmes proposés (SNP-VSS-N et SNP-VSS-PU-S) en comparaison avec l'algorithme classique SNLMS. Le signal d'excitation est la parole présentée à la figure 5.3, les réponses impulsionnelles sont courtes de taille $L=256$, le rapport signal à bruit $RSB=50$ dB, le pas d'adaptation est choisi égale à 1 pour l'algorithme SNLMS, le paramètre Q de la technique NP-VSS est choisi $Q=300$. Le facteur de non-linéarité est $\alpha_r=0,5$. Les résultats de cette expérience sont donnés à la figure 5.26 pour l'évaluation de Misalignement et à la figure 5.27 pour l'évaluation de l'EQM.

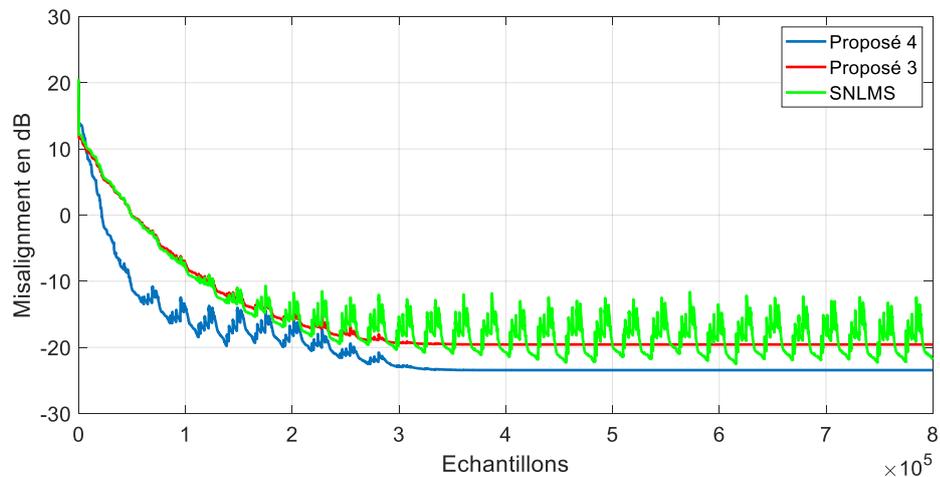


Figure 5.26. Evaluation du Misalignement obtenu avec les algorithmes proposés en utilisant le signal de parole en entrée et une RI de taille $L=256$.

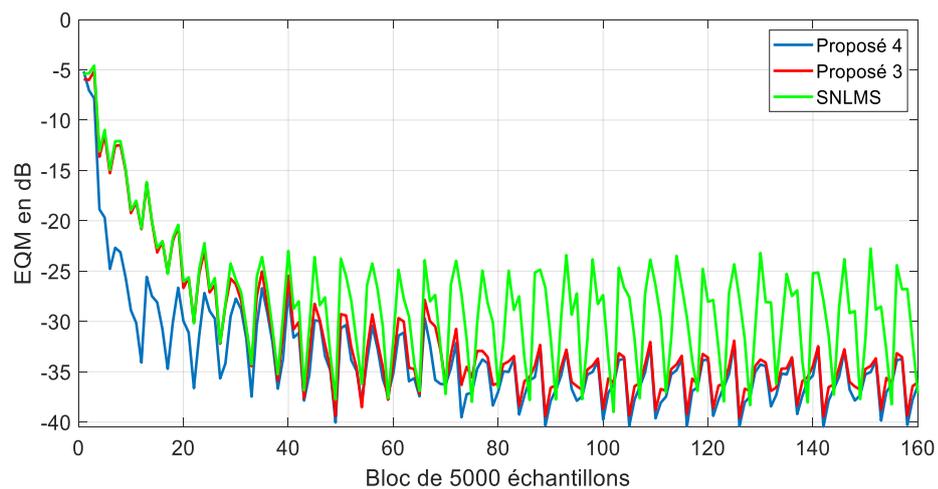


Figure 5.27. Evaluation de l'EQM obtenue avec les algorithmes proposés en utilisant le signal de parole en entrée et une RI de taille $L=256$.

D'après les figures 5.26 et 5.27, nous notons que les algorithmes proposés ont conservé leur supériorité, en matière de l'EQM et le Misalignement lorsqu'on utilise le signal de parole en entrée avec un léger changement dû au choix des paramètres de l'entrée. Nous remarquons aussi que les algorithmes proposés sont plus stables en état permanent lors de l'évaluation de l'EQM finale.

5.6.3. Complexité de calcul

La complexité de calcul (nombre d'opérations numériques : les multiplications, les additions et les divisions) des algorithmes d'AEAS proposés est présentée dans le Tableau 5.1 suivant :

Tableau 5.1 : Complexité de calcul des algorithmes proposés

Algorithme	Nombre de multiplications	Nombre d'addition	Nombre de division
SNLMS	$6L+2$	$6L$	1
SFNLMS	$6L+22$	$4L+21$	8
ES-SM-NLMS	$6L+2$	$6L+1$	2
SNP-VSS-N	$6L+5$	$6L+4$	3
SM-PU-SNLMS	$5L+2$	$5L+2$	2
SNP-VSS-PU-N	$5L+5$	$5L+5$	3

5.6.4. Synthèse de l'évaluation des performances des algorithmes proposés

Le Tableau 5.2 résume les résultats les plus importants obtenus lors des tests précédents des algorithmes proposés. L'évaluation peut varier de 5 à 1, où une valeur de 5 est donnée pour la meilleure performance et une valeur de 1 pour la mauvaise performance.

Tableau 5.2 : Synthèse de l'évaluation des performances des algorithmes proposés

Algorithme	Complexité de calcul	Vitesse de convergence	Niveau final de EQM et ERLE	complexité moyenne (Nbr de mises à jour nécessaires)
SFNLMS	1	5	5	1
ES-SM-NLMS	3	1	1	2
SNP-VSS-N	2	2	2	3
SM-PU-SNLMS	5	3	3	4
SNP-VSS-PU-N	4	4	4	5

5.7. Conclusion

Dans ce chapitre, nous avons décrit les signaux de test ainsi que les critères objectifs de qualité. A travers les simulations, nous évalué l'effet des paramètre (taille du filtre et le pas d'adaptation) des algorithmes sur la performance de la vitesse de convergence ainsi que leur comportement dans le régime transitoire et permanent et ceci dans le cas monophonique. Nous avons, par la suite, présenté les résultats de simulation obtenus avec

les algorithmes stéréophoniques proposés, et nous avons utilisé l'algorithme classique SNLMS comme un le référentiel de performance. Ces résultats ont bien montré que les algorithmes proposés sont bien supérieurs à l'algorithme stéréophonique classique SNLMS. Nous avons aussi montré que les algorithmes proposés peuvent être des choix intéressants pour l'AEAS car ils permettent de faire un compromis entre la complexité, le niveau de l'EQM finale, le niveau de l'ERLE et le niveau du Misalignement.

CHAPITRE 6: Implémentation sur FPGA « Description VHDL » d'un Algorithme d'Annulation d'Écho Acoustique Stéréophonique

6.1. Introduction

Le traitement du signal fait l'objet de recherches dans la plupart de laboratoires d'électronique. Le filtrage adaptatif est une technique indispensable dans le traitement du signal. L'implémentation de cette technique a été réalisé dans plusieurs langages et logiciels tels que MATLAB et langage de description matériel VHDL ou VERILOG.

Quelle que soit la cible, FPGA ou DSP, il demeure une phase de transformation sémantique difficile qui consiste à passer du type abstrait des variables manipulées par notre algorithme (variables réelles, complexes, entières...etc.) à un type logique comme, par exemple, le vecteur de bits qui sera admis pour le système RTL (Register Transfert Logic).

Plusieurs travaux ont traité de la conception VHDL pour les filtres adaptatifs. Par exemple, dans [52-58], les algorithmes de filtrage adaptatif les plus connu, LMS, NLMS et RLS ont été mis en œuvre pour l'AEA et l'identification des systèmes inconnus. La validation et la comparaison entre ces algorithmes de base a été effectuée dans les deux contextes, c'est-à-dire sous MATLAB et VHDL, ainsi que l'avantage de chacun par rapport aux autres. Ces implémentations ont été expliqués et testés en terme de vitesse de convergence et de complexité. Des techniques plus avancées sont ensuite proposées et mises en œuvre pour un algorithme plus robuste et complexe comme dans les références [58-61]. Dans ces références, les auteurs ont apporté des modifications en ajoutant une nouvelle technique permettant de mettre en œuvre des filtres adaptatifs pour l'AEAS et le système multicanal. Contrairement à la monophonie, dans la stéréophonie, nous avons besoin de techniques supplémentaires pour fonctionner correctement, et cela est dû à l'apparition de nouveaux défis, tels que la taille étendue des données et le problème de cohérence entre les canaux. Des résultats plus précis ont été présentés dans les références [62-66]. Dans [62], un nouvel algorithme appelé algorithme RLS à régularisation variable (VR-RLS) a été présenté pour la suppression du bruit et de l'AEA, basé sur la méthode de la réduction de coordonnées dichotomiques (DCD). Sur la base de [62], les auteurs dans la référence [63], ont présenté une idée originale pour obtenir une implémentation efficace sur FPGA de l'algorithme VR-RLS. Un autre exemple pratique d'implémentation avancée a été présenté dans l'article [64], il consiste en un noyau d'estimation par moindres carrés

récurifs (RLSE) pour l'activation du cerveau en temps réel. Dans l'article [65], un algorithme LMS à pas variable a été mis en œuvre sur la base d'une nouvelle structure, cette structure a été comparée à la forme directe et à la structure de forme transposée. En outre, la conception proposée a été testée dans différents types de carte FPGA afin de sélectionner le meilleur choix (en termes de vitesse, puissance et ressources) pour une mise en œuvre en temps réel. Dans la référence [66], un filtre adaptatif multi-ordres LMS pour l'annulation d'écho a été conçu et développé autour du SoPc (conception du système sur puce programmable) à l'aide de la plateforme Xilinx Zynq FPGA avec le microcontrôleur cortex double ARM, il présente un chemin de passage aisé à partir de microcontrôleur à la conception basée sur FPGA.

Dans ce chapitre, nous avons implémenté l'algorithme SFNLMS proposé et discuté dans le chapitre 3 en code VHDL pour FPGA [67]. Les résultats obtenus sont comparés avec celles obtenues avec MATLAB. Nous avons toujours pris l'algorithme SNLMS classique comme référence dans le test des performances.

6.2. FPGAs

Le FPGA (Field Programmable GateArray) a été présenté par Xilinx en 1984. C'est des circuits en silicium programmables. Ils sont devenus au milieu des années 90 plus complexes grâce à l'intégration des ressources spécifiques dédiées, associées à une mémoire interne et à des ports d'entrées/sorties flexibles [68].

Les FPGAs se composent d'une matrice de blocs logiques configurables (CLB) élémentaires permettant de réaliser des fonctions combinatoires et des fonctions séquentielles. Des blocs entrées/sorties IOB dont le rôle est de gérer les entrées-sorties réalisant l'interface avec les modules extérieurs et des ressources d'interconnexion entre les CLB. La Figure 6.1 montre un exemple schématique de cette structure des FPGAs [68].

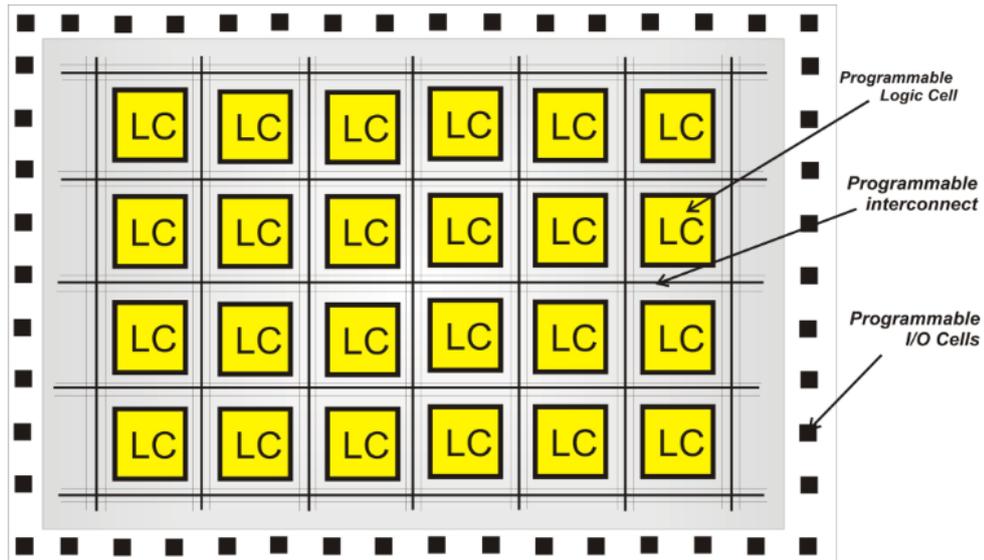


Figure 6.1. Structure interne de la carte FPGA

Les principales firmes de fabrications des circuits FPGAs sont Xilinx, Altera, Actel, Atmel, Quicklogic et autres.

6.3. Les langages de description matériel

Actuellement, ce n'est plus possible d'utiliser les outils de saisie d'un schéma pour développer les circuits logiques grâce à la densité de fonctions logiques (ports et bascules) intégrées dans le FPGA. Les sociétés de développement et les ingénieurs ont donc créé des langages dits de haut niveau à savoir aux langages de description matérielle de haut niveau (Hardware description language HDL). Deux d'entre eux sont couramment utilisés : VHDL (VHSIC « Very High Speed Integrated Circuit » Hardware Description Language) et Verilog. Par la suite de nos travaux en s'intéresse au langage VHDL pour le développement de nos algorithmes.

6.4. Structure d'une description VHDL

En VHDL, les structures logiques sont décrites sous la forme d'une paire de fonctions, l'entité (ENTITY) et l'architecture (ARCHITECTURE). Nous trouvons aussi des bibliothèques (LIBRARY) et des fois des paquets (package). De la façon suivante :

```
LIBRARY IEEE;

use IEEE.STD_LOGIC_1164.all;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_SIGNED.ALL;

PACKAGE FNLMS_FILT IS
-- DECLARE CONSTANTS
CONSTANT TAP : INTEGER := 24;
```

```

CONSTANT SCALE: INTEGER :=1000;
CONSTANT MU: INTEGER :=1;
CONSTANT LAMBDA_A: INTEGER :=9985;      --
LAMBDA_A*SCALE*10
CONSTANT LAMBDA: INTEGER :=970;        --
LAMBDA*SCALE
CONSTANT C0: INTEGER :=SCALE;
CONSTANT CA: INTEGER :=(12*SCALE);
--SUBTYPE BITS32 IS INTEGER RANGE -2**32 TO 2**31-1;
SUBTYPE BITS32 IS INTEGER RANGE -429496729 TO
429496729;
TYPE TMP_REG IS ARRAY (0 TO TAP-1) OF BITS32;
TYPE TMP_REG_PLUS_ONE IS ARRAY (0 TO TAP) OF BITS32;
END FNLMS_FILT;
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_ARITH.ALL;
USE IEEE.STD_LOGIC_SIGNED.ALL;
LIBRARY WORK;
USE WORK.FNLMS_FILT.ALL;

ENTITY FNLMS IS
PORT ( CLK : IN STD_LOGIC;
DATA_IN : IN BITS32;
DESIRED : IN BITS32;
ERROR_OUT : OUT BITS32;
DATA_OUT : OUT BITS32 );
END FNLMS;

ARCHITECTURE BEHAVIORAL OF FNLMS IS
DECLARATION DE COMPOSANTS
+ DECLARATION DES SIGANUX INTERNES
+ AUTRE DECALARATION ...
BEGIN
    INSTRUCTION CONCURENTES;
PROCESSUS;
    INSTANCES DE COMPOSANT;
END ARCHITECTURE;

```

L'entité peut être vue comme une boîte noire, elle donne les informations concernant les signaux d'entrées et de sorties de la structure, ainsi que leurs noms et leurs types. L'architecture décrit le comportement à l'intérieur de l'entité. Il est possible de créer plusieurs architectures différentes pour une même entité. Chacune de ces architectures peut décrire l'entité de façon différente.

6.5. Conception sur FPGA et mise en œuvre

6.5.1. Technique de codage de données

Sous MATLAB, nous lisons le signal vocal d'un fichier pour générer les signaux d'entrée $x_1(n)$ et $x_2(n)$. Ces signaux d'entrées passent par des réponses impulsions acoustiques de taille L , puis sommés pour générer le signal souhaité $d(n)$. Tous ces signaux sont multipliés par une échelle = 1000 et tronquent les résultats en une valeur entière. Cette échelle sera prise en compte une fois le processus terminé. De plus, pour une simple convolution, la multiplication par une échelle prédéfinie et la division du résultat par une même échelle n'ont aucun effet sur le calcul du processus. Pour un algorithme complexe, des modifications supplémentaires sont nécessaires pour obtenir des résultats corrects. Ces modifications sont répertoriées dans le tableau 6.1 suivant :

Tableau 6.1 : Modification requise pour les algorithmes complexes

Technique d'échelle

- À travers des simulations, trouvez l'échelle de précision requise.
- Multipliez tous les signaux de l'échantillon par l'échelle.
- Diviser par échelle lorsque deux valeurs mises à l'échelle sont multipliées.
- Multipliez par l'échelle lorsque deux valeurs mises à l'échelle sont divisées.

Remplacement

- Pour une multiplication par constante inférieure à 1 → remplacez par division par sa valeur réciproque.
 - Pour une division par constante inférieure à 1 → remplacez par multiplication par sa valeur réciproque.
-

Cette technique est nécessaire pour gérer la perte de précision lors du passage de la précision en virgule flottante à la précision en virgule fixe.

Pour effectuer une simulation en langage VHDL (logiciel ModelSim), les signaux d'entrée ont été générés à l'aide du logiciel MATLAB, puis sauvegardés dans un fichier spécifique via une fonction d'appel spécifique qui utilise la technique d'échelle précédente.

Ce fichier contient des valeurs entières de données. Après utilisation de ces données dans ModelSim, les résultats de sortie sont enregistrés dans un fichier spécifique, puis lus avec MATLAB et comparés aux résultats simulés du logiciel (MATLAB et ModelSim). La Figure 6.2 illustre le flux de données de MATLAB à ModelSim.

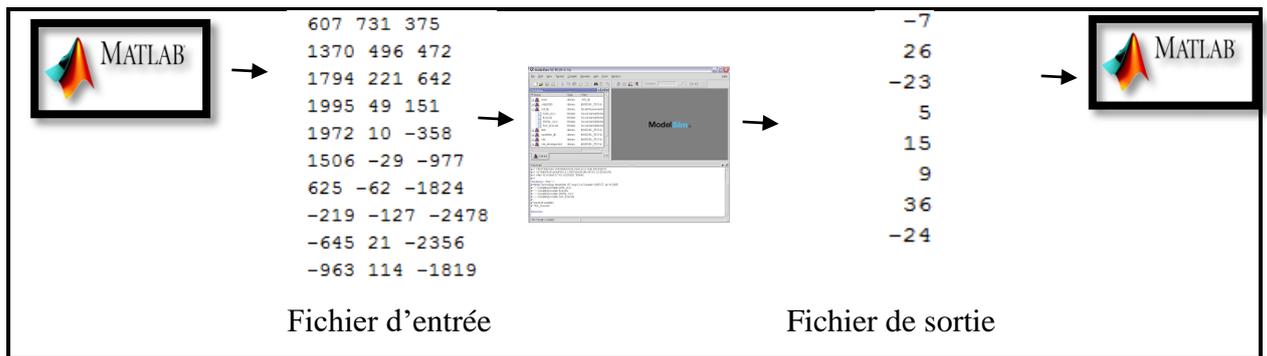


Figure 6.2. Les flux des données de MATLAB vers ModelSim.

6.5.2. Organigramme de l'AEAS par l'algorithme SNLMS

L'organigramme de l'algorithme SNLMS dans l'application AEAS est basé sur les points suivants :

- Lire les données échantillonnées et mises à l'échelle (entrée 1, entrée 2 et le signal souhaité) à partir du fichier « Fichier d'entrée » et stockées dans des registres à décalages.
- Calculez la sortie du filtre en additionnant les deux produits des deux entrées avec les poids des deux filtres.
- Calcule l'erreur entre le signal souhaité et la sortie du filtre.
- Mettez à jour les poids de filtres par l'équation de mise à jour SNLMS.
- Écrivez les valeurs d'erreur dans le fichier de sortie « Fichier de sortie » .

L'organigramme de l'algorithme SNLMS dans l'application AEAS est présenté dans la Figure 6.3.

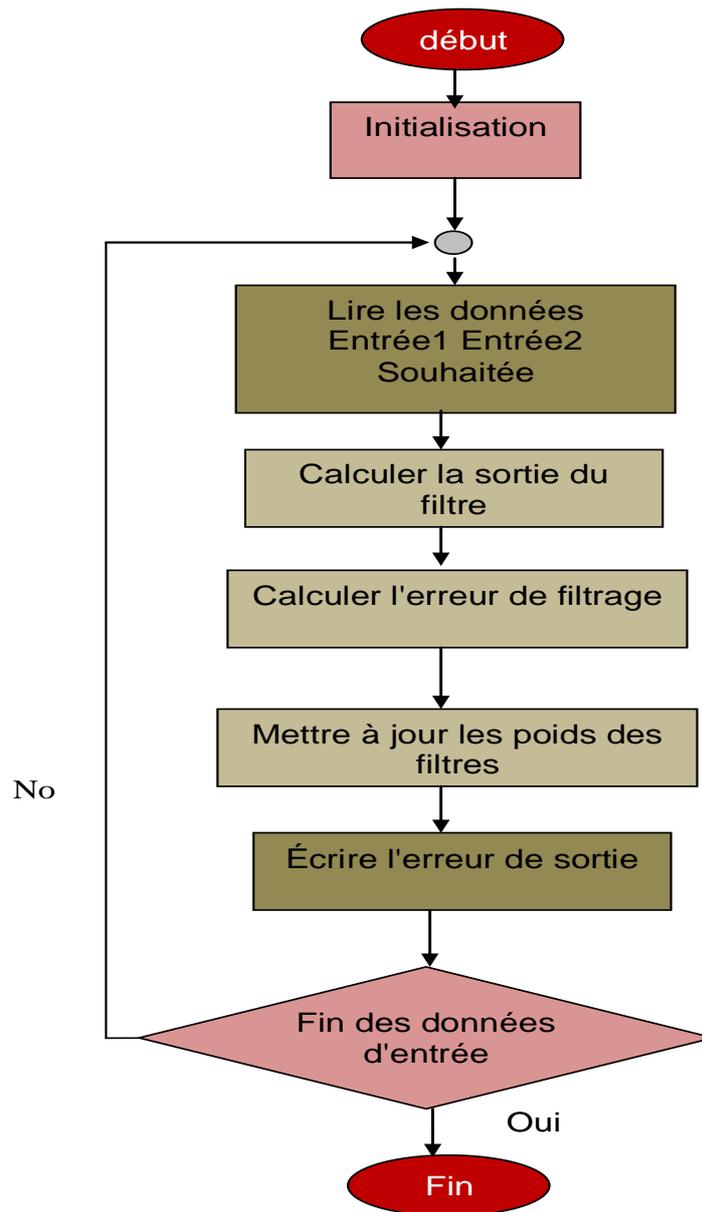


Figure 6.3. Organigramme de l'algorithme adaptatif SNLMS dans l'application AEAS.

6.5.3. Organigramme de l'AEAS par l'algorithme SFNLMS

L'organigramme de l'algorithme SFNLMS dans l'application d'AEAS est basé sur les points suivants :

- Lire les données échantillonnées et mises à l'échelle (entrée1, entrée2 et le signal souhaité) à partir du fichier « Fichier d'entrée » et stockées dans des registres à décalages.
- Calculez tous les paramètres nécessaires dans l'algorithme SFNLMS.

- Calculez la sortie du filtre en additionnant les deux produits des deux entrées avec les poids des deux filtres.
- Calcule l'erreur entre le signal souhaité et la sortie du filtre.
- Mettez à jour les poids de filtres par l'équation de mise à jour SFNLMS.
- Écrivez les valeurs d'erreur dans le fichier de sortie « Fichier de sortie ».

L'organigramme de l'algorithme SFNLMS dans l'application AEAS est présenté dans la Figure 6.4.

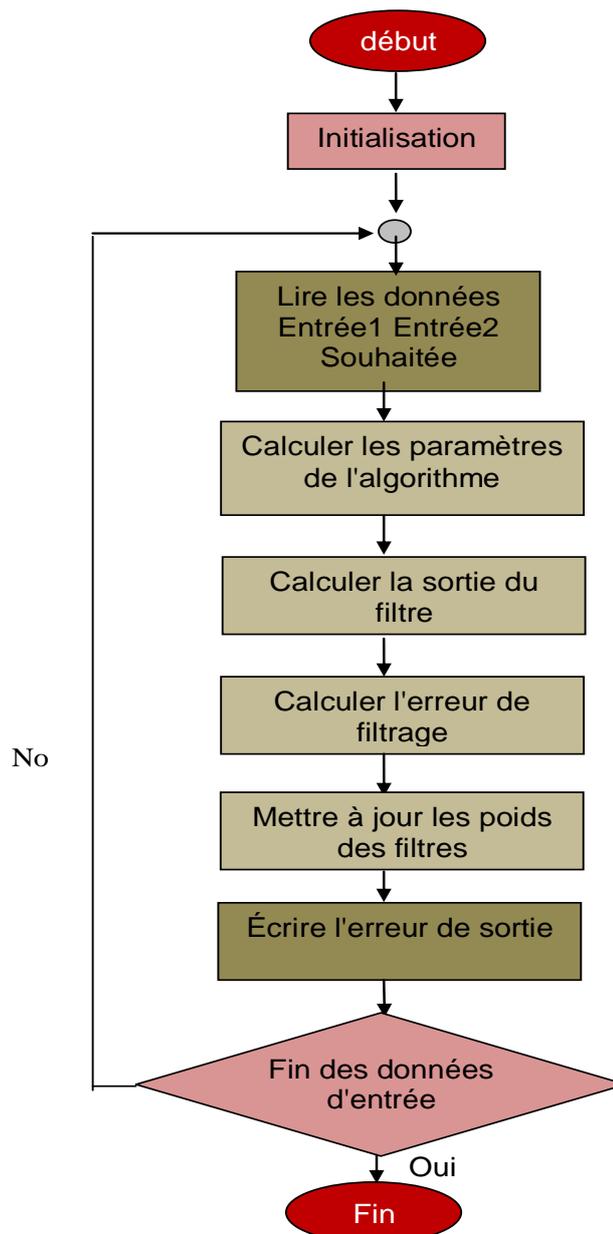


Figure 6.4. Organigramme de l'algorithme adaptatif SFNLMS dans l'application d'AEAS.

6.5.4. Astuces de conception VHDL et pièces de déclaration

Le code VHDL comprend au moins trois parties : les bibliothèques, les entités et les architectures. La bibliothèque fournit un ensemble de paquets, composants et fonctions qui simplifient la tâche de conception du matériel. Une entité est une spécification de l'interface externe de la conception, l'architecture spécifiant l'implémentation interne.

Dans ce travail, nous utilisons une entité appelée SNLMS ou SFNLMS tous dépend de l'algorithme implémenté, et une architecture appelée comportementale. L'interface d'entité est illustrée à la Figure 6.5.



Figure 6.5. Interface externe de l'algorithme SNLMS/ SFNLMS en VHDL.

Dans l'architecture de l'algorithme SNLMS, nous utilisons deux processus parallèles. Le premier utilisé pour stocker chaque donnée à venir dans un registre à décalage. Pour les entrées $x_1(n)$ et $x_2(n)$, les registres à décalage contenant les derniers L échantillons de données. Cependant, le registre de signal désiré ne contient que l'échantillon précédent. Ce processus sera activé chaque fois que l'horloge de front montant est surveillée.

Les registres à décalage sont déclarés en tant que types « signaux » dans la déclaration d'architecture SNLMS «Le type signal en VHDL représente un fil et un élément de stockage».

Le deuxième processus traite les calculs des opérations SNLMS. L'erreur et la sortie de filtrage sont déclarées en tant que variables dans la déclaration de processus « Le type variable en VHDL fournit une affectation instantanée du résultat, variable locale». Cependant, les poids des filtres sont déclarés en tant que type signal dans la déclaration

d'architecture «variable globale» afin d'enregistrer les valeurs précédentes de ces poids. Puis utilisé dans la prochaine itération.

Pour l'architecture SFNLMS, nous utilisons également deux processus parallèles. Le premier est similaire au processus de stockage SNLMS avec le même rôle et la même instruction. Où, le second est légèrement différent pour gérer l'opération SFNLMS : La sortie de filtrage, l'erreur de filtrage, les paramètres de prédiction, les erreurs de prédiction directe et les variables de vraisemblance sont déclarés dans le processus en tant que variables. Alors que les poids de filtres, les gains de Kalman et les variances d'erreur de prédiction directe sont déclarés en tant que signaux dans la déclaration d'architecture «variable globale». Des variables et des signaux plus implicites sont utilisées pour atteindre les variables connues.

Certains de ces signaux et variables sont des images des signaux d'entité. Par conséquent, nous avons associé les entrées de l'entité à ses images au début du processus. Puis, à la fin, la sortie de l'entité à ses images à la fin de l'architecture.

Ces deux processus sont en pipeline pour accélérer la vitesse de calcul. La Figure 6.6 montre bien l'étape en cascade du processus de traitement.

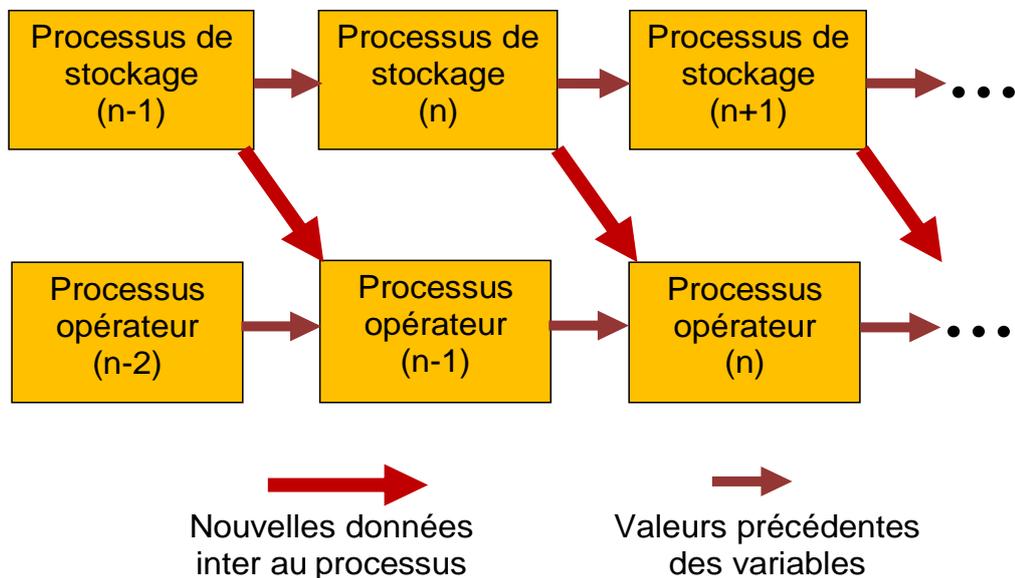


Figure 6.6. Les étapes en cascade du processus de traitement.

6.6. Résultats de simulation

Dans cette section, nous présentons les résultats des expériences réalisées sur MATLAB et ModelSim, dans lesquels les algorithmes SNLMS et SFNLMS proposés sont simulés en MATLAB (code MATLAB) et en ModelSim (code VHDL) avec différents signaux d'entrée.

Dans notre simulation, nous utilisons une réponse impulsionnelle simulée de longueur $L = 256$, et tous les signaux d'entrée sont échantillonnés à 16 kHz. Le SNR de sortie est choisi égal à 30 dB; les pas d'adaptations sont sélectionnées $\mu_{\text{SNLMS}} = \mu_{\text{SFNLMS}} = 1$. Le reste des paramètres de l'algorithme SFNLMS proposés est donné dans le tableau 6.2.

Tableau 6.2. Les paramètres de réglage SFNLMS

Paramètre	Valeur
Facteur d'oubli exponentiel λ_a	0.9985
Facteur d'oubli λ	0.99
Constante positive c_0	1
Constante positive c_a	1

Dans la première expérience, nous testons l'algorithme SFNLMS proposé et l'algorithme SNLMS classique dans l'application d'AEAS avec un signal d'entrée non corrélé sous MATLAB et ModelSim. Le signal d'entrée est un bruit blanc gaussien (WGN) stationnaire qui donne des informations sur la stabilité numérique des algorithmes simulés. Les résultats obtenus par ModelSim pour SNLMS (Figure 6.7) et SFNLMS (Figure 6.8). Les résultats de comparaison entre ces deux algorithmes en termes de MSE et ERLE sont donnés respectivement par la Figure 6.9 et Figure 6.10.

Ces résultats montrent que les résultats de simulations sous MATLAB et ModelSim sont comparables avec les mêmes paramètres de réglage. De plus, l'algorithme SFNLMS proposé a montré une grande stabilité par rapport à l'algorithme classique SNLMS dans les environnements MATLAB et ModelSim.

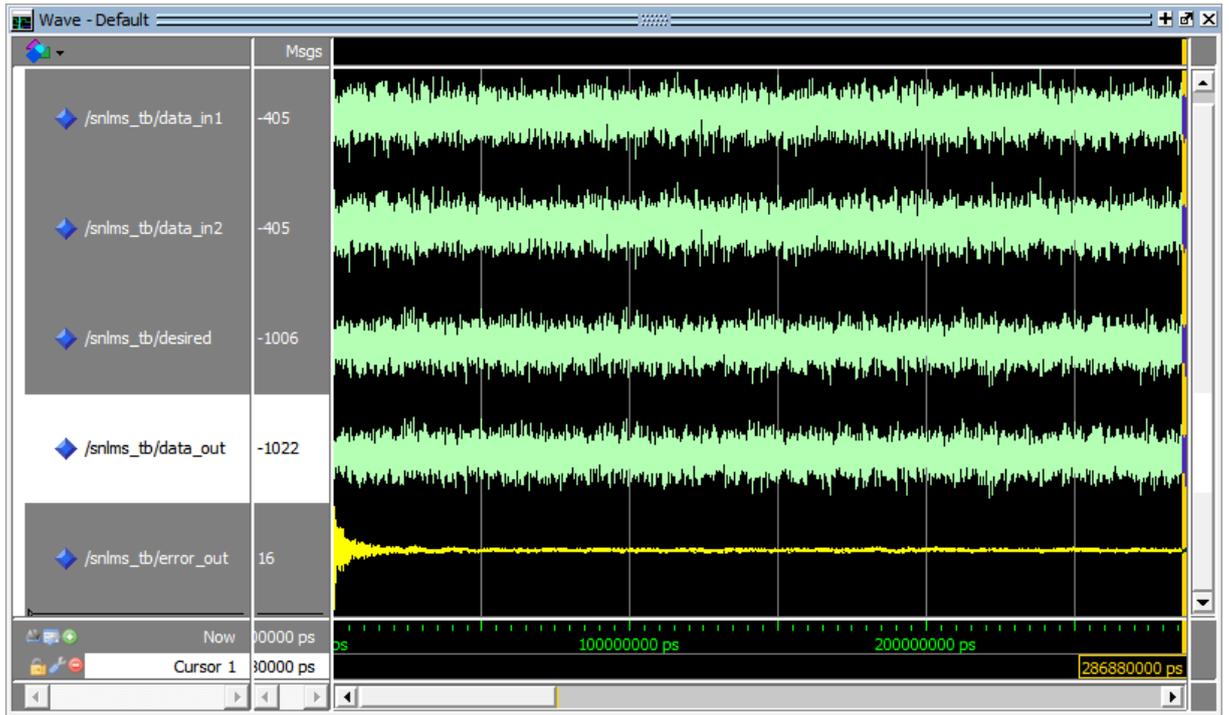


Figure 6.7. Simulation de l'algorithme SNLMS pour AEAS en ModelSim avec bruit blanc en entrée.

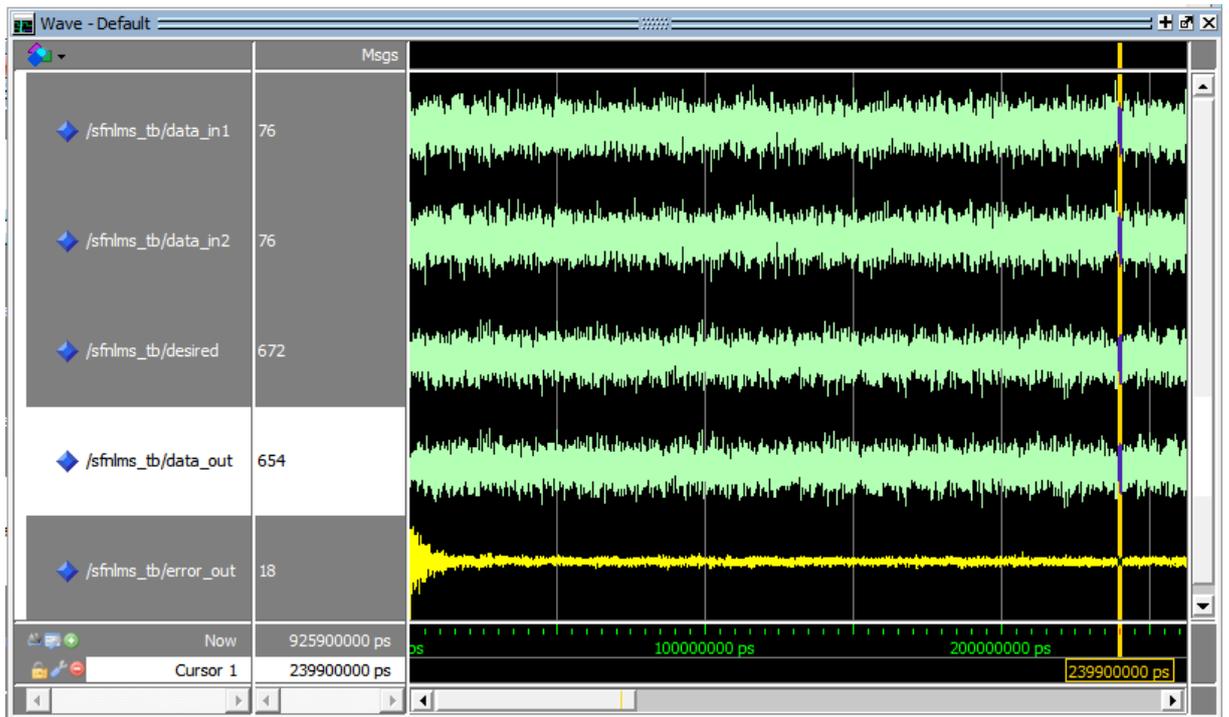


Figure 6.8. La simulation de l'algorithme SFNLMS pour AEAS en ModelSim avec bruit blanc en entrée.

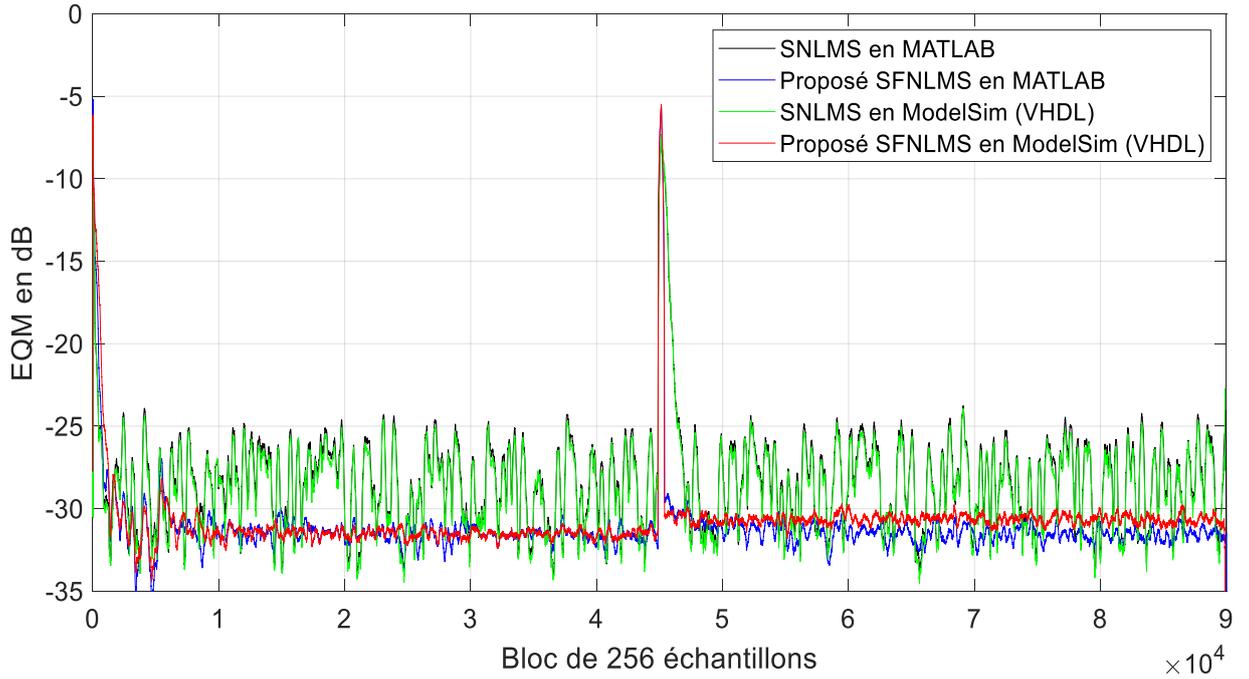


Figure 6.9. Évaluation de EQM obtenu par les deux algorithmes SNLMS et SFNLMS sur MATLAB et ModelSim avec bruit blanc en entrée.

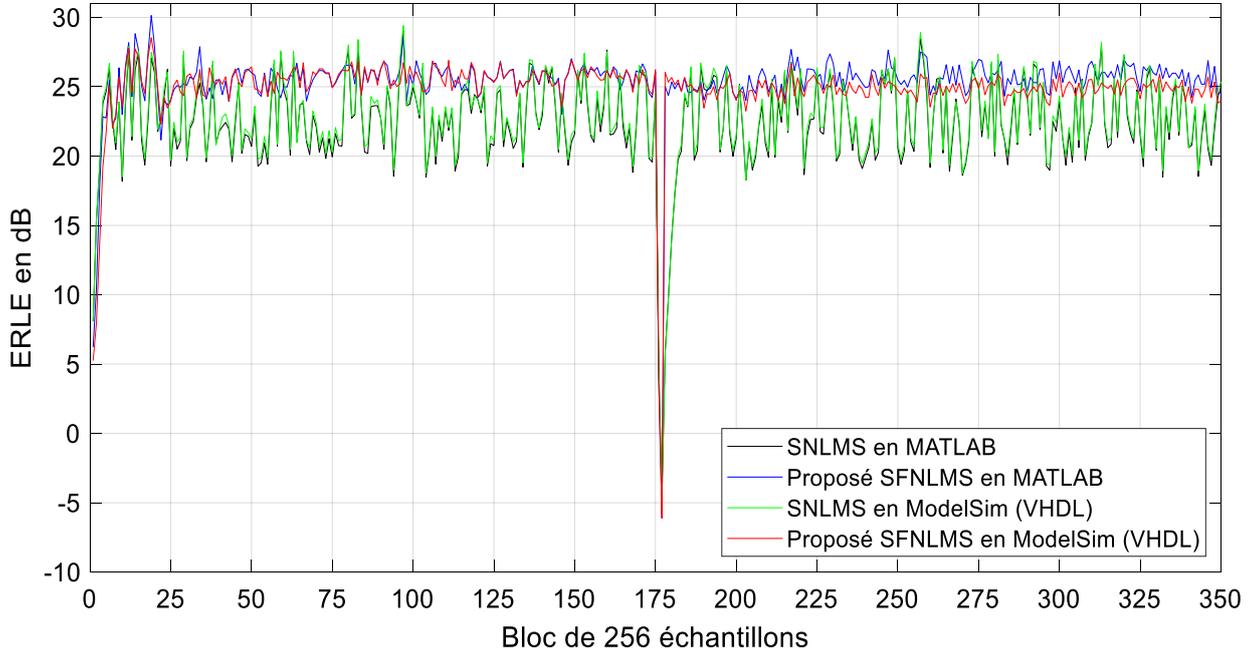


Figure 6.10. Évaluation de l'ERLE obtenu par les deux algorithmes SNLMS et SFNLMS sur MATLAB et ModelSim avec bruit blanc en entrée.

Dans la deuxième expérience, nous testons l'algorithme SFNLMS proposé et l'algorithme classique SNLMS dans l'application d'AEAS avec signal d'entrée corrélé sous MATLAB et ModelSim. Le signal d'entrée est le bruit USASI (United State of America Standards Institut maintenant ANSI) qui donne des informations sur la vitesse de convergence des algorithmes simulés en phase transitoire.

Les résultats des signaux E/S obtenus par ModelSim sont illustrés à la figure 6.11 pour SNLMS et à la Figure 6.12 pour SFNLMS. Les résultats de comparaison entre ces deux algorithmes en termes de MSE et ERLE sont donnés respectivement à la Figure 6.13 et la Figure 6.14. Ces résultats permettent de constater que les résultats de simulations obtenus sous MATLAB et ModelSim sont très proches en utilisant les mêmes paramètres de réglage.

De plus, l'algorithme proposé SFNLMS a une plus grande vitesse de convergence plus grande que celle du SNLMS et cela dans la phase transitoire et sous les deux environnements de travail, c'est-à-dire sous MATLAB et ModelSim. Nous pouvons remarquer une légère différence entre les résultats obtenus sous par MATLAB et ModelSim en phase transitoire de l'algorithme proposé SFNLMS. Cette différence dépend des conditions initiales. En effet, nous pouvons remarquer l'absence de cette différence dans la phase transitoire après le saut simulé au milieu du temps d'exécution.

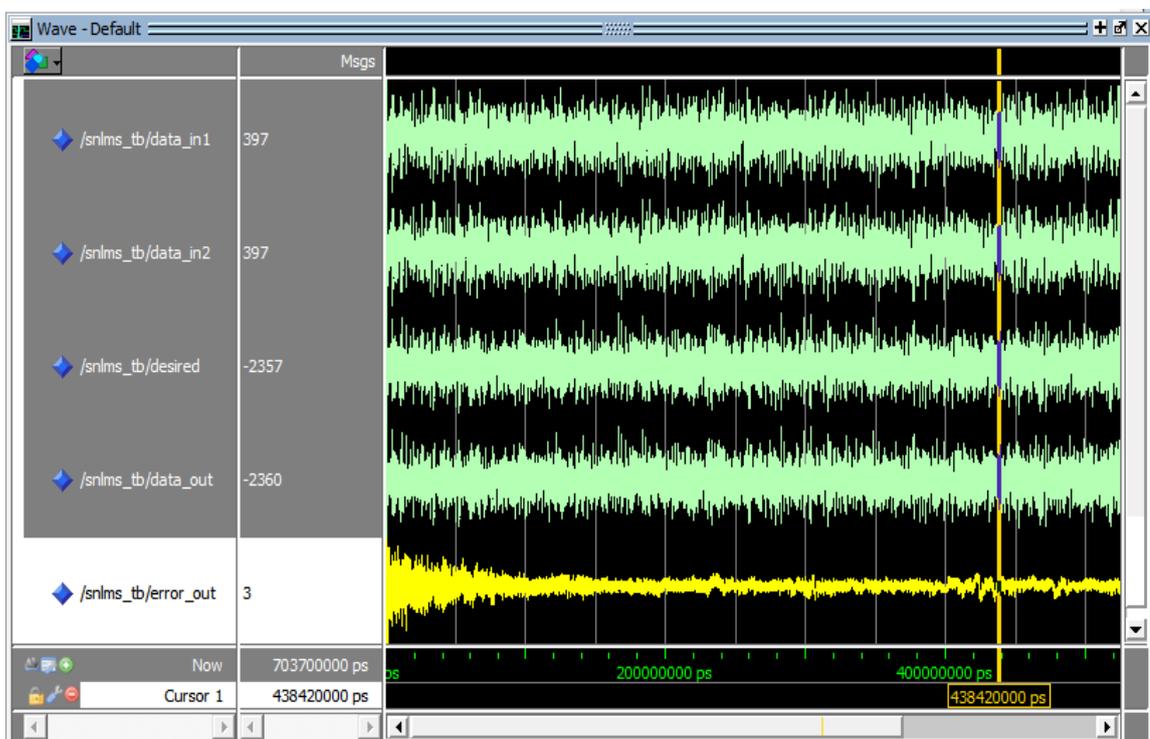


Figure 6.11. La simulation de l'algorithme SNLMS pour AEAS en ModelSim avec le bruit USASI en entrée.

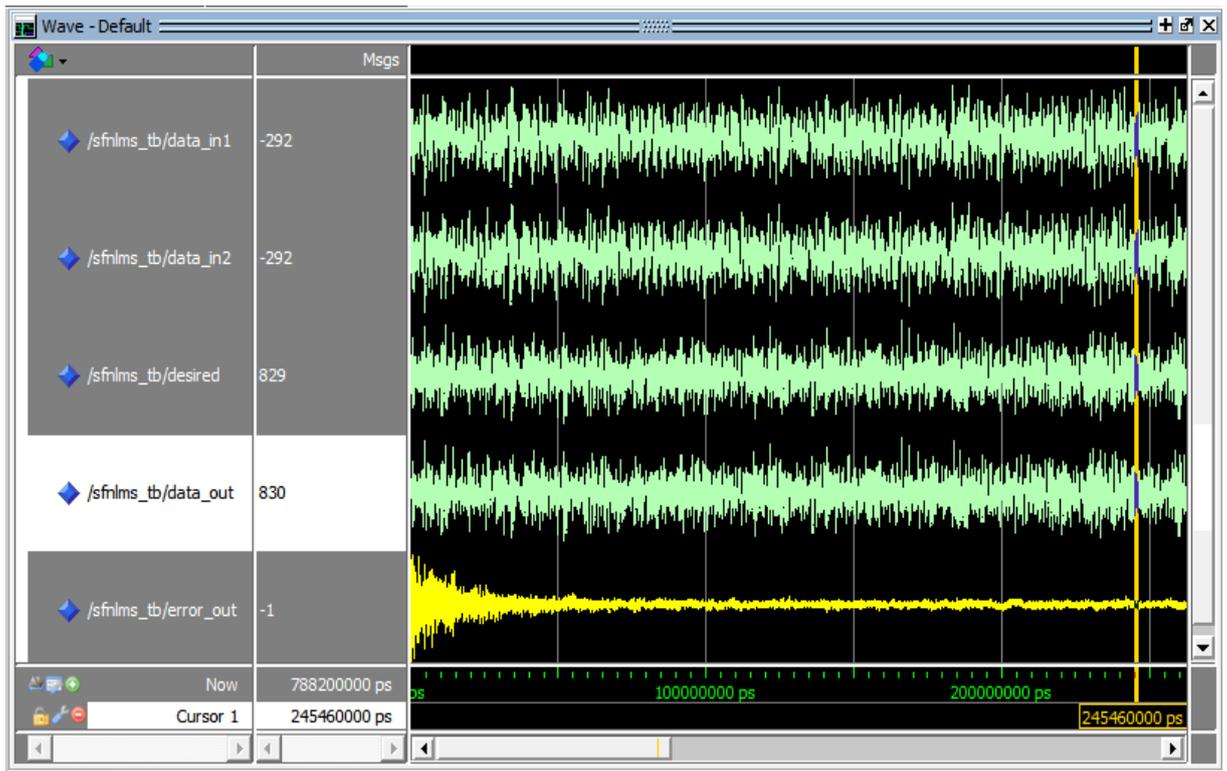


Figure 6.12. La simulation de l'algorithme SFNLMS pour AEAS en ModelSim avec le bruit USASI en entrée.

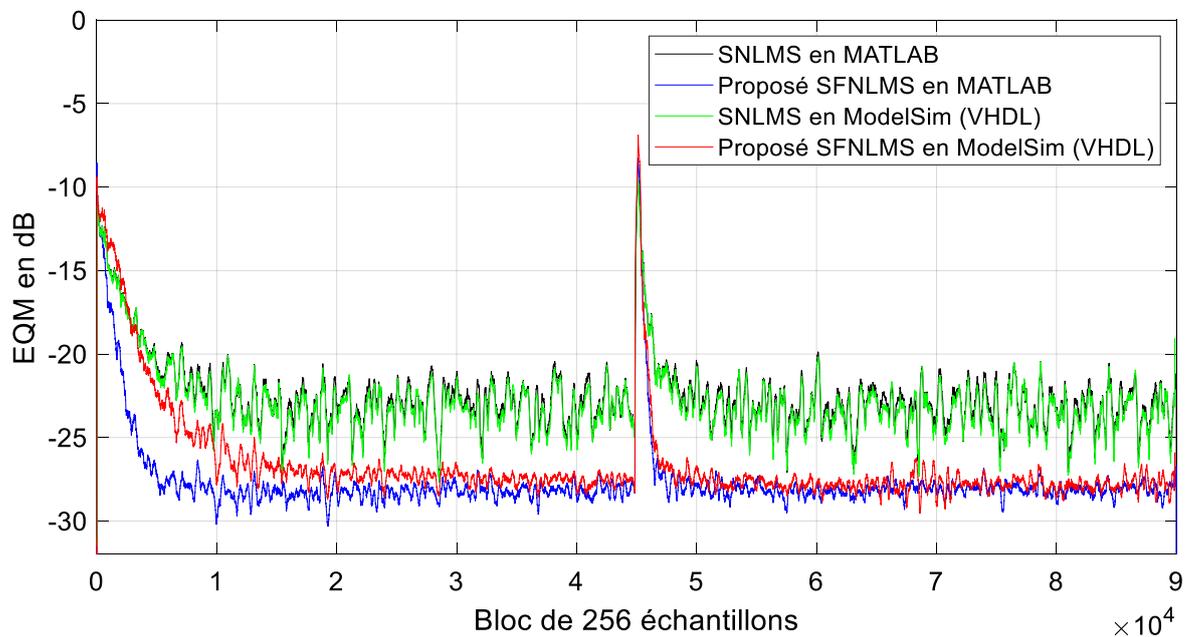


Figure 6.13. Évaluation de EQM obtenu par les deux algorithmes SNLMS et SFNLMS sur MATLAB et ModelSim avec le bruit USASI en entrée.

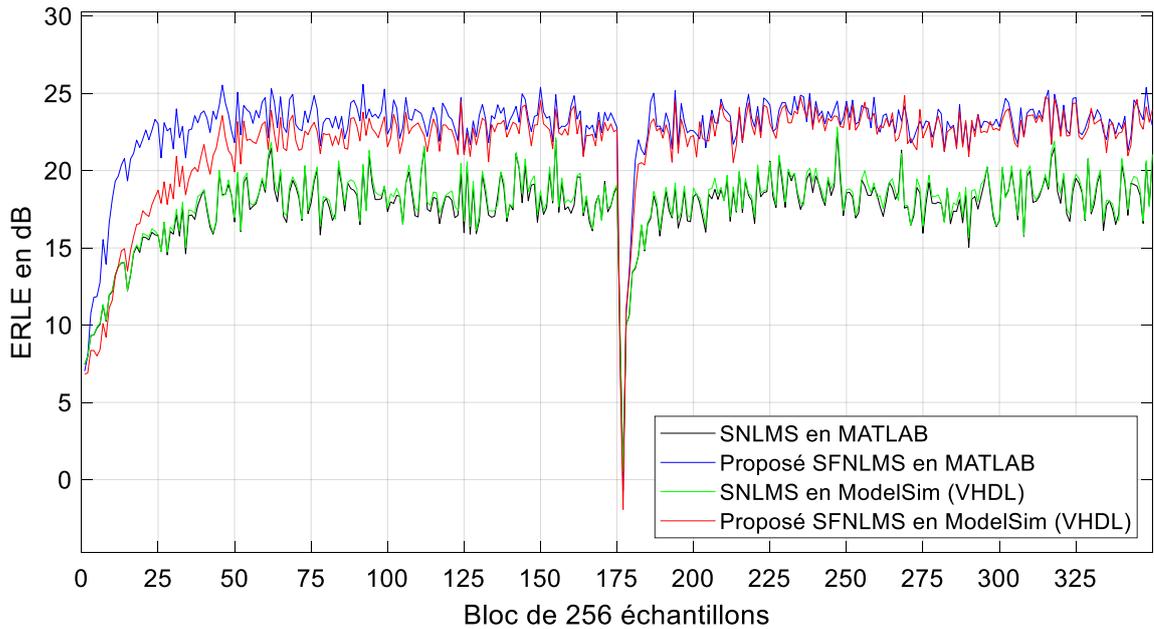


Figure 6.14. Évaluation de l'ERLE obtenu par les deux algorithmes SNLMS et SFNLMS sur MATLAB et ModelSim avec le bruit USASI en entrée.

Dans la dernière expérience, et après avoir testé la stabilité et la vitesse de convergence du nouvel algorithme SFNLMS, nous le testons maintenant avec un signal d'entrée corrélé (signal de parole) sous MATLAB et ModelSim. Les résultats obtenus par ModelSim sont illustrés à la figure 6.15 pour SNLMS et à la Figure 6.16 pour SFNLMS. Ainsi, pour les résultats de comparaison du nouvel SFNLMS et SNLMS en termes de MSE et ERLE, les résultats obtenus sous Matlab et ModelSim sont donnés respectivement à la Figure 6.17 et la Figure 6.18. Ces résultats montrent que le comportement des deux algorithmes sous MATLAB et ModelSim est très proches en utilisant les mêmes paramètres. De plus, l'algorithme SFNLMS proposé est stable et présente une vitesse de convergence rapide dans la phase transitoire par rapport à l'algorithme classique SNLMS sous les deux environnements MATLAB et ModelSim.

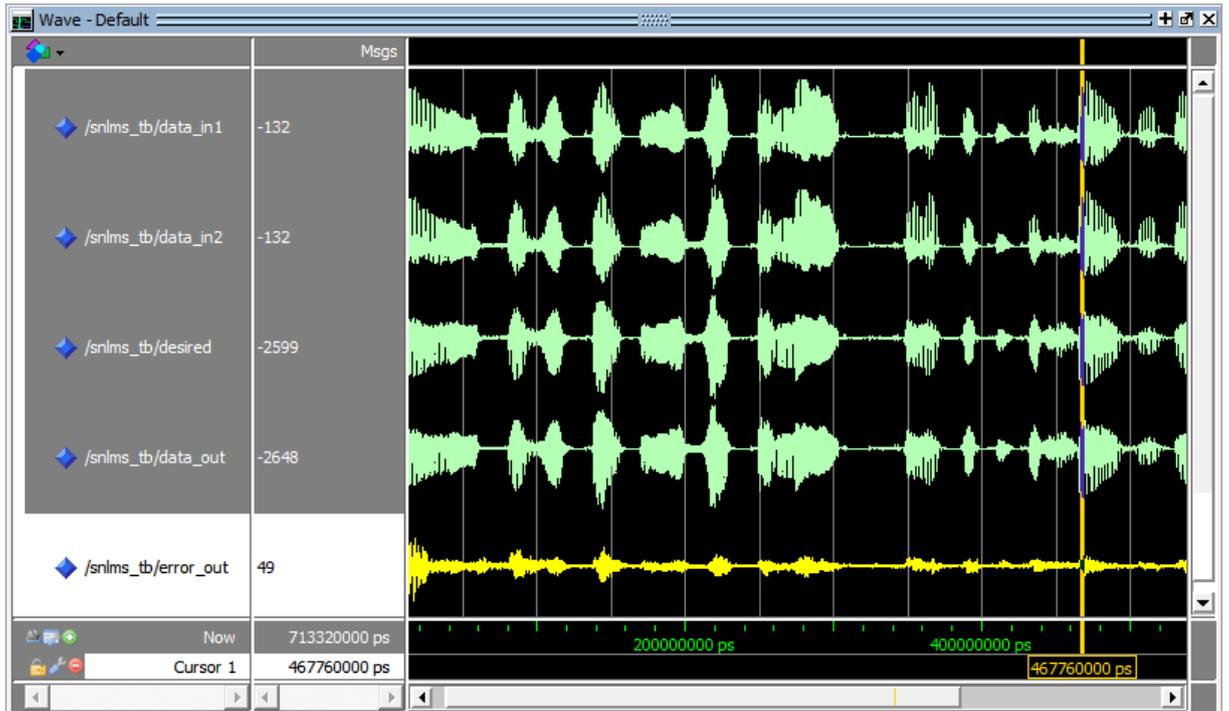


Figure 6.15. La simulation de l'algorithme SNLMS pour AEAS en ModelSim avec le signal parole en entrée.

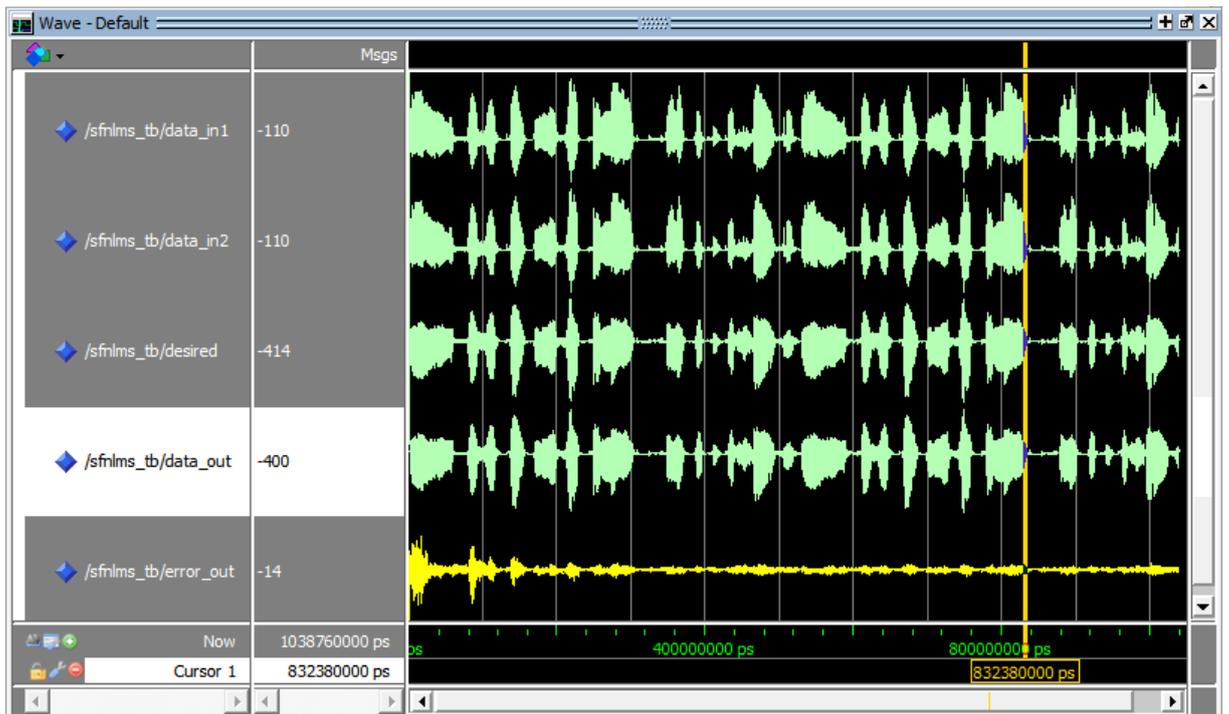


Figure 6.16. La simulation de l'algorithme SFNLMS pour AEAS en ModelSim avec le signal parole en entrée.

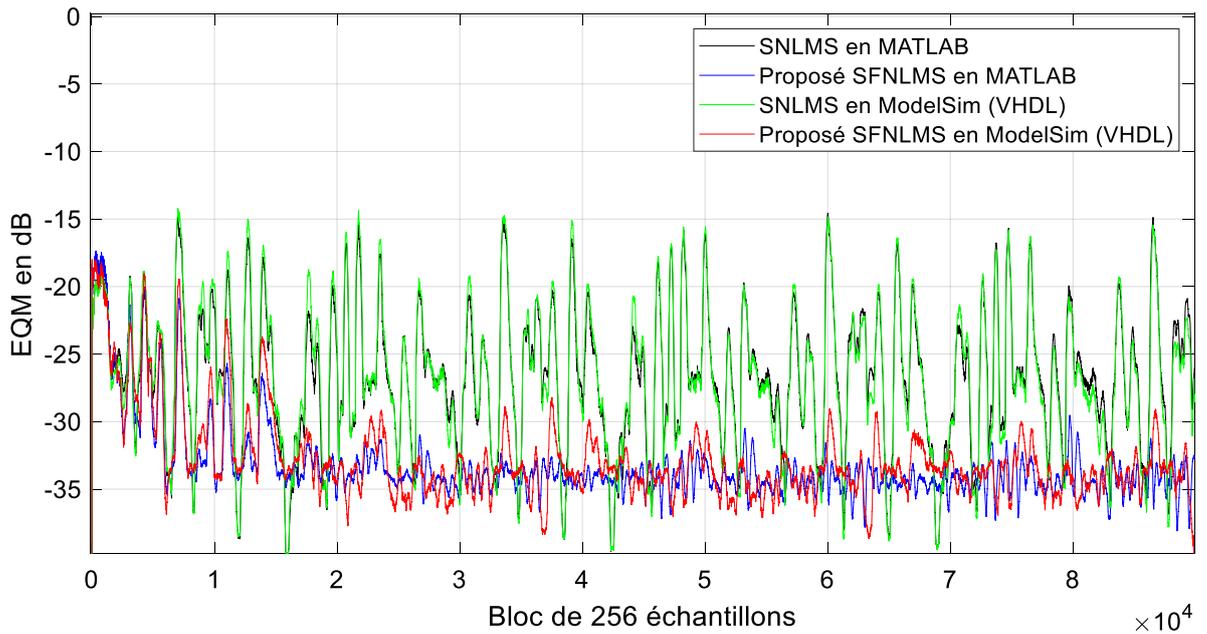


Figure 6.17. Évaluation de l'EQM obtenu par les deux algorithmes SNLMS et SFNLMS sur MATLAB et ModelSim avec le signal parole en entrée.

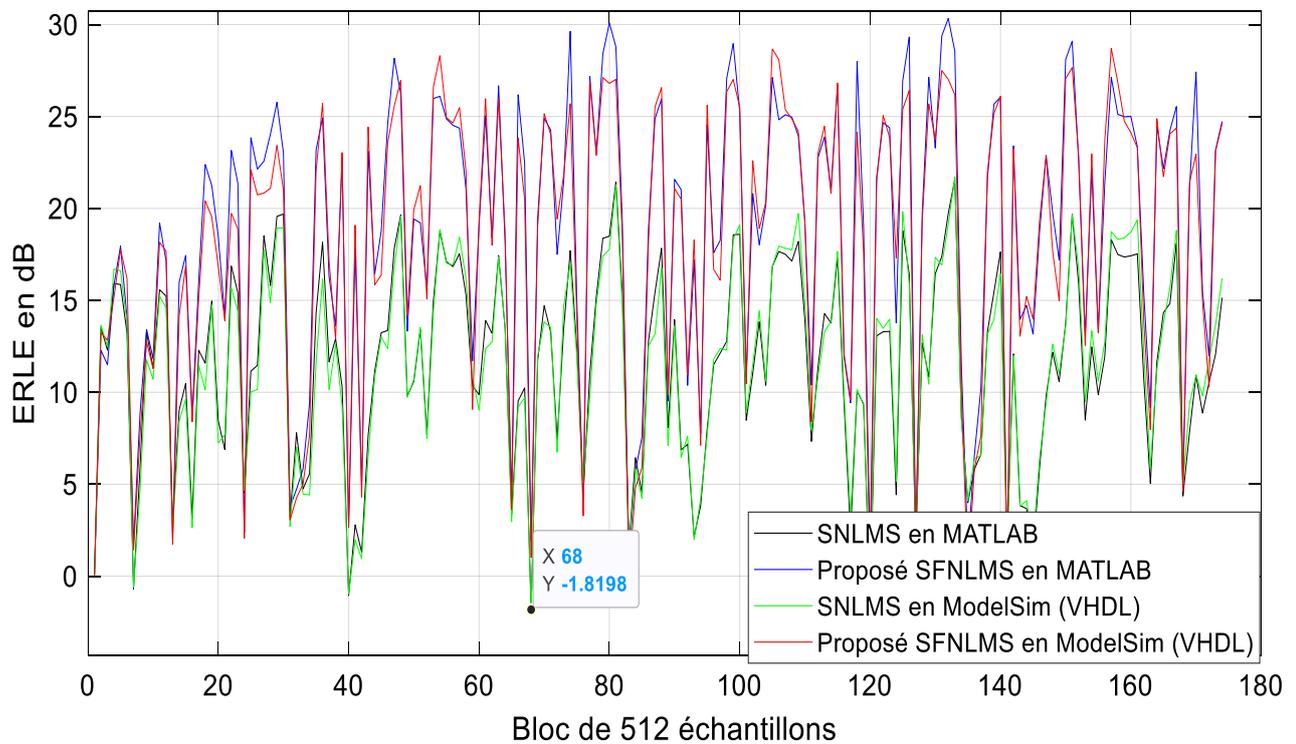


Figure 6.18. Évaluation de l'ERLE obtenu par les deux algorithmes SNLMS et SFNLMS sur MATLAB et ModelSim avec le signal parole en entrée.

6.7. Conclusion

Dans ce chapitre, nous avons proposé une implémentation efficace et simple de la technique d'AEAS par les algorithmes adaptatifs SNLMS classique et SFNLMS proposé. La conception proposée a été testée et confirmée en utilisant des outils offerts par MATLAB. En plus, le logiciel ModelSim a été utilisé pour vérifier la bonne conception de l'ensemble des programmes VHDL développés. Les résultats obtenus par les deux logiciels sont très proches et qui nous révèlent l'efficacité de notre conception.

CONCLUSION GENERALE

L'étude que nous avons présentée dans cette thèse de doctorat LMD consiste en l'annulation stéréophonique d'écho acoustique dans le domaine temporel en utilisant des nouvelles techniques, récemment développées, de traitement du signal afin d'améliorer la qualité de l'annulation stéréophonique de l'écho acoustiques. Nous nous sommes intéressés aussi à l'implémentation des algorithmes de l'annulation stéréophonique d'écho acoustique sur la carte FPGA.

Dans ce contexte, nous avons présenté en premier lieu une vision générale relative à l'annulation d'écho acoustique par le filtrage adaptatif. Comme nous avons vu le filtrage adaptatif ses approches et structures d'adaptation et ses algorithmes du gradients stochastiques (LMS et sa version normalisée NLMS) ainsi qu'algorithme des moindres carrés récursifs RLS.

Dans la deuxième partie du travail présenté dans cette thèse, nous avons introduit une revue générale sur l'annulation stéréophonique d'écho acoustique qui inclut, la configuration générale, les défis rencontrés lorsqu'on passe d'un système d'annulation monophonique à un système d'annulation stéréophonique tel que le problème de non-unicité des réponses impulsionnelles estimées dues au couplage acoustique entre les deux canaux stéréo, par la suite, un ensemble des techniques de décorrélation ont été traités et analysés dans le but de surmonter cette difficulté.

Pour améliorer la performance d'annulation d'écho acoustique stéréophonique, nous avons proposé, dans un premier temps, un nouvel algorithme stéréophonique dénoté SFNLMS. Ce nouvel algorithme stéréophonique SFNLMS a été proposé pour augmenter la vitesse de convergence par rapport à celle obtenue par l'algorithme stéréophonique SNLMS classique. Les résultats de simulations ont également prouvé que le nouvel algorithme SFNLMS est supérieur à l'algorithme SNLMS en termes de critères de performance à savoir l'EQM et ERLE. En utilisant les mêmes paramètres de réglages pour les deux algorithmes, nous avons noté que la convergence de l'algorithme SFNLMS est plus rapide par rapport à l'algorithme SNLMS.

Nous avons également proposé dans le chapitre 4, quatre nouveaux algorithmes d'annulation d'écho acoustique stéréophonique en se basant sur la sélection des données. Ces algorithmes sont ES-SM-NLMS, SM-PU-SNLMS, SNP-VSS-N et SNP-VSS-PU-S.

Dans les deux algorithmes ES-SM-NLMS et SM-PU-SNLMS, nous avons proposé à utiliser le filtrage par appartenance à un ensemble SMF pour réduire la complexité de calcul et augmenter la vitesse de convergence. Dans le nouvel algorithme dénoté SM-PU-SNLMS, nous avons proposé d'utiliser la combinaison de deux techniques, la technique SMF et la technique de sélection de prises avec maximum exclusif non linéaire XMNL en faisant référence à la technique du PUF. Cette dernière permet une amélioration remarquable sur la réduction de la cohérence entre les signaux d'entrés et la réduction de complexité de calcul.

Dans les deux algorithmes SNP-VSS-N et SNP-VSS-PU-S, nous avons proposé à utiliser le filtrage par un pas variable non paramétrique NP-VSS pour réduire la complexité de calcul et augmenter la vitesse de convergence. Dans le nouvel algorithme dénoté SNP-VSS-PU-S, nous avons proposé d'utiliser la combinaison de deux techniques, la technique NP-VSS et la technique de sélection de prises avec maximum exclusif non linéaire XMNL faisant référence à la technique du PUF.

Les résultats de simulations ont bien prouvé la supériorité des algorithmes proposés en matière d'EQM et le Mis-alignement par rapport à l'algorithme SNLMS classique pour l'annulation d'écho acoustique stéréophonique.

Dans les derniers travaux présentés dans cette thèse, nous avons implémenté l'algorithme SFNLMS proposé et discuté dans le chapitre 3 en code VHDL pour la carte FPGA. La conception proposée a été testé et vérifiée en utilisant le logiciel ModelSim et des outils offerts par MATLAB. Les résultats obtenus par les deux logiciels sont comparables et qui nous révèlent l'efficacité de notre conception.

Comme perspectives de notre travail, nous proposons d'implémenter la description VHDL proposé sur un banc d'essai d'un système d'annulation écho acoustique stéréophonique.

RÉFÉRENCES

- [1] S. R. D. Poulou, “Adaptive filtering, algorithms and practical implementations.” Springer Publisher, New York, 2008.
- [2] M. M. Sondhi, D. R. Morgan, and J. L. Hall, “Acoustic echo cancellation for stereophonic teleconferencing,” *J Acoust Soc Am*, vol. 94, no. 3_Supplement, pp. 1826–1826, Sep. 1993, doi: 10.1121/1.407801.
- [3] F. Amand, A. Gilloire, and J. Benesty, “Identifying the true echo path impulse responses in stereophonic acoustic echo cancellation,” in *1996 8th European Signal Processing Conference (EUSIPCO 1996)*, 1996, pp. 1–4.
- [4] J. Benesty, D. R. Morgan, and M. M. Sondhi, “A better understanding and an improved solution to the specific problems of stereophonic acoustic echo cancellation,” *IEEE Transactions on Speech and Audio Processing*, vol. 6, no. 2, pp. 156–165, Mar. 1998, doi: 10.1109/89.661474.
- [5] M. M. Sondhi, D. R. Morgan, and J. L. Hall, “Stereophonic acoustic echo cancellation-an overview of the fundamental problem,” *IEEE Signal Process Lett*, vol. 2, no. 8, pp. 148–151, Aug. 1995, doi: 10.1109/97.404129.
- [6] A. Hirano and A. Sugiyama, “A compact multi-channel echo canceller with a single adaptive filter per channel,” in *[Proceedings] 1992 IEEE International Symposium on Circuits and Systems*, IEEE, pp. 1922–1925. doi: 10.1109/ISCAS.1992.230434.
- [7] A. Hirano, A. Sugiyama, Y. Arasawa, and N. Kawayachi, “DSP implementation and performance evaluation of a compact stereo echo canceller,” in *Proceedings of ICASSP '94. IEEE International Conference on Acoustics, Speech and Signal Processing*, 1994, p. II/245-II/248 vol.2. doi: 10.1109/ICASSP.1994.389674.
- [8] S. Minami, “A stereophonic echo canceler using single adaptive filter,” in *1995 International Conference on Acoustics, Speech, and Signal Processing*, 1995, pp. 3027–3030 vol.5. doi: 10.1109/ICASSP.1995.479483.
- [9] B. C. Bispo and D. da S. Freitas, “Hybrid pre-processor based on frequency shifting for stereophonic acoustic echo cancellation,” in *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, 2012, pp. 2447–2451.
- [10] M. Djendi and A. Guessoum, “A new fast Newton-type adaptive filtering algorithm for stereophonic acoustic echo cancellation (SAEC),” *Int J Adapt Control Signal Process*, vol. 24, no. 6, pp. 435–444, Jun. 2010, doi: 10.1002/acs.1134.
- [11] M. Djendi, “An efficient stabilized fast Newton adaptive filtering algorithm for stereophonic acoustic echo cancellation SAEC,” *Computers & Electrical Engineering*, vol. 38, no. 4, pp. 938–952, Jul. 2012, doi: 10.1016/j.compeleceng.2012.02.010.

- [12] J. Benesty, C. Paleologu, T. Gänslér, and S. Ciochină, *A Perspective on Stereophonic Acoustic Echo Cancellation*, vol. 4. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. doi: 10.1007/978-3-642-22574-1.
- [13] C. Stanciu, C. Paleologu, J. Benesty, S. Ciochina, and F. Albu, “Variable-forgetting factor RLS for stereophonic acoustic echo cancellation with widely linear model,” in *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, 2012, pp. 1960–1964.
- [14] C. Paleologu, J. Benesty, and S. Ciochină, “Widely linear general Kalman filter for stereophonic acoustic echo cancellation,” *Signal Processing*, vol. 94, pp. 570–575, Jan. 2014, doi: 10.1016/j.sigpro.2013.08.001.
- [15] N. K. Desiraju, S. Doclo, T. Gerkmann, and T. Wolff, “Efficient Multi-Channel Acoustic Echo Cancellation Using Constrained Sparse Filter Updates in the Subband Domain,” in *Speech Communication; 11. ITG Symposium*, 2014, pp. 1–4.
- [16] M. Schneider and W. Kellermann, “Multichannel Acoustic Echo Cancellation in the Wave Domain With Increased Robustness to Nonuniqueness,” *IEEE/ACM Trans Audio Speech Lang Process*, vol. 24, no. 3, pp. 518–529, Mar. 2016, doi: 10.1109/TASLP.2016.2515502.
- [17] L. Romoli, S. Cecchi, and F. Piazza, “Multichannel acoustic echo cancellation exploiting effective fundamental frequency estimation,” *Speech Commun*, vol. 86, pp. 97–106, Feb. 2017, doi: 10.1016/j.specom.2016.11.009.
- [18] F. South, T. Usf, D. Graduate, G. Usf, D. Theses, and S. Raghavendran, “Implementation of an Acoustic Echo Canceller Using Matlab,” 2003. [Online]. Available: <https://digitalcommons.usf.edu/etd>
- [19] L. Lu, “Implementation of Acoustic Echo Cancellation for PC Applications using Matlab.” 2007. [Online]. Available: <http://web.it.kth.se/~axel/papers/2007/MSc-lu-lu.pdf>
- [20] S. Gupta, “Acoustic Echo Cancellation using Conventional Adaptive Algorithms and Modified Variable Step Size LMS Algorithm,” 2007. [Online]. Available: <https://api.semanticscholar.org/CorpusID:64313243>
- [21] B. Widrow and S. ~D. Stearns, *Adaptive signal processing*. 1985.
- [22] R. C. Bilcu, P. Kuosmanen, and K. Egiazarian, “A new variable length LMS algorithm: theoretical analysis and implementations,” in *9th International Conference on Electronics, Circuits and Systems*, 2002, pp. 1031–1034 vol.3. doi: 10.1109/ICECS.2002.1046426.
- [23] P. S. R. Diniz, *Adaptive Filtering*. Boston, MA: Springer US, 1997. doi: 10.1007/978-1-4419-8660-3.
- [24] L. Brnak, M. Svarc, V. Platenka, and M. Richterova, “Blind Equalization of Carrier in Carrier Signals by Using Adaptive Filters,” in *2018 New Trends in Signal Processing (NTSP)*, IEEE, Oct. 2018, pp. 1–6. doi: 10.23919/NTSP.2018.8524037.
- [25] W. Ma, D. Zheng, Z. Zhang, J. Duan, and B. Chen, “Robust proportionate adaptive filter based on maximum correntropy criterion for sparse system identification in impulsive noise environments,”

- Signal Image Video Process*, vol. 12, no. 1, pp. 117–124, Jan. 2018, doi: 10.1007/s11760-017-1137-0.
- [26] M. Djendi and R. Bendoumia, “A new adaptive filtering subband algorithm for two-channel acoustic noise reduction and speech enhancement,” *Computers & Electrical Engineering*, vol. 39, no. 8, pp. 2531–2550, Nov. 2013, doi: 10.1016/j.compeleceng.2013.09.009.
- [27] N. Kumar, B. Singh, and B. K. Panigrahi, “Integration of Solar PV Generation with Distribution Grid: Using Novel Adaptive Filter Based Control Technique,” in *2018 4th International Conference on Universal Village (UV)*, IEEE, Oct. 2018, pp. 1–6. doi: 10.1109/UV.2018.8642120.
- [28] B. Farhang-Boroujeny, *Adaptive filters: theory and applications*. John wiley & sons, 2013.
- [29] N. Wiener, *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*. The MIT Press, 1949. doi: 10.7551/mitpress/2946.001.0001.
- [30] J. F. Bercher and P. Jardin, “Introduction au filtrage adaptatif,” *ESIEE Paris*, vol. 2, pp. 2008–2009, 2003.
- [31] D. T. M. Slock, “On the convergence behavior of the LMS and the normalized LMS algorithms,” *IEEE Transactions on Signal Processing*, vol. 41, no. 9, pp. 2811–2825, 1993, doi: 10.1109/78.236504.
- [32] L. Ljung and T. Söderström, *Theory and practice of recursive identification*. MIT press, 1983.
- [33] M. Messini and M. Djendi, “A new adaptive filtering algorithm for stereophonic acoustic echo cancellation,” *Applied Acoustics*, vol. 146, pp. 345–354, Mar. 2019, doi: 10.1016/j.apacoust.2018.11.033.
- [34] P. Enderoth, *Stereophonic Acoustic Echo Cancellation: Theory and Implementation*. Univ.-bibl., 2001.
- [35] J. Benesty, P. Duhamel, and Y. Grenier, “Multi-channel adaptive filtering applied to multi-channel acoustic echo cancellation,” in *EUSIPCO*, 1996, pp. 1–4.
- [36] J. Benesty, J. Chen, and Y. Huang, “Linear prediction,” *Springer Handbook of Speech Processing*, pp. 121–134, 2008.
- [37] A. W. H. Khong and P. A. Naylor, “Stereophonic acoustic echo cancellation employing selective-tap adaptive algorithms,” *IEEE Trans Audio Speech Lang Process*, vol. 14, no. 3, pp. 785–796, May 2006, doi: 10.1109/TSA.2005.858065.
- [38] A. Benallal and M. Arezki, “A fast convergence normalized least-mean-square type algorithm for adaptive filtering,” *Int J Adapt Control Signal Process*, vol. 28, no. 10, pp. 1073–1080, Oct. 2014, doi: 10.1002/acs.2423.
- [39] M. Djendi, M. Bouchard, A. Guessoum, A. Benallal, and D. Berkani, “Improvement of the convergence speed and the tracking ability of the fast Newton type adaptive filtering (FNMF) algorithm,” *Signal Processing*, vol. 86, no. 7, pp. 1704–1719, Jul. 2006, doi: 10.1016/j.sigpro.2005.09.012.

- [40] J. Cioffi and T. Kailath, "Fast, recursive-least-squares transversal filters for adaptive filtering," *IEEE Trans Acoust*, vol. 32, no. 2, pp. 304–337, Apr. 1984, doi: 10.1109/TASSP.1984.1164334.
- [41] A. Benallal and A. Benkrid, "A simplified FTF-type algorithm for adaptive filtering," *Signal Processing*, vol. 87, no. 5, pp. 904–917, May 2007, doi: 10.1016/j.sigpro.2006.08.013.
- [42] M. Rupp, "A family of adaptive filter algorithms with decorrelating properties," *IEEE Transactions on Signal Processing*, vol. 46, no. 3, pp. 771–775, Mar. 1998, doi: 10.1109/78.661344.
- [43] Y.-R. Chien and S.-I. Chu, "A Fast Converging Partial Update LMS Algorithm with Random Combining Strategy," *Circuits Syst Signal Process*, vol. 33, no. 6, pp. 1883–1898, Jun. 2014, doi: 10.1007/s00034-013-9724-y.
- [44] Md. Z. A. Bhotto and A. Antoniou, "A new partial-update NLMS adaptive-filtering algorithm," in *2014 IEEE 27th Canadian Conference on Electrical and Computer Engineering (CCECE)*, IEEE, May 2014, pp. 1–5. doi: 10.1109/CCECE.2014.6901048.
- [45] T. Aboulnasr and K. Mayyas, "Complexity reduction of the NLMS algorithm via selective coefficient update," *IEEE Transactions on Signal Processing*, vol. 47, no. 5, pp. 1421–1424, May 1999, doi: 10.1109/78.757235.
- [46] 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 Process Lett*, vol. 5, no. 5, pp. 111–114, May 1998, doi: 10.1109/97.668945.
- [47] J. Benesty, H. Rey, L. R. Vega, and S. Tressens, "A Nonparametric VSS NLMS Algorithm," *IEEE Signal Process Lett*, vol. 13, no. 10, pp. 581–584, Oct. 2006, doi: 10.1109/LSP.2006.876323.
- [48] M. Messini and M. Djendi, "A new data-selective adaptive algorithm for stereophonic acoustic echo cancellation,"
- [49] M. Messini and M. Djendi, "a new stereophonic echo suppression base on the variable step size approach," in *International Conference on Electrical Engenieering Biskra. ICEEB*, Biskra, 2018.
- [50] M. Messini and M. Djendi, "Non parametric variable step size partial update normalized least mean square algorithm for stereophonic acoustic echo cancellation," in *4th International Conference on Embedded Systems in telecommunication and instrumentation* , Annaba, Algeria, 2019.
- [51] S. Werner and P. S. R. Diniz, "Set-membership affine projection algorithm," *IEEE Signal Process Lett*, vol. 8, no. 8, pp. 231–235, Aug. 2001, doi: 10.1109/97.935739.
- [52] I. Homana, I. Muresan, M. Topa, and C. Contan, "FPGA implementation of LMS and NLMS adaptive filters for acoustic echo cancellation," *Acta Technica Napocensis*, vol. 52, no. 4, p. 13, 2011.
- [53] O. S. Tehrani, M. Ashourian, and P. Moallem, "An FPGA-based implementation of fixed-point standard-LMS algorithm with low resource utilization and fast convergence," *International Review on Computers and Software*, vol. 5, no. 4, pp. 436–444, 2010.

- [54] K. Rekha, B. Nagabushan, and K. Nataraj, "FPGA Implementation of NLMS Algorithm for Receiver in wireless communication system," *IJCSNS International Journal of Computer Science and Network Security*, vol. 10, no. 12, 2010.
- [55] C. Choo and S. Mutsuddy, "An embedded adaptive filtering system on fpga," *Department of Electrical Engineering, San Jose State University, San Jose, CA*, pp. 95184–95198, 2006.
- [56] Z. J. Mou and P. Duhamel, "Fast FIR filtering: Algorithms and implementations," *Signal Processing*, vol. 13, no. 4, pp. 377–384, Dec. 1987, doi: 10.1016/0165-1684(87)90019-3.
- [57] M. Djendi, M. Messini, and A. Guessoum, "A VHDL implementation of NLMS and RLS adaptive algorithms for acoustic echo cancellation," in *2017 5th International Conference on Electrical Engineering - Boumerdes (ICEE-B)*, IEEE, Oct. 2017, pp. 1–6. doi: 10.1109/ICEE-B.2017.8192091.
- [58] M. Borhani and V. Sedghi, "An acoustic echo canceller chip," in *Fifth International Workshop on System-on-Chip for Real-Time Applications (IWSOC'05)*, IEEE, 2005, pp. 193–198. doi: 10.1109/IWSOC.2005.36.
- [59] S. S. Khanande, "Survey Paper on Recursive Least Square Adaptive Filters Using Block DCD Approach," *International Journal on Recent and Innovation Trends in Computing and Communication*, 2015. [Online]. Available: <https://api.semanticscholar.org/CorpusID:212605206>
- [60] S. Khanande and S. J. Honade, "Design and Implementation of Recursive Least Square Adaptive Filter Using Block DCD approach," *International Journal of Reconfigurable and Embedded Systems (IJRES)*, vol. 4, no. 3, p. 209, Nov. 2015, doi: 10.11591/ijres.v4.i3.pp209-212.
- [61] C. Stanciu, C. Anghel, and L. Stanciu, "Efficient FPGA implementation of the DCD-RLS algorithm for stereo acoustic echo cancellation," in *2015 International Symposium on Signals, Circuits and Systems (ISSCS)*, IEEE, Jul. 2015, pp. 1–4. doi: 10.1109/ISSCS.2015.7204008.
- [62] C. Anghel and S. Ciochina, "On the FPGA Implementation of the VR-RLS Algorithms," *ICN 2017*, p. 109, 2017.
- [63] C. Elisei-Iliescu, C. Stanciu, C. Paleologu, J. Benesty, C. Anghel, and S. Ciochina, "Robust variable-regularized RLS algorithms," in *2017 Hands-free Speech Communications and Microphone Arrays (HSCMA)*, IEEE, 2017, pp. 171–175. doi: 10.1109/HSCMA.2017.7895584.
- [64] M. S. Nazir, A. Akram, B. C. Maheshwari, M. U. Hadi, M. Aqil, and H. Khan, "Real-time brain activation detection by FPGA implemented adaptive algorithm," in *2017 International Conference on Innovations in Electrical Engineering and Computational Technologies (ICIEECT)*, IEEE, Apr. 2017, pp. 1–3. doi: 10.1109/ICIEECT.2017.7916522.
- [65] C. Stanciu, C. Anghel, M. Udrea, and L. Stanciu, "Variable-regularized low complexity RLS algorithm for stereophonic acoustic echo cancellation," in *2017 International Symposium on Signals, Circuits and Systems (ISSCS)*, IEEE, Jul. 2017, pp. 1–4. doi: 10.1109/ISSCS.2017.8034933.

- [66] S. Shaikh and S. Pujari, "Migration from microcontroller to FPGA based SoPC design: Case study: LMS adaptive filter design on Xilinx Zynq FPGA with embedded ARM controller," in *2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT)*, IEEE, Sep. 2016, pp. 129–134. doi: 10.1109/ICACDOT.2016.7877565.
- [67] M. Messini and M. Djendi, "VHDL implementation of FNLMS algorithm".
- [68] E. E. Swartzlander and C. E. Lemonds, "Computer Arithmetic: volume III (vol. 3)." World Scientific, 2015.
- [69] B. Widrow, M. E. Hoff, and others, "Adaptive switching circuits," in *IRE WESCON convention record*, 1960, pp. 96–104.