

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي والبحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البليدة
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique



Mémoire de Master

En Télécommunication
Spécialité : Réseaux & Télécommunications

Présenté par

Bensidi Aissa Ouassim Ahmed

&

Moussaoui Nazim

Un Algorithme LMF Automatique Pour La Réduction Du Bruit Acoustique

Proposé par : Pr. DJENDI Mohamed

Année Universitaire 2022-2023

Remerciements

Nous sommes profondément reconnaissants envers plusieurs personnes qui ont rendu possible la réalisation de ce mémoire. En premier lieu, nous tenons à exprimer notre gratitude envers Monsieur DJENDI Mohamed, notre directeur de mémoire, qui a été le premier à nous introduire au sujet qui a guidé notre travail. Sa patience, sa disponibilité et ses conseils avisés ont grandement enrichi notre réflexion.

Nous voudrions aussi exprimer notre reconnaissance envers les amis et collègues qui ont apporté leur soutien moral et intellectuel tout au long de notre démarche

Dédicaces

Je souhaite dédier ce travail à mes parents bien-aimés. Aucune dédicace ne peut véritablement exprimer le respect et l'amour que j'ai pour les efforts qu'ils ont déployés pour mon éducation et mon bien-être. J'espère pouvoir leur rendre un jour une partie de ce qu'ils ont fait pour moi.

À tous mes amis et à mon binôme, je tiens à exprimer ma gratitude pour leur volonté et leur soutien tout au long de ce mémoire.

Un grand merci à vous tous.

Ouassim

Dédicaces

Ce travail est dédié à mes chers parents, dont aucun hommage ne peut véritablement exprimer le respect et l'amour que j'ai pour les efforts qu'ils ont déployés pour mon éducation et mon bien-être. J'espère pouvoir leur rendre un peu de ce qu'ils ont fait pour moi un jour.

Je tiens également à remercier ma chère petite sœur qui a toujours été présente lors des moments difficiles.

Je souhaite exprimer ma gratitude à tous mes amis, ainsi qu'à mon binôme, pour leur volonté et leur soutien tout au long de ce mémoire.

Je vous suis reconnaissant à tous.

Nazim

ملخص: الهدف من هذا المشروع التخرج هو تصميم خوارزمية جديدة للتعديل التلقائي للمستوى السمعي الذي تصل إليه الكلمة بهدف تحسين جودتها المدركة عن طريق تقليل الضوضاء الصوتية.

يتم توضيح كفاءة هذه الخوارزمية من خلال نتائج المحاكاة التي تم الحصول عليها من خلال مقارنة الخوارزمية (NLMS) المقترحة بخوارزمية التدرج العشوائي المعياري.

كلمات المفاتيح: الخوارزمية الرابعة الأقل المعدلة؛ تحسين الكلام.

Résumé : L'objectif de ce projet de fin d'études est de concevoir un nouvel algorithme NLMF automatique visant à améliorer la qualité perçue de la parole en réduisant le bruit acoustique.

L'efficacité de cet algorithme est démontrée par les résultats de simulation obtenus en comparant l'algorithme proposé à l'algorithme de gradient stochastique normalisé (NLMS).

Mots clés : NLMF ; NLMS ; Algorithme adaptatif.

Abstract : The objective of this end-of-studies project is to design a new automatic NLMF algorithm aimed at improving the perceived quality of speech by reducing acoustic noise.

The effectiveness of this algorithm is demonstrated by the simulation results obtained by comparing the proposed algorithm to the normalized stochastic gradient algorithm (NLMS).

Keywords : NLMF ; NLMS ; adaptive algorithm.

Listes des acronymes et abréviations

- LMF : Least Mean Fourth
- NLMF : Normalized Least Mean Fourth
- DAV : Détection d'Activité Vocale.
- EQM : Erreur Quadratique Moyenne.
- LMS : Least Mean Squares.
- NLMS : Normalized Least Mean Squares.
- RSB : Rapport Signal à Bruit
- RLS : Recursive least squares
- SM : System Mismatch
- *dB* : décibel.

Table des matières

Introduction générale	1
Chapitre 1 Technique de réduction de bruit	2
1.1 Introduction	2
1.2 Méthode de réduction de bruit	2
1.2.1 Méthode mono-capteur	2
1.2.2 Méthode bi-capteurs :	3
1.2.3 Méthode bi-capteurs simplifiée :	4
1.2.4 Méthode multi-capteurs	4
1.2.5 La réduction de bruit par filtrage	5
1.2.6 Réduction de bruit adaptative	5
1.3 Conclusion.....	6
Chapitre 2 Filtres et filtrage adaptatifs.....	7
2.1 Introduction	7
2.2 Filtrage de Wiener	7
2.2.1 L'erreur quadratique moyenne	8
2.2.2 Principe d'orthogonalité	8
2.3 Filtrage adaptatif.....	9
2.3.1 Définition.....	9
2.3.2 Principe du filtrage adaptatif.....	10
2.3.3 Application du Filtrage adaptatif.....	10
2.4 Algorithmes adaptatifs	10
2.4.2 Critères de comparaison et choix des algorithmes adaptatifs	10
2.4.3 Algorithme du gradient stochastique LMS.....	11
2.4.4 Algorithme du gradient stochastique normalisé NLMS	12
2.4.5 Algorithme à méthode récursive à moindre carré RLS.....	13
2.5 Conclusion.....	13
Chapitre 3 Proposition du nouvel algorithme NLMF automatique.....	14
3.1 Introduction	14
3.2 Principe de nouvel algorithme NLMF	14
3.3 Problème de stabilité d'algorithme NLMF	14
3.3.1 Notions et hypothèses	14
3.3.2 Facteur affectant la stabilité d'algorithme LMF	15
3.3.3 Algorithmes LMF normalisés	16
3.4 L'algorithme proposé	18
3.5 Conclusion.....	19

Chapitre 4 Résultats Des Simulations	20
4.1 Introduction	20
4.2 Signaux utilisés aux simulations	20
4.2.2 Bruit blanc	21
4.2.3 Réponses impulsionnelles	21
4.2.4 Signaux de mélange	22
4.3 Principe de détecteur d'activité locale (DAV)	22
4.4 Résultat de simulation	23
4.5 Evaluation du nouvel algorithme proposé	23
4.5.1 Critère du MSE segmentée (SegMSE)	23
4.5.2 Le critère de désajustement (System Mismatch)	23
4.5.3 L'effet du <i>RSB</i> avec une taille de filtre $L = 32$	24
4.5.4 L'effet du <i>RSB</i> avec une taille de filtre $L = 128$	28
4.6 Résultat de simulation en changeant un paramètre primordial	32
4.7 Conclusion	40
Conclusion générale	41
Bibliographie	42

Liste des Figures

Figure 1.1	Principe de débruitage mono-capteur.	2
Figure 1.2	La structure complète d'un mélange convolutif bi-capteurs	3
Figure 1.3	La structure simplifiée d'un mélange convolutif bi capteurs	4
Figure 1.4	La structure d'un mélange convolutif multi capteurs (cas général)	5
Figure 2.1	Représentation schématique d'un filtre Wiener.	7
Figure 3.1	Identification du plan adaptatif	14
Figure 4.1	Signal de parole utilisé	20
Figure 4.2	Signal du bruit blanc utilisé	21
Figure 4.3	La réponse impulsionnelle h_{12} (à gauche) et la réponse impulsionnelle h_{21} (à droite) avec $L = 64$.	21
Figure 4.4	Le signal du premier mélange $m_1(n)$ (en haut) et le signal du deuxième mélange $m_2(n)$ (en bas).	22
Figure 4.5	Signal de la parole avec sa DAV.	22
Figure 4.6	Evaluation temporelle entre le NLMS et le NLMF	24
Figure 4.7	Comparaison entre RI réelle et RI du NLMS	24
Figure 4.8	Comparaison entre RI réelle et RI du NLMF	25
Figure 4.9	Critère SM	25
Figure 4.10	Critère EQM du NLMS ET NLMF	25
Figure 4.11	Evaluation temporelle entre le NLMS et le NLMF	26
Figure 4.12	Comparaison entre RI réelle et RI du NLMS	26

Figure 4.13	Comparaison entre RI réelle et RI du NLMF	26
Figure 4.14	Critère SM	26
Figure 4.15	Critère EQM du NLMS ET NLMF	26
Figure 4.16	Evaluation temporelle entre le NLMS et le NLMF	27
Figure 4.17	Comparaison entre RI réelle et RI du NLMS	27
Figure 4.18	Comparaison entre RI réelle et RI du NLMF	27
Figure 4.19	Critère SM	27
Figure 4.20	Critère EQM du NLMS ET NLMF	27
Figure 4.21	Evaluation temporelle entre le NLMS et le NLMF	28
Figure 4.22	Comparaison entre RI réelle et RI du NLMS	28
Figure 4.23	Comparaison entre RI réelle et RI du NLMF	29
Figure 4.24	Critère SM	29
Figure 4.25	Critère EQM du NLMS ET NLMF	29
Figure 4.26	Evaluation temporelle entre le NLMS et le NLMF	30
Figure 4.27	Comparaison entre RI réelle et RI du NLMS	30
Figure 4.28	Comparaison entre RI réelle et RI du NLMF	30
Figure 4.29	Critère SM	30
Figure 4.30	Critère EQM du NLMS ET NLMF	30

Figure 4.31	Evaluation temporelle entre le NLMS et le NLMF	31
Figure 4.32	Comparaison entre RI réelle et RI du NLMS	31
Figure 4.33	Comparaison entre RI réelle et RI du NLMF	31
Figure 4.34	Critère SM	31
Figure 4.35	Critère EQM du NLMS ET NLMF	31
Figure 4.36	Evaluation temporelle entre le NLMS et le NLMF	33
Figure 4.37	Comparaison entre RI réelle et RI du NLMS	33
Figure 4.38	Comparaison entre RI réelle et RI du NLMF	33
Figure 4.39	Critère SM	33
Figure 4.40	Critère EQM du NLMS ET NLMF	34
Figure 4.41	Evaluation temporelle entre le NLMS et le NLMF	34
Figure 4.42	Comparaison entre RI réelle et RI du NLMS	34
Figure 4.43	Comparaison entre RI réelle et RI du NLMF	35
Figure 4.44	Critère SM	35
Figure 4.45	Critère EQM du NLMS ET NLMF	35
Figure 4.46	Evaluation temporelle entre le NLMS et le NLMF	36
Figure 4.47	Comparaison entre RI réelle et RI du NLMS	36
Figure 4.48	Comparaison entre RI réelle et RI du NLMF	36
Figure 4.49	Critère SM	36
Figure 4.50	Critère EQM du NLMS ET NLMF	36
Figure 4.51	Evaluation temporelle entre le NLMS et le NLMF	37
Figure 4.52	Comparaison entre RI réelle et RI du NLMS	37
Figure 4.53	Comparaison entre RI réelle et RI du NLMF	37
Figure 4.54	Critère SM	37
Figure 4.55	Critère EQM du NLMS ET NLMF	37
Figure 4.56	Evaluation temporelle entre le NLMS et le NLMF	38
Figure 4.57	Comparaison entre RI réelle et RI du NLMS	38

Figure 4.58	Comparaison entre RI réelle et RI du NLMF	38
Figure 4.59	Critère SM	38
Figure 4.60	Critère EQM du NLMS ET NLMF	38
Figure 4.61	Evaluation temporelle entre le NLMS et le NLMF	39
Figure 4.62	Comparaison entre RI réelle et RI du NLMS	39
Figure 4.63	Comparaison entre RI réelle et RI du NLMF	39
Figure 4.64	Critère SM	39
Figure 4.65	Critère EQM du NLMS ET NLMF	40

Liste des tableaux

Figure 4.1	Paramètre de Simulation	23
Figure 4.2	Paramètre de Simulation	35

Introduction générale

Le bruit est omniprésent dans de nombreuses applications liées à l'acoustique et à la parole, telles que les communications mains libres, les systèmes de téléconférences et de télécollaboration, parmi d'autres. Ce bruit peut avoir un impact négatif à la fois sur la qualité perçue du signal de parole et sur son intelligibilité.

La gêne causée par cette perturbation, qualifiée de bruit, est généralement source d'inconfort et de fatigue pour les interlocuteurs. Il est donc essentiel de développer des systèmes de réduction du bruit qui visent à extraire l'information utile en traitant le signal bruité. De plus, les équipements d'acquisition et de traitement des signaux peuvent générer un bruit supplémentaire lors de l'amplification, ce qui rend le bruit omniprésent.

L'objectif principal de ce mémoire est de proposer une solution efficace pour la réduction du bruit acoustique. Nous présentons un nouvel algorithme automatique NLMF (Normalized Least Mean Fourth) basé sur un système bi-capteurs simplifié, ainsi qu'un détecteur automatique d'activité vocale utilisant le rapport signal sur bruit (SNR) et le coefficient de contrôle d'interférence comme paramètres clés de la détection.

Pour cela, nous présentons ci-après les chapitres de ce mémoire :

Le premier chapitre aborde les méthodes de mélange convolutif linéaire dans des configurations mono-capteur, bi-capteurs et multi-capteurs, ainsi que la réduction du bruit par filtrage.

Dans le deuxième chapitre, nous nous concentrons sur le filtrage adaptatif et présentons les algorithmes les plus connus de la famille des gradients stochastiques : LMS, NLMS et RLS.

Le troisième chapitre présente notre nouvel algorithme NLMF automatique, en détaillant son développement mathématique et son principe de fonctionnement.

Le quatrième et dernier chapitre est consacré aux résultats de simulation, permettant d'étudier les performances de notre nouvel algorithme proposé en utilisant des critères objectifs de qualité.

En conclusion, nous résumons l'ensemble de notre mémoire dans une conclusion générale.

Chapitre 1 Technique de réduction de bruit

1.1 Introduction

La réduction de bruit (ou noise réduction en anglais) est une technique qui consiste à réduire les bruits indésirables présents dans un signal audio, vidéo ou électrique, tout en préservant le signal utile. Cette technique est utilisée dans de nombreux domaines, tels que la communication, la musique, l'audiovisuel, la médecine, l'industrie.

Le but de la réduction de bruit est d'améliorer la qualité du signal en éliminant ou en atténuant les bruits parasites qui peuvent nuire à la compréhension, à la lisibilité ou à l'analyse du signal. Les bruits parasites peuvent provenir de diverses sources, telles que l'environnement acoustique, les interférences électromagnétiques, les imperfections du matériel ou du traitement.

1.2 Méthode de réduction de bruit

La réduction de bruit peut être réalisée à l'aide de diverses techniques, telles que :

- Méthode mono-capteur
- Méthode bi-capteurs
- Méthode bi-capteurs simplifiée
- Méthode multi-capteurs
- La réduction de bruit par filtrage
- Réduction de bruit adaptative

Chaque technique a ses avantages et ses limites, et doit être choisie en fonction du type de bruit, du contexte et de la qualité souhaitée.

1.2.1 Méthode mono-capteur

La méthode décrite utilise un algorithme adaptatif pour séparer la parole du bruit en se basant sur une seule observation. Le signal capté par le microphone est considéré comme la combinaison d'un signal utile et d'un bruit indésirable, que l'on cherche à éliminer. La figure ci-dessous illustre le schéma de la technique de réduction de bruit mono-capteur [1].

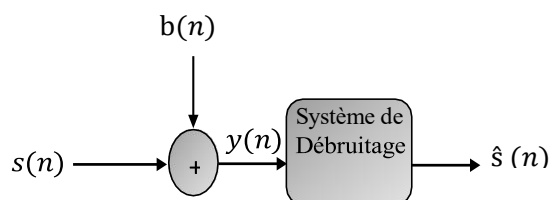


Fig. 1.1 - Principe de débruitage mono-capteur.

Où :

- $s(n)$ est le signal de la parole.
- $b(n)$ est le bruit.
- $y(n)$ est le signal de la parole bruité tel que : $y(n) = s(n) + b(n)$ (signal observé).
- $\hat{s}(n)$ est le signal de la parole après débruitage.

1.2.2 Méthode bi-capteurs :

Contrairement à la méthode précédente qui requiert un seul microphone, la méthode bi-capteurs fait appel à l'utilisation de deux microphones pour supprimer le bruit. Dans ce cas, un mélange convolutif se produit, qui est influencé par la propagation des signaux dans l'environnement, la disposition des microphones et des sources sonores, ainsi que les caractéristiques de la salle. Ce mélange convolutif linéaire entre le signal vocal et le bruit peut être représenté par un modèle complet, tel qu'illustré dans la figure suivante [3] :

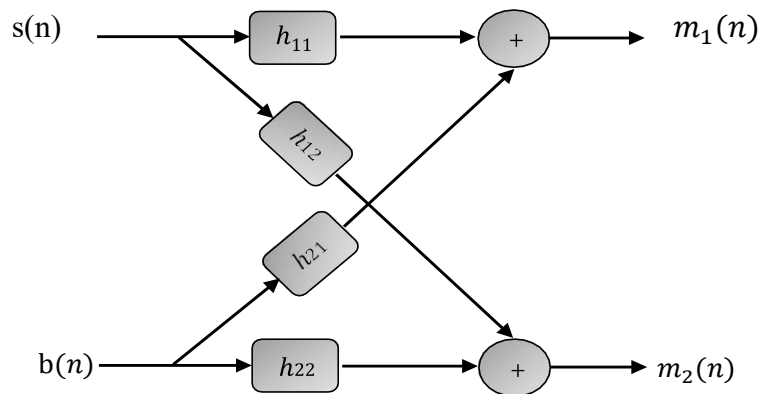


Fig. 1.2 - La structure complète d'un mélange convolutif bi-capteurs.

Les équations des signaux observés à la sortie du mélange sont données par :

$$m_1(n) = h_{11}(n) * s(n) + h_{21}(n) * b(n) \quad (1.1)$$

$$m_2(n) = h_{22}(n) * b(n) + h_{12}(n) * s(n) \quad (1.2)$$

En supposant que le bruit est ponctuel, et la distance entre la source de la parole et la source du bruit et les microphones est petite, on peut dire alors que :

$$h_{11}(n) = h_{22}(n) = \delta(n).$$

1.2.3 Méthode bi-capteurs simplifiée :

En supposant que $h_{11}(n) = h_{22}(n) = \delta(n)$, le schéma de la méthode bi-capteurs simplifiée sera comme suit [1] :

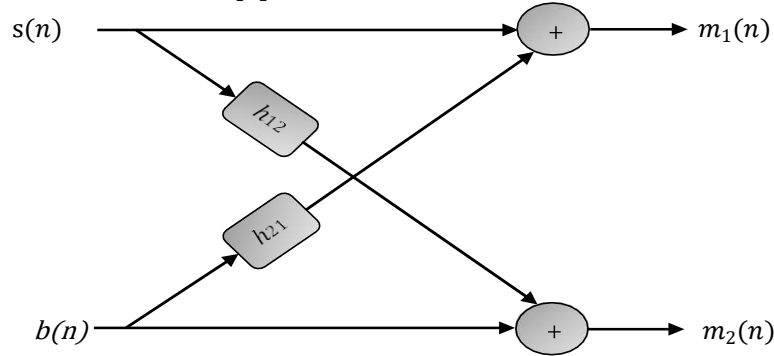


Fig. 1.3 - La structure simplifiée d'un mélange convolutif bi capteurs

Les équations des signaux observés à la sortie du mélange sont données par :

$$m_1(n) = s(n) + \alpha_1 b(n) * h_{21}(n) \quad (1.3)$$

$$m_2(n) = b(n) + \alpha_2 s(n) * h_{12}(n) \quad (1.4)$$

Où $h_{12}(n)$ et $h_{21}(n)$ représentent les effets du couplage mutuel entre les signaux, $s(n)$ et $b(n)$ sont respectivement, deux sources ponctuelles de parole et de bruit.

α_1 : Coefficient pour contrôler l'interférence du bruit dans la voie 1.

α_2 : Coefficient pour contrôler l'interférence de la parole dans la voie 2.

Avec :

$$RSB_1 = 10 \text{Log}_{10} \left(\frac{E_s}{\alpha_1^2 E_b} \right) \quad \Longrightarrow \quad \alpha_1 = \sqrt{\frac{E_s}{E_b} 10^{\frac{-RSB_1}{10}}}$$

$$RSB_2 = 10 \text{Log}_{10} \left(\frac{\alpha_2^2 E_s}{E_b} \right) \quad \Longrightarrow \quad \alpha_2 = \sqrt{\frac{E_b}{E_s} 10^{\frac{+RSB_2}{10}}}$$

1.2.4 Méthode multi-capteurs

Le capteur reçoit plusieurs signaux venus des sources différentes, parole ou bruit. Dans ce cas, φ signaux de sources $s(n) = [s_1(n), s_2(n) \dots \dots s_\varphi(n)]$ subissent une convolution avec φ canaux de $h_{ip}(k)$ (réponses impulsionnelles). Les relations entre les signaux de source et les observations sont données par la relation suivante :

$$m_p(n) = \sum_{i=1}^{\varphi} \sum_{k=0}^{M-1} h_{ip}(k) s_i(n-k), \quad p = 1, 2, \dots, \Psi$$

Où Ψ , φ et M représentent respectivement, le nombre de mélanges, le nombre de sources et la longueur de réponses d'impulsionnelles [2].

Voici ci-dessous le schéma général du mélange :

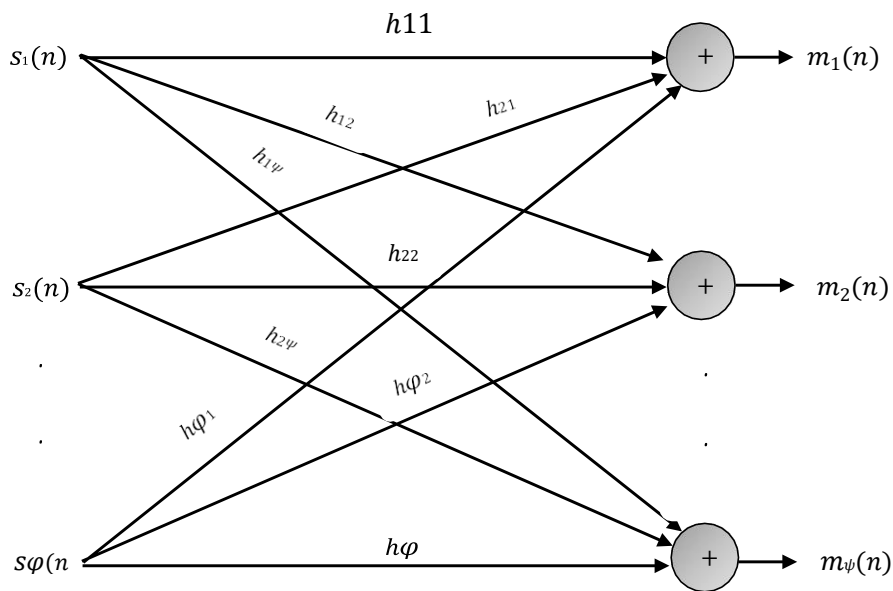


Fig1.4 - La structure d'un mélange convolutif multi capteurs (cas général)

1.2.5 La réduction de bruit par filtrage

Est une méthode de réduction du bruit qui consiste à utiliser des filtres pour éliminer les fréquences indésirables du signal audio. Concrètement, cela se fait en utilisant un filtre qui permet de supprimer les fréquences spécifiques du bruit ambiant. Par exemple, si le bruit ambiant est constitué principalement de basses fréquences, un filtre passe-bas peut être utilisé pour éliminer ces fréquences et ainsi réduire le bruit. Cependant, il convient de noter que cette technique ne fonctionne que si le bruit ambiant est constitué principalement de fréquences spécifiques et peut ne pas être efficace pour éliminer les bruits plus complexes et aléatoires. De plus, l'utilisation de filtres peut également affecter la qualité du signal audio et peut donc nécessiter une expertise technique pour trouver un équilibre entre la réduction du bruit et la qualité audio [6].

1.2.6 Réduction de bruit adaptative

La réduction de bruit adaptative (ou Adaptive Noise Cancelling en anglais) est une méthode de réduction de bruit qui utilise un microphone pour capturer les bruits environnants et un algorithme pour analyser et adapter la réponse du système de réduction de bruit en temps réel. La réduction de bruit adaptative peut s'adapter à des bruits changeants et imprévus. Par exemple, si l'utilisateur est dans un environnement bruyant et qu'un bruit soudain se produit, comme une annonce dans un aéroport ou un

klaxon de voiture, la réduction de bruit adaptative peut rapidement ajuster la réponse du système pour réduire l'impact de ce bruit spécifique. Cependant, il est important de noter que la réduction de bruit adaptative peut être plus coûteuse que d'autres méthodes de réduction de bruit et peut également nécessiter des algorithmes complexes et des composants matériels avancés pour fonctionner efficacement [7].

1.3 Conclusion

Dans ce chapitre, nous avons examiné les principales méthodes et techniques utilisées pour réduire le bruit. Ces méthodes comprennent l'utilisation d'un seul capteur (mono capteur), de deux capteurs (bi capteurs) ou de plusieurs capteurs (multi-capteurs), ainsi que les réductions de bruit par filtrage et la réduction de bruits adaptative. L'objectif de cette partie est de définir les concepts que nous utiliserons dans notre travail.

2.1 Introduction

Au cours de ce deuxième chapitre, nous aborderons le fonctionnement du filtrage de Wiener et du filtrage adaptatif. Nous examinerons également les principaux algorithmes associés au filtrage adaptatif, tels que les algorithmes de gradient stochastique LMS (Least Mean Square) et sa variante normalisée NLMS (Normalized Least Mean Square), ainsi que l'algorithme des moindres carrés RLS (Recursive Least Square).

2.2 Filtrage de Wiener

Le filtrage de Wiener est une technique de traitement de signal utilisée pour restaurer un signal perturbé par du bruit ou d'autres interférences. Cette technique est basée sur la minimisation de l'erreur quadratique moyenne (EQM) entre le signal d'origine et le signal de sortie filtré.

Le filtre de Wiener utilise une approche mathématique qui prend en compte la connaissance préalable de la structure statistique du signal d'origine et du bruit qui le perturbe. En utilisant cette connaissance, le filtre peut estimer la contribution du bruit et la soustraire du signal d'origine pour récupérer une version plus propre du signal. La figure ci-dessous montre une représentation schématique d'un filtre de Wiener [9] :

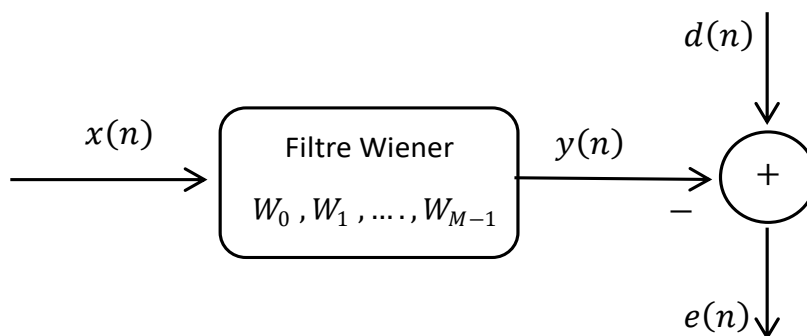


Fig2.1 Représentation schématique d'un filtre Wiener

On définit :

- $x(n)$: signal d'entrée.
- w : filtre Wiener (filtre d'estimation).
- $y(n)$: sortie du filtre adaptatif.
- $e(n)$: signal d'erreur entre la réponse désirée et la sortie du filtre.

- $d(n)$: signal désiré.

Le signal de sortie $y(n)$ est donnée par [1] :

$$y(n) = \sum_{k=0}^{M-1} w(k)x(n-k), n = 0,1, \dots \infty \quad (2.1)$$

Avec :

M : le nombre des coefficients de la réponse impulsionnelle.

On peut exprimer $y(n)$ en notation vectorielle [6] :

$$y(n) = \mathbf{w}(n)^T \mathbf{x}(n) \quad (2.2)$$

Où :

- $\mathbf{w} = [w(0), w(1) \dots \dots, w(M-1)]^T$: est un vecteur de longueur M contenant les coefficients du filtre.
- $\mathbf{x}(n) = [x(n), x(n-1) \dots \dots x(n-M+1)]^T$: est le vecteur de M données d'entrée les plus récentes.
- T : désigne l'opérateur de la transposition.

Le signal d'erreur du filtre de Wiener qui représente la différence entre le signal désiré $d(n)$ et le signal $y(n)$ est donnée par l'équation suivante :

$$e(n) = d(n) - y(n) = d(n) - \mathbf{w}^T \mathbf{x}(n) \quad (2.3)$$

2.2.1 L'erreur quadratique moyenne

L'objectif des systèmes de communication numérique est de réduire les erreurs lors de la transmission de l'information. Pour mesurer cette erreur, il est possible d'utiliser l'erreur quadratique moyenne. La fonction de coût est utilisée comme référence pour minimiser l'erreur en utilisant des filtres optimaux [1] :

$$J = E\{e^2(n)\} \quad (2.4)$$

Où :

J : la fonction cout.

E : L'espérance mathématique.

2.2.2 Principe d'orthogonalité

Les coefficients du filtre optimal, notés \mathbf{w}_{opt} , peuvent être obtenus en annulant le gradient de l'erreur quadratique moyenne par rapport aux coefficients du filtre [6].

$$\nabla J = \frac{\partial J}{\partial \mathbf{w}} \quad (2.5)$$

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

Pour $\nabla J = 0$ on obtient :

$$E\{e_{min}\mathbf{x}(n)\} = 0 \quad (2.7)$$

Lorsque l'erreur $e(n)$ est minimale, cela implique que $e_{min}(n)$ est atteint, et à ce stade, $\mathbf{x}(n)$ et $e(n)$ sont orthogonaux, ce qui représente l'état optimal [3].

Le principe d'orthogonalité stipule que les entrées $\mathbf{x}(n-k)$, où $0 < k < M-1$, sont indépendantes de l'erreur $e_{min}(n)$ et donc décorréélées.

Ainsi, dans le cas optimal nous avons :

$$e_{min} = d(n) - [\mathbf{x}(n)^T \mathbf{w}_{opt}] \quad (2.8)$$

On développe l'équation 2.7, on aura :

$$E\{\mathbf{x}(n)[d(n) - \mathbf{x}(n)^T \mathbf{w}_{opt}]\} = 0 \quad (2.9)$$

$$E[\mathbf{x}(n)d(n)] = E[\mathbf{x}(n)\mathbf{x}(n)^T] \mathbf{w}_{opt} \quad (2.10)$$

Il est possible d'écrire :

$$R_{xd} = R_{xx} \mathbf{w}_{opt} \quad (2.9)$$

Ou encore l'équation de Wiener peut s'écrire sous la forme [1] :

$$\mathbf{w}_{opt} = R_{xx}^{-1} R_{xd} \quad (2.11)$$

Avec :

$$\text{La matrice d'autocorrélation : } R_{xx} = E\{\mathbf{x}(n)\mathbf{x}(n)^T\} \quad (2.12)$$

$$\text{La matrice d'intercorrélacion : } R_{xd} = E\{\mathbf{x}(n)d(n)\} \quad (2.13)$$

2.3 Filtrage adaptatif

2.3.1 Définition

Le filtrage adaptatif est une méthode employée pour régler automatiquement les paramètres d'un filtre lorsque ceux-ci sont instables dans le temps ou difficiles à déterminer. Les coefficients sont ajustés en utilisant un algorithme d'adaptation afin de minimiser les erreurs associées à l'algorithme du gradient, ce qui n'est pas possible avec les filtres classiques. Ce type de filtre numérique est couramment utilisé pour réduire les nuisances sonores et les échos acoustiques.

On peut catégoriser les filtres adaptatifs selon les décisions prises quant aux éléments suivants [9] :

- Les critères d'optimisation.
- L'algorithme de mise à jour des coefficients.
- La structure du filtre programmable.

- Le type de signal traité (mono ou multidimensionnel).

2.3.2 Principe du filtrage adaptatif

Le filtrage adaptatif fonctionne selon les mêmes principes qu'un algorithme classique. Le processus implique deux étapes [4] :

- Tout d'abord, le signal d'entrée $x(n)$ est traité avec un filtre $w(n)$, produisant ainsi un signal de sortie $y(n)$ qui est ensuite comparé au signal souhaité $d(n)$.
- Ensuite, un algorithme prédéfini d'adaptation est utilisé pour calculer les coefficients du filtre $w(n)$, afin de minimiser le signal d'erreur $e(n) = d(n) - y(n)$. Dans certains cas, il est nécessaire de réduire $e(n)$ jusqu'à ce qu'il atteigne la valeur nulle. À ce stade, les coefficients du filtre adaptatif ne s'adaptent plus.

2.3.3 Application du Filtrage adaptatif

L'utilisation du filtrage adaptatif est largement répandue dans les domaines du traitement de signal, des communications numériques et du contrôle automatique. En voici quelques exemples [3] :

- L'identification du système.
- La prédiction.
- L'annulation du bruit et le contrôle.
- L'identification du système.
- L'annulation d'écho.
- Le rehaussement du signal.
- Modélisation inverse.

2.4 Algorithmes adaptatifs

2.4.1 Présentation

Les algorithmes adaptatifs sont basés sur les mêmes principes que les algorithmes classiques. Il se traduit par une séquence finie de valeurs et d'instructions. Comme son nom l'indique, un algorithme adaptatif ajuste et modifie automatiquement son comportement en fonction du contexte lors de son exécution [5].

2.4.2 Critères de comparaison et choix des algorithmes adaptatifs

- **Taux de convergence** : Nombre d'itérations pour converger suffisamment près de la solution de Wiener.

- **Désajustement** : La différence entre la moyenne d'ensemble de l'erreur quadratique et l'erreur quadratique minimale obtenue avec le filtre Wiener.
- **Robustesse** : Résistance au mauvais conditionnement des données.
- **Complexité** : Nombre d'opérations par itération + place mémoire nécessaire (programme et données).
- **Stabilité Numérique** : Influence des erreurs de quantification, problème de la propagation des erreurs.
- **Structure** : Aspect hardware, complexité de l'implantation matérielle.

2.4.3 Algorithme du gradient stochastique LMS

Widrow-Hoff a introduit l'algorithme LMS dans les années 60, qui est considéré comme une version simplifiée de l'algorithme du gradient déterministe. L'algorithme du gradient stochastique est devenu l'algorithme adaptatif le plus couramment utilisé en raison de sa faible complexité. Son objectif est de minimiser l'erreur quadratique moyenne entre un signal estimé et un signal désiré, notamment dans les applications de filtrage adaptatif.

Les performances du LMS sont influencées par trois facteurs : le taux d'adaptation, les valeurs propres de la matrice et la longueur du filtre [1].

On a l'équation de $e(n)$:

$$e(n) = d(n) - y(n) = d(n) - \mathbf{x}(n)^T \mathbf{w}(n) \quad (2.14)$$

L'équation de l'algorithme du gradient déterministe est donnée par [2] :

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \frac{1}{2} \mu \frac{\partial j[\mathbf{w}(n)]}{\partial [\mathbf{w}(n)]} \quad (2.15)$$

L'algorithme inclut une étape d'ajustement appelée pas d'adaptation μ , qui permet de réguler la vitesse de convergence. Ce paramètre est déterminé en fonction de l'énergie de $x(n)$ tel que :

$$0 < \mu < \frac{1}{L\sigma_x^2} \quad (2.16)$$

Où :

- σ_x^2 : L'Energie du signal d'entrée $x(n)$.
- L : La taille du filtre adaptatif.

On peut écrire aussi :

$$\frac{\partial j[\mathbf{w}(n)]}{\partial [\mathbf{w}(n)]} = -2R_{xd} + 2R_{xx}\mathbf{w}(n) \quad (2.17)$$

En simplifiant l'équation $\mathbf{w}(n+1)$ on trouve :

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu \mathbf{x}(n)[d(n) - \mathbf{x}(n)^T \mathbf{w}(n)] \quad (2.18)$$

Donc la nouvelle équation de mise à jour de l'algorithme LMS est donnée par [4] :

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu \mathbf{x}(n)e(n) \quad (2.19)$$

2.4.4 Algorithme du gradient stochastique normalisé NLMS

Le NLMS, une version normalisée de l'algorithme LMS, est conçu pour mieux converger avec des signaux d'entrée (tels que la parole) qui ont une grande variation d'énergie (appelés signaux non-stationnaires). Cette amélioration est obtenue grâce à la normalisation du pas d'adaptation μ de l'algorithme LMS, qui est divisé par l'énergie du signal d'entrée $\mathbf{x}(n)$. Un des inconvénients de l'algorithme NLMS par rapport au LMS est l'augmentation de la complexité de calcul au niveau du nombre de multiplications. L'algorithme NLMS présente l'avantage de rendre le pas d'adaptation indépendant de la variance du signal d'entrée, contrairement à l'algorithme LMS [5].

Le pas d'adaptation μ est remplacé par :

$$\mu_n = \frac{\mu}{\mathbf{x}(n)^T \mathbf{x}(n)} \quad (2.20)$$

Donc l'équation de mise à jour des coefficients du filtre adaptatif par l'algorithme NLMS est donnée par :

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\mu}{\mathbf{x}(n)^T \mathbf{x}(n) + \varepsilon} e(n) \mathbf{x}(n) \quad (2.21)$$

Où ε désigne une constante positive pour éviter la division par zéro.

Pour étudier la stabilité de NLMS, on compare l'erreur a priori $e(n) = d(n) - \mathbf{w}(n)^T \mathbf{x}(n)$ à l'erreur a posteriori $\epsilon(n) = d(n) - \mathbf{w}(n+1)^T \mathbf{x}(n)$ pour voir si la première est plus grande que la seconde [4].

On remplace la formule de mise à jour de L'NLMS dans $\epsilon(n)$ et en supposant que $\varepsilon = 0$ on trouve [3] :

$$\epsilon(n) = d(n) - \left(\mathbf{w}(n) + \frac{\mu}{\mathbf{x}(n)^T \mathbf{x}(n)} e(n) \mathbf{x}(n) \right)^T \mathbf{x}(n) \quad (2.22)$$

$$\begin{aligned} \epsilon(n) &= d(n) - \mathbf{w}(n)^T \mathbf{x}(n) - \mu e(n) \\ \epsilon(n) &= e(n)(1 - \mu) \end{aligned} \quad (2.23)$$

Et comme :

$$\begin{aligned} |\epsilon(n)| &< |e(n)| \\ e(n)(1 - \mu) &< 1 \\ 0 &< \mu < 2 \end{aligned} \quad (2.24)$$

Si la valeur du pas d'adaptation μ se situe dans l'intervalle $]0,2[$, l'algorithme NLMS atteint une convergence.

2.4.5 Algorithme à méthode récursive à moindre carré RLS

L'algorithme RLS est plus complexe que celui du LMS, mais il présente une convergence plus rapide, une erreur asymptotique plus faible et une faible sensibilité aux variations des valeurs propres du système. Les étapes de mise en œuvre sont similaires à celles du LMS, mais avec quelques différences notables, notamment la mise à jour des poids et la présence du gain [4].

L'objectif de cet algorithme est de calculer les valeurs des paramètres $w(n)$ du filtre adaptatif en utilisant la méthode des moindres carrés.

$$J(n) = \sum_{i=0}^n \lambda^{n-i} [d(i) - \mathbf{w}(n)^T \mathbf{x}(i)]^2 \quad (2.25)$$

Avec :

λ : Facteur de pondération ou d'oubli compris entre 0 et 1.

Ce facteur permet d'ignorer les données correspondant à un passé lointain.

Pour trouver les coefficients de $w(n)$, il faut minimiser la fonction $J(n)$, ce qui revient à calculer la dérivée partielle de cette fonction par rapport à $w(n)$ et on obtient [4] :

$$\mathbf{w}(n) = \mathbf{w}(n-1)k(n)e(n) \quad (2.26)$$

Avec :

$k(n)$: est le gain du Kalaman qui est définie par :

$$k(n) = \frac{\lambda^{-1}Q(n-1)\mathbf{x}(n)}{1+\lambda^{-1}\mathbf{x}(n)^T Q(n-1)\mathbf{x}(n)} \quad (2.27)$$

Où :

$$Q(n) : \lambda^{-1}Q(n-1) - \lambda^{-1}k(n)\mathbf{x}(n)^T Q(n-1) \quad (2.28)$$

2.5 Conclusion

Ce chapitre nous a permis d'avoir un aperçu du filtrage adaptatif et de ses diverses applications. Ensuite, nous avons examiné en détail les algorithmes de la famille de gradient stochastique, notamment les algorithmes LMS et NLMS, ainsi que l'algorithme RLS basé sur la méthode récursive des moindres carrés.

Chapitre 3 Proposition du nouvel algorithme NLMF automatique

3.1 Introduction

L'algorithme du moindre quatrième moyenne (LMF) est une amélioration de l'algorithme du moindre carré moyen (LMS) qui offre un meilleur équilibre entre les performances transitoires et stationnaires d'un filtre adaptatif. Cependant, l'algorithme LMF présente des problèmes de stabilité qui peuvent limiter son utilisation dans certaines applications. La stabilité de l'algorithme dépend de la puissance du signal d'entrée et du bruit, ainsi que de la configuration initiale des poids du filtre.

Un nouvel algorithme NLMF est proposé, dans lequel le terme de mise à jour des poids est normalisé par la quatrième puissance du régresseur. Cette normalisation permet d'améliorer la stabilité de l'algorithme en évitant la divergence lorsque la puissance du signal d'entrée dépasse une certaine valeur seuil. Cependant, il est démontré que la stabilité de cet algorithme dépend toujours de la puissance du signal d'entrée.

3.2 Principe de nouvel algorithme NLMF

Pour améliorer la stabilité de l'algorithme LMF, des versions normalisées appelées NLMF ont été proposées. Cependant, aucun des algorithmes NLMF existants ne résout complètement les problèmes de stabilité mais ces algorithmes normalisés ont des termes de mise à jour des poids qui sont normalisés par des mesures de puissance du régresseur et de l'erreur d'estimation.

3.3 Problème de stabilité d'algorithme NLMF

3.3.1 Notions et hypothèses

Prenons le cas de l'identification adaptative des plans décrite par Figure suivante [10] :

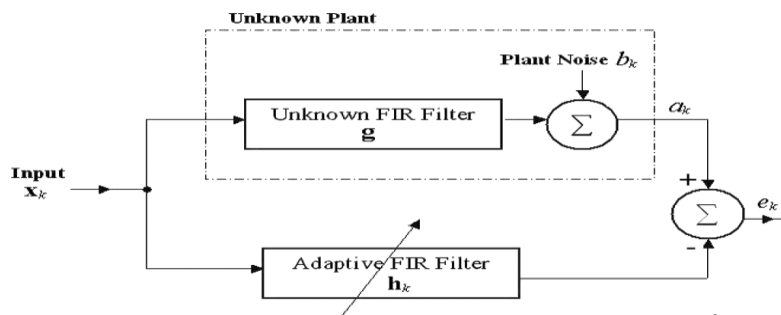


Fig. 3.1 Identification du plan adaptatif

Le signal de sortie a_n est donnée par :

$$a(n) = g(t)x(n) + \mathbf{b}(n) \quad (3.1)$$

Où :

$$\mathbf{g} \equiv (g(1), g(2), \dots, g(N))^T \quad (3.2)$$

Et :

$x(n)$ Est le signal d'entrée

$$\mathbf{x}(n) = (x(n), x(n-1), \dots, x(n-N+1))^T \quad (3.3)$$

Est le vecteur de régression au moment n où x_n est l'entrée du plan, n est le nombre de paramètres du plan, $\mathbf{b}(n)$ est le bruit de la plante, et $(.)^T$ est la transposition de $(.)$. L'identification du plan se fait par un filtre FIR adaptatif dont la longueur est supposée égale à celle du plan. Le poids du vecteur $\mathbf{w}(n)$ du filtre adaptatif est adapté en fonction de l'erreur donné par :

$$e(n) = a(n) - \mathbf{w}(n)^T \mathbf{x}(n) \quad (3.4)$$

Où l'algorithme d'adaptation considéré dans cette correspondance est l'algorithme LMF écrit par

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e(n) \mathbf{x}(n) \quad (3.5)$$

Où $\mu > 0$ est la taille du pas de l'algorithme, le vecteur de l'écart de poids est défini par :

$$\mathbf{v}(n) \equiv \mathbf{w}(n) - \mathbf{g} \quad (3.6)$$

La division carrée moyenne instantanée est donnée par $\|\mathbf{v}(n)\|^2$ où E dénote l'espérance mathématique et $\|\mathbf{v}(n)\|$ est l'Euclidien Norme du vecteur $\mathbf{v}(n)$ qui est $\|\mathbf{v}(n)\| = \sqrt{\mathbf{v}(n)^T \mathbf{v}(n)}$. Les hypothèses suivantes sont utilisées :

Hypothèses A1 : Les séquences $\mathbf{x}(n)$ et $\mathbf{v}(n)$ sont indépendantes les unes des autres

Hypothèses A2 : $\mathbf{x}(n)$ est une séquence stationnaire de variables aléatoires moyennes nulles avec une variance définie et une matrice de covariance définie positive E

Hypothèses A3 : $\mathbf{x}(n)$ est une suite stationnaire de variables aléatoires de moyenne nulle avec une variance finie et zéro moment d'ordre impair.

3.3.2 Facteur affectant la stabilité d'algorithme LMF

La condition suffisante de convergence $E(\mathbf{v}(n))$ suivante de de l'algorithme LMF est dérivé :

$$0 < \mu < \frac{1}{2E(b(n)^2)E(\|\mathbf{x}(n)\|^2)} \quad (3.7)$$

L'équation (3.7) implique que la stabilité de l'algorithme LMF dépend sur la puissance d'entrée du filtre adaptatif et la puissance de bruit. Cependant, la condition (3.7) est dérivée sous l'hypothèse que l'écart vecteur est proche de zéro, c'est-à-dire :

$$\mathbf{v}(n) \approx \mathbf{0} \quad (3.8)$$

Ainsi, (3.7) ne concerne que la stabilité $E(\mathbf{v}(n))$ autour du filtre de Wiener, il ne concerne pas la stabilité dans la phase initiale d'adaptation, où l'amplitude du vecteur de déviation n'est généralement pas petite et alors (3.8) ne tient pas. Des études complémentaires de l'algorithme LMF ont montré que la stabilité de l'algorithme LMF dépend sur le réglage initial des poids du filtre adaptatif. L'algorithme LMF a une probabilité de divergence non nulle, quelle que soit la taille du pas, pour les régresseurs dont la probabilité les densités ont un support infini, comme dans le cas des régresseurs gaussiens. D'après la discussion ci-dessus, la stabilité de l'algorithme LMF dépend sur les facteurs suivants :

- Puissance d'entrée du filtre adaptatif
- Bruit puissance
- Réglage initial des poids du filtre adaptatif
- Limite de la distribution de l'entrée du filtre adaptatif

Ainsi, le LMF algorithme souffre de nombreux problèmes de stabilité par rapport à l'Algorithme LMS dont la stabilité dépend uniquement de la puissance d'entrée du filtre adaptatif.

3.3.3 Algorithmes LMF normalisés

Un outil bien connu qui peut être utilisé pour améliorer la stabilité des algorithmes de filtrage adaptatif est la normalisation. Rappelons d'abord brièvement les normalisations de l'algorithme LMS. Il est bien connu que la stabilité de l'algorithme LMS dépend de la puissance d'entrée du filtre adaptatif. Cela rend très difficile, voire impossible, le choix d'une taille de pas qui garantit la stabilité de l'algorithme en cas de manque connaissance de la puissance d'entrée. Ceci est résolu en normalisant le terme de mise à jour du vecteur de poids par $\|\mathbf{x}(n)\|^2$. L'algorithme résultant est appelé comme l'algorithme LMS normalisé (NLMS). Cet algorithme est stable pour toute la puissance d'entrée, la puissance de bruit et le réglage initial des poids de filtre adaptatifs, tant que la taille du pas est comprise entre 0 et 2. Maintenant, nous souhaitons une normalisation de l'algorithme LMF qui produit un algorithme avec des caractéristiques de stabilité similaires. Trois algorithmes NLMF sont disponibles. Le premier est décrit par [10] :

$$\mathbf{w}(n + 1) = \mathbf{w}(n) + \mu \frac{e(n)^3 \mathbf{x}(n)}{\|\mathbf{x}(n)\|^2} \quad (3.9)$$

Un exemple est utilisé pour montrer que l'algorithme (3.9) diverge lorsque la puissance d'entrée dépasse une valeur seuil qui dépend de la taille du pas de l'algorithme. La stabilité quadratique moyenne de l'algorithme (3.9) est analysé pour l'entrée gaussienne blanche et le bruit. L'analyse montre que la stabilité de (3.9) dépend de la puissance d'entrée, du bruit d'initialisation de la puissance et du poids. Le deuxième algorithme NLMF est décrit par [10] :

$$\mathbf{w}(n + 1) = \mathbf{w}(n) + \mu \frac{e(n)^3 \mathbf{x}(n)}{\delta + \lambda \|\mathbf{x}(n)\|^2 + (1-\lambda) \|e(n)\|^2} \quad (3.10)$$

λ est le vecteur d'erreur incluant la puissance d'erreur dans le terme de normalisation de (3.10) peut atténuer l'erreur d'estimation aberrantes, améliorant ainsi la stabilité. Le paramètre λ est appelé paramètre de puissance de mélange. Le choix de λ est un compromis entre une convergence rapide et une faible erreur en régime permanent. Une version modifiée de (3.10) avec un adaptatif. On montre que l'algorithme (3.10) diverge lorsque la puissance d'entrée dépasse une valeur seuil qui dépend de la taille du pas de l'algorithme. A raison de la divergence de (3.9) et (3.10) lorsque la puissance d'entrée augmente est le fait que dans (3.9) et (3.10), le numérateur du terme de mise à jour du vecteur de poids est du quatrième ordre dans $\mathbf{x}(n)$ tandis que le dénominateur est du second ordre dans $\mathbf{x}(n)$ Pour surmonter ce problème, le troisième NLMF Il est décrit par [10] :

$$\mathbf{w}(n + 1) = \mathbf{w}(n) + \mu \frac{e(n)^3 \mathbf{x}(n)}{\|\mathbf{x}(n)\|^4} \quad (3.11)$$

À la fois le numérateur et le dénominateur du terme de mise à jour du vecteur de poids dans (3.11) sont du quatrième ordre dans le régresseur. L'algorithme (3.11) a une meilleure stabilité que les algorithmes (3.9) et (3.10). la stabilité quadratique moyenne de l'algorithme (3.11) est analysée pour une entrée gaussienne blanche et bruit. La stabilité de (3.11) dépend toujours de la puissance d'entrée, de la puissance de bruit, et initialisation du poids. Ainsi, en résumé, aucune des techniques de normalisation disponibles n'atteint une stabilisation globale de l'algorithme LMF [14].

Dans la section suivante, nous présentons une normalisation du LMF algorithme qui réalise une stabilisation globale de l'algorithme LMF. A savoir, l'algorithme NLMF résultant est stable pour toutes les valeurs de :

- Puissance d'entrée du filtre adaptatif,
- Puissance de bruit,

- Réglage initial des poids du filtre adaptatif,
- Le type de distribution de l'entrée du filtre adaptatif.

3.4 L'algorithme proposé

L'algorithme NLMF proposé est décrit par

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu \frac{e(n)^3 \mathbf{x}(n)}{\|\mathbf{x}(n)\|^2 (\|\mathbf{x}(n)\|^2 + e(n)^2)}, \quad 0 < \mu < 2 \quad (3.12)$$

L'idée de cet algorithme est une combinaison entre les équations (3.10) et (3.11). Le terme normalisant dans (3.12) est un quatrième ordre polynôme en $x(n)$. Cette propriété stabilise l'algorithme contre l'augmentation de la variance d'entrée, puisque $e(n)^3 \mathbf{x}(n)$ est un polynôme du quatrième degré en $\mathbf{x}(n)$. La même propriété stabilise également l'algorithme pour les entrées avec des distributions illimitées, telles que les entrées gaussiennes. Le terme normalisant dans (3.12) comprend également $e(n)^2$. Cette propriété stabilise l'algorithme contre l'augmentation de la variance du bruit et contre l'augmentation du carré écart de poids, puisque $e(n) = b(n) + \mathbf{v}(n)^T \mathbf{x}(n)$. Maintenant, nous montrons que l'algorithme (3.12) est stable pour toutes les valeurs de la puissance d'entrée, de la puissance de bruit, initialisation du vecteur de poids et type de distribution de l'entrée. Une preuve rigoureuse de cette caractéristique est difficile du fait que le poids le terme de mise à jour du vecteur dans (3.12) est hautement non linéaire dans le régresseur et le poids de filtre adaptatifs. Par conséquent, nous suivrons une procédure heuristique pour montrer la stabilité de (3.12). Deux approches sont proposées. La première approche est la suivante. Réécrire l'équation (3.12) sous la forme :

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \alpha \frac{e(n)^3 \mathbf{x}(n)}{\|\mathbf{x}(n)\|^2} \quad (3.13)$$

Où

$$\alpha(n) \equiv \frac{\mu e(n)^2}{\|\mathbf{x}(n)\|^2 + e(n)^2}, \quad 0 < \mu < 2 \quad (3.14)$$

En gros, l'équation (3.12) peut être considéré comme une équation de l'algorithme NLMS (3.13) avec une taille de pas variant dans le temps α_n défini par (3.14). En raison de (3.14), pour tout : $0 < \mu < 2$ on a : $0 < \alpha(n) < 2$ Pour tout : $\mathbf{x}(n)$, $b(n)$ et $\mathbf{v}(n)$. (3.15)

Pour une matrice de covariance définie positive $E(\mathbf{x}(n)\mathbf{x}(n)^T)$, l'algorithme NLMS est stable tant que sa taille de pas est comprise entre 0 et 2. Cette condition est satisfaite pour l'algorithme (3.13) en raison de (3.15). L'algorithme est donc stable. Le fait que (3.15) soit satisfait pour tous $\mathbf{x}(n)$, $b(n)$ et $\mathbf{v}(n)$ implique que l'algorithme est stable pour

toutes les entrées variance, variance du bruit, initialisation du vecteur de poids et type de distribution de l'entrée.

La seconde approche montre la stabilité de l'algorithme par contradiction selon les arguments heuristiques suivants. Si l'algorithme (3.12) est divergent, $e(n)^2$ tend vers l'infini comme n tend vers l'infini. Alors, (3.14) implique que α_n tend à μ comme n tend vers l'infini. Par conséquent, (3.13) implique que l'équation (3.12) tend vers l'équation de l'algorithme NLMS $\mathbf{w}(n+1) = \mathbf{w}(n) + \mu \frac{e(n)^3 \mathbf{x}(n)}{\|\mathbf{x}(n)\|^2}$, qui est stable pour tout :

$0 < \mu < 2$, Cela contredit l'hypothèse ci-dessus selon laquelle l'algorithme (3.12) est divergent. L'algorithme (3.12) est donc non divergent pour tout $0 < \mu < 2$.

Il convient de mentionner que la plage de stabilité par pas est entre $0 < \mu < 2$. De l'algorithme NLMF proposé (3.12) ne nécessite aucune connaissance sur les statistiques d'initialisation d'entrée, de bruit et de poids, qui est un avantage significatif par rapport aux versions disponibles de l'Algorithme NLMF. Enfin, il convient de mentionner que la proposition L'algorithme NLMF peut être régularisé afin de contourner le problème quand $\|\mathbf{x}(n)\|$ est proche de zéro. Cela se fait en ajoutant un petit positif δ au dénominateur du terme de mise à jour des poids d'algorithme (3.12). Dans ce cas, l'algorithme devient :

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu \frac{e(n)^3 \mathbf{x}(n)}{\delta + \|\mathbf{x}(n)\|^2 (\|\mathbf{x}(n)\|^2 + e(n)^2)}, \quad 0 < \mu < 2 \quad (3.16)$$

La valeur de δ est choisie telle que $\delta \ll E(\|\mathbf{x}(n)\|^4)$, et alors elle n'aura pas eu un effet significatif sur la performance moyenne de l'algorithme

3.5 Conclusion

Un nouvel algorithme de LMF normalisé (NLMF) est donné, l'algorithme est stable pour toutes les statistiques de l'entrée, du bruit et de l'initialisation des poids. L'algorithme est obtenu en normalisant le terme de mise à jour du vecteur de poids par un terme qui est de quatrième ordre dans le régresseur et de second ordre dans l'erreur d'estimation. Le fait que le terme de normalisation soit de quatrième ordre dans le régresseur stabilise l'algorithme contre l'augmentation de la variance d'entrée et contre le caractère non borné de la distribution d'entrée. Le fait que le terme de normalisation soit de second ordre dans l'erreur d'estimation stabilise l'algorithme contre l'augmentation de la variance du bruit et contre l'augmentation de l'écart initial des poids. L'algorithme est stable pour toutes les valeurs de la taille du pas dans la plage entre 0 et 2.

Chapitre 4 Résultats Des Simulations

4.1 Introduction

Dans ce chapitre nous allons présenter les résultats de simulation obtenus à l'aide du logiciel de simulation Matlab. Tout d'abord, une brève introduction est faite sur les signaux source utilisés dans cette étude, à savoir la parole et le bruit, ainsi que les réponses impulsionnelles correspondantes. Ensuite, une comparaison entre les deux algorithmes adaptatifs NLMS et le nouvel algorithme NLMF automatique est présentée. Les performances des algorithmes sont évaluées selon deux critères : le système Mismatch (SM) et l'erreur quadratique moyenne (EQM).

4.2 Signaux utilisés aux simulations

4.2.1 Signal de la parole

Dans cette étude, nous avons utilisé une séquence échantillonnée à une fréquence d'échantillonnage de $f_e = 8000$ Hz. Cette séquence a été prononcée par un locuteur avec la phrase suivante : "Un loup s'est jeté immédiatement sur la petite chèvre". La figure ci-dessous illustre le signal de la parole utilisé.

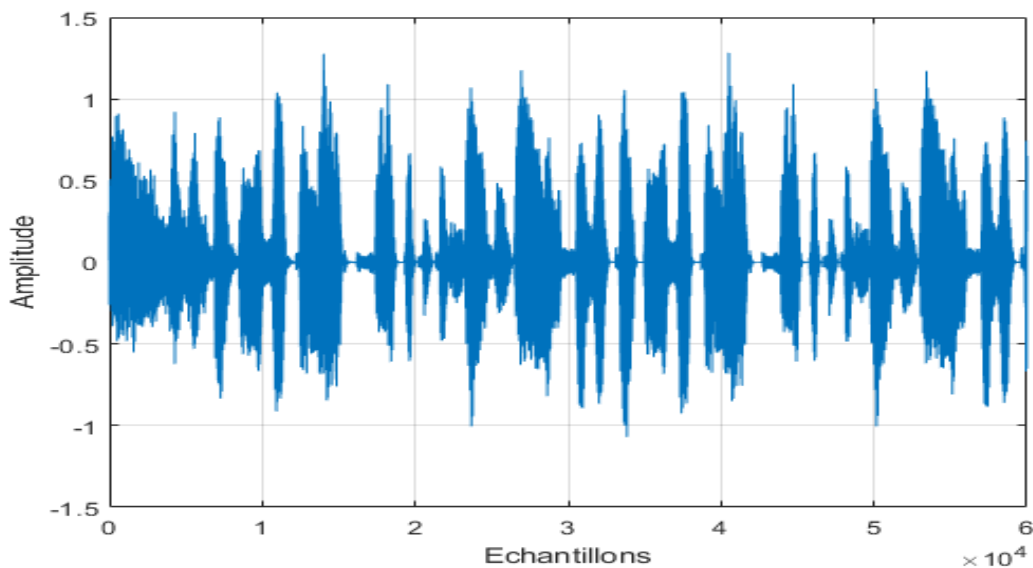


Fig4.1 - Signal de parole utilisé

4.2.2 Bruit blanc

Le bruit blanc est un processus aléatoire stationnaire dans lequel les échantillons sont des variables aléatoires non corrélées entre elles. Il se distingue par sa densité spectrale de puissance constante pour toutes les fréquences. Ce type de bruit est souvent utilisé pour évaluer la stabilité des algorithmes adaptatifs. La figure ci-dessous illustre le signal de bruit blanc

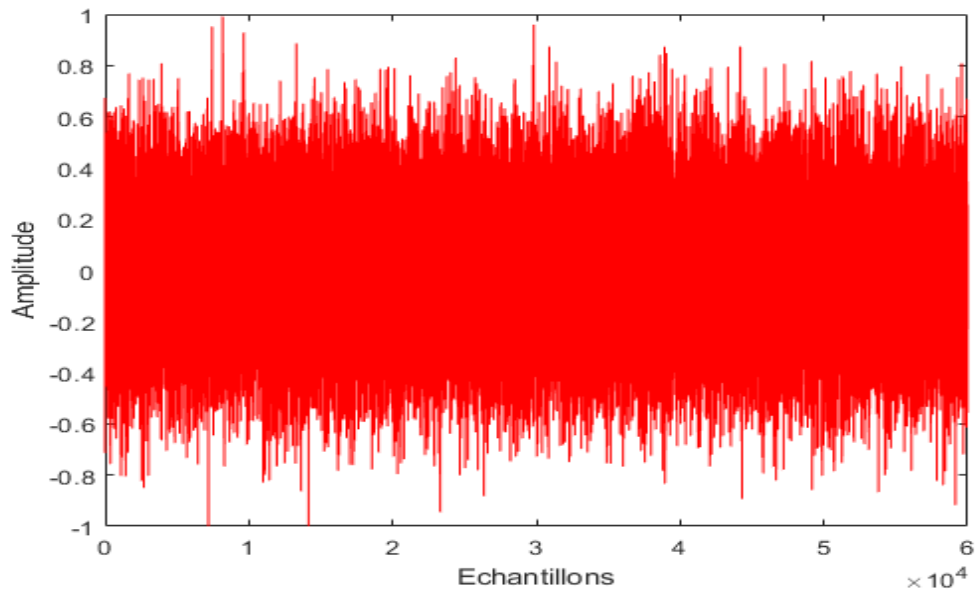


Fig4.2 – Signal du bruit blanc utilisé

4.2.3 Réponses impulsionnelles

Les figures ci-dessous illustrent la modélisation des réponses impulsionnelles des filtres h_{12} et h_{21} . Ces réponses impulsionnelles sont obtenues en utilisant un bruit gaussien (GN) modulé par une fonction exponentielle avec une durée approximative de $L = 64$ échantillons et une fréquence d'échantillonnage $f_e = 8 \text{ kHz}$. Ces deux modèles permettent de mieux représenter les conditions réelles de propagation des signaux à travers une salle donnée et de mettre en évidence l'effet de la distance.

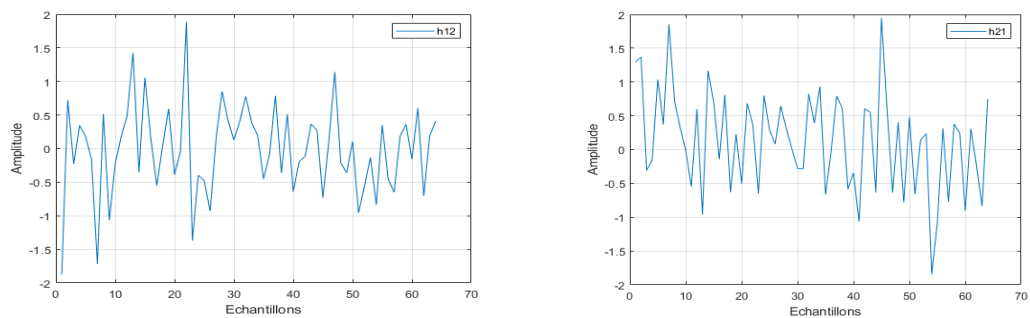


Fig4.3 – La réponse impulsionnelle h_{12} (à gauche) et la réponse impulsionnelle h_{21} (à droite) avec $L = 64$.

4.2.4 Signaux de mélange

Les signaux présentés dans cette section sont les sorties de la structure du mélange convolutif $m1(n)$ et $m2(n)$. Les signaux d'entrée utilisés sont le signal de parole et le signal de bruit, avec un rapport signal/bruit de 3 dB pour les deux voies et une longueur de filtre $L = 64$.

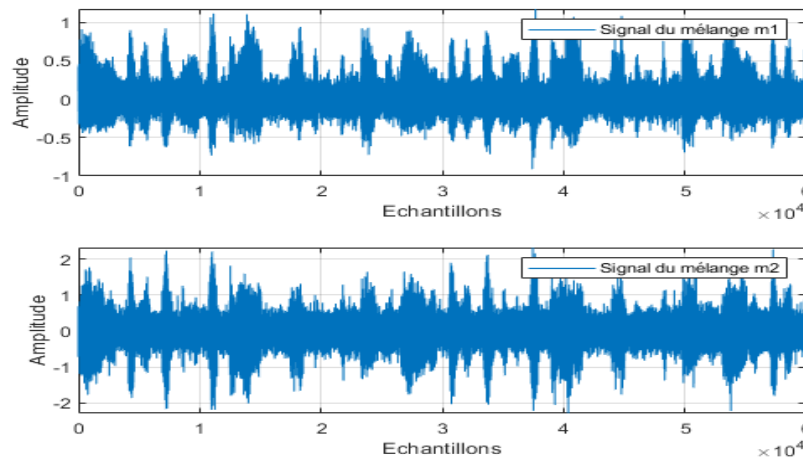


Fig4.4 – Le signal du premier mélange $m1(n)$ (en haut) et le signal du deuxième mélange $m2(n)$ (en bas).

4.3 Principe de détecteur d'activité locale (DAV)

Un algorithme de détection d'activité vocale (DAV) peut faire la distinction entre les moments où il y a de l'activité vocale et ceux où il n'y en a pas, afin de contrôler l'adaptation du filtre $w(n)$. Ce contrôle adaptatif fonctionne de la manière suivante : le filtre $w(n)$ s'ajuste uniquement pendant les périodes de silence

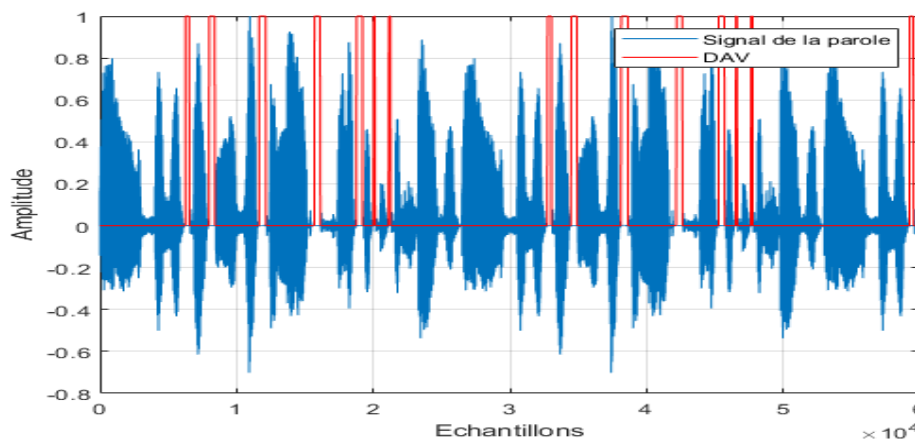


Fig4.5 – Signal de la parole avec sa DAV.

4.4 Résultat de simulation

Le tableau (4.1) résume toutes les valeurs des paramètres utilisées dans cette simulation :

Tableau 4.1 - Paramètre de Simulation

Paramètre	Signification	Valeur
L	Longueur du filtre	32,128
N	Nombre d'itérations	130000
Fe	Fréquence d'échantillonnage	8000
RSB1=RSB2	Rapport Signal Bruit	3dB, 0dB et -3dB
α	Coefficient de contrôle d'interférence	0.999
μ	Pas d'adaptation	0.9

4.5 Evaluation du nouvel algorithme proposé

Nous allons maintenant examiner les performances du nouvel algorithme NLMF dans cette section. Nous allons le faire en étudiant les effets des différentes tailles de filtre, des niveaux variés de RSB d'entrée, ainsi que l'utilisation de deux critères objectifs de qualité.

4.5.1 Critère du MSE segmentée (SegMSE)

La mesure de l'erreur quadratique moyenne segmentée (Segmental-MSE) permet d'évaluer le bruit résiduel restant dans le signal. Elle est calculée pendant les périodes de silence. L'estimation du *SegMSE* repose sur la relation suivante :

$$SegMSE_{dB} = 10 \log_{10}(Eb)$$

Avec : Eb est l'Energie du signal de bruit.

4.5.2 Le critère de désajustement (System Mismatch)

Le critère de désajustement, également appelé Mismatch, est un critère robuste utilisé pour évaluer la rapidité de convergence des systèmes de réduction de bruit grâce à des algorithmes adaptatifs. Il est quantifié par la formule suivante :

$$SM_{dB} = 10 \log_{10} \frac{|h_{21} - w|^2}{|h_{21}|^2}$$

Avec : h_{21} est la réponse impulsionnelle du filtre réel.

4.5.3 L'effet du *RSB* avec une taille de filtre $L = 32$

En utilisant un bruit blanc, et pour une longueur de filtre fixée à $L = 32$, nous avons varié le *RSB* d'entrée.

- $L=32$ et $RSB= 3dB$

La figure (4.6) représente une évolution temporelle où nous faisons une comparaison entre l'algorithme NLMS et le nouvel algorithme NLMF. Nous remarquons dans la figure (4.6) une qualité acceptable du signal de sortie de l'algorithme NLMF proposé par rapport à NLMS.

La figure (4.7) représente l'évolution des réponses impulsionnelles réelles avec les réponses impulsionnelles de l'algorithme NLMS. Nous remarquons dans la figure (4.7) que les réponses impulsionnelles réelles sont presque identiques avec celles de l'algorithme NLMS sauf qu'elles se différencient d'amplitudes sur quelques échantillons.

La figure (4.8) représente l'évolution des réponses impulsionnelles réelles avec les réponses impulsionnelles de l'algorithme NLMF. Nous remarquons dans la figure (4.8) que les réponses impulsionnelles réelles sont identiques avec celles de l'algorithme NLMF.

La figure (4.9) représente le critère System Mismatch des algorithmes NLMS et NLMF. Nous remarquons dans cette figure que notre algorithme proposé converge jusqu'à qu'il dépasse la valeur $-30dB$ puis il reste varié entre $-20db$ et $-30dB$ de la valeur du SM alors que l'algorithme NLMS converge dans un petit laps de temps en atteignant la valeur $-10dB$ puis reste instable.

La figure (4.10) représente la variation de l'erreur quadratique moyenne (EQM) des algorithmes NLMS et NLMF. Nous remarquons que l'algorithme proposé, le NLMF a pu atteindre une valeur de $-85dB$.

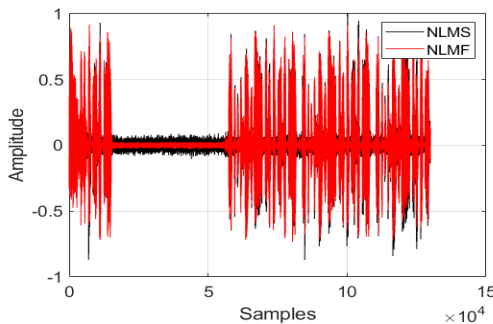


Fig4.6- Evaluation temporelle entre le NLMS et le NLMF

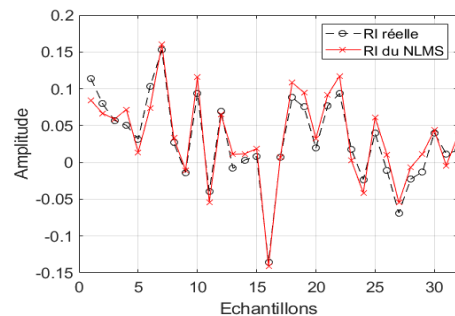


Fig4.7- Comparaison entre RI réelle et RI du NLMS

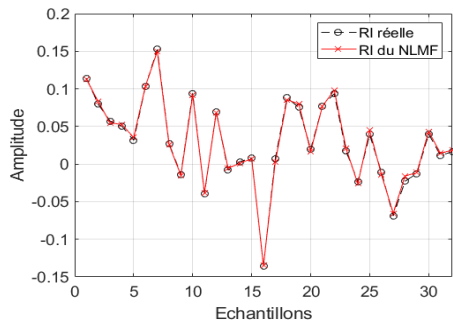


Fig4.8- Comparaison entre RI réelle et RI du NLMF

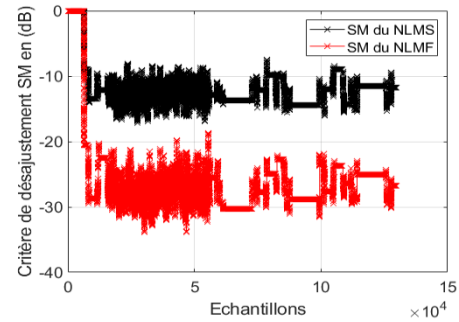


Fig4.9- Critère SM

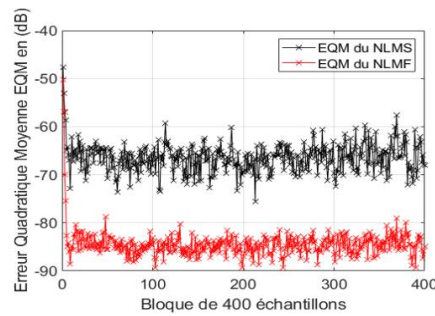


Fig10- Critère EQM Du NLMS ET NLMF

- **L= 32 et RSB=0dB**

La figure (4.11) représente une évolution temporelle où nous faisons une comparaison entre l'algorithme NLMS et le nouvel algorithme NLMF. Nous remarquons dans la figure une qualité meilleure du signal de sortie de l'algorithme NLMF proposé par rapport à NLMS.

La figure (4.12) représente l'évolution des réponses impulsionnelles réelles avec les réponses impulsionnelles de l'algorithme NLMS. Nous remarquons dans la figure (4.12) que les réponses impulsionnelles réelles et les réponses impulsionnelles de l'algorithme NLMS sont presque superposées.

La figure (4.13) représente l'évolution des réponses impulsionnelles réelles avec les réponses impulsionnelles de l'algorithme NLMF. Nous remarquons dans la figure (4.13) que les réponses impulsionnelles réelles et les réponses impulsionnelles du NLMF sont superposées.

La figure (4.14) représente le critère System Mismatch des algorithmes NLMS et NLMF. Nous remarquons dans cette figure que notre algorithme proposé maintient un niveau très faible de SM qui a pu atteindre une valeur de -30dB.

La figure (4.15) représente la variation de l'erreur quadratique moyenne (EQM) des algorithmes NLMS et NLMF. Nous remarquons que l'algorithme NLMS converge puis reste stable vers une valeur de l'EQM qui dépasse les -60dB et touche -70dB à

quelques reprises alors que l'algorithme proposé converge jusqu'il atteigne la valeur de -80dB

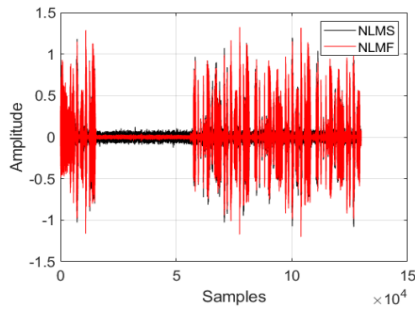


Fig4.11 - Evaluation temporelle entre le NLMS et le NLMF

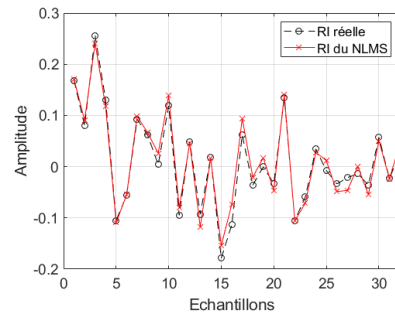


Fig4.12- Comparaison entre RI réelle et RI du NLMS

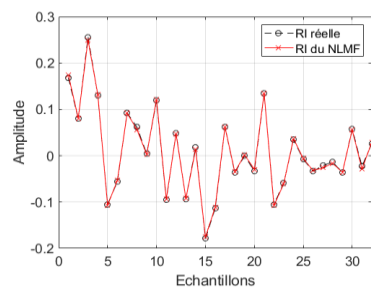


Fig4.13- Comparaison entre RI réelle et RI du NLMF

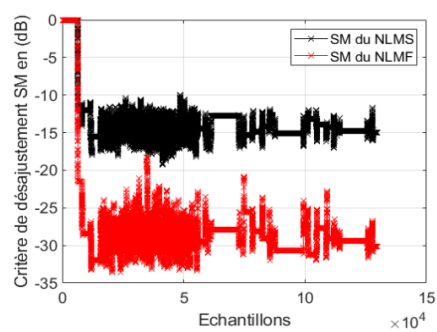


Fig4.14- Critère SM

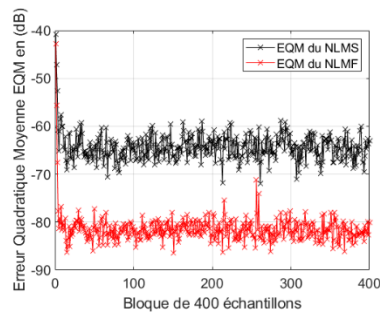


Fig15-Critère EQM Du NLMS ET NLMF

- $L=32$ et $RSB= -3dB$

La figure (4.16) représente une évolution temporelle où nous faisons une comparaison entre l'algorithme NLMS et le nouvel algorithme NLMF. Nous remarquons dans la figure une qualité meilleure du signal de sortie de l'algorithme NLMF proposé par rapport à NLMS.

La figure (4.17) représente l'évolution des réponses impulsionnelles réelles avec les réponses impulsionnelles de l'algorithme NLMS. Nous remarquons dans la figure (4.17) que les réponses impulsionnelles réelles et les réponses impulsionnelles de l'algorithme NLMS sont presque superposées.

La figure (4.18) représente l'évolution des réponses impulsionnelles réelles avec les réponses impulsionnelles de l'algorithme NLMF. Nous remarquons dans la figure (4.18) que les réponses impulsionnelles réelles et les réponses impulsionnelles du NLMF sont superposées.

La figure (4.19) représente le critère System Mismatch des algorithmes NLMS et NLMF. Nous remarquons dans cette figure que notre algorithme proposé maintient un niveau très faible de SM qui a pu atteindre une valeur de -30dB mais en restant varié entre les valeurs -20dB et -30dB.

La figure (4.20) représente la variation de l'erreur quadratique moyenne (EQM) des algorithmes NLMS et NLMF. Nous remarquons que l'algorithme NLMS converge puis reste stable vers une valeur de l'EQM qui atteint la valeur -60dB et reste instable alors que l'algorithme proposé converge jusqu'il atteint la valeur de -80dB.

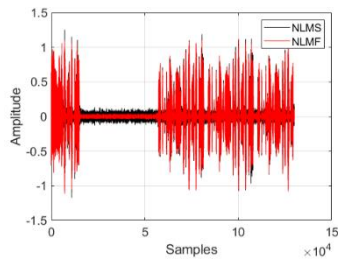


Fig4.16- Evaluation temporelle entre le NLMS et le NLMF

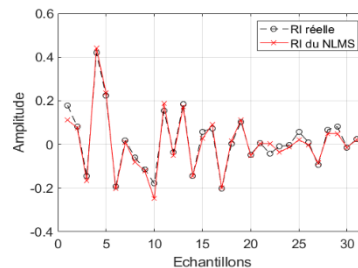


Fig4.17- Comparaison entre RI réelle et RI du NLMS

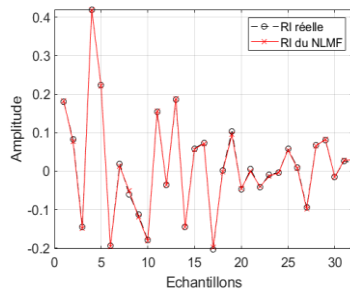


Fig4.18- Comparaison entre RI réelle et RI du NLMF

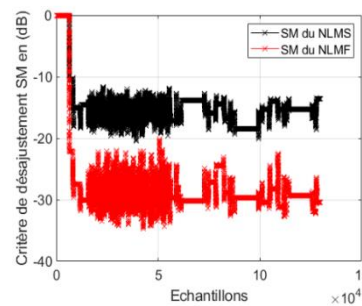


Fig4.19- Critère SM

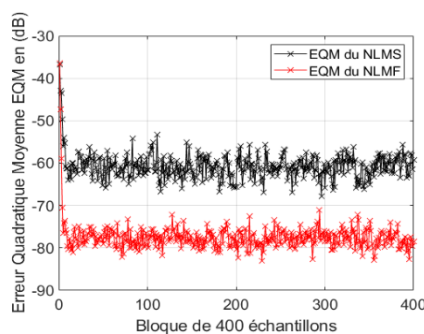


Fig20- Critère EQM Du NLMS ET NLMF

4.5.4 L'effet du *RSB* avec une taille de filtre $L = 128$

En utilisant toujours un bruit blanc, et pour une longueur de filtre fixée à $L = 128$, nous avons varié le *RSB* d'entrée.

- $L= 128$ et $RSB=3$

La figure (4.21) représente une évolution temporelle où nous faisons une comparaison entre l'algorithme NLMS et le nouvel algorithme NLMF. Nous remarquons dans la figure une qualité meilleure du signal de sortie de l'algorithme NLMF proposé par rapport à NLMS.

La figure (4.22) représente l'évolution des réponses impulsionnelles réelles avec les réponses impulsionnelles de l'algorithme NLMS. Nous remarquons dans la figure (4.22) que les réponses impulsionnelles réelles et les réponses impulsionnelles de l'algorithme NLMS sont dispersées.

La figure (4.23) représente l'évolution des réponses impulsionnelles réelles avec les réponses impulsionnelles de l'algorithme NLMF. Nous remarquons dans la figure (4.23) que les réponses impulsionnelles réelles et les réponses impulsionnelles du NLMF sont superposées.

La figure (4.24) représente le critère System Mismatch des algorithmes NLMS et NLMF. Nous remarquons dans cette figure que notre algorithme proposé maintient un niveau très faible de SM qui a pu atteindre une valeur de -25dB mais en restant varié entre les valeurs -25dB et -30dB.

La figure (4.25) représente la variation de l'erreur quadratique moyenne (EQM) des algorithmes NLMS et NLMF. Nous remarquons que l'algorithme NLMS converge puis reste stable vers une valeur de l'EQM de -60dB et reste instable dans l'intervalle [-60dB ; -70dB] alors que l'algorithme proposé converge jusqu'il dépasse la valeur de -80dB et atteint presque la valeur -90dB.

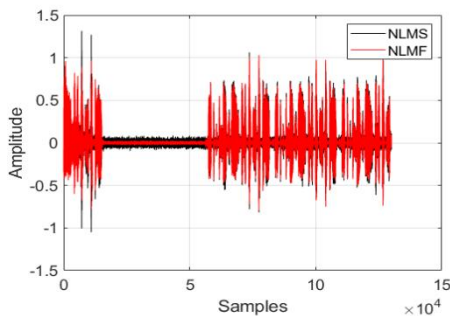


Fig4.21- Evaluation temporelle entre le NLMS et le NLMF

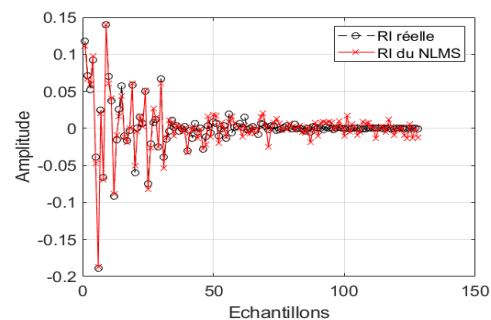


Fig4.22- Comparaison entre RI réelle et RI du NLMS

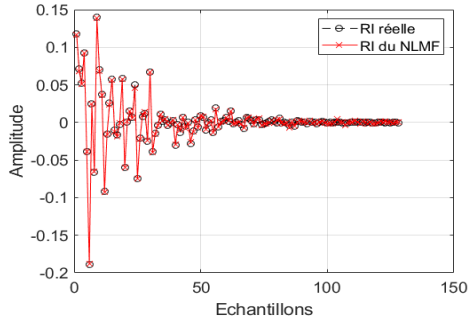


Fig4.23- Comparaison entre RI réelle et RI du NLMF

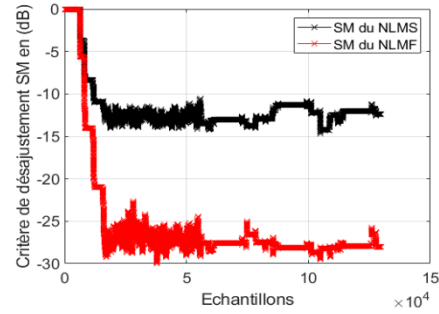


Fig4.24- Critère SM

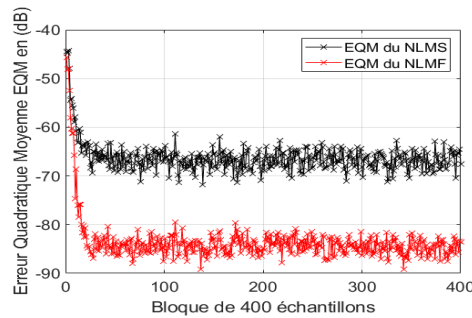


Fig25-Critère EQM du NLMS ET NLMF

- **L=128 et RSB=0**

La figure (4.26) représente une évolution temporelle où nous faisons une comparaison entre l'algorithme NLMS et le nouvel algorithme NLMF. Nous remarquons dans la figure une qualité acceptable du signal de sortie de l'algorithme NLMF proposé par rapport à NLMS.

La figure (4.27) représente l'évolution des réponses impulsionnelles réelles avec les réponses impulsionnelles de l'algorithme NLMS. Nous remarquons dans la figure (4.27) que les réponses impulsionnelles réelles et les réponses impulsionnelles de l'algorithme NLMS sont dispersées.

La figure (4.28) représente l'évolution des réponses impulsionnelles réelles avec les réponses impulsionnelles de l'algorithme NLMF. Nous remarquons dans la figure (4.28) que les réponses impulsionnelles réelles et les réponses impulsionnelles du NLMF sont superposées.

La figure (4.29) représente le critère System Mismatch des algorithmes NLMS et NLMF. Nous remarquons dans cette figure que notre algorithme proposé maintient un niveau très faible de SM qui a pu atteindre une valeur de -25dB mais en restant varié entre les valeurs -25dB et -30dB.

La figure (4.30) représente la variation de l'erreur quadratique moyenne (EQM) des algorithmes NLMS et NLMF. Nous remarquons que l'algorithme NLMS converge vers

une valeur EQM qui dépasse les -60dB puis reste instable dans l'intervalle [-60dB ; -70dB] alors que l'algorithme proposé converge jusqu'il atteigne la valeur -80dB.

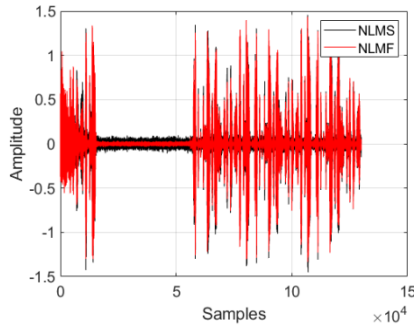


Fig4.26- Evaluation temporelle entre le NLMS et le NLMF

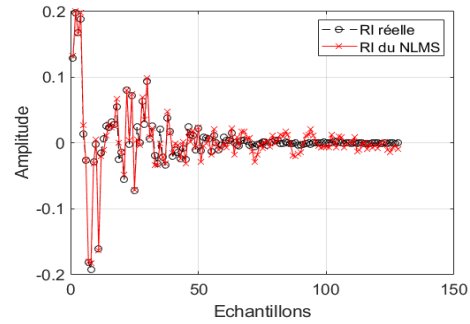


Fig4.27- Comparaison entre RI réelle et RI du NLMS

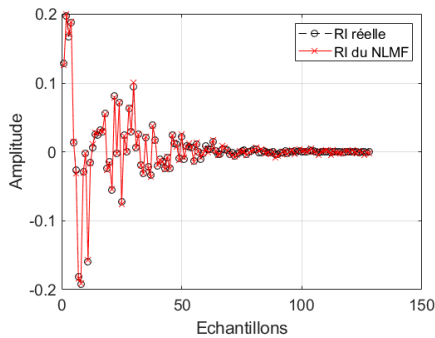


Fig4.28- Comparaison entre RI réelle et RI du NLMF

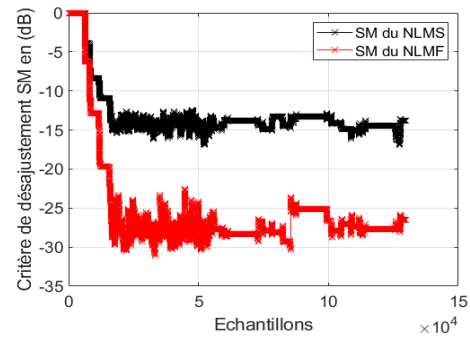


Fig4.29- Critère SM

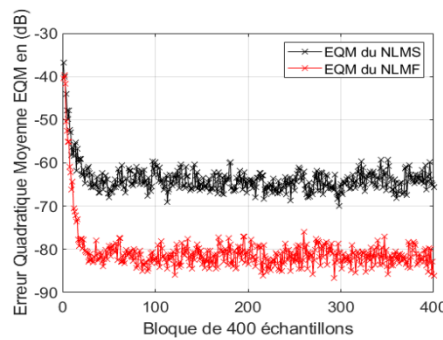


Fig30- Critère EQM du NLMS ET NLMF

- **L=128 et RSB= -3dB**

La figure (4.31) représente une évolution temporelle où nous faisons une comparaison entre l'algorithme NLMS et le nouvel algorithme NLMF. Nous remarquons dans la figure une qualité presque parfaite du signal de sortie de l'algorithme NLMF proposé par rapport à NLMS.

La figure (4.32) représente l'évolution des réponses impulsionnelles réelles avec les réponses impulsionnelles de l'algorithme NLMS. Nous remarquons dans la figure

(4.32) que les réponses impulsionnelles réelles et les réponses impulsionnelles de l'algorithme NLMS sont presque superposées.

La figure (4.33) représente l'évolution des réponses impulsionnelles réelles avec les réponses impulsionnelles de l'algorithme NLMF. Nous remarquons dans la figure (4.33) que les réponses impulsionnelles réelles et les réponses impulsionnelles du NLMF sont superposées.

La figure (4.34) représente le critère System Mismatch des algorithmes NLMS et NLMF. Nous remarquons dans cette figure que notre algorithme proposé maintient un niveau très faible de SM qui a pu atteindre une valeur de -25dB mais en restant varié entre les valeurs -25dB et -30dB.

La figure (4.35) représente la variation de l'erreur quadratique moyenne (EQM) des algorithmes NLMS et NLMF. Nous remarquons que l'algorithme NLMS converge vers une valeur EQM à -60dB puis reste instable alors que l'algorithme proposé converge jusqu'il atteigne la valeur -80dB.

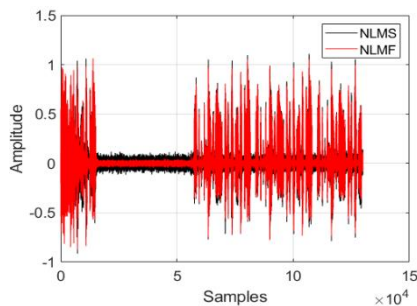


Fig4.31- Evaluation temporelle entre le NLMS et le NLMF

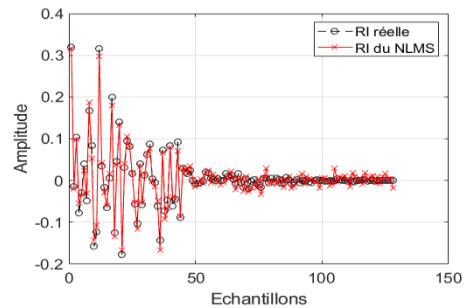


Fig4.32- Comparaison entre RI réelle et RI du NLMS

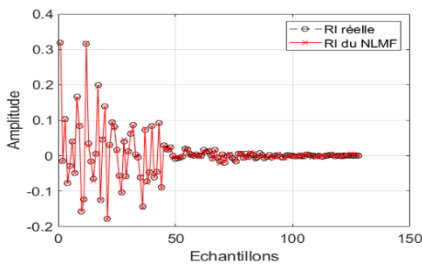


Fig4.33- Comparaison entre RI réelle et RI du NLMF

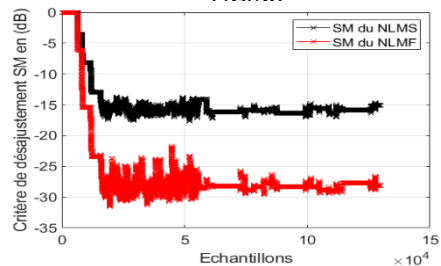


Fig4.34- Critère SM

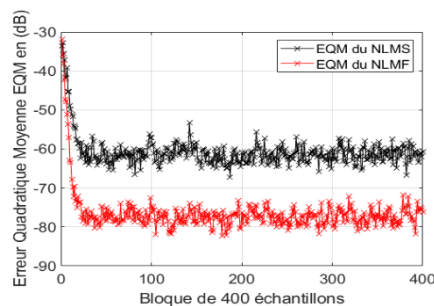


Fig35-Critère EQM du NLMS ET NLMF

Figures (4.6), (4.11), (4.16), (4.21), (4.26), (4.31) montrent la qualité satisfaisante que délivre notre algorithme proposé (NLMF) à la sortie.

Les figures (4.8), (4.13), (4.18), (4.23), (4.28) et (4.33) montre clairement que les réponses impulsionnelles du NLMF sont très bien superposées sur les réponses impulsionnelles réelles.

Il est clairement démontré par les figures (4.9), (4.14), (4.19), (4.24), (4.29) et (4.34) que le nouvel algorithme NLMF surpasse l'algorithme NLMS en termes de vitesse de convergence. De plus, ces mêmes figures mettent en évidence que le comportement du NLMF est plus favorable pour les filtres de petite taille par rapport aux filtres de grande taille. D'après les résultats présentés dans les figures (4.10), (4.15), (4.20), (4.25), (4.30) et (4.35) il est observé que le SegMSE du NLMF automatique reste presque constant pour les tailles de filtre $L=32$ et $L=128$.

4.6 Résultat de simulation en changeant un paramètre primordial

Dans cette partie, nous allons présenter les résultats de simulation avec le même type de bruit mais en changeant un seul paramètre qui est le coefficient α qui permet de contrôler l'interférence du bruit et de la parole.

Pour la simulation, on a fixé les paramètres comme suit :

Tableau 4.2 – Paramètre de Simulation

Paramètre	Signification	Valeur
L	Longueur du filtre	32
N	Nombre d'itérations	130000
Fe	Fréquence d'échantillonnage	8000
RSB1=RSB2	Rapport Signal Bruit	3dB, 0dB et -3dB
α	Coefficient de contrôle d'interférence	0.5 et 0.2
μ	Pas d'adaptation	0.9

En utilisant toujours un bruit blanc, et pour une longueur de filtre fixée à $L = 32$, nous avons varié le RSB et le coefficient α .

- $L=32$ et $RSB1=RSB2=3$ et $\alpha=0.5$

La figure (4.36) représente une évolution temporelle où nous faisons une comparaison entre l'algorithme NLMS et le nouvel algorithme NLMF. Nous remarquons dans la figure une qualité meilleure du signal de sortie de l'algorithme NLMF proposé par rapport à NLMS.

La figure (4.37) représente l'évolution des réponses impulsionnelles réelles avec les réponses impulsionnelles de l'algorithme NLMS. Nous remarquons dans la figure (4.2) que les réponses impulsionnelles réelles et les réponses impulsionnelles de l'algorithme NLMS sont presque superposées.

La figure (4.38) représente l'évolution des réponses impulsionnelles réelles avec les réponses impulsionnelles de l'algorithme NLMF. Nous remarquons dans la figure (4.3) que les réponses impulsionnelles réelles et les réponses impulsionnelles du NLMF sont superposées.

La figure (4.39) représente le critère System Mismatch des algorithmes NLMS et NLMF. Nous remarquons dans cette figure que notre algorithme dépasse la valeur -30dB puis il reste constant.

La figure (4.40) représente la variation de l'erreur quadratique moyenne (EQM) des algorithmes NLMS et NLMF.

Nous remarquons que l'algorithme NLMS converge puis reste stable vers une valeur de l'EQM qui dépasse les -60dB et touche -70dB à quelques reprises alors que l'algorithme proposé dépasse la valeur de -80dB et converge vers la valeur -90dB.

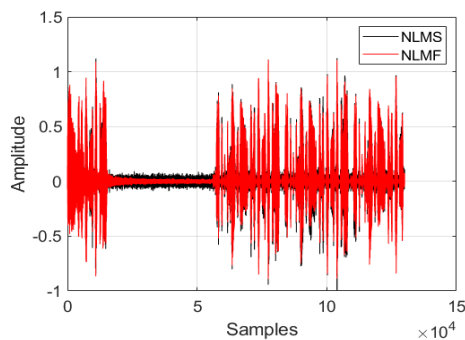


Fig4.36- Evaluation temporelle entre le NLMS et le NLMF

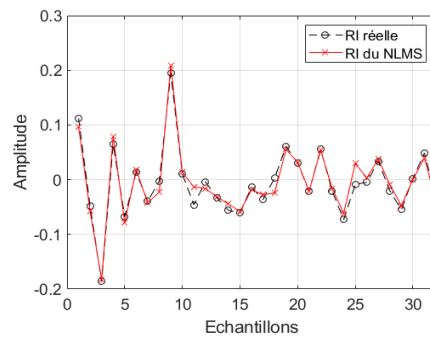


Fig4.37- Comparaison entre RI réelle et RI du NLMS

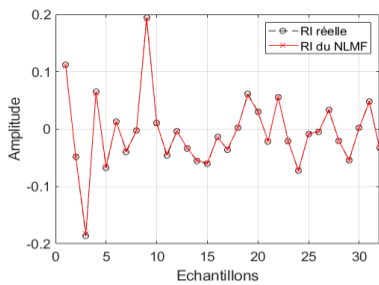


Fig4.38- Comparaison entre RI réelle et RI du NLMF

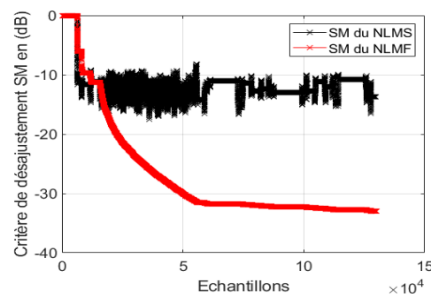


Fig4.39- Critère SM

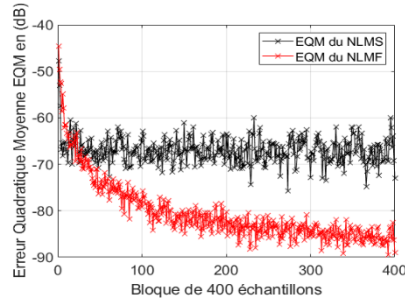


Fig4.40-Critère EQM du NLMS ET NLMF

- $L=32$ et $RSB1=RSB2=0$ et $\alpha=0.5$

La figure (4.41) représente une évolution temporelle où nous faisons une comparaison entre l’algorithme NLMS et le nouvel algorithme NLMF. Nous remarquons dans la figure une qualité meilleure du signal de sortie de l’algorithme NLMF par rapport à NLMS.

La figure (4.42) représente l’évolution des réponses impulsionnelles réelles avec les réponses impulsionnelles de l’algorithme NLMS. Nous remarquons dans la figure (4.42) que les réponses impulsionnelles réelles et les réponses impulsionnelles de l’algorithme NLMS sont presque superposées.

La figure (4.43) représente l’évolution des réponses impulsionnelles réelles avec les réponses impulsionnelles de l’algorithme NLMF. Nous remarquons dans la figure (4.43) que les réponses impulsionnelles réelles et les réponses impulsionnelles du NLMF sont superposées.

La figure (4.44) représente le critère System Mismatch des algorithmes NLMS et NLMF. Nous remarquons dans cette figure que notre algorithme dépasse la valeur -30dB puis il reste constant en atteignant la valeur -35dB.

La figure (4.45) représente la variation de l’erreur quadratique moyenne (EQM) des algorithmes NLMS et NLMF. Nous remarquons que l’algorithme NLMS converge puis reste stable vers une valeur de l’EQM qui dépasse les -60dB et touche -70dB à quelques reprises alors que l’algorithme proposé NLMF atteint la valeur de -80dB.

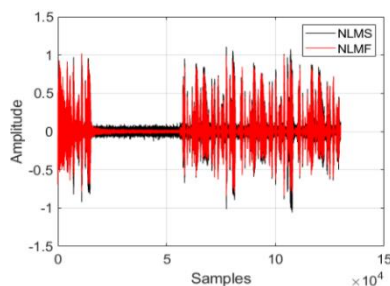


Fig4.41 - Evolution Temporelle entre NLMS et NLMF

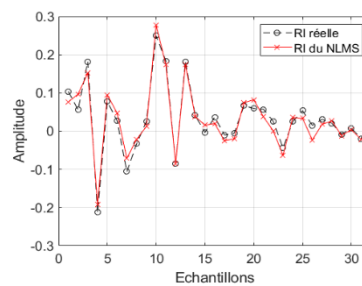


Fig4.42 – Comparaisons entre RI réelles et RI du NLMS

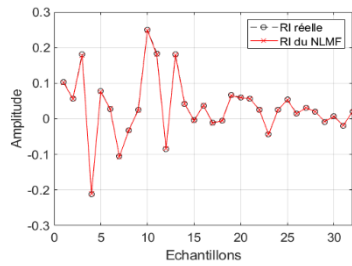


Fig4.43- Comparaison entre RI réelle et RI du NLMF

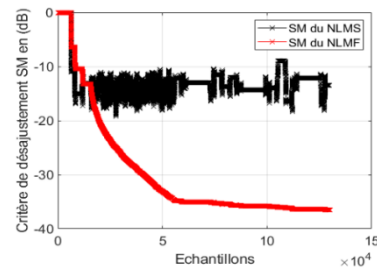


Fig4.44 – Critère SM

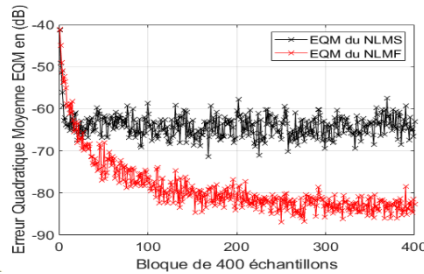


Fig4.45-Critère EQM du NLMS ET NLMF

- **$L=32$ et $RSB1=RSB2=-3dB$ et $\alpha=0.5$**

La figure (4.46) représente une évolution temporelle où nous faisons une comparaison entre l'algorithme NLMS et le nouvel algorithme NLMF. Nous remarquons dans la figure une qualité meilleure du signal de sortie de l'algorithme NLMF proposé par rapport à NLMS.

La figure (4.47) représente l'évolution des réponses impulsionnelles réelles avec les réponses impulsionnelles de l'algorithme NLMS. Nous remarquons dans la figure (4.2) que les réponses impulsionnelles réelles et les réponses impulsionnelles de l'algorithme NLMS sont presque superposées.

La figure (4.48) représente l'évolution des réponses impulsionnelles réelles avec les réponses impulsionnelles de l'algorithme NLMF. Nous remarquons dans la figure (4.3) que les réponses impulsionnelles réelles et les réponses impulsionnelles du NLMF sont superposées.

La figure (4.49) représente le critère System Mismatch des algorithmes NLMS et NLMF. Nous remarquons dans cette figure que notre algorithme dépasse la valeur -40dB puis il reste constant.

La figure (4.50) représente la variation de l'erreur quadratique moyenne (EQM) des algorithmes NLMS et NLMF. Nous remarquons que l'algorithme NLMS converge puis reste stable vers une valeur de l'EQM qui touche les -60dB alors que l'algorithme proposé dépasse la valeur de -80dB et converge vers la valeur -90dB

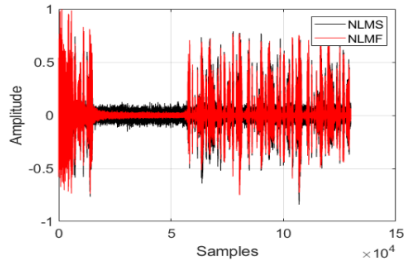


Fig4.46 - Evolution Temporelle entre NLMS et NLMF

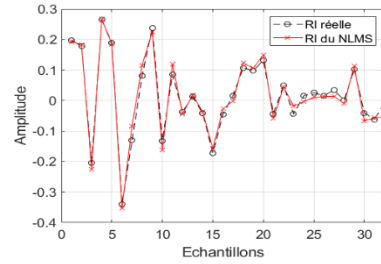


Fig4.47 – Comparaisons entre RI réelles et RI du NLMS

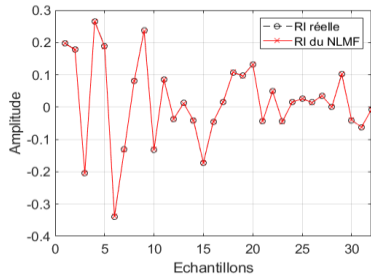


Fig4.48- Comparaison entre RI réelle et RI du NLMF

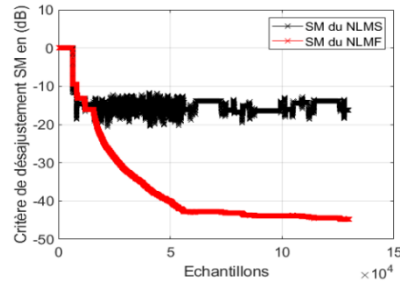


Fig4.49 – Critère SM

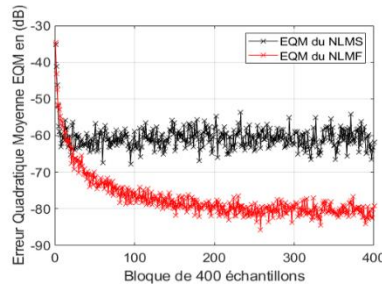


Fig4.50-Critère EQM du NLMS ET NLMF

- $L=32$ et $RSB1=RSB2=3dB$ et $\alpha=0.2$

La figure (4.51) représente une évolution temporelle où nous faisons une comparaison entre l'algorithme NLMS et le nouvel algorithme NLMF. Nous remarquons dans la figure une qualité meilleure du signal de sortie de l'algorithme NLMF proposé par rapport à NLMS.

La figure (4.52) représente l'évolution des réponses impulsionnelles réelles avec les réponses impulsionnelles de l'algorithme NLMS. Nous remarquons dans la figure (4.2) que les réponses impulsionnelles réelles et les réponses impulsionnelles de l'algorithme NLMS sont presque superposées.

La figure (4.53) représente l'évolution des réponses impulsionnelles réelles avec les réponses impulsionnelles de l'algorithme NLMF. Nous remarquons dans la figure (4.3) que les réponses impulsionnelles réelles et les réponses impulsionnelles du NLMF sont superposées.

La figure (4.54) représente le critère System Mismatch des algorithmes NLMS et NLMF. Nous remarquons dans cette figure que notre algorithme dépasse la valeur -25dB puis il reste constant.

La figure (4.55) représente la variation de l'erreur quadratique moyenne (EQM) des algorithmes NLMS et NLMF. Nous remarquons que l'algorithme NLMS converge puis reste stable vers une valeur de l'EQM qui dépasse les -60dB et atteint -70dB à quelques reprises alors que l'algorithme proposé dépasse la valeur de -80dB et converge vers la valeur -85dB.

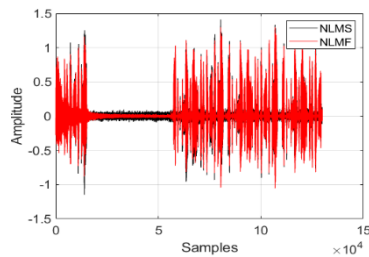


Fig4.51 - Evolution Temporelle entre NLMS et NLMF

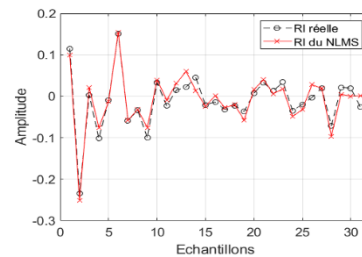


Fig4.52 - Comparaisons entre RI réelles et RI du NLMS

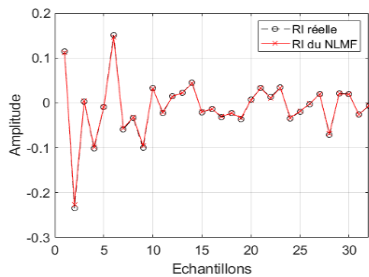


Fig4.53 - Comparaison entre RI réelle et RI du NLMF

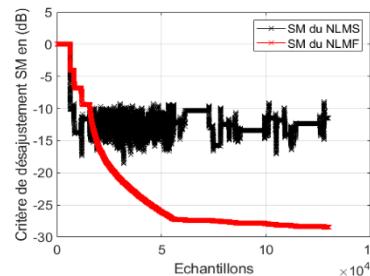


Fig4.54 - Critère SM

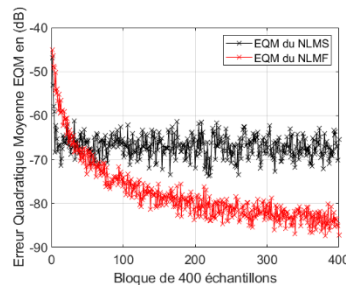


Fig4.55 - Critère EQM du NLMS ET NLMF

- $L=32$ et $RSB1=RSB2=0dB$ et $\alpha=0.2$

La figure (4.56) représente une évolution temporelle où nous faisons une comparaison entre l'algorithme NLMS et le nouvel algorithme NLMF. Nous remarquons dans la figure une qualité meilleure du signal de sortie de l'algorithme NLMF proposé par rapport à NLMS.

La figure (4.57) représente l'évolution des réponses impulsionnelles réelles avec les réponses impulsionnelles de l'algorithme NLMS. Nous remarquons dans la figure (4.2) que les réponses impulsionnelles réelles et les réponses impulsionnelles de l'algorithme NLMS sont presque superposées.

La figure (4.58) représente l'évolution des réponses impulsionnelles réelles avec les réponses impulsionnelles de l'algorithme NLMF. Nous remarquons dans la figure (4.3) que les réponses impulsionnelles réelles et les réponses impulsionnelles du NLMF sont superposées.

La figure (4.59) représente le critère System Mismatch des algorithmes NLMS et NLMF. Nous remarquons dans cette figure que notre algorithme dépasse la valeur -30dB atteignant la valeur de -35dB puis il reste constant.

La figure (4.60) représente la variation de l'erreur quadratique moyenne (EQM) des algorithmes NLMS et NLMF. Nous remarquons que l'algorithme NLMS converge puis reste stable vers une valeur de l'EQM qui dépasse les -60dB et touche -70dB à quelques reprises alors que l'algorithme NLMF converge vers les -80dB.

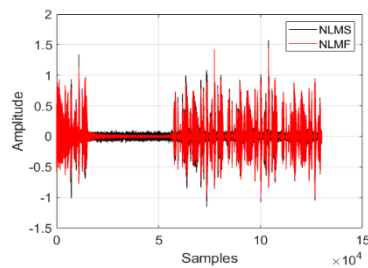


Fig4.56 - Evolution Temporelle entre NLMS et NLMF

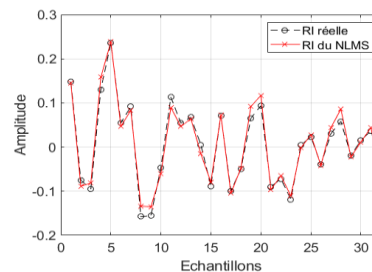


Fig4.57 - Comparaisons entre RI réelles et RI du NLMS

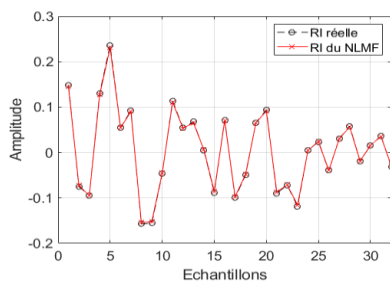


Fig4.58 - Comparaison entre RI réelle et RI du NLMF

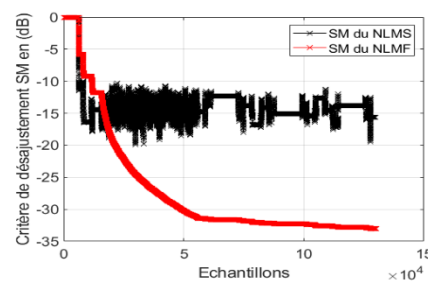


Fig4.59 - Critère SM

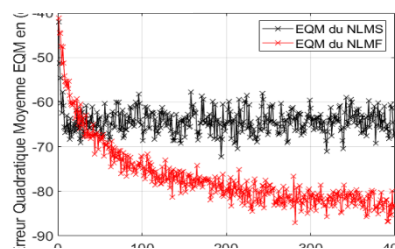


Fig4.60 - Critère EQM du NLMS ET NLMF

- $L=32$ et $RSB1=RSB2=-3dB$ et $\alpha=0.2$

La figure (4.61) représente une évolution temporelle où nous faisons une comparaison entre l'algorithme NLMS et le nouvel algorithme NLMF. Nous remarquons dans la figure une qualité meilleure du signal de sortie de l'algorithme NLMF proposé par rapport à NLMS.

La figure (4.62) représente l'évolution des réponses impulsionnelles réelles avec les réponses impulsionnelles de l'algorithme NLMS. Nous remarquons dans la figure (4.2) que les réponses impulsionnelles réelles et les réponses impulsionnelles de l'algorithme NLMS sont presque superposées.

La figure (4.63) représente l'évolution des réponses impulsionnelles réelles avec les réponses impulsionnelles de l'algorithme NLMF. Nous remarquons dans la figure (4.3) que les réponses impulsionnelles réelles et les réponses impulsionnelles du NLMF sont superposées.

La figure (4.64) représente le critère System Mismatch des algorithmes NLMS et NLMF. Nous remarquons dans cette figure que notre algorithme dépasse la valeur -30dB puis il reste constant.

La figure (4.65) représente la variation de l'erreur quadratique moyenne (EQM) des algorithmes NLMS et NLMF. Nous remarquons que l'algorithme NLMS converge puis reste stable vers une valeur de l'EQM qui dépasse les -60dB et touche -70dB à quelques reprises alors que l'algorithme proposé dépasse la valeur de -80dB et converge vers la valeur -85dB.

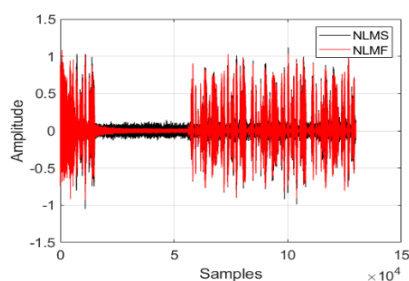


Fig4.61 - Évolution Temporelle entre NLMS et NLMF

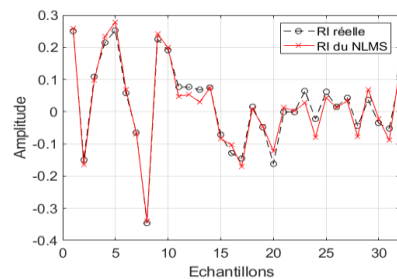


Fig4.62 – Comparaisons entre RI réelles et RI du NLMS

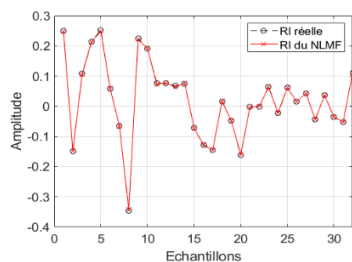


Fig4.63 – Comparaisons entre RI réelles et RI du NLMF

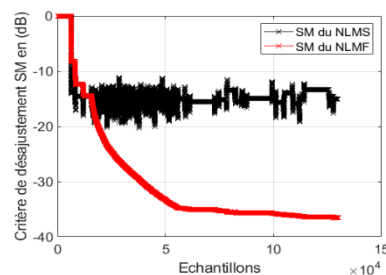


Fig4.64 – Critère SM

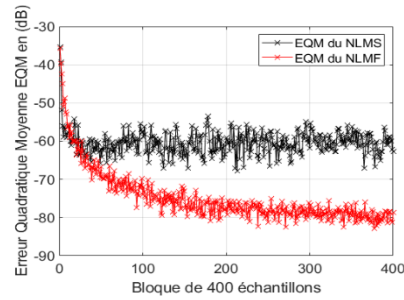


Fig4.65 – Critère EQM

Les figures (4.40), (4.45), (4.50), (4.55), (4.60), (4.65) nous montrent que l’algorithme perd sa stabilité pour $\alpha=0.5$ et $\alpha=0.2$. Nous pouvons conclure que lorsque la valeur de α se situe entre 0,9 et 0,99, l'algorithme offre de meilleurs résultats en termes de stabilité. Cependant, lorsque α est égal à 1, l'algorithme ne fonctionne pas.

4.7 Conclusion

Au cours de ce chapitre, nous avons effectué une comparaison entre le NLMF automatique et le NLMS. Nous avons présenté nos résultats en utilisant un signal de parole dans un environnement bruité (bruit blanc) et en variant les paramètres de simulation. Les résultats de nos simulations ont clairement démontré les performances favorables de l'algorithme proposé en termes de convergence et de réduction du bruit. Ces résultats témoignent de la capacité de l'algorithme à atteindre rapidement la solution optimale du filtrage adaptatif.

Conclusion générale

L'objectif principal de ce projet de fin d'études était de résoudre les problèmes causés par le bruit acoustique lors des transmissions du signal utile en utilisant différentes techniques de réduction du bruit. Nous avons commencé par une étude théorique sur le filtrage et les algorithmes de filtrage adaptatif tels que LMS et NLMS. Ensuite, nous avons exploré les principales méthodes et techniques de réduction du bruit, en utilisant des acquisitions mono-capteur, bi-capteurs et multi-capteurs.

Dans une deuxième partie, nous avons présenté le développement de notre algorithme proposé, le LMF, et son fonctionnement basé sur la détection du rapport signal sur bruit (RSB) comme paramètre clé. En effectuant plusieurs tests et en évaluant les critères de performance tels que l'EQM (Erreur Quadratique Moyenne) et le SM (System Mismatch) pour différentes valeurs des paramètres de simulation (longueur du filtre, type de bruit, coefficient de contrôle d'interférence) à l'aide du logiciel MATLAB, nous avons pu démontrer que notre algorithme surpassait l'algorithme de gradient stochastique NLMS en termes de convergence, de stabilité et de réduction du bruit.

En conclusion, ce travail nous a permis d'explorer une facette du domaine de la réduction du bruit, qui est un domaine vaste et en constante évolution.

Bibliographie

- [1] F. Naer, S. RABEE, "Un algorithme efficace pour le rehaussement du signal de parole", Mémoire de master, Université de Blida1, 2021-2022.
- [2] A. S. Rezak, A. Koulougli, "Proposition d'un nouvel algorithme adaptatif pour la réduction automatique du bruit acoustique", mémoire de Master, Université de Blida1, 2019-2020.
- [3] N. BELALIA, M.A. HADJALLA, "Développement d'un algorithme RNQ efficace pour la réduction bi-capteurs du bruit acoustique", mémoire de Master, Université de Blida1, 2021-2022.
- [4] F. ZAHI, C. CHIHEB, "Etude et simulation de l'algorithme NLMS dans l'annulation de l'écho acoustique des salles", mémoire de Master, Université de Blida1, 2019-2020.
- [5] F.T. NAIT AOURAGH, B.AIT ALI, " Algorithme Backward P-NLMS à pas d'adaptation variables en sous-bande pour la réduction du bruit et le rehaussement de la parole ", mémoire de Master, Université de Blida1, 2021-2022.
- [6] I. TCHIALI, A. BOUAICHAOUI, " Rehaussement des images bruitées par une approche adaptative ", mémoire de Master, Université de Blida1, 2015-2016.
- [7] F. BOUDERBALA, O. CHABOUNI, " Proposition d'un nouvel algorithme rapide à deux canaux pour la réduction du bruit dans les systèmes téléphoniques à main-libres ", mémoire de Master, Université de Blida1, 2017-2018.
- [8] D.F.EKE, " Méthode d'identification en virgule fixe d'un modèle non linéaire basé sur les algorithmes génétiques", mémoire de Master, Université Du Québec, 2007-2008.

- [9] J.F. Bercher, P. Jardin, “ Introduction au filtrage adaptatif“, cours de Master, ESIEE de Paris, 2003.
- [10] Eweda Eweda, “ Global Stabilization of the Least Mean Fourth Algorithm“, IEEE transactions on signal processing, vol. 60, no. 3, March 2012.
- [11] A. KEDJAR, M.A. RAMDANE, “ Algorithmes rapides pour l’identification de systèmes parcimonieux “, mémoire de Master, Université de Blida1, 2017-2018.
- [12] L. CHERGUI, “ Débruitage de la parole par des méthodes basées sur les transformées discrètes “, mémoire de Master, Université de Sétif 1, 2016-2017.
- [13] K. NABGHA, M. KHANOUSSI, “ Bruit et filtrage : filtre de Wiener “, mémoire de Master, Université de Adrar, 2017-2018.
- [14] A. ZERGUINE, “Convergence and steady-state analysis of the normalized least mean fourth algorithm “Electrical Engineering Department, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia, February 2006.

