

**UNIVERSITE BLIDA-1**

**Faculté de Technologie**  
Département d'Electronique

**THESE DE DOCTORAT**

En Electronique  
Spécialité : Instrumentation

**METHODES DE REDUCTION DE BRUIT MULTI-CANAL POUR  
TELEPHONIE MOBILE**

Par

**Mohamed ZEROUALI**

Devant le jury composé de :

<b>Nom et Prénom</b>	<b>Grade</b>	<b>Etablissement</b>	<b>Qualité</b>
YKHLEF Farid	Pr	Université Blida 1	Président
BOUHEDDA Mounir	Pr	Université Médéa	Examineur
BENDOUMIA Redha	MCA	Université Blida 1	Examineur
DJENDI Mohamed	Pr	Université Blida 1	Directeur de Thèse

Blida, 2024

## RESUME

Dans le cadre de cette thèse, nous nous sommes intéressés aux méthodes de réduction du bruit multi-canaux. Dans ce contexte, quatre nouveaux algorithmes ont été développés. Tout d'abord, nous nous sommes penchés sur le cas bi-capteurs, pour lequel nous avons proposé un algorithme bi-capteurs (TC-PU-SFTF) dédié aux systèmes de séparation aveugle des sources (BSS), implémenté dans la structure directe (FBSS). Cet algorithme repose sur la mise à jour partielle des filtres adaptatifs de l'algorithme (TC-SFTF) récemment développé, selon un choix de sélection basé sur les coefficients de gain de Kalman. Ce choix a permis de réduire considérablement la complexité de calcul sans dégrader la vitesse de convergence de l'algorithme. Ensuite, nous nous sommes tournés vers le cas multi-canaux, pour lequel trois nouveaux algorithmes multi-canaux ont été développés. Le deuxième algorithme (MC-FNLMS) est une version multi-canaux rapide et de faible complexité de l'algorithme FNLMS. Le troisième et le quatrième algorithme sont deux versions en sous-bandes (SB-MC-FNLMS) du deuxième algorithme proposé (MC-FNLMS), implémentés dans deux structures en sous-bandes différentes ; la structure multi-bande et la structure en boucle ouverte. Les résultats des simulations présentés dans ce travail, confirment la supériorité des algorithmes proposés pour réduire le bruit acoustique dans le contexte multi-canaux.

**Mots clés :** Filtrage adaptatif ; TC-PU-SFTF ; BSS ; MC-FNLMS ; SB-MC-FNLMS ; Rapport signal sur bruit ; Erreur quadratique moyenne.

# ABSTRACT

In this thesis, we focused on multi-channel noise reduction methods. In this context, four new algorithms are developed. Firstly, we focused on the two-sensor case, for which we proposed a two-channel algorithm (TC-PU-SFTF) for blind source separation systems, implemented in forward structure (FBSS). This algorithm relies on the partial update of two adaptive filters of the recently developed algorithm (TC-SFTF), according to a selection choice based on the Kalman gain coefficients. This choice of selection coefficients has significantly reduced the computational complexity without degrading the convergence rate of the algorithm. Next, we turned to the multi-channel case, for which three new algorithms were developed. The second algorithm (MC-FNLMS) has fast convergence speed and low computational complexity. The third and the fourth algorithms are two sub-band versions (SB-MC-FNLMS) of the second proposed algorithm (MC-FNLMS), implemented in two different sub-band structures; the multi-band structure and the open-loop structure. The simulation results presented in this study confirm the superiority of the proposed algorithms for acoustic noise reduction in the multi-sensor context.

**Keywords:** Adaptive filtering; TC-PU-SFTF; BSS; MC-FNLMS; SB-MC-FNLMS; Signal to noise ratio; Mean squared error.

## ملخص

ركزنا في هذه الأطروحة على طرق تقليل الضوضاء متعدد القنوات. وفي هذا السياق، تم تطوير أربع خوارزميات جديدة. أولاً، ركزنا على حالة الملتقطين، والتي اقترحنا من أجلها خوارزمية ثنائية القناة (TC-PU-SFTF) لأنظمة فصل المصادر والتي يتم تنفيذها في البنية المباشرة (FBSS). تعتمد هذه الخوارزمية على التحديث الجزئي لإثنين من المرشحات التكيفية للخوارزمية المطورة حديثاً (TC-SFTF)، بناءً على معاملات متجه كالمان. مكنا التحديث الجزئي اعتماداً على متجه كالمان إلى تقليل التعقيد الحسابي بشكل كبير دون التقليل من سرعة تقارب الخوارزمية. بعد ذلك، انتقلنا إلى حالة متعددة القنوات، حيث تم تطوير ثلاث خوارزميات جديدة. تتميز الخوارزمية الثانية (MC-FNLMS) بسرعة تقارب عالية وتعقيد حسابي منخفض. الخوارزميات الثالثة والرابعة عبارة عن نسختين من النطاق الطيفي النحتي (SB-MC-FNLMS) من الخوارزمية المقترحة الثانية (MC-FNLMS)، ويتم تنفيذها في هيكلين مختلفين؛ هيكل متعدد النطاقات وهيكل الحلقة المفتوحة. تؤكد نتائج المحاكاة المقدمة في هذه الدراسة تفوق الخوارزميات المقترحة لتقليل الضوضاء الصوتية في سياق تقليل الضوضاء متعدد الملتقطات.

**الكلمات المفتاحية:** التصفية التكيفية؛ MC-FNLMS؛ SB-MC-FNLMS؛ نسبة الإشارة إلى نسبة الضوضاء؛ خطأ تربيع المتوسط.

# REMERCIEMENTS

En ce moment significatif de mon parcours académique, je tiens à exprimer ma profonde gratitude envers toutes les personnes qui m'ont soutenu et inspiré tout au long de ce cursus de recherche. Leurs contributions ont été inestimables.

Je tiens tout d'abord à exprimer ma gratitude envers mon directeur de thèse, professeur DJENDI Mohamed, pour sa guidance experte, son soutien constant et ses précieux conseils tout au long de ce parcours de recherche scientifique. Sa vision éclairée et son dévouement ont été des piliers essentiels de ce travail.

Je souhaite également remercier chaleureusement les membres de mon jury de thèse, professeur Farid YKHLEF, professeur Mounir BOUHEDDA, et docteur Redha BENDOUMIA pour leur temps, leur expertise et leurs commentaires constructifs qui ont contribué à enrichir ce travail.

Mes remerciements vont également à tous les professeurs du département d'électronique de l'université de Blida 1 qui ont enrichi mes connaissances tout au long de ma carrière d'étude universitaire.

Je tiens à exprimer ma gratitude envers mes parents pour leur soutien indéfectible, leur amour et leur encouragement tout au long de ce parcours académique. Leur soutien inconditionnel a été ma source de force et de motivation.

Enfin, je tiens à exprimer ma reconnaissance envers toutes les personnes qui, de près ou de loin, ont contribué à la réalisation de cette thèse. Leur soutien a été précieux et a énormément contribué à la réussite de ce travail.

Merci du fond du cœur à tous ceux qui ont rendu cette réalisation possible.

ZEROUALI Mohamed.

## TABLE DES MATIERES

RESUME

ABSTRACT

ملخص

REMERCIEMENTS

TABLE DES MATIERES

LISTE DES FIGURES ET DES TABLEAUX

Liste des figures

Liste des tableaux

**INTRODUCTION** ..... 1

**CHAPITRE 1 : FILTRAGE ADAPTATIF, ALGORITHMES ET REDUCTION DE BRUIT MULTI-CANAUX** ..... 4

1.1. Introduction ..... 4

1.2. Filtre de Wiener ..... 5

1.2.1. Principe d'orthogonalité ..... 6

1.2.2. Équations de Wiener-Hopf et filtre optimal de Wiener ..... 8

1.2.3. Expression de la fonction cout minimal *J<sub>min</sub>* ..... 9

1.3. Filtrage adaptatif, algorithmes adaptatifs et applicaions. .... 10

1.3.1. Algorithme des moindres carrés moyennes (LMS)..... 12

1.3.2. Algorithme des moindres carrés moyennes normalisé (NLMS) ..... 13

1.3.3. Algorithme des moindres carrés récursif (RLS) ..... 14

1.3.4. Exemple comparative entre les trois algorithmes LMS, NLMS et RLS ..... 16

1.4. Méthode de réduction de bruit multi-canaux ..... 18

1.4.1. Annulation de bruit par la séparation aveugle des sources ..... 18

1.4.2. Annulation de bruit par le filtrage adaptatif multi-canaux ..... 21

1.5. Conclusion ..... 24

**CHAPITRE 2 : NOUVEAU ALGORITHME BI-CAPTEURS POUR LE REHAUSSEMENT DU SIGNAL DE PAROLE** ..... 25

2.1. Introduction ..... 25

2.2. Algorithme FTF et sa version simplifiée (SFTF) ..... 25

2.3. Dérivation et présentation de l'algorithme proposé (Algorithme 1) ..... 28

2.4. Simulation et résultats ..... 36

2.5. Conclusion ..... 48

**CHAPITRE 3 : NOUVEAU ALGORITHME MULTI-CANAUX POUR LA REDUCTION DE BRUIT** ..... 50

3.1.	Introduction .....	50
3.2.	Les algorithmes FNLMS, MC-APA et MC-NLMS .....	51
3.2.1.	L'algorithme NLMS rapide (FNLMS).....	51
3.2.2.	Algorithmes (MC-APA) et (MC-NLMS) .....	54
3.3.	Algorithme FNLMS multi-canaux proposé (MC-FNLMS) .....	56
3.3.1.	Dérivation de l'algorithme proposé.....	57
3.3.2.	Analyse de la convergence .....	58
3.4.	Simulation et résultats .....	61
3.5.	Conclusion .....	74
<b>CHAPITRE 4 : DEUX NOUVAUX ALGORITHMES MULTI-CANAU</b>		
<b>BANDE.....</b>		<b>75</b>
4.1.	Introduction .....	75
4.2.	Principe du filtrage adaptatif en sous-bande.....	76
4.3.	Les algorithmes en sous-bandes proposés .....	78
4.3.1.	Algorithme MC-SB-FNLMS basé sur la réalisation multi-bandes (Algorithme 3 proposé)79	
4.3.2.	Algorithme MC-SB-FNLMS basé sur la réalisation en boucle ouverte (Algorithme 4 proposé) .....	82
4.4.	Simulation et résultats .....	85
4.4.1.	Identification multi-canaux .....	88
4.4.2.	Application au Beamforming adaptatif GSC .....	96
4.5.	Conclusion .....	99
<b>CONCLUSION GENERALE ET PERSPECTIVES.....</b>		<b>100</b>
Liste des notations et des symboles.....		102
Liste des abréviations .....		103
REFERENCES.....		105

# LISTE DES FIGURES ET DES TABLEAUX

## Liste des figures

<b>Figure 1.1.</b> Représentation sous forme de schéma bloc du problème de filtrage optimal. ....	5
<b>Figure 1.2.</b> Interprétation géométrique de la relation entre la réponse désiré, l'estimation à la sortie du filtre et l'erreur d'estimation à l'optimum. ....	7
<b>Figure 1.3.</b> Surface d'erreur. ....	9
<b>Figure 1.4.</b> Principe de filtrage adaptatif.....	10
<b>Figure 1.5.</b> Les quatre fonctions de base du filtrage adaptatif. ....	11
<b>Figure 1.6.</b> EQM Seg cas d'un signal d'excitation stationnaire.....	17
<b>Figure 1.7.</b> EQM Seg cas d'un signal d'excitation non-stationnaire. ....	18
<b>Figure 1.8.</b> Le mélange convolutive.....	18
<b>Figure 1.9.</b> Mélange convolutive simplifié. ....	19
<b>Figure 1.10.</b> Structure de séparation directe (FBSS). ....	20
<b>Figure 1.11.</b> Structure de séparation récursif (BBSS).....	21
<b>Figure 1.12.</b> Filtrage multi-canaux.....	22
<b>Figure 2.1.</b> Direction de mis à jour de $c_{SFTF,M}(n)$ , $c_{FTF,M}(n)$ et $c_{SFTF,M}(n)$ pour $M = 3$ et $N = 2$ . ....	31
<b>Figure 2.2.</b> Direction de mis à jour de $c_{SFTF,M}(n)$ , $c_{FTF,M}(n)$ et $c_{SFTF,M}(n)$ pour $M = 3$ et $N = 1$ . ....	31
<b>Figure 2.3.</b> Modèle convolutive simplifié de simulation. ....	36
<b>Figure 2.4.</b> Séparateur des sources aveugle, structure direct (FBSS). ....	36
<b>Figure 2.5.</b> Les réponses impulsionnelles $h_1(n)$ et $h_2(n)$ . ....	37
<b>Figure 2.6.</b> Évolution temporelle de signal parole pur et son spectrogramme.....	37
<b>Figure 2.7.</b> Les spectrogrammes des bruits.....	38
<b>Figure 2.8.</b> Les 30 premiers coefficients de prédiction avant et arrière. ....	39
<b>Figure 2.9.</b> L'indicateur BFR pour différents bruits. ....	40
<b>Figure 2.10.</b> Evolution temporelle obtenu par l'algorithme proposé (TC-PU-SFTF). Le bruit est de type : USAS. Bloc d'adaptation $N = 80$ et $RSB$ d'entrée = $0\text{ dB}$ .....	42
<b>Figure 2.11.</b> L'évaluation du désajustement d'algorithme proposé (Algorithme 1) pour différents choix de $N$ , le $RSB$ d'entrée = $0\text{ dB}$ .....	43

<b>Figure 2.12.</b> Comparaison de désajustement pour les algorithmes TC-NLMS, TC-SFTF et l’algorithme proposé (Algorithme 1) pour différents types de bruit, $N = 80$ , $RSB$ d’entré=0 dB. ....	44
<b>Figure 2.13.</b> Fréquence d’adaptation pour chaque coefficient du filtre adaptative $w_1(n)$ pour les 500 première itération avec $N = 80$ , $RSB$ d’entré=0 dB. ....	45
<b>Figure 2.14.</b> Fréquence d’adaptation pour chaque coefficient du filtre adaptative $w_1(n)$ pour les 500 dernière itération, avec $N=M/2$ , $RSB$ d’entré=0dB. ....	45
<b>Figure 2.15.</b> Evaluation de l’EQM Seg pour les algorithmes TC-NLMS, TC-SFTF et l’algorithme 1 proposé pour $N = 80$ et $RSB$ d’entré=0dB. ....	46
<b>Figure 2.16.</b> L’évaluation du $RSB$ Seg de sortie pour les algorithmes TC-NLMS, TC-SFTF et Algorithme 1 proposé, $N = 80$ et $RSB$ d’entré = 0 dB. ....	47
<b>Figure 3.1.</b> Filtrage adaptatif multi-canaux. ....	55
<b>Figure 3.2.</b> Réduction de bruit adaptatif multi-canaux. ....	62
<b>Figure 3.3.</b> Signal de parole nette et son spectrogramme. ....	62
<b>Figure 3.4.</b> Spectrogrammes des bruits. ....	63
<b>Figure 3.5.</b> Réponses impulsionnelles acoustiques courtes. ....	64
<b>Figure 3.6.</b> Réponses impulsionnelles acoustiques longues et dispersées. ....	64
<b>Figure 3.7.</b> Signal de parole bruité ( $RSB = 0dB$ ). ....	65
<b>Figure 3.8.</b> Signaux rehaussés obtenus en sortie des différents algorithmes (à gauche : Simulation des RIs courtes ; à droite : Simulation des RIs longues et dispersées). ....	68
<b>Figure 3.9.</b> Désajustement obtenu par les différents algorithme (Simulation des RIs courtes). ....	69
<b>Figure 3.10.</b> Désajustement obtenu par les différents algorithme (Simulation des RIs longues et dispersées). ....	70
<b>Figure 3.11.</b> EQM Segmentée obtenue à la sortie par les différents algorithmes. ....	71
<b>Figure 3.12.</b> RSB Segmenté obtenu en sortie des différents algorithmes. ....	72
<b>Figure 3.13.</b> RSB obtenu à la sortie par les différents algorithme (Simulation des RI longues et dispersées). ....	72
<b>Figure 3.14.</b> Évaluation la complexité de calcul (CC) pour les algorithmes MC-NLMS, MC-APA et MC-FNLMS (version non réduite ainsi que la version réduite). ....	73
<b>Figure 4.1.</b> Exemple de découpage d’un signal en quatre sous-bandes. ....	77
<b>Figure 4.2.</b> Opération de décimation par un facteur $D$ . ....	77
<b>Figure 4.3.</b> Opération d’interpolation par un facteur $I$ . ....	77

<b>Figure 4.4.</b> Schéma global d'un traitement en sous-bande.....	78
<b>Figure 4.5.</b> Structure multi-bandes de filtrage adaptatif multi-canaux en sous-bande. ....	80
<b>Figure 4.6.</b> Structure en boucle ouverte de filtrage adaptatif multi-canaux en sous-bandes.	83
<b>Figure 4.7.</b> Densité spectrale d'amplitude ( <i>DSA</i> ) des signaux d'entrées <i>x<sub>in</sub></i> .....	86
<b>Figure 4.8.</b> Réponses fréquentielles d'amplitude des filtres d'analyses. ....	86
<b>Figure 4.9.</b> Densité spectrale d'amplitude ( <i>DSA</i> ) des signaux sous-bande après l'analyse. .	87
<b>Figure 4.10.</b> <i>EQM Seg</i> obtenu par l'algorithme 3, à gauche : <i>EQM Seg</i> différentes décomosition en sous-bandes ( $D = 1$ ), à droite : 4 sous-bandes avec différentes valeurs de $D$ ( $\mu = 0.05$ ).....	89
<b>Figure 4.11.</b> Comparaison de l' <i>EQM Seg</i> obtenue par l'algorithme 3 et l'algorithme MC- SB-NLMS (multi-bandes) pour différentes décomositions en sous-bandes. ....	90
<b>Figure 4.12.</b> Comparaison de la complexité de calcul de l'algorithme 3.....	91
<b>Figure 4.13.</b> <i>EQM Seg</i> obtenue par l'algorithme 4, à gauche : <i>EQM Seg</i> différentes décomosition en sous-bandes avec $D = 1$ , à droite :4 sous-bandes avec différentes valeurs de $D$ .....	93
<b>Figure 4.14.</b> Comparaison de l' <i>EQM Seg</i> obtenue par l'algorithme 4 et l'algorithme MC- SB-NLMS (multi-bandes) pour différentes décomositions en sous-bandes. ....	93
<b>Figure 4.15.</b> Comparaison de la complexité de calcul de l'algorithme 4.....	95
<b>Figure 4.16.</b> GSC basé sur les algorithmes proposés.....	96
<b>Figure 4.17.</b> Dimensions de la salle. ....	97
<b>Figure 4.18.</b> Réponses impulsionnelles acoustiques entre les sources et les microphones. .	97
<b>Figure 4.19.</b> Comparaison de l' <i>EQM Seg</i> obtenue à la sortie du GSC pour les algorithmes proposés et l'algorithme MC-SB-NLMS. ....	98

## Liste des tableaux

<b>Tableau 1.1.</b> L'algorithme des moindres carrés (LMS).....	12
<b>Tableau 1.2.</b> L'algorithme des moindres carrés normalisé (NLMS).....	13
<b>Tableau 1.3.</b> L'algorithme des moindres carrés récursif (RLS). ....	16
<b>Tableau 2.1.</b> L'algorithme FTF.....	27
<b>Tableau 2.2.</b> L'algorithme des filtres transversaux rapide simplifié (SFTF).....	27
<b>Tableau 2.3.</b> L'algorithme 1 (PU-SFTF) proposé (mono canal). ....	34
<b>Tableau 2.4.</b> L'algorithme 1 proposé pour la séparation aveugle des source FBSS (TC-PU-SFTF) .....	35
<b>Tableau 2.5.</b> Paramètres de simulation. ....	41
<b>Tableau 2.6.</b> Complexité de calcul.....	48
<b>Tableau 2.7.</b> Nombre des multiplications à chaque itération.....	48
<b>Tableau 3.1.</b> L'algorithme NLMS rapide (FNLMS) [22].....	54
<b>Tableau 3.2.</b> L'algorithme de projection affine multi-canaux (MC-APA).....	56
<b>Tableau 3.3.</b> L'algorithme MC-NLMS.....	56
<b>Tableau 3.4.</b> L'algorithme MC-FNLMS proposé (Algorithme 2).....	58
<b>Tableau 3.5.</b> Paramètres de simulation. ....	66
<b>Tableau 4.1.</b> L'algorithme MC-SB-FNLMS multi-bandes (Algorithme 3). ....	81
<b>Tableau 4.2.</b> L'algorithme MC-SB-FNLMS en boucle ouverte (Algorithme 4).....	84
<b>Tableau 4.3.</b> Paramètres de simulation. ....	87

# INTRODUCTION

La réduction du bruit acoustique et le rehaussement du signal parole, dans les systèmes de communication mobiles à mains libres, occupent une place très importante dans le traitement du signal. Dans un système de communication mobiles à mains libres, il est fréquent que le signal de parole soit altéré par des bruits acoustiques. Par conséquent, la qualité du signal reçu se dégrade. Cette dégradation de la qualité du signal constitue un défi majeur pour les chercheurs, qui s'efforcent quotidiennement de chercher des solutions à ce problème. La recherche scientifique de ce domaine adopte une approche pragmatique ; diviser le bruit acoustique en plusieurs catégories basées sur le mécanisme de sa génération. Généralement, le bruit est divisé en trois catégories de base à savoir le [1]:

- Bruit additif
- Réverbération
- Interférence

L'objectif de la réduction du bruit est de restaurer le signal d'origine (parole) à partir d'une observation du signal bruité. Cela peut être décrit comme un problème d'estimation de paramètres d'un système de débruitage, l'estimation optimale de la parole originale peut donc être obtenue en optimisant un critère à la sortie de ce système, tel que l'erreur quadratique moyenne (EQM) ou le rapport signal sur bruit (RSB) [1].

Le problème de la réduction de bruit a été largement abordé dans la littérature et se base généralement sur un système constitué d'un simple microphone (mono-capteur) ou plusieurs (multi-capteurs). La méthode classique et la plus ancienne de débruitage est basée sur un seul microphone ; c'est la méthode de soustraction spectrale de puissance (SSP) ou d'amplitude (SSA) [2]. C'est une méthode fréquentielle développée à partir d'une seule et unique observation constituée d'un mélange de parole et de bruit. Le système doit être en mesure de supprimer de réduire le bruit en vue de restituer un signal moins bruité. Dans cette approche, le bruit est supposé stationnaire et additif, ce qui permet son estimation pendant les périodes de silence. Cependant, cette méthode mono-capteur reste efficace pour un faible rapport signal sur bruit (RSB) [2]. Dans le cas de deux microphones, avec un microphone captant le signal parole bruité et l'autre mesurant le bruit, nous pouvons utiliser le signal du deuxième microphone comme référence de bruit et éliminer le bruit dans le premier microphone au moyen d'une annulation adaptative [3].

La technique de séparation aveugle des sources (SAS) [4, 5, 6, 7, 8] permet de séparer des mélanges convolutifs des signaux à partir des observations en utilisant deux capteurs. De nombreux chercheurs ont suggéré l'utilisation des structures de séparation adaptatif, avec les algorithmes du filtrage adaptatif dans différents domaines : temporel et fréquentiel [9, 10, 11, 12]. Cependant, les méthodes mono- et bi-capteurs ne permettent pas d'avoir le meilleur traitement dans les environnements où plusieurs sources de bruit peuvent être présentes en même temps. Un exemple de telle situation est le bruit à l'intérieur de véhicule qui provient de multiples sources, comme la motorisation, le roulement des pneus, etc. En général, plus le nombre de microphones est élevé, plus le système de réduction du bruit est efficace [1].

Le filtrage adaptatif joue un rôle crucial dans la réduction de bruit, en effet les systèmes de réduction de bruit doivent être capables de s'adapter à l'environnement dans lequel ils sont mis en œuvre. Les premiers algorithmes de filtrage adaptatif développés étaient à la moitié du XX<sup>ème</sup> siècle, période pendant laquelle plusieurs chercheurs travaillaient sur le développement des nouveaux algorithmes efficaces pour différentes applications [3, 13, 14, 15]. Jusqu'à ce jour, une grande variété d'algorithme a été développé. Le choix de l'algorithme dépend des performances visées, y compris la vitesse de convergence vers la solution optimale, la capacité à suivre la non-stationnarité, la complexité de calcul et la robustesse [16, 17].

Le but principal de ce travail est la réduction du bruit acoustique et le rehaussement de la parole bi- et multi-capteurs en utilisant le filtrage adaptatif multi-canaux. Dans ce manuscrit, nous proposons un algorithme à deux canaux et trois autres algorithmes multi-canaux, rapide en termes de vitesse de convergence et à faible complexité de calcul.

Dans le premier chapitre, nous présentons une étude approfondie sur la théorie du filtrage optimal de Wiener, le principe du filtrage adaptatif et les algorithmes de référence du filtrage adaptatif. Ensuite, nous examinons les diverses techniques classiques de réduction du bruit bi- et multi-capteurs.

Le chapitre 2 présente le premier algorithme bi-capteur proposé, nommé TC-PU-SFTF [18]. Cet algorithme représente une version à faible complexité de l'algorithme bi-capteur SFTF récemment développé [19] pour la réduction du bruit acoustique par la séparation aveugle des sources. L'algorithme proposé repose sur la technique de mise à jour partielle du filtre adaptatif afin de réduire la complexité de calcul. Nous détaillerons l'analyse, les formulations mathématiques et les résultats de simulations de ce nouveau algorithme. Nous présenterons aussi des comparaisons avec d'autres algorithmes de l'état de l'art.

Le chapitre 3 est consacré au développement de nouvel algorithme multi-canaux (MC-FNLM) [20] rapide et de complexité réduite dédiée aux applications de réduction de bruit par le filtrage adaptatif multicanal. Cet algorithme est une version multi-canaux à faible complexité de l'algorithme FNLMS [21, 22]. L'étude théorique de l'algorithme et sa stabilité est détaillée. Les résultats de simulation, les comparaisons avec d'autres algorithmes de l'état de l'art seront aussi présentées.

Dans le chapitre 4 nous introduisons deux algorithmes en sous-bande, dérivés de l'algorithme MC-FNLMS. L'objectif est de développer des variantes en sous-bande de l'algorithme MC-FNLMS pour obtenir des nouveaux algorithmes offrant des performances optimales en termes de taux de convergence et de complexité de calcul. Ainsi que leurs résultats de simulation seront présentés.

Nous terminons ce travail de thèse par une conclusion générale et nous exposerons les perspectives pour les futurs travaux envisagés dans la lignée de notre recherche.

# CHAPITRE 1 : FILTRAGE ADAPTATIF, ALGORITHMES ET REDUCTION DE BRUIT MULTI-CANAU

## 1.1.Introduction

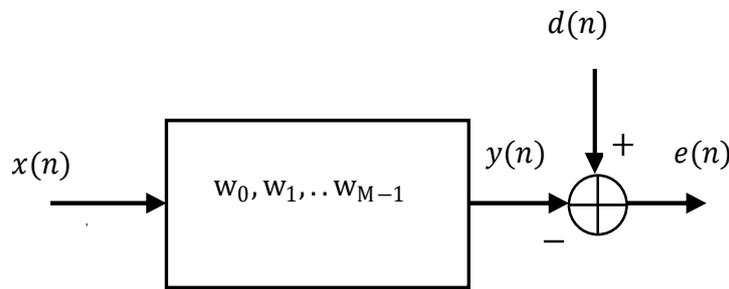
La théorie du filtre de Wiener a été initialement développée pour les signaux en temps continu dans le contexte du problème de prédiction linéaire. Norbert Wiener a également formulé une méthode explicite pour le prédicteur optimal, il s'est aussi intéressé au problème du filtrage optimal, qui consiste à estimer un processus altéré par un bruit additif. La formule explicite pour l'estimation optimale nécessitait la résolution d'une équation intégrale connue sous le nom d'équation de Wiener-Hopf (1931). En 1947, Levinson a reformulé le problème du filtrage de Wiener dans le contexte discret [16, 23].

L'application directe de la théorie du filtrage de Wiener à certains problèmes requiert la connaissance préalable des caractéristiques statistiques des signaux. Cependant, dans la plupart des cas, ces informations statistiques ne sont pas disponibles. C'est pourquoi le filtrage adaptatif revêt une grande importance car il permet, dans des conditions appropriées, de faire converger le filtre adaptatif vers la solution optimale de Wiener. Un exemple très intéressant du filtrage adaptatif est l'annulation adaptatif de bruit (ANC). De nombreuses structures sont proposées, en utilisant un seul capteur (mono-canal), deux capteurs ou plusieurs capteurs (multi-canaux), et dans lesquelles des algorithmes adaptatifs efficaces sont utilisés. Dans la littérature, on trouve de nombreux algorithmes, le choix de l'algorithme dépend des performances visées ; y compris la vitesse de convergence vers la solution optimale, la capacité à suivre la non-stationnarité, la complexité de calcul et la robustesse [16, 17].

Dans ce chapitre, nous allons présenter le principe du filtrage de Wiener appliqué au cas d'un filtre transversal à réponse impulsionnelle finie (RIF). Ensuite, nous introduirons le principe du filtrage adaptatif ainsi que la présentation des algorithmes classiques utilisés dans le filtrage adaptatif à savoir les algorithmes de gradient stochastique (c'est-à-dire LMS et NLMS), et l'algorithme des moindres carrés récurrents RLS. Nous présenterons également le principe de base des systèmes de réduction de bruit bi- capteurs par la technique de séparation aveugle des source (SAS) et multi-capteurs par le filtrage adaptatif multi-canaux.

## 1.2. Filtre de Wiener

On considère le schéma bloc de la **Figure 1.1** construit d'un filtre linéaire à temps discret caractérisé par une réponse impulsionnelle finie de taille  $M$ , représentée par les poids  $w_0, w_1, \dots, w_{M-1}$ . À un instant discret  $n$ , le filtre produit un signal de sortie noté  $y(n)$ . Ce signal de sortie est utilisé pour fournir une estimation d'une réponse désirée désignée par  $d(n)$ . L'estimation est généralement accompagnée d'une erreur d'estimation, notée  $e(n)$ , est définie comme la différence entre la réponse désirée  $d(n)$  et la sortie du filtre  $y(n)$ . L'objectif du filtrage de Wiener est de rendre l'erreur d'estimation  $e(n)$  aussi petite que possible dans le sens statistique des moindres carrés.



**Figure 1.1.** Représentation sous forme de schéma bloc du problème de filtrage optimal.

La sortie du filtre  $y(n)$ , est donnée par l'équation suivante :

$$y(n) = \sum_{k=0}^{M-1} x(n-k)w_k \quad (1.1)$$

Cette équation peut aussi s'écrire sous la forme matricielle ci-dessous :

$$y(n) = \mathbf{x}^T(n)\mathbf{w} \quad (1.2)$$

Où les vecteur  $\mathbf{x}(n)$  et  $\mathbf{w}$  sont définis comme ci-dessous :

$$\mathbf{x}(n) = [x(n), x(n-1), \dots, x(n-M+1)]^T \quad (1.3)$$

$$\mathbf{w} = [w_0, w_1, \dots, w_{M-1}]^T \quad (1.4)$$

L'estimation du signal désiré  $d(n)$  est évidemment accompagnée d'une erreur, qui est définie par la différence :

$$\begin{aligned} e(n) &= d(n) - y(n) \\ e(n) &= d(n) - \mathbf{x}^T(n)\mathbf{w} \end{aligned} \quad (1.5)$$

### 1.2.1. Principe d'orthogonalité

Les signaux  $x(n)$  et  $d(n)$  sont considérés comme des réalisations d'un processus aléatoire réel stationnaire au sens large de moyenne statistique nulle (c-à-d.  $\forall n \in \mathbb{Z}, E[x(n)] = 0$  et  $E[y(n)] = 0$ ). L'erreur d'estimation  $e(n)$  à l'instant  $n$  est une variable aléatoire. Pour optimiser la conception du filtre, on préfère de minimiser la valeur quadratique moyenne de  $e(n)$ . Ce choix d'optimisation facilite les manipulations mathématiques. En particulier, il conduit à une dépendance du second ordre (fonctions de corrélation et d'intercorrélation) pour optimiser la fonction coût par rapport aux poids de filtre. En plus, la minimisation de la valeur quadratique moyenne de la fonction coût, possède un minimum distinct qui définit de manière unique la conception optimale du filtre. Cette fonction de coût est définie par l'expression ci-dessous :

$$J = E[e^2(n)] \quad (1.6)$$

Où  $E[.]$  représente l'opérateur d'espérance mathématique. Le but consiste donc à déterminer le filtre représenté par le vecteur  $\mathbf{w}$  pour lequel la fonction coût  $J$  atteint sa valeur minimale. On développe cette fonction coût en fonction de vecteur  $\mathbf{w}$  :

$$J = E[(d(n) - \mathbf{x}^T(n)\mathbf{w})^2] \quad (1.7)$$

On peut dériver la fonction  $J$  selon le vecteur  $\mathbf{w}$  en utilisant la règle de la dérivation matricielle suivante :

$$\frac{\partial g(f(\mathbf{w}))}{\partial \mathbf{w}} = \frac{\partial f(\mathbf{w})}{\partial \mathbf{w}} \frac{\partial g(f)}{\partial f} \quad (1.8)$$

Où  $g(f(\mathbf{w})) = (f(\mathbf{w}))^2$  et  $f(\mathbf{w}) = d(n) - \mathbf{x}^T(n)\mathbf{w}$ .

Le calcul de la dérivée conduit donc à l'expression écrite ci-dessous :

$$\begin{aligned} \frac{\partial J}{\partial \mathbf{w}} &= E[-2\mathbf{x}(n)(d(n) - \mathbf{x}^T(n)\mathbf{w})] \\ &= -2E[\mathbf{x}(n)e(n)] \end{aligned} \quad (1.9)$$

Le filtre optimal au sens des moindres carrés correspond à une dérivée nulle de la fonction coût :

$$\left. \frac{\partial J}{\partial \mathbf{w}} \right|_{\mathbf{w}=\mathbf{w}_{opt}} = \mathbf{0}_{M \times 1} \quad (1.10)$$

Ce qui est équivalente à :

$$E[\mathbf{x}(n)e_o(n)] = \mathbf{0}_{M \times 1} \quad (1.11)$$

Ou encore :

$$E[x(n-k)e_o(n)] = 0, \quad k = 0, 1, 2, \dots, M-1 \quad (1.12)$$

Où  $e_o(n)$  est l'erreur d'estimation à l'optimum.

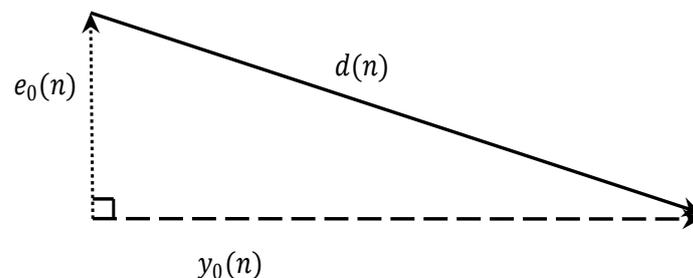
En vertu de l'équation (1.12), la condition nécessaire et suffisante pour que la fonction de coût  $J$  atteigne sa valeur minimale implique que l'erreur  $e(n)$  soit orthogonale avec l'échantillon  $x(n-k)$  de signal d'entrée qui entre dans l'estimation de la réponse désirée à l'instant  $n$ . Cette assertion constitue le théorème du principe de l'orthogonalité, ce théorème permet de vérifier si le filtre linéaire fonctionne dans sa condition optimale en observant que l'erreur  $e(n)$  et les échantillons  $x(n-k)$ . En multipliant les deux côtés de l'équation (1.11) par le filtre optimal  $\mathbf{w}_{opt}^T$  on obtient :

$$\begin{aligned} \mathbf{w}_0^T E[\mathbf{x}(n)e_o(n)] &= \mathbf{w}_{opt}^T \mathbf{0}_{M \times 1} \\ &= E[\mathbf{w}_{opt}^T \mathbf{x}(n)e_o(n)] = 0 \end{aligned} \quad (1.13)$$

Sachant que  $y_o(n) = \mathbf{w}_{opt}^T \mathbf{x}(n)$ , alors

$$E[y_o(n)e_o(n)] = 0 \quad (1.14)$$

On peut donc à partir de l'équation (1.14), donc énoncer le corollaire du principe de l'orthogonalité comme suit : lorsque le filtre fonctionne dans sa condition optimale, l'estimation de la réponse à la sortie du filtre  $y_o(n)$  et l'erreur d'estimation correspondante  $e(n)$  sont mutuellement orthogonales. L'interprétation géométrique des signaux à l'optimum est donnée par la figure ci-dessous [16, 17]:



**Figure 1.2.** Interprétation géométrique de la relation entre la réponse désirée, l'estimation à la sortie du filtre et l'erreur d'estimation à l'optimum.

### 1.2.2. Équations de Wiener-Hopf et filtre optimal de Wiener

On développe l'équation (1.9) pour faire intervenir la matrice d'autocorrélation  $\mathbf{R}$  d'ordre  $M$  du signal  $x(n)$  et le vecteur d'intercorrélacion  $\mathbf{p}$  d'ordre  $M$  des signaux  $x(n)$  et  $d(n)$ , Ces deux derniers sont définis par les deux équations ci-dessous :

$$\frac{\partial J}{\partial \mathbf{w}} = -2E[(x(n)d(n) - x(n)x^T(n)\mathbf{w})] \quad (1.15)$$

En sachant que

$$\mathbf{R} = E[\mathbf{x}^T(n)\mathbf{x}(n)] \quad (1.16)$$

et

$$\mathbf{p} = E[\mathbf{x}(n)d(n)] \quad (1.17)$$

On obtient donc l'expression de la quantité  $\frac{\partial J}{\partial \mathbf{w}}$  en fonction de  $\mathbf{R}$  et  $\mathbf{p}$  :

$$\frac{\partial J}{\partial \mathbf{w}} = -2(\mathbf{p} - \mathbf{R}\mathbf{w}) \quad (1.18)$$

Le filtre optimal qui minimise la fonction cout correspond donc à la nouvelle équation donnée ci-dessous :

$$-2(\mathbf{p} - \mathbf{R}\mathbf{w}_{opt}) = \mathbf{0}_{M \times 1} \quad (1.19)$$

La solution de cette dernière équation conduit à l'équation de Wiener-Hopf pour le filtrage optimal à réponse impulsionnelle fini (RIF) :

$$\mathbf{R}\mathbf{w}_{opt} = \mathbf{p} \quad (1.20)$$

Pour résoudre les équations (1.20), nous supposons que la matrice d'autocorrélation  $\mathbf{R}$  est inversible, ce qui est le cas pour les processus stationnaires au sens large. On peut donc multiplier les deux côtés de l'équation (1.20) par l'inverse de la matrice d'autocorrélation  $\mathbf{R}^{-1}$ , l'expression du filtre optimal de Wiener est donnée par l'équation ci-dessous :

$$\mathbf{w}_{opt} = \mathbf{R}^{-1}\mathbf{p} \quad (1.21)$$

### 1.2.3. Expression de la fonction cout minimal $J_{min}$

On peut écrire la dépendance de la fonction coût avec le filtre  $\mathbf{w}$  en fonction de la variance du signal désiré et des matrices d'autocorrélation  $\mathbf{R}$  et la matrice d'intercorrélation  $\mathbf{p}$ . En développant l'équation (1.7) :

$$J = E[d^2(n) - d(n)\mathbf{x}^T(n)\mathbf{w} - \mathbf{x}^T(n)\mathbf{w}d(n) + \mathbf{x}^T(n)\mathbf{w}\mathbf{x}^T(n)\mathbf{w}] \quad (1.22)$$

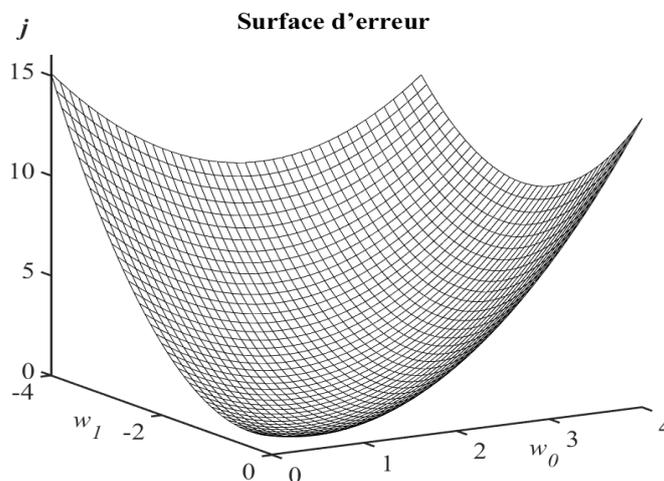
En exploitant la propriété de la linéarité de l'espérance mathématique, on obtient la nouvelle expression de  $J$ :

$$\begin{aligned} J &= E[d^2(n)] - E[d(n)\mathbf{x}^T(n)\mathbf{w}] - E[\mathbf{x}^T(n)\mathbf{w}d(n)] + E[\mathbf{x}^T(n)\mathbf{w}\mathbf{x}^T(n)\mathbf{w}] \\ &= E[d^2(n)] - E[d(n)\mathbf{x}^T(n)\mathbf{w}] - E[\mathbf{w}^T\mathbf{x}(n)d(n)] + E[\mathbf{w}^T\mathbf{x}(n)\mathbf{x}^T(n)\mathbf{w}] \\ &= E[d^2(n)] - E[d(n)\mathbf{x}^T(n)]\mathbf{w} - \mathbf{w}^T E[\mathbf{x}(n)d(n)] + \mathbf{w}^T E[\mathbf{x}(n)\mathbf{x}^T(n)]\mathbf{w} \\ &= \sigma_d^2 - \mathbf{p}^T\mathbf{w} - \mathbf{w}^T\mathbf{p} + \mathbf{w}^T\mathbf{R}\mathbf{w} \end{aligned} \quad (1.23)$$

D'après la paragraphe 1.2.2 ,cette fonction est minimale pour  $\mathbf{w} = \mathbf{w}_{opt} = \mathbf{R}^{-1}\mathbf{p}$ . Dans ce cas, on peut donc calculer  $J_{min}$  :

$$\begin{aligned} J_{min} &= \sigma_d^2 - \mathbf{p}^T\mathbf{w} - \mathbf{w}^T\mathbf{p} + (\mathbf{R}^{-1}\mathbf{p})^T\mathbf{R}(\mathbf{R}^{-1}\mathbf{p}) \\ &= \sigma_d^2 - \mathbf{p}^T\mathbf{R}^{-1}\mathbf{p} \end{aligned} \quad (1.24)$$

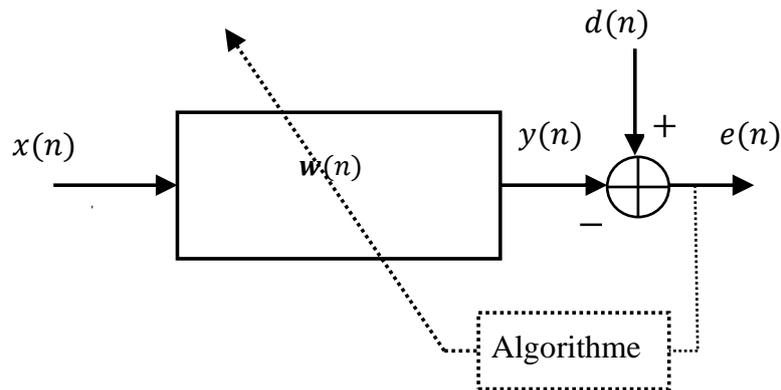
Un exemple illustrative de variation de la fonction coût en fonction des poids du filtre  $\mathbf{w}$  pour un filtre  $\mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}$  ( $M = 2$ ),  $\sigma_d^2 = 0.9486$ ,  $\mathbf{R} = \begin{bmatrix} 1.1 & 0.5 \\ 0.5 & 1.1 \end{bmatrix}$  et  $\mathbf{p} = [0.5272, -0.4458]$ . Avec ces choix, le tracé de la surface d'erreur quadratique moyenne  $J(w_0, w_1)$  est représenté dans la **Figure 1.3** [16] :



**Figure 1.3.** Surface d'erreur.

### 1.3. Filtrage adaptatif, algorithmes adaptatifs et applications.

Le principe du filtrage adaptatif est représenté à la **Figure 1.4**. Il correspond à une opération effectuée sur un signal d'entrée  $x(n)$  pour fournir à chaque instant un signal de sortie dont la différence  $e(n) = d(n) - y(n)$  soit minimale. Cette minimisation est obtenue en mettant à jour les coefficients  $w_i$  (poids) du filtre adaptatif  $\mathbf{w}$  à l'instant  $n$  utilisant le nouvel ensemble de données du signal désiré  $d(n)$ ,  $x(n)$  et l'erreur de filtrage à priori.

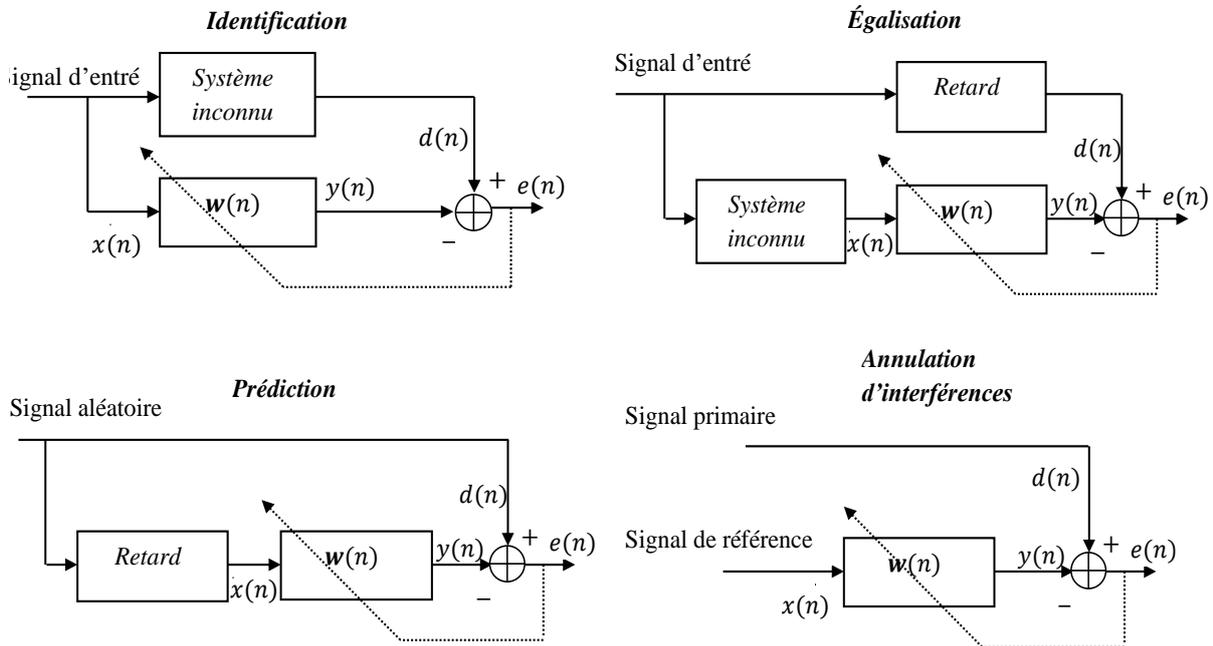


**Figure 1.4.** Principe de filtrage adaptatif.

On peut distinguer quatre classes de fonction de base du filtrage adaptatif [16, 17]:

- *Identification* : Dans l'identification, un filtre adaptatif est utilisé pour fournir un modèle linéaire qui représente le meilleur ajustement (dans un certain sens) à un système inconnu (à identifier).
- *Égalisation (modélisation inverse)* : Dans cette deuxième classe d'applications le filtre adaptatif fournit un modèle de fonction de transfert égale à l'inverse de système inconnu.
- *Prédiction* : Dans ce cas, le filtre adaptatif fournit une meilleure prédiction (dans un certain sens) de la valeur actuelle à l'instant  $n$  du signal  $x(n)$  en connaissant les valeurs passées  $x(n-1), x(n-2) \dots x(n-M)$ . La valeur actuelle du signal sert ainsi de réponse désirée pour le filtre adaptatif (c-à-d.  $d(n) = x(n)$ ).
- *Annulation d'interférences* : Dans cette classe d'applications, le filtre adaptatif est utilisé pour annuler les interférences superposées sur un signal primaire, avec une annulation optimisée dans un certain sens. Le signal primaire sert de réponse désirée pour le filtre adaptatif.

Les structures de ces quatre classes de filtrage adaptatif de base sont représentées dans la **Figure 1.5**.



**Figure 1.5.** Les quatre fonctions de base du filtrage adaptatif.

Une grande variété d'algorithmes de filtrage adaptatif a été développée dans la littérature. Le choix d'un algorithme est déterminé par un ou plusieurs des critères suivants [16]:

- *Taux de convergence* : Se définit comme le nombre de cycles d'adaptation nécessaires pour que l'algorithme atteigne la solution optimale. Cela signifie qu'un taux de convergence élevé permet à l'algorithme de s'adapter rapidement à des signaux de forte corrélation.
- *Le désajustement* : Ce critère fournit une mesure quantitative de la distance entre le vecteur du filtre adaptatif par rapport au vecteur de la solution optimale de Wiener.
- *La complexité de calcul* : C'est le nombre d'opérations (c'est-à-dire, les multiplications, les divisions et les additions/soustractions) nécessaires pour effectuer un cycle d'adaptation complet de l'algorithme, ainsi que la taille de la mémoire requise.
- *La capacité à poursuivre la non-stationnarité* : Lorsqu'un algorithme de filtrage adaptatif fonctionne dans un environnement non stationnaire, il est nécessaire que l'algorithme suive les variations statistiques de signal.

- *Robustesse* : Afin qu'un algorithme soit considéré robuste, il est essentiel que de légères perturbations n'engendrent que de légères erreurs d'estimation. Ces perturbations peuvent provenir de divers facteurs, internes ou externes au filtre.

### 1.3.1. Algorithme des moindres carrés moyennes (LMS)

On peut obtenir l'algorithme des moindres carrés moyennes (LMS) en utilisant plusieurs approches, la plus simple est basée sur la méthode de descente de gradient ; en adaptant les coefficients du filtre adaptatif à chaque instant dans la direction inverse de l'estimation du gradient de la fonction coût  $J$ . Une manière de faire converger le filtre adaptatif vers la solution optimale est d'ajouter à chaque itération de l'algorithme de descente de gradient, l'opposé du gradient. En utilisant l'expression du gradient donnée par l'équation (1.9), on forme ainsi l'équation de la mise à jour de l'algorithme de descente de gradient :

$$\begin{aligned} \mathbf{w}(n) &= \mathbf{w}(n-1) - \frac{\mu_{LMS}}{2} \frac{\partial J}{\partial \mathbf{w}} \\ &= \mathbf{w}(n-1) + \mu_{LMS} E[\mathbf{x}(n)e(n)] \end{aligned} \quad (1.25)$$

Malheureusement, cet algorithme n'est pas réalisable tant que nous ne connaissons pas en général l'espérance ci-dessus (c-à-d.  $E[\mathbf{x}(n)e(n)]$ ). Au lieu de cela, nous pouvons l'estimer en ignorant l'opérateur d'espérance dans l'équation (1.9) en utilisant une estimation instantanée de cette espérance en remplaçant la quantité  $E[\mathbf{x}(n)e(n)]$  par  $\mathbf{x}(n)e(n)$ . Cette estimation conduit à l'algorithme LMS. L'algorithme LMS est donné dans le tableau ci-dessous :

**Tableau 1.1.** L'algorithme des moindres carrés moyennes (LMS).

---

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

---

Pour garantir la convergence de l'algorithme, la condition pratique, imposée sur le pas d'adaptation  $\mu$  est exprimée par la double inégalité suivante :

$$0 < \mu_{LMS} < \frac{2}{\text{tr}[\mathbf{R}]} \quad (1.26)$$

Ou encore :

$$0 < \mu_{LMS} < \frac{2}{M\sigma_x^2} \quad (1.27)$$

Où  $\text{tr}[\mathbf{R}]$  désigne la trace de la matrice d'autocorrélation du signal  $x(n)$  à l'ordre de la taille du filtre adaptatif. Cette limite empêche la divergence du filtre adaptatif (en pratique, la valeur

de  $\mu_{LMS}$  ne devrait pas être choisie près de cette borne supérieure. La constante de temps de la courbe d'apprentissage  $\tau_{MSE}$  peut être approximée par la formule suivante [24]:

$$\tau_{MSE} \approx \frac{1}{2\mu_{LMS}\lambda_{min}} \quad (1.28)$$

$\lambda_{min}$  représente la plus petite valeur propre de la matrice d'autocorrélation  $\mathbf{R}$ . En se basant sur cette approximation, on peut déduire que pour un pas d'adaptation fixe, plus la valeur de  $\lambda_{min}$  est grande, plus la vitesse de convergence sera importante. Cela explique pourquoi un signal d'entrée de type bruit blanc permet d'obtenir une vitesse de convergence optimale, par le fait que toutes les valeurs propres d'un tel signal sont égales à la puissance du signal d'entrée.

### 1.3.2. Algorithme des moindres carrés moyennes normalisé (NLMS)

Le principal inconvénient de l'algorithme LMS est sa mauvaise stabilisée lorsque le signal d'entrée du filtré est de statistiques variables. Cela rend très difficile (voire impossible) de la mise en œuvre de l'algorithme LMS. Un autre inconvénient de cet algorithme, est sa grande erreur résiduelle en régime permanent [16, 17].

L'algorithme des moindres carrés moyennes normalisés (NLMS) est une variante de l'algorithme LMS qui résout ce problème en normalisant le pas d'adaptation par la norme Euclidienne de vecteur  $\mathbf{x}(n)$  (c-à-d.  $\|\mathbf{x}(n)\|$ ) du signal d'entrée. L'algorithme NLMS est résumé dans le tableau ci-dessous :

**Tableau 1.2.** L'algorithme des moindres carrés moyennes normalisé (NLMS).

---


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

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


---

Où  $\delta_{NLMS}$  est une petite constante de régularisation, empêche la division par zéro.

En comparant l'équation de mise à jour de l'algorithme NLMS avec celle de l'algorithme LMS, et en sachant la condition de convergence de l'algorithme LMS (équation (1.27)) on peut déduire que la convergence de l'algorithme NLMS est assurée si seulement si :

$$0 < \frac{\mu_{NLMS}}{\mathbf{x}^T(n)\mathbf{x}(n)} < \frac{2}{M\sigma_x^2} \quad (1.29)$$

En vertu de l'hypothèse de l'ergodicité on peut approximer la quantité  $\mathbf{x}^T(n)\mathbf{x}(n)$  par la quantité  $M\sigma_x^2$ , ainsi, la condition de convergence de l'algorithme NLMS est donc donnée par :

$$0 < \mu_{NLMS} < 2. \quad (1.30)$$

### 1.3.3. Algorithme des moindres carrés récursif (RLS)

L'algorithme RLS est connu par sa convergence rapide, même si la répartition des valeurs propres de la matrice d'autocorrélation de signal d'entrée sont dispersées. Cet algorithme peut avoir des performances compétitives lorsque on travaille dans un environnement non stationnaire ou avec des signaux de forte corrélation. Tous ces avantages s'accompagnent d'une augmentation de la complexité de calcul et de certains problèmes de stabilité pour les versions à complexité réduite de RLS (c-à-d. famille FTF), qui ne sont pas aussi critiques que de NLMS. L'objectif de l'algorithme RLS est de choisir les coefficients du filtre adaptatif de telle manière que le signal de sortie  $y(n)$ , pendant la période d'observation, correspond le plus possible au signal désiré dans le sens des moindres carrés donné par l'équation ci-dessous. Les coefficients de filtre adaptatif, sont adaptés dans le but de minimiser à chaque instant  $n$  par rapport aux coefficients  $w_i$ , la fonction cout déterministe donnée :

$$\tilde{J}(n) = \sum_{k=1}^n \lambda^{n-k} [d(k) - \mathbf{x}^T(k)\mathbf{w}(n)]^2 \quad (1.31)$$

Avec :

$$\mathbf{w}(n) = [w_0(n), w_1(n), \dots, w_{M-1}(n)]^T \quad (1.32)$$

Le paramètre  $\lambda$  est un facteur de pondération exponentielle qui devrait être vérifié  $0 < \lambda < 1$ , ce paramètre est appelé facteur d'oubli, il ignore l'effets du passé sur la mise à jour des coefficients du filtre adaptatif. Le résultat de l'optimisation de la fonction cout conduit à l'équation normal suivante :

$$\mathbf{w}(n) = \tilde{\mathbf{R}}^{-1}(n)\tilde{\mathbf{r}}(n) \quad (1.33)$$

Avec

$$\tilde{\mathbf{R}}(n) = \sum_{k=1}^n \lambda^{n-k} \mathbf{x}(k)\mathbf{x}^T(k) \quad (1.34)$$

Et

$$\tilde{\mathbf{r}}(n) = \sum_{k=1}^n \lambda^{n-k} \mathbf{x}(k)d(k) \quad (1.35)$$

L'équation (1.33) à peut être résolue à chaque instant  $n$  afin obtenir le filtre optimal, cependant, l'inversion de la matrice  $\tilde{\mathbf{R}}(n)$  à chaque instant  $n$ , rend la complexité de calcul trop élevé

(nécessite un nombre d'opérations de multiplication proportionnel à  $M^3$ ) ce qui n'est pas pratique pour les applications temps réel. On préfère alors de réduire la complexité de l'algorithme RLS en utilisant un calcul récursif de la matrice  $\tilde{\mathbf{R}}(n)$  est préféré. L'algorithme peut être déduit en écrivant les équations de récurrences suivantes :

$$\tilde{\mathbf{R}}(n) = \lambda \tilde{\mathbf{R}}(n-1) + \mathbf{x}(n)\mathbf{x}^T(n) \quad (1.36)$$

et

$$\tilde{\mathbf{r}}(n) = \lambda \tilde{\mathbf{r}}(n-1) + \mathbf{x}(n)d(n) \quad (1.37)$$

D'où, d'après (1.33) :

$$\mathbf{w}(n) = \tilde{\mathbf{R}}^{-1}(n)(\lambda \tilde{\mathbf{r}}(n-1) + \mathbf{x}(n)d(n)) \quad (1.38)$$

Comme :

$$\tilde{\mathbf{r}}(n-1) = \tilde{\mathbf{R}}(n-1)\mathbf{w}(n-1) = \left(\lambda^{-1}\tilde{\mathbf{R}}(n) - \lambda^{-1}\mathbf{x}(n)\mathbf{x}^T(n)\right)\mathbf{w}(n-1) \quad (1.39)$$

En remplaçant cette dernière expression dans (1.38), on trouve la nouvelle équation de la mise jours du filtre :

$$\mathbf{w}(n) = \tilde{\mathbf{R}}^{-1}(n) \left( \lambda \left( \lambda^{-1}\tilde{\mathbf{R}}(n) - \lambda^{-1}\mathbf{x}(n)\mathbf{x}^T(n) \right) \mathbf{w}(n-1) + \mathbf{x}(n)d(n) \right) \quad (1.40)$$

En simplifiant cette dernière expression :

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \tilde{\mathbf{R}}^{-1}(n)\mathbf{x}(n)(d(n) - \mathbf{x}^T(n)\mathbf{w}(n-1)) \quad (1.41)$$

La quantité  $d(n) - \mathbf{x}^T(n)\mathbf{w}(n-1)$  étant l'erreur d'estimation à priori, l'équation de mise à jour donc s'écrit comme suite :

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \tilde{\mathbf{R}}^{-1}(n)\mathbf{x}(n)e(n) \quad (1.42)$$

Vu l'inversion de la matrice  $\tilde{\mathbf{R}}(n)$ , cette dernière équation de mise à jour reste encore de complexité élevée [16], cependant on peut utiliser dans ce cas le lemme d'inversion des matrices de la forme  $(A + BCD)$  :

$$(\mathbf{A} + \mathbf{BCD})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{BC}(\mathbf{C}^{-1} + \mathbf{DA}^{-1}\mathbf{B})^{-1}\mathbf{DA}^{-1} \quad (1.43)$$

En appliquant le lemme d'inversion de la matrice avec :

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

On obtient ainsi :

$$\tilde{\mathbf{R}}^{-1}(n) = \lambda^{-1}\tilde{\mathbf{R}}^{-1}(n-1) - \lambda^{-2} \frac{\tilde{\mathbf{R}}^{-1}(n-1)\mathbf{x}(n)\mathbf{x}^T(n)\tilde{\mathbf{R}}^{-1}(n-1)}{1 + \lambda^{-1}\mathbf{x}^T(n)\tilde{\mathbf{R}}^{-1}(n-1)\mathbf{x}(n)} \quad (1.44)$$

L'équation (1.44) calcule récursivement la matrice  $\tilde{\mathbf{R}}^{-1}(n)$  en connaissant  $\tilde{\mathbf{R}}^{-1}(n-1)$ . Cette technique est beaucoup moins complexe que l'inversion directe de  $\tilde{\mathbf{R}}(n)$ .

Nous posant en fin :

$$\mathbf{P}(n) = \tilde{\mathbf{R}}^{-1}(n) \quad (1.45)$$

On définit le gain de Kalman à priori  $\mathbf{k}(n)$  par :

$$\mathbf{k}(n) = \tilde{\mathbf{R}}^{-1}(n-1)\mathbf{x}(n) \quad (1.46)$$

On résume l'algorithme RLS, qui était proposé, pour la première fois par Godard [10] :

**Tableau 1.3.** L'algorithme des moindres carrés récursif (RLS).

---


$$\mathbf{P}(0) = \delta^{-1}\mathbf{I}$$

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

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

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

$$\mathbf{P}(n) = \lambda^{-1}\mathbf{P}(n-1) - \lambda^{-1}\mathbf{k}(n)\mathbf{x}^T(n)\mathbf{P}(n-1)$$


---

$\delta$  : petite constante positive.

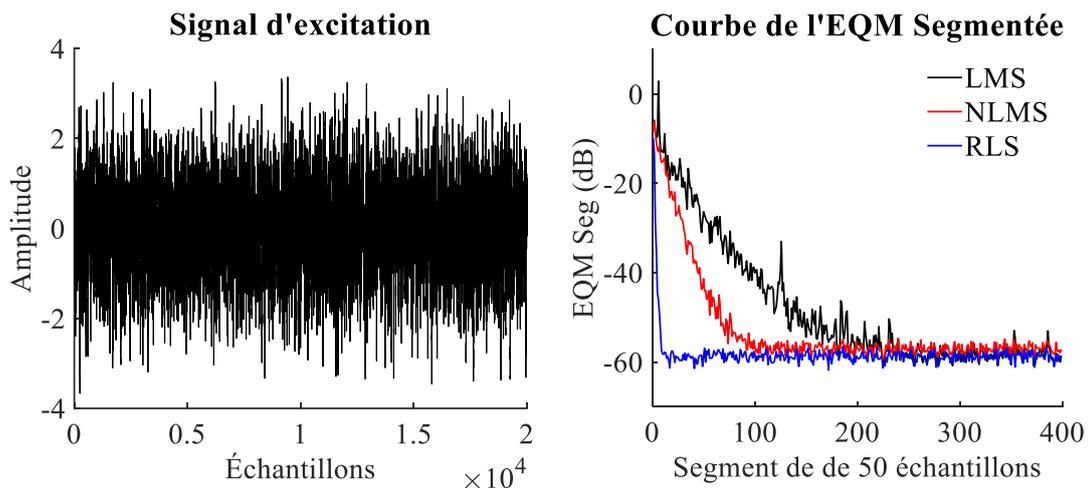
L'utilisation du lemme d'inversion permet de réduire la complexité de calcul de l'algorithme RLS de  $O(M^3)$  à  $O(M^2)$ . Cependant, il existe un algorithme encore moins complexe appelé l'algorithme des filtres transversaux rapides (Fast Transversal Filter Algorithm (FTF)). Cet algorithme fait apparaître les paramètres des prédictions avant et arrière pour calculer le gain de Kalman  $\mathbf{k}(n)$  sans passer par l'inversion de la matrice  $\tilde{\mathbf{R}}(n)$ .

#### 1.3.4. Exemple comparative entre les trois algorithmes LMS, NLMS et RLS

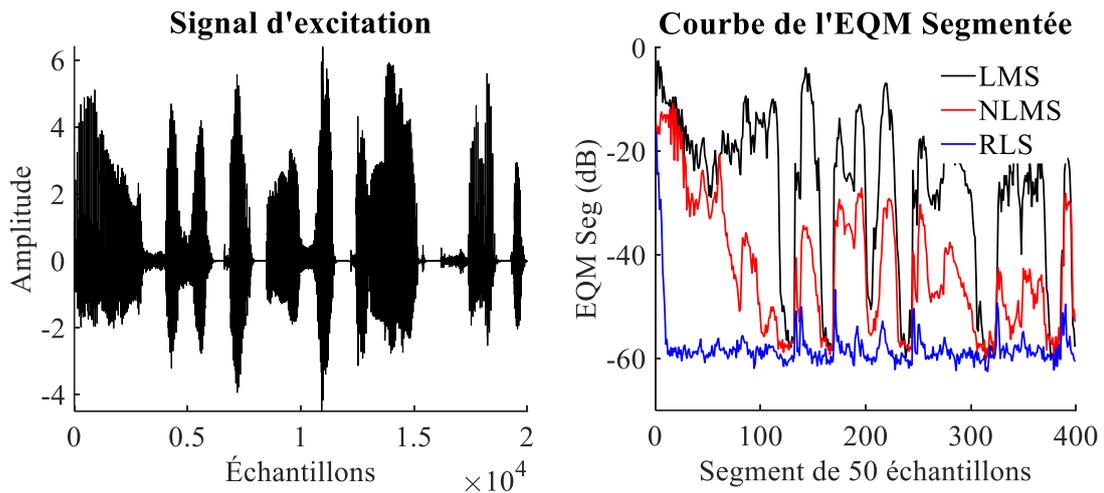
Ce paragraphe présente une démonstration des courbes d'apprentissage pour les trois algorithmes (LMS, NLMS et RLS), évaluées en termes de l'erreur quadratique moyenne segmentée (EQM Seg). Nous nous intéressons au contexte d'identification (**Figure 1.5**) d'une

réponse impulsionnelle aléatoire d'une longueur égale à 64 échantillons, à travers deux types de signaux ; le premier signal est un signal (autorégressif) stationnaire sur toute la durée de simulation (**Figure 1.6**), le deuxième est un signal de parole possédant des intervalles de non-stationnarité (**Figure 1.7**). Pour l'algorithme LMS, les pas d'adaptation qui garantissent une stabilité avec vitesse de convergence optimale sont choisis  $\mu_{LMS} = 0.01$  pour le cas d'un signal stationnaire et  $\mu_{LMS} = 0.001$  pour le deuxième cas. Le pas d'adaptation pour l'algorithme NLMS est fixé à  $\mu_{NLMS} = 1$ , le facteur d'oubli de l'algorithme RLS est  $\lambda=0.99$ .

Comme illustré dans les deux figures **Figure 1.6** et **Figure 1.7** ci-dessous, l'algorithme RLS présente le meilleur taux de convergence et un faible EQM en phase permanente, même en cas de signal non stationnaire, il reste robuste et moins sensible à la non-stationnarité. En revanche, l'algorithme LMS présente un faible taux de convergence et une sensibilité élevée à la non-stationnarité du signal. Les performances de l'algorithme NLMS se trouvent entre les deux algorithmes LMS et RLS.



**Figure 1.6.** EQM Seg cas d'un signal d'excitation stationnaire.



**Figure 1.7.** EQM Seg cas d'un signal d'excitation non-stationnaire.

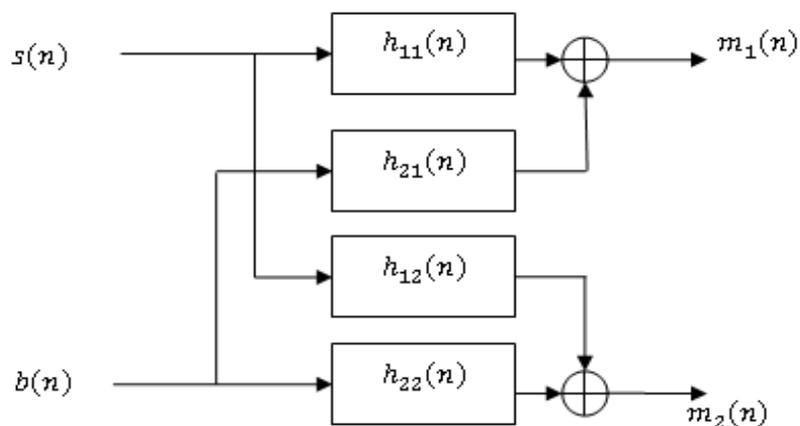
## 1.4. Méthode de réduction de bruit multi-canaux

### 1.4.1. Annulation de bruit par la séparation aveugle des sources

La séparation aveugle des sources (SAS ou BSS) a fait l'objet de nombreuses applications dans le domaine de rehaussement du signal de parole et la réduction du bruit. Elle consiste à retrouver les signaux sources à partir des signaux mélangés (observés). Dans cette thèse, nous allons nous intéresser au cas de deux sources de signaux ; l'une des sources est la parole, l'autre consiste le bruit. Le signal capté par les microphones contient donc un mélange convolutif de deux signaux sources avec lesquels on place près de chacune de ces deux sources un microphone [4, 5, 6, 7, 8, 25].

#### 1.4.1.1. Mélange convolutif

Le signal capté par chacun des microphones forme le mélange convolutif décrit dans **Figure 1.8** :



**Figure 1.8.** Le mélange convolutive.

À partir de la **Figure 1.8**, nous obtenons les signaux observables bruités au niveau du modèle de mélange :

$$m_1(n) = s(n) * h_{11}(n) + b(n) * h_{21}(n) \quad (1.47)$$

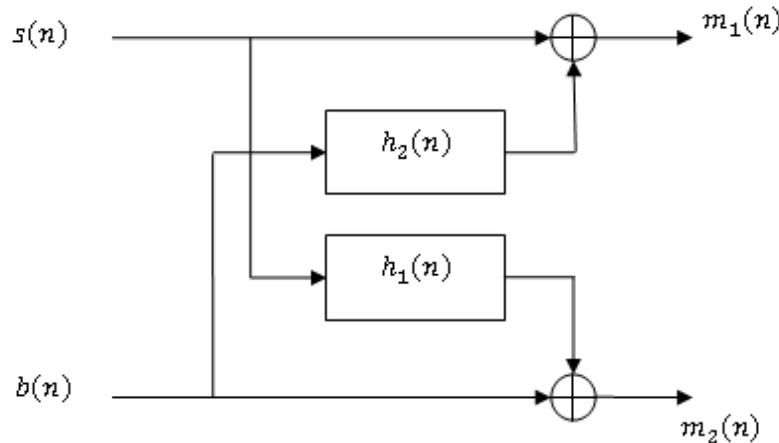
$$m_2(n) = s(n) * h_{12}(n) + b(n) * h_{22}(n) \quad (1.48)$$

Où  $h_{11}(n)$  et  $h_{22}(n)$  représentent les réponses (RIs) des canaux directs, tandis que  $h_{12}(n) = h_{21}(n) = h_2(n)$  correspondent aux réponses impulsionnelles entre les canaux. L'opération '\*' représente le produit de convolution. Les observations bruitées  $m_1(n)$  et  $m_2(n)$  résultent du mélange des signaux sources avec leurs réponses impulsionnelles respectives. Dans notre travail, nous considérons la situation où le locuteur est proche du microphone  $m_1(n)$  et la source de bruit est placée près du microphone  $m_2(n)$ . Cela conduit à  $h_{11}(n) = h_{22}(n) = \delta(n)$ , où  $\delta(n)$  est l'impulsion de Kronecker. Les équations simplifiées des observations bruitées sont données comme suit :

$$m_1(n) = s(n) + b(n) * h_2(n) \quad (1.49)$$

$$m_2(n) = b(n) + s(n) * h_1(n) \quad (1.50)$$

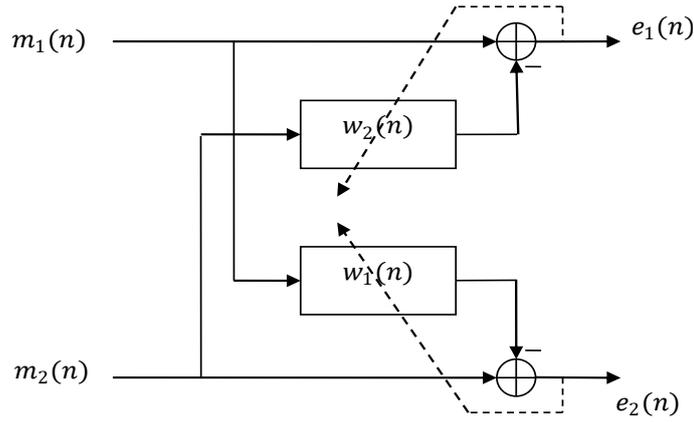
Le nouveau modèle simplifié du signal de mélange est représenté par la **Figure 1.9** [6] :



**Figure 1.9.** Mélange convolutif simplifié.

#### 1.4.1.2. Structure directe de séparation aveugle de source (FBSS)

La **Figure 1.10** montre le FBSS. Il permet de récupérer le signal des sources originales à partir d'observations bruitées en utilisant deux filtres adaptatifs [4, 6]. Cette structure est largement utilisée pour la réduction du bruit. Les deux signaux de sortie du FBSS sont donnés par les équations suivantes :



**Figure 1.10.** Structure de séparation directe (FBSS).

L'expression des signaux  $e_1(n)$  et  $e_2(n)$  est donnée par :

$$e_1(n) = s(n) + b(n) * h_2(n) - (b(n) + s(n) * h_1(n)) * w_2(n) \quad (1.51)$$

$$e_2(n) = b(n) + s(n) * h_1(n) - (s(n) + b(n) * h_2(n)) * w_1(n) \quad (1.52)$$

Lorsque les deux filtres adaptatifs  $w_1(n)$  et  $w_2(n)$  convergent vers les deux solutions optimales suivantes :

$$w_1^{opt}(n) = h_1(n) \quad (1.53)$$

$$w_2^{opt}(n) = h_2(n) \quad (1.54)$$

Ces solutions optimales conduisent aux nouvelles expressions de sortie :

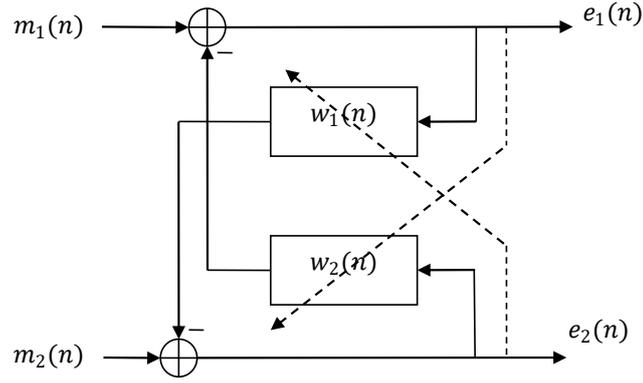
$$e_1(n) = s(n) * (\delta(n) - h_1(n) * h_2(n)) \quad (1.55)$$

$$e_2(n) = b(n) * (\delta(n) - h_1(n) * h_2(n)) \quad (1.56)$$

L'expression (1.55) montre que le signal de sortie  $e_1(n)$  est une version distordue du signal original  $s(n)$ . Nous considérons que les deux microphones sont tellement proches, alors l'effet de distorsion peut être donc négligé.

#### 1.4.1.3. Structure récursive de séparation aveugle de source

La seconde configuration de séparation de sources aveugles est la méthode récursive (BBSS), illustrée dans la **Figure 1.11**. Dans cette configuration, le signal estimé  $e_1(n)$  est obtenu en soustrayant le mélange  $m_1(n)$  du signal à la sortie du deuxième filtre adaptatif  $w_2(n)$ . De même, le signal estimé  $e_2(n)$  est obtenu en soustrayant le premier signal du mélange  $m_2(n)$  du signal à la sortie du premier filtre adaptatif  $w_1(n)$ .



**Figure 1.11.** Structure de séparation récursif (BBSS).

Les expressions des signaux estimés à la sortie, sont donnés par les deux équations suivantes :

$$e_1(n) = m_1(n) - e_2(n) * w_2(n) \quad (1.57)$$

$$e_2(n) = m_2(n) - e_1(n) * w_1(n) \quad (1.58)$$

En remplaçant les deux expressions de  $m_1(n)$  et  $m_2(n)$  dans les équations de  $e_1(n)$   $e_2(n)$ , nous obtenons les deux expressions suivantes :

$$e_1(n) = [b(n) * (h_2(n) - w_2(n)) + s(n) * (\delta(n) - h_1(n) * h_2(n))] * (\delta(n) - w_1(n) * w_2(n))^{-1} \quad (1.59)$$

$$e_2(n) = [s(n) * (h_1(n) - w_1(n)) + b(n) * (\delta(n) - h_2(n) * h_1(n))] * (\delta(n) - w_2(n) * w_1(n))^{-1} \quad (1.60)$$

Sous l'hypothèse d'optimalité pour les deux filtres adaptatifs  $w_1^{opt} = h_1(n)$  et  $w_2^{opt} = h_2(n)$  on obtient les deux signaux source original :

$$e_1(n) = s(n) \quad (1.61)$$

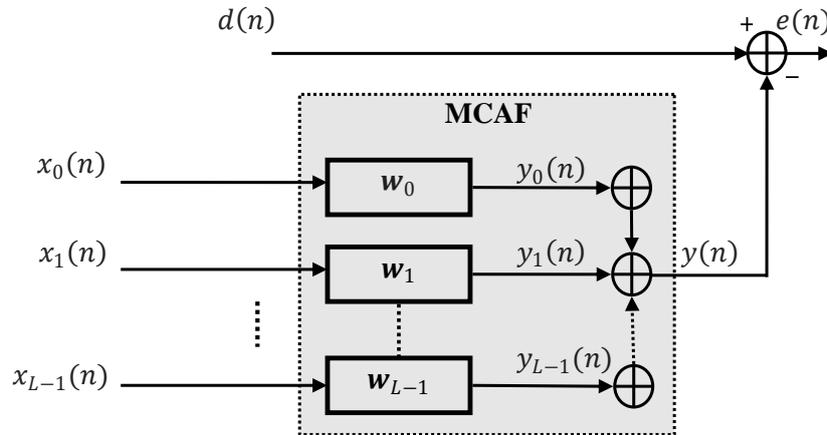
$$e_2(n) = b(n) \quad (1.62)$$

D'après ces deux dernières équations, on conclut que la structure récursive permet de récupérer les signaux source sans distorsion, contrairement à la structure directe.

#### 1.4.2. Annulation de bruit par le filtrage adaptatif multi-canaux

Le filtrage adaptatif multicanal **Figure 1.12** consiste à adapter l'ensemble des filtres adaptatifs  $w_i$  (les poids de chaque filtre adaptatif) pour faire converger ces derniers à une solution optimale. Les algorithmes utilisés dans le filtrage adaptatif multicanal peuvent être

dérivés à partir des algorithmes monocanaux en tenant compte des nouvelles conditions de convergence.



**Figure 1.12.** Filtrage multi-canaux.

Les filtres RIF optimaux qui minimisent l'erreur quadratique moyenne sont donnés par l'équation matricielle suivante :

$$\begin{bmatrix} \mathbf{w}_0^{opt} \\ \mathbf{w}_1^{opt} \\ \vdots \\ \mathbf{w}_{L-1}^{opt} \end{bmatrix} = \mathbf{R}^{-1} \mathbf{p} \quad (1.63)$$

avec :

$$\mathbf{R} = E[\mathbf{X}^T(n)\mathbf{X}(n)] \quad (1.64)$$

et

$$\mathbf{p} = E[\mathbf{X}(n)d(n)] \quad (1.65)$$

$\mathbf{X}(n)$  est un vecteur colonne de taille  $ML \times 1$  :

$$\mathbf{X}(n) = \begin{bmatrix} \mathbf{x}_0(n) \\ \mathbf{x}_1(n) \\ \vdots \\ \mathbf{x}_{L-1}(n) \end{bmatrix} \quad (1.66)$$

Dans la réduction de bruit multi-canaux,  $L$  signal de référence (bruits)  $x_i(n)$  ( $i = 1, 2, \dots$ ), et un signal primaire contenant le signal de parole  $s(n)$  et une composante de bruit  $b(n)$ , corrélé avec ceux  $x_i(n)$ . Le signal primaire constitue le signal désiré  $d(n)$  pour les filtres adaptatifs multi-canaux, est donné par  $d(n) = s(n) + b(n)$ . La sortie du filtre multicanal est donnée par :

$$e(n) = d(n) - \sum_{k=0}^{L-1} x_k(n) * w_k(n) \quad (1.67)$$

Dans le cas où le bruit  $b(n)$  résulte des transformations linéaires par des réponses impulsionnelles finies (RIF) (c-à-d.  $b(n) = \sum_{k=0}^{L-1} x_k(n) * h_k(n)$ ). Le signal désiré est donc écrit comme :

$$d(n) = s(n) + \sum_{k=0}^{L-1} x_k(n) * h_k(n)$$

Ou sous forme matricielle :

$$d(n) = s(n) + \mathbf{X}^T(n) \begin{bmatrix} \mathbf{h}_0 \\ \mathbf{h}_1 \\ \cdot \\ \cdot \\ \mathbf{h}_{L-1} \end{bmatrix} \quad (1.68)$$

La matrice d'autocorrélation  $\mathbf{p}$  est définie par :

$$\begin{aligned} \mathbf{p} &= E[\mathbf{X}(n)d(n)] \\ &= E \left[ \mathbf{X}(n) \left( s(n) + \mathbf{X}^T(n) \begin{bmatrix} \mathbf{h}_0 \\ \mathbf{h}_1 \\ \cdot \\ \cdot \\ \mathbf{h}_{L-1} \end{bmatrix} \right) \right] \\ &= E \left[ \mathbf{X}(n)s(n) + \mathbf{R} \begin{bmatrix} \mathbf{h}_0 \\ \mathbf{h}_1 \\ \cdot \\ \cdot \\ \mathbf{h}_{L-1} \end{bmatrix} \right] \end{aligned} \quad (1.69)$$

En exploitant la linéarité de l'espérance mathématique après développement, on obtient la nouvelle expression du vecteur d'intercorrélacion :

Ou encore :

$$\mathbf{p} = \left[ E[\mathbf{X}(n)s(n)] + \mathbf{R} \begin{bmatrix} \mathbf{h}_0 \\ \mathbf{h}_1 \\ \cdot \\ \cdot \\ \mathbf{h}_{L-1} \end{bmatrix} \right] \quad (1.70)$$

Les composantes du vecteur  $\mathbf{X}(n)$  contiennent le bruit à l'instant  $n, n - 1 \dots n - (M - 1)$ , ces composants sont évidemment décorrélés du signal de parole  $s(n)$ , cela implique  $E[\mathbf{X}(n)s(n)] = \mathbf{0}$

L'expression de la matrice d'intercorrélation se réduit donc à la matrice suivante :

$$\mathbf{p} = \mathbf{R} \begin{bmatrix} \mathbf{h}_0 \\ \mathbf{h}_1 \\ \cdot \\ \cdot \\ \mathbf{h}_{L-1} \end{bmatrix} \quad (1.71)$$

Les filtres optimaux  $\mathbf{w}_i^{opt}$  au sens des moindres carrés sont déterminés en remplaçant cette dernière expression de vecteur d'intercorrélation dans l'équation (1.63) :

$$\begin{bmatrix} \mathbf{w}_0^{opt} \\ \mathbf{w}_1^{opt} \\ \cdot \\ \cdot \\ \mathbf{w}_{L-1}^{opt} \end{bmatrix} = \mathbf{R}^{-1} \mathbf{R} \begin{bmatrix} \mathbf{h}_0 \\ \mathbf{h}_1 \\ \cdot \\ \cdot \\ \mathbf{h}_{L-1} \end{bmatrix} = \begin{bmatrix} \mathbf{h}_0 \\ \mathbf{h}_1 \\ \cdot \\ \cdot \\ \mathbf{h}_{L-1} \end{bmatrix} \quad (1.72)$$

L'expression du signal de la sortie du filtre  $e(n)$  à l'optimum peut donc être facilement trouvée :

$$\begin{aligned} e(n) &= d(n) - \sum_{k=0}^{L-1} x_k(n) * w_k^{opt}(n) \\ &= s(n) + \sum_{k=0}^{L-1} x_k(n) * h_k(n) - \sum_{k=0}^{L-1} x_k(n) * w_k^{opt}(n) \end{aligned} \quad (1.73)$$

Les filtres optimaux  $w_k^{opt}(n) = h_k(n)$  permettent ainsi de récupérer à la sortie le signal de parole original sans composants de bruits.

$$e(n) = s(n) \quad (1.74)$$

## 1.5. Conclusion

Dans ce premier chapitre, nous avons présenté le problème du filtrage optimal de Wiener pour les filtres à réponse impulsionnelle finie (RIF). Nous avons également présenté le principe du filtrage adaptatif permettant de s'approcher de la solution optimale de Wiener. Une étude détaillée est réalisée sur les algorithmes adaptatifs classiques, tels que les algorithmes de gradient stochastique (c'est-à-dire LMS et NLMS) et l'algorithme des moindres carrés récursifs RLS. Ensuite, nous avons présenté quelques méthodes (bi- et multi-capteurs) utilisées pour la réduction du bruit et le rehaussement de signal de la parole. Dans le prochain chapitre, nous allons proposer un algorithme bi-capteur FBSS rapide et de complexité réduite pour la réduction du bruit et le rehaussement du signal de parole.

# CHAPITRE 2 : NOUVEAU ALGORITHME BI-CAPTEURS POUR LE REHAUSSEMENT DU SIGNAL DE PAROLE

## 2.1.Introduction

Plusieurs algorithmes de filtrage adaptatif ont été proposés dans le but d'améliorer la qualité du signal parole, en réduisant le niveau de bruit acoustique auquel ont été superposés. La technique de filtrage partiel a été proposée dans la littérature [26, 27, 28] afin de diminuer la complexité de calcul. Elle est basée sur l'adaptation partielle des poids du filtre adaptatif selon un critère d'optimisation. Cependant, il est important de noter que cette technique peut dégrader d'autres performances, tel que la vitesse de convergence [29]. Néanmoins, la technique de filtrage partiel reste primordiale pour l'identification des longues réponses impulsionnelles qui nécessitent l'utilisation des filtres adaptatifs de longues tailles. Le filtrage partiel est aussi utilisé lorsqu'on emploie des algorithmes rapides qui possède une complexité très élevée tel que les algorithmes FTF, APA, SFTF, etc.

L'algorithme FTF est basé sur le calcul des prédicteurs avant et arrière pour la mise à jour du filtre adaptatif [30, 31], à l'opposé de l'algorithme SFTF qui n'utilise que des prédicteurs avant [32, 19, 33]. Dans l'algorithme SFTF, les prédicteurs arrière sont complètement supprimés dans le processus d'adaptation, ce qui réduit la complexité par rapport à l'algorithme FTF qui nécessite  $8M$  multiplications à  $7M$  multiplications par itération. Cependant, la suppression de prédicteur arrière  $\mathbf{b}(n)$  génère une erreur dans le calcul du gain de Kalman, par conséquent, réduit les performances de la convergence de l'algorithme.

Dans ce chapitre, nous proposons une solution pour compenser l'erreur générée par l'élimination des prédicteurs arrière dans l'algorithme SFTF, par une réduction de complexité de calcul en utilisant la technique de filtrage partielle sélectif. L'algorithme proposé est basé sur la mise à jour uniquement des coefficients des filtres transversaux  $\mathbf{a}(n)$  et  $\mathbf{w}(n)$  qui ne s'affectent pas de l'élimination du prédicteur arrière, cela va nous permettre de garder la même vitesse de convergence par rapport à l'algorithme original SFTF avec un bénéfice dans la complexité des calculs. L'algorithme proposé sera combiné avec le séparateur des sources aveugle pour obtenir une version à deux canaux TC-PU-SFTF (Algorithme 1) adapté à la séparation des mélanges convolutifs dans le cadre de la réduction de bruit.

## 2.2. Algorithme FTF et sa version simplifiée (SFTF)

L'algorithme FTF est un algorithme rapide en termes de complexité de calcul par rapport à l'algorithme RLS. Cet algorithme utilise une équation de récurrence pour calculer le gain de Kalman dual en utilisant les filtres prédicteurs avant-arrière afin d'éviter l'inversion de la matrice  $\tilde{\mathbf{R}}^{-1}(n)$  de taille  $M \times M$ , comme dans le cas de l'algorithme RLS [30, 31]. Cependant, l'algorithme FTF simplifié (SFTF), n'utilise que le filtre prédicteur avant pour la mise à jour de gain de Kalman dual [32]. Dans cette section, nous présentons l'algorithme FTF et sa version simplifiée (SFTF), ce qui nous inspirera à dériver l'algorithme proposé dans la section 2.3.

La relation de mise à jour de gain de Kalman dual de l'algorithme FTF est donnée par ci-dessous :

$$\tilde{\mathbf{c}}_{M+1}(n) = \begin{bmatrix} 0 \\ \tilde{\mathbf{c}}_M(n-1) \end{bmatrix} - \frac{\bar{e}(n)}{\lambda\alpha(n-1)} \begin{bmatrix} 1 \\ -\mathbf{a}(n-1) \end{bmatrix} \quad (2.1)$$

$$\begin{bmatrix} \tilde{\mathbf{c}}_M(n) \\ 0 \end{bmatrix} = \tilde{\mathbf{c}}_{M+1}(n) - \tilde{\mathbf{c}}_{M+1}^{M+1}(n) \begin{bmatrix} -\mathbf{b}(n-1) \\ 1 \end{bmatrix} \quad (2.2)$$

Où :

$\mathbf{a}(n)$  et  $\mathbf{b}(n)$  sont respectivement le prédicteur avant et arrière d'ordre  $M$ .

$\bar{e}(n)$  et  $\alpha(n)$  sont respectivement l'erreur de prédiction avant et sa variance.

$\lambda$  c'est le facteur d'oubli ( $0 \ll \lambda < 1$ ).

$\tilde{\mathbf{c}}_{M+1}(n)$  est le gain de Kalman dual d'ordre  $M + 1$ .

Ces dernières quantités peuvent être mises à jour en introduisant le facteur de conversion  $\gamma(n)$ . Dans l'algorithme FTF, les équations (2.1) et (2.2) de la mise à jour de gain de Kalman dual  $\tilde{\mathbf{c}}_M(n)$ , utilisent à la fois le vecteur de prédiction avant et arrière. Des travaux [34, 32] ont démontré que le gain de Kalman dual  $\tilde{\mathbf{c}}_M(n)$  peut être estimé dans le cas général, uniquement à partir prédicteur avant. Cette hypothèse repose sur le faible impact du vecteur de prédiction arrière sur les composants de Kalman dual  $\tilde{\mathbf{c}}_M(n)$ , en particulier pour le cas des signaux autorégressifs (AR) de faible ordre, tels que la parole. L'algorithme SFTF [32] est basé sur ces dernières hypothèses qui justifient l'élimination des prédicteurs arrière. Afin d'assurer la convergence de l'algorithme SFTF, deux paramètres sont introduits ; le premier est une petite constante  $c$ , sert à éviter la division par zéro lors de la mise à jour de  $\tilde{\mathbf{c}}_M(n)$  (quand  $\alpha(n-1) = 0$ ), et le second paramètre, est un facteur de fuite  $\eta$  très proche de 1, ce facteur sert à réinitialiser les coefficients de prédicteur avant  $\mathbf{a}(n)$  en cas de divergence critique. L'algorithme des filtres transversaux rapide (FTF) et sa version simplifiée (SFTF) sont donnés respectivement dans le **Tableau 2.1** et **Tableau 2.2** ci-dessous :

**Tableau 2.1.** L'algorithme FTF.

---

**Initialisation :**

$$\mathbf{a}(0) = \mathbf{0}_{M \times 1}, \mathbf{b}(0) = \mathbf{0}_{M \times 1}, \tilde{\mathbf{c}}_M(0) = \mathbf{0}_{M \times 1}, \gamma(0) = 1, \\ \alpha(0) = E_0 \lambda^N, \beta(0) = E_0, \mathbf{w}(0) = \mathbf{0}_{M \times 1}$$

**Prédiction :**

$$\bar{e}(n) = \mathbf{x}_{M+1}^T(n) \begin{bmatrix} 1 \\ -\mathbf{a}(n-1) \end{bmatrix} \\ \bar{r}(n) = \mathbf{x}_{M+1}^T(n) \begin{bmatrix} -\mathbf{b}(n-1) \\ 1 \end{bmatrix} \\ \tilde{\mathbf{c}}_{M+1}(n) = \begin{bmatrix} 0 \\ \tilde{\mathbf{c}}_M(n-1) \end{bmatrix} - \frac{\bar{e}(n)}{\lambda \alpha(n-1)} \begin{bmatrix} 1 \\ -\mathbf{a}(n-1) \end{bmatrix} \\ \begin{bmatrix} \tilde{\mathbf{c}}_M(n) \\ 0 \end{bmatrix} = \tilde{\mathbf{c}}_{M+1}(n) - \tilde{\mathbf{c}}_{M+1}^{M+1}(n) \begin{bmatrix} -\mathbf{b}(n-1) \\ 1 \end{bmatrix} \\ \alpha(n) = \lambda \alpha(n-1) + \gamma(n-1) \bar{e}^2(n) \\ \mathbf{a}(n) = \mathbf{a}(n-1) - \bar{e}(n) \gamma(n-1) \tilde{\mathbf{c}}_M(n-1) \\ \gamma(n) = \frac{1}{1 - \mathbf{x}_M^T(n) \tilde{\mathbf{c}}_M(n)} \\ \mathbf{b}(n) = \mathbf{b}(n-1) - \bar{r}(n) \gamma(n) \tilde{\mathbf{c}}_M(n)$$

**Filtrage :**

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


---

$\bar{r}(n)$  et  $\beta(n)$  sont respectivement l'erreur de prédiction avant et sa variance.

**Tableau 2.2.** L'algorithme des filtres transversaux rapide simplifié (SFTF).

---

**Initialisation**

$$\mathbf{a}(0) = \mathbf{0}_{M \times 1}, \tilde{\mathbf{c}}_M(0) = \mathbf{0}_{M \times 1}, \gamma(0) = 1, \alpha(0) = E_0 \lambda^N, \mathbf{w}(0) = \mathbf{0}_{M \times 1}$$

**Prédiction**

$$\bar{e}(n) = \mathbf{x}_{M+1}^T(n) \begin{bmatrix} 1 \\ -\mathbf{a}(n) \end{bmatrix} \\ \tilde{\mathbf{c}}_{M+1}(n) = \begin{bmatrix} 0 \\ \tilde{\mathbf{c}}_M(n-1) \end{bmatrix} - \frac{\bar{e}(n)}{\lambda \alpha(n-1) + c} \begin{bmatrix} 1 \\ -\mathbf{a}(n-1) \end{bmatrix}$$

$$\begin{bmatrix} \tilde{\mathbf{c}}_M(n) \\ * \end{bmatrix} = \tilde{\mathbf{c}}_{M+1}(n)$$

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

$$\mathbf{a}(n) = \eta[\mathbf{a}(n-1) - \bar{e}(n)\gamma(n-1)\tilde{\mathbf{c}}_M(n-1)]$$

$$\gamma(n) = \frac{1}{1 - \mathbf{x}_M^T(n)\tilde{\mathbf{c}}_M(n)}$$

### Filtrage

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

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


---

### 2.3. Dérivation et présentation de l'algorithme proposé (Algorithme 1)

L'algorithme SFTF est couramment utilisé pour l'annulation d'écho acoustique (AEC) car, dans ce contexte, les signaux d'entrée possèdent généralement un modèle AR de faible ordre, c'est pourquoi les performances de l'algorithme SFTF restent très proches de l'algorithme FTF original. Par contre, dans le contexte de réduction adaptatif du bruit acoustique (ANR), les signaux de bruit ne possèdent pas nécessairement un modèle AR de faible ordre, par conséquent, la suppression du prédicteur arrière  $\mathbf{b}(n)$  dans ce cas, va introduire alors une erreur d'estimation de gain de Kalman dual par rapport à l'algorithme FTF, ce qui conduit ainsi à une sous-estimation du prédicteur avant  $\mathbf{a}(n)$ , du filtre adaptatif  $\mathbf{w}(n)$  et du facteur de conversion  $\gamma(n)$ . Par conséquent, les performances de l'algorithme SFTF peuvent se dégrader par rapport à celles de l'algorithme FTF original. Pour compenser ce problème, nous proposons une nouvelle version de l'algorithme SFTF avec une complexité de calcul réduite tout en maintenant ou voire même améliorant la vitesse de convergence pour certains types de signaux, voici la démarche :

Supposons à un instant  $n$ , les deux algorithmes FTF et SFTF possèdent les mêmes coefficients de gain de Kalman dual d'ordre  $M + 1$   $\tilde{\mathbf{c}}_{M+1}(n)$ , alors les gains de Kalman dual d'ordre  $M$  pour les deux algorithmes sont mises à jour respectivement par les équations suivantes :

$$\begin{bmatrix} \tilde{\mathbf{c}}_{FTF,M}(n) \\ 0 \end{bmatrix} = \tilde{\mathbf{c}}_{M+1}(n) - \tilde{\mathbf{c}}_{M+1}^{M+1}(n) \begin{bmatrix} -\mathbf{b}(n-1) \\ 1 \end{bmatrix} \quad (2.3)$$

$$\begin{bmatrix} \tilde{\mathbf{c}}_{SFTF,M}(n) \\ * \end{bmatrix} = \tilde{\mathbf{c}}_{M+1}(n) \quad (2.4)$$

On définit ci-dessous le vecteur d'erreur  $\Delta(n)$  de l'estimation du gain de Kalman de l'algorithme FTF par rapport à sa version simplifiée (c-à-d SFTF) :

$$\Delta(n) \triangleq \tilde{\mathbf{c}}_{SFTF,M}(n) - \tilde{\mathbf{c}}_{FTF,M}(n) = -\tilde{c}_{M+1}^{M+1}(n)\mathbf{b}(n-1) \quad (2.5)$$

$$\Delta(n) = -\tilde{c}_{M+1}^{M+1}(n)[b_0(n-1)b_1(n-1) \dots b_{M-1}(n-1)]^T \quad (2.6)$$

À l'instant  $n$ ,  $\tilde{c}_{M+1}^{M+1}(n)$  représente la dernière composante de gain de Kalman d'ordre  $M + 1$ , sa valeur change à chaque instant  $n$ , et le vecteur  $\mathbf{b}(n-1)$  correspond au prédicteur arrière à l'instant  $n-1$ . Étant donné que ce dernier (c-à-d.  $\mathbf{b}(n-1)$ ) est complètement supprimé dans le processus de mise à jour dans l'algorithme SFTF, alors l'estimation du gain de Kalman dual  $\tilde{\mathbf{c}}_{SFTF,M}(n)$  pour l'algorithme SFTF sera moins précise par rapport à l'algorithme FTF. Les composants de cette erreur d'estimation  $\Delta_i(n)$  du vecteur d'erreur  $\Delta(n)$  sont donnés par l'expression ci-dessous :

$$\Delta_i(n) = \tilde{c}_{M+1}^{M+1}(n)b_i(n-1) \quad (2.7)$$

L'estimation du carré (puissance) de l'erreur relative associée à chaque composante  $\tilde{c}_{SFTF,M}^i(n)$  par rapport à  $\tilde{c}_{FTF,M}^i(n)$  est donnée comme suit :

$$\varepsilon_i^2(n) = \frac{[\Delta_i(n)]^2}{[\tilde{c}_{FTF,M}^i(n)]^2} = \frac{[\Delta_i(n)]^2}{[\tilde{c}_{SFTF,M}^i(n) - \Delta_i(n)]^2} \quad (2.8)$$

Selon les deux cas particuliers, ce dernier peut être approximé par 0 ou 1 :

$$Si \quad [\tilde{c}_{SFTF,M}^i(n)]^2 \gg [\Delta_i(n)]^2 \quad alors \quad \varepsilon_i^2(n) \approx 0 \quad (2.9)$$

$$Si \quad [\tilde{c}_{SFTF,M}^i(n)]^2 \ll [\Delta_i(n)]^2 \quad alors \quad \varepsilon_i^2(n) \approx 1 \quad (2.10)$$

Étant donné pour le cas d'un processus AR, la puissance de coefficients optimaux  $a_i$  est plus importante pour les premiers coefficients rapport aux majorités de coefficients restants ( $a_i^2 \approx 0$ ). Cependant, pour les coefficients optimaux  $b_i$  c'est le cas inverse, les derniers coefficients optimaux  $b_i$  sont ceux de puissance la plus élevée par rapport aux majorités de coefficients restants ( $b_i^2 \approx 0$ ). Ce qui signifie selon l'équation (2.11) que l'inégalité (2.12) en dessous est vérifiée pour la plupart des instants  $n$ .

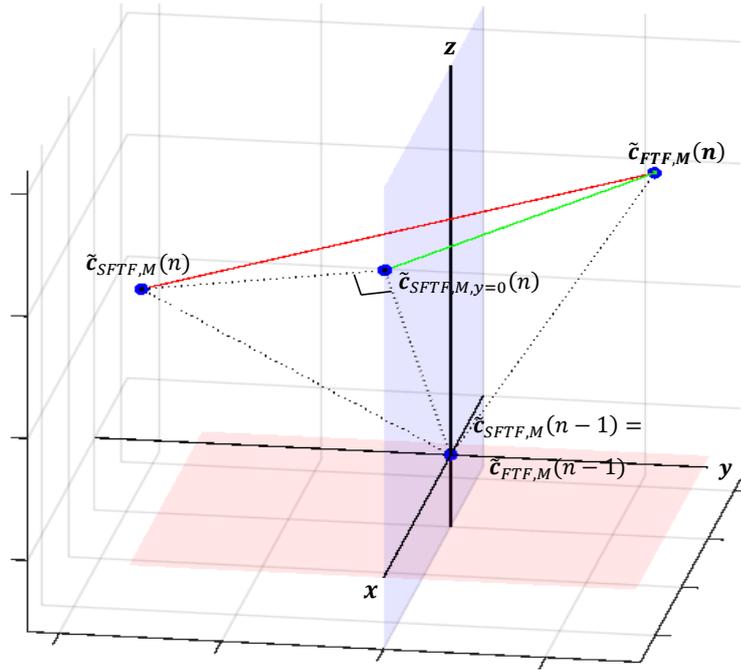
$$[\tilde{c}_{M+1}^{M+1}(n)]^2 \gg [\Delta_i(n)]^2 \quad (2.13)$$

On peut donc déduire que les coefficients  $\tilde{c}_{SFTF,M}^i(n)$  de l'algorithme *SFTF* de carré supérieur à  $[\tilde{c}_{M+1}^{M+1}(n)]^2$  (c-à-d  $[\tilde{c}_{SFTF,M}^i(n)]^2 > [\tilde{c}_{M+1}^{M+1}(n)]^2$ ), subissent moins l'impact de l'erreur d'élimination de prédicteur  $\mathbf{b}(n)$  :

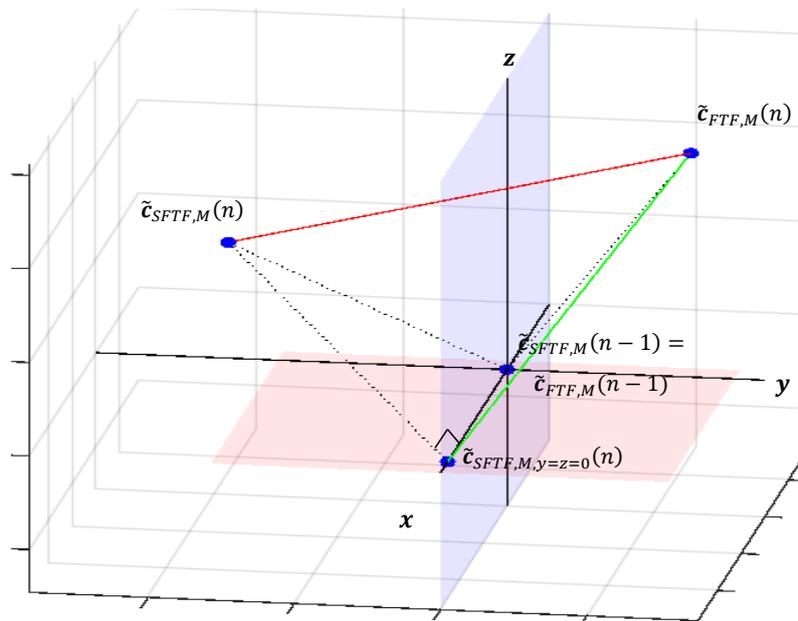
$$([\tilde{c}_{SFTF,M}^i(n)]^2 > [\tilde{c}_{M+1}^{M+1}(n)]^2 \text{ et } [\tilde{c}_{M+1}^{M+1}(n)]^2 \gg [\Delta_i(n)]^2) \Rightarrow ([\tilde{c}_{SFTF,M}^i(n)]^2 \gg [\Delta_i(n)]^2) \\ \Leftrightarrow (\varepsilon_i^2(n) \approx 0)$$

Cependant, les autres coefficients  $\tilde{c}_{SFTF,M}^i(n)$ , qui possèdent une faible puissance, ne contribuent pas à l'adaptation des filtres transversaux et par conséquent la vitesse convergence de l'algorithme peut être ralenti. Cette analyse montre comment sélectionner les coefficients  $\tilde{c}_{SFTF,M}^i(n)$  de vecteur  $\tilde{\mathbf{c}}_{SFTF,M}(n)$  qui minimisent la quantité  $\varepsilon_i^2(n)$ , qui sont les coefficients de carré plus grand que  $[\tilde{c}_{M+1}^{M+1}(n)]^2$ , les autres coefficients ne contribuent pas à l'estimation du gain de Kalman dual, donc ils peuvent être ignorés en les remplaçant par des zéros. Si on souhaite sélectionner un nombre fixe  $N$  de coefficients du vecteur  $\tilde{\mathbf{c}}_{SFTF,M}(n)$ , les  $N$  coefficients d'indices  $i$  devraient correspondre aux coefficients de plus grande puissance.

Dans les figures ci-dessous, on montre la direction de mise à jour du gain de Kalman dual dans l'espace  $\mathbb{R}^3$  à trois dimensions ( $M = 3$ ). Dans la **Figure 2.1** on choisit  $N = 2$ , on met à jour deux coefficients de plus grande puissance (composantes "x" et "z") et la composante "y" est met à zéro. Dans la **Figure 2.2** on choisit  $N = 1$ , on mise à jour qu'un seul coefficient de plus grande puissance (composante "x") et on met à zéro les composantes "y" et "z".



**Figure 2.1.** Direction de mis à jour de  $\tilde{c}_{SFTF,M}(n)$ ,  $\tilde{c}_{FTF,M}(n)$  et  $\tilde{c}_{SFTF,N}(n)$  pour  $M = 3$  et  $N = 2$ .



**Figure 2.2.** Direction de mis à jour de  $\tilde{c}_{SFTF,M}(n)$ ,  $\tilde{c}_{FTF,M}(n)$  et  $\tilde{c}_{SFTF,N}(n)$  pour  $M = 3$  et  $N = 1$ .

D'après les figures ci-dessus, on peut constater que pour  $N = 2$ , la distance entre les vecteurs  $\tilde{\mathbf{c}}_{FTF,M}(n+1)$  et  $\tilde{\mathbf{c}}_{SFTF,N}(n+1)$  est optimale, d'autre part, pour le cas  $N = M$  et  $N = 1$  cette distance n'est plus optimale, cela signifie qu'une augmentation de nombre  $N$  il n'implique pas nécessairement une meilleure estimation du gain de Kalman dual par rapport à l'algorithme FTF. De même, réduire  $N$  ne l'est plus.

En gardant  $N$  coefficients de vecteur  $\tilde{\mathbf{c}}_{SFTF,M}^i(n)$  et en mettant les  $(M - N)$  coefficients restants à zéro (car ils n'influencent pas l'estimation du vecteur de gain de Kalman dual), on obtient les deux résultats suivants :

- Le calcul de produit scalaire du facteur  $\gamma(n)$  (c-à-d  $\mathbf{x}_M^T(n)\tilde{\mathbf{c}}_M(n)$ ) peut s'effectuer dans l'espace  $\mathbb{R}^N$  au lieu de  $\mathbb{R}^M$ , ce qui nous permet de réduire la complexité de calcul à  $(M - N)$  multiplications au lieu de  $M$ .
- La mise à jour des filtres transversaux  $\mathbf{a}(n)$  et  $\mathbf{w}(n)$  peut s'effectuer de manière partielle en mettant à jour uniquement les coefficients d'indice correspondent aux  $N$  coefficients significatifs de  $\tilde{\mathbf{c}}_M(n)$ , ce qui nous permet de réduire encore la complexité de calcul de mise à jour en effectuant uniquement  $3(M - N)$  multiplications au lieu de  $3M$ .

Avant de prouver les deux résultats mentionnés ci-dessus, commençons par décomposer les vecteurs  $\tilde{\mathbf{c}}_M(n)$ ,  $\mathbf{x}(n)$ ,  $\mathbf{w}(n)$  et  $\mathbf{a}(n)$  en deux vecteurs,  $\tilde{\mathbf{c}}_N(n)$ ,  $\tilde{\mathbf{c}}_{M-N}(n)$ ,  $\mathbf{x}_N(n)$ ,  $\mathbf{x}_{M-N}(n)$ ,  $\mathbf{w}_N(n)$ ,  $\mathbf{w}_{M-N}(n)$  et  $\mathbf{a}_N(n)$ ,  $\mathbf{a}_{M-N}(n)$ , respectivement avec :

- $\tilde{\mathbf{c}}_N(n)$  est un vecteur de  $N$  coefficients significatifs sélectionnés à partir de  $\tilde{\mathbf{c}}_{SFTF,M}(n)$ , les  $M - N$  coefficients restants, sont mises à zéro et placés dans le vecteur  $\tilde{\mathbf{c}}_{M-N}(n)$ .
- $\mathbf{x}_N(n)$  est un vecteur contient  $N$  coefficients sélectionnés à partir de  $\mathbf{x}(n)$ , de mêmes indices que les  $N$  coefficients significatifs triés de  $\tilde{\mathbf{c}}_{SFTF,M}(n)$  et placés dans le vecteur  $\tilde{\mathbf{c}}_N(n)$ . Les  $M - N$  coefficients de  $\mathbf{x}(n)$  qui restent, sont placés dans le vecteur  $\mathbf{x}_{M-N}(n)$ .
- Avec la même manière de décomposition de  $\mathbf{x}(n)$ , les vecteurs  $\mathbf{w}(n)$  et  $\mathbf{a}(n)$  sont décomposés en  $\mathbf{w}_N(n)$ ,  $\mathbf{w}_{M-N}(n)$  et  $\mathbf{a}_N(n)$ ,  $\mathbf{a}_{M-N}(n)$ , respectivement.

➤ La preuve du premier résultat :

En vertu de ces dernières décompositions nous écrivons l'expression de produit  $\mathbf{x}_M^T(n)\tilde{\mathbf{c}}_M(n)$  comme suite :

$$\begin{aligned}\mathbf{x}_M^T(n)\tilde{\mathbf{c}}_M(n) &= \begin{bmatrix} \mathbf{x}_N(n) \\ \mathbf{x}_{M-N}(n) \end{bmatrix}^T \begin{bmatrix} \tilde{\mathbf{c}}_N(n) \\ \tilde{\mathbf{c}}_{M-N}(n) \end{bmatrix} \\ &= \mathbf{x}_N^T(n)\tilde{\mathbf{c}}_N(n) + \mathbf{x}_{M-N}^T(n)\tilde{\mathbf{c}}_{M-N}(n)\end{aligned}\quad (2.14)$$

Puisque  $\tilde{\mathbf{c}}_{M-N}(n) = \mathbf{0}_{(M-N)\times 1}$ , On aura donc  $\mathbf{x}_M^T(n)\tilde{\mathbf{c}}_M(n) = \mathbf{x}_N^T(n)\tilde{\mathbf{c}}_N(n)$ , le nouveau facteur  $\gamma_{new}$  s'écrit comme suit :

$$\gamma_{new}(n) = \frac{1}{1 - \mathbf{x}_N^T(n)\tilde{\mathbf{c}}_N(n)} \quad (2.15)$$

➤ La preuve du deuxième résultat :

Les  $N$  et les  $M - N$  coefficients des vecteurs  $\mathbf{w}(n)$  et  $\mathbf{a}(n)$  sont respectivement notés  $\mathbf{w}_N(n)$ ,  $\mathbf{w}_{M-N}(n)$  et  $\mathbf{a}_N(n)$ ,  $\mathbf{a}_{M-N}(n)$ . Selon la définition de ces vecteurs, leurs équations de mis à jour s'écrit comme suite :

$$\begin{bmatrix} \mathbf{a}_N(n) \\ \mathbf{a}_{M-N}(n) \end{bmatrix} = \eta \left[ \begin{bmatrix} \mathbf{a}_N(n-1) \\ \mathbf{a}_{M-N}(n-1) \end{bmatrix} - \bar{e}(n)\gamma(n-1) \begin{bmatrix} \tilde{\mathbf{c}}_N(n-1) \\ \tilde{\mathbf{c}}_{M-N}(n-1) \end{bmatrix} \right] \quad (2.16)$$

Puisque  $\tilde{\mathbf{c}}_{M-N}(n-1) = \mathbf{0}_{(M-N)\times 1}$ , on obtient :

$$\begin{bmatrix} \mathbf{a}_N(n) \\ \mathbf{a}_{M-N}(n) \end{bmatrix} = \eta \left[ \begin{bmatrix} \mathbf{a}_N(n-1) - \bar{e}(n)\gamma(n-1)\tilde{\mathbf{c}}_N(n-1) \\ \mathbf{a}_{M-N}(n-1) \end{bmatrix} \right] \quad (2.17)$$

Où  $\eta$  est un scalaire choisit très proche de 1, nous obtenons donc  $\mathbf{a}_{M-N}(n) \approx \mathbf{a}_{M-N}(n-1)$ . C'est-à-dire le calcul du vecteur  $\mathbf{a}(n)$ , nécessite que l'estimation de  $\mathbf{a}_N(n)$  qui est donnée par l'équation suivante :

$$\mathbf{a}_N(n) = \eta[\mathbf{a}_N(n-1)\bar{e}(n)\gamma(n-1)\tilde{\mathbf{c}}_N(n-1)] \quad (2.18)$$

Le vecteur  $\mathbf{a}_{M-N}(n)$  est ignoré et n'est pas mis à jour. De la même manière, nous mettons à jour  $\mathbf{w}_N(n)$  et  $\mathbf{w}_{M-N}(n)$  comme suit :

$$\begin{bmatrix} \mathbf{w}_N(n) \\ \mathbf{w}_{M-N}(n) \end{bmatrix} = \begin{bmatrix} \mathbf{w}_N(n-1) \\ \mathbf{w}_{M-N}(n-1) \end{bmatrix} - e(n)\gamma(n) \begin{bmatrix} \tilde{\mathbf{c}}_N(n) \\ \tilde{\mathbf{c}}_{M-N}(n) \end{bmatrix} \quad (2.19)$$

Puisque  $\tilde{\mathbf{c}}_{M-N}(n) = \mathbf{0}_{(M-N)\times 1}$ , on obtient :

$$\begin{bmatrix} \mathbf{w}_N(n) \\ \mathbf{w}_{M-N}(n) \end{bmatrix} = \begin{bmatrix} \mathbf{w}_N(n-1) - e(n)\gamma(n)\tilde{\mathbf{c}}_N(n) \\ \mathbf{w}_{M-N}(n-1) \end{bmatrix} \quad (2.20)$$

Pour calculer  $\mathbf{w}(n)$ , nous avons donc besoin uniquement l'estimation de  $\mathbf{w}_N(n)$ , qui est donnée par l'équation suivante :

$$\mathbf{w}_N(n) = \mathbf{w}_N(n-1)e(n)\gamma(n)\mathbf{w}_N(n) \quad (2.21)$$

On résume l'algorithme proposé nommé PU-SFTF (cas mono canal) dans le **Tableau 2.3** et l'algorithme TC-PU-SFTF pour la séparation aveugle des source FBSS dans **Tableau 2.4**.

**Tableau 2.3.** Algorithme 1 (PU-SFTF) proposé (mono canal).

---

**Initialisation**

$$\mathbf{a}(0) = \mathbf{0}_{M \times 1}, \tilde{\mathbf{c}}_M(0) = \mathbf{0}_{M \times 1}, \gamma(0) = 1, \alpha(0) = E_0\lambda^N, \mathbf{w}(0) = \mathbf{0}_{M \times 1}, \lambda \approx \eta \approx 1, c \approx 0$$


---

**Prédiction**

$$\bar{\mathbf{e}}(n) = \mathbf{x}_{M+1}^T(n) \begin{bmatrix} 1 \\ -\mathbf{a}(n) \end{bmatrix}$$

$$\begin{bmatrix} \tilde{\mathbf{c}}_M(n) \\ * \end{bmatrix} = \begin{bmatrix} 0 \\ \tilde{\mathbf{c}}_M(n-1) \end{bmatrix} - \frac{\bar{\mathbf{e}}(n)}{\lambda\alpha(n-1) + c} \begin{bmatrix} 1 \\ -\mathbf{a}(n-1) \end{bmatrix}$$

Sortir  $\tilde{\mathbf{c}}_N(n)$  à partir de  $\tilde{\mathbf{c}}_M(n)$

Sortir  $\mathbf{a}_N(n-1)$  à partir de  $\mathbf{a}(n-1)$

Sortir  $\mathbf{w}_N(n-1)$  à partir de  $\mathbf{w}(n-1)$ .

$$\alpha(n) = \lambda\alpha(n-1) + \gamma_{new}(n-1)\bar{\mathbf{e}}^2(n)$$

$\mathbf{a}_N(n) = \eta[\mathbf{a}_N(n-1)\bar{\mathbf{e}}(n)\gamma_{new}(n-1)\tilde{\mathbf{c}}_N(n-1)]$ ; Mis à jour uniquement les  $N$  coefficients de  $\mathbf{a}(n)$ .

$$\gamma_{new}(n) = \frac{1}{1 - \mathbf{x}_N^T(n)\tilde{\mathbf{c}}_N(n)}$$

**Filtrage**

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

$\mathbf{w}_N(n) = \mathbf{w}_N(n-1)e(n)\gamma_{new}(n)\mathbf{w}_N(n)$  ; Mis à jour uniquement les  $N$  coefficients de  $\mathbf{w}(n)$

---

**Tableau 2.4.** Algorithme 1 proposé pour la séparation aveugle des source FBSS (TC-PU-SFTF)

---

**Initialisation**

$$\mathbf{a}_1(0) = \mathbf{a}_2(0) = \mathbf{0}_{M \times 1}, \tilde{\mathbf{c}}_{1,M}(0) = \tilde{\mathbf{c}}_{2,M}(0) = \mathbf{0}_{M \times 1}, \gamma_{new,1}(0) = \gamma_{new,2}(0) = 1, \alpha_1(0) = \alpha_2(0) = E_0 \lambda^N, \mathbf{w}_1(n) = \mathbf{w}_2(n) = \mathbf{0}_{M \times 1}, \lambda_1 \approx \lambda_2 \approx \eta_1 \approx \eta_2 \approx 1. c \approx 0;$$


---

**Prédiction**

$$\bar{\mathbf{e}}_1(n) = \mathbf{m}_{2,M+1}^T(n) \begin{bmatrix} 1 \\ -\mathbf{a}_1(n) \end{bmatrix}; \bar{\mathbf{e}}_2(n) = \mathbf{m}_{1,M+1}^T(n) \begin{bmatrix} 1 \\ -\mathbf{a}_2(n) \end{bmatrix}$$

$$\begin{bmatrix} \tilde{\mathbf{c}}_{1,M}(n) \\ * \end{bmatrix} = \begin{bmatrix} 0 \\ \tilde{\mathbf{c}}_{1,M}(n-1) \end{bmatrix} - \frac{\bar{\mathbf{e}}_1(n)}{\lambda_1 \alpha_1(n-1) + c} \begin{bmatrix} 1 \\ -\mathbf{a}_1(n-1) \end{bmatrix}$$

$$\begin{bmatrix} \tilde{\mathbf{c}}_{2,M}(n) \\ * \end{bmatrix} = \begin{bmatrix} 0 \\ \tilde{\mathbf{c}}_{2,M}(n-1) \end{bmatrix} - \frac{\bar{\mathbf{e}}_2(n)}{\lambda_2 \alpha_2(n-1) + c} \begin{bmatrix} 1 \\ -\mathbf{a}_2(n-1) \end{bmatrix}$$

Sortir  $\tilde{\mathbf{c}}_{1,N}(n)$  et  $\tilde{\mathbf{c}}_{2,N}(n)$  à partir de  $\tilde{\mathbf{c}}_{1,M}(n)$  et  $\tilde{\mathbf{c}}_{2,M}(n)$  respectivement.

Sortir  $\mathbf{a}_{1,N}(n-1)$  et  $\mathbf{a}_{2,N}(n-1)$  à partir de  $\mathbf{a}_1(n-1)$  et  $\mathbf{a}_2(n-1)$  respectivement.

Sortir  $\mathbf{w}_{1,N}(n-1)$  et  $\mathbf{w}_{2,N}(n-1)$  à partir de  $\mathbf{w}_1(n-1)$  et  $\mathbf{w}_2(n-1)$  respectivement.

$$\alpha_1(n) = \lambda \alpha_1(n-1) + \gamma_{1,new} \bar{\mathbf{e}}_1^2(n); \alpha_2(n) = \lambda \alpha_2(n-1) + \gamma_{2,new} \bar{\mathbf{e}}_2^2(n)$$

$\mathbf{a}_{1,N}(n) = \eta_1 [\mathbf{a}_{1,N}(n-1) - \bar{\mathbf{e}}_1(n) \gamma_{new,1}(n-1) \tilde{\mathbf{c}}_{1,N}(n-1)]$ ; Mis à jour uniquement les  $N$  coefficients.

$\mathbf{a}_{2,N}(n) = \eta_2 [\mathbf{a}_{2,N}(n-1) - \bar{\mathbf{e}}_2(n) \gamma_{new,2}(n-1) \tilde{\mathbf{c}}_{2,N}(n-1)]$ ; Mis à jour uniquement les  $N$  coefficients.

$$\gamma_{new,1}(n) = \frac{1}{1 - \mathbf{m}_{2,N}^T \tilde{\mathbf{c}}_{1,N}(n)}; \gamma_{new,2}(n) = \frac{1}{1 - \mathbf{m}_{1,N}^T \tilde{\mathbf{c}}_{2,N}(n)}$$

**Filtrage**

$$e_1(n) = m_1(n) - \mathbf{w}_1^T(n-1) \mathbf{m}_{2,M}(n)$$

$$e_2(n) = m_2(n) - \mathbf{w}_2^T(n-1) \mathbf{m}_{1,M}(n)$$

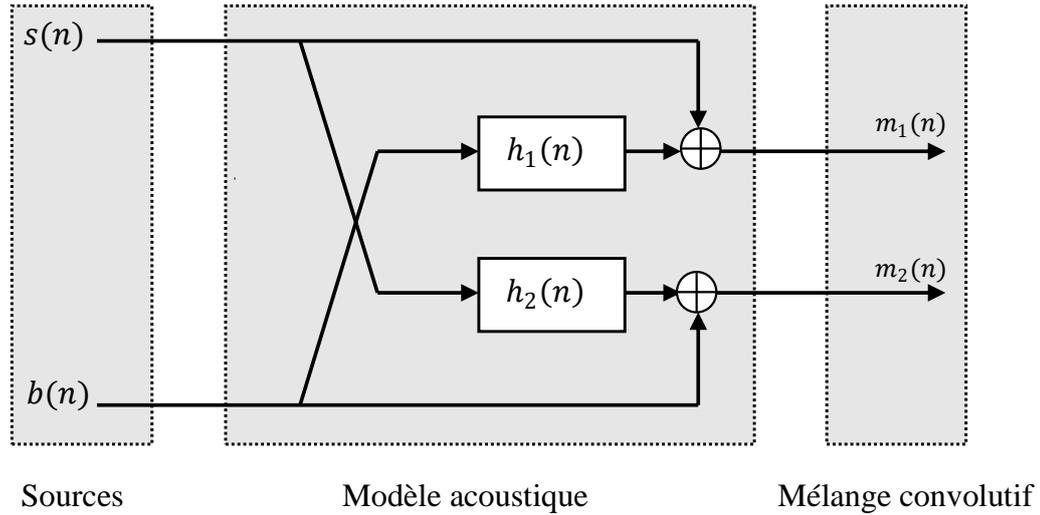
$\mathbf{w}_{1,N}(n) = \mathbf{w}_{1,N}(n-1) - e_1(n) \gamma_{new,1}(n) \tilde{\mathbf{c}}_{1,N}(n)$ ; Mis à jour uniquement les  $N$  coefficients.

$\mathbf{w}_{2,N}(n) = \mathbf{w}_{2,N}(n-1) - e_2(n) \gamma_{new,2}(n) \tilde{\mathbf{c}}_{2,N}(n)$ ; Mis à jour uniquement les  $N$  coefficients.

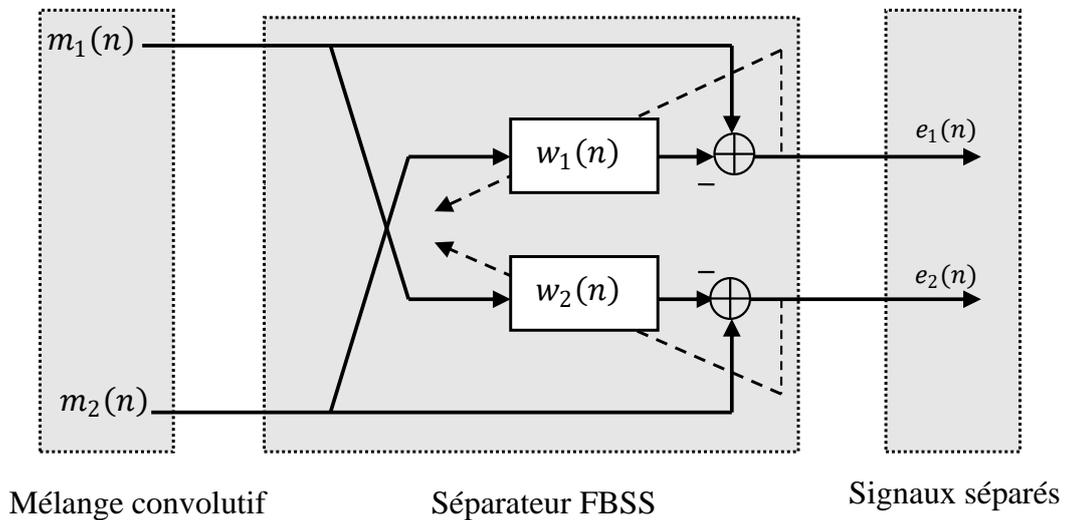
---

## 2.4. Simulation et résultats

Dans cette section, nous présentons les résultats de simulation de l'algorithme proposé (c-à-d Algorithme 1) pour la réduction de bruit par la séparation aveugle des sources, ainsi que la comparaison avec les algorithmes (c-à-d TC-NLMS et TC-SFTF [19]). Les mélanges sont obtenus en utilisant le modèle convolutive présenté dans la **Figure 2.3**. Lors de la simulation, nous nous concentrons sur le signal de parole estimé à la sortie  $e_1(n)$  du séparateur **Figure 2.4**.

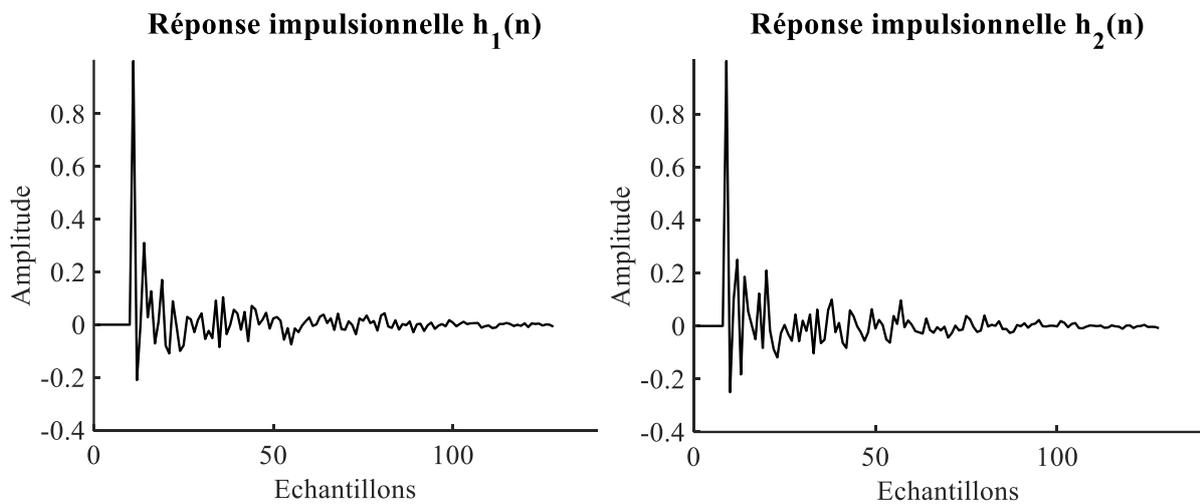


**Figure 2.3.** Modèle convolutive simplifié de simulation.

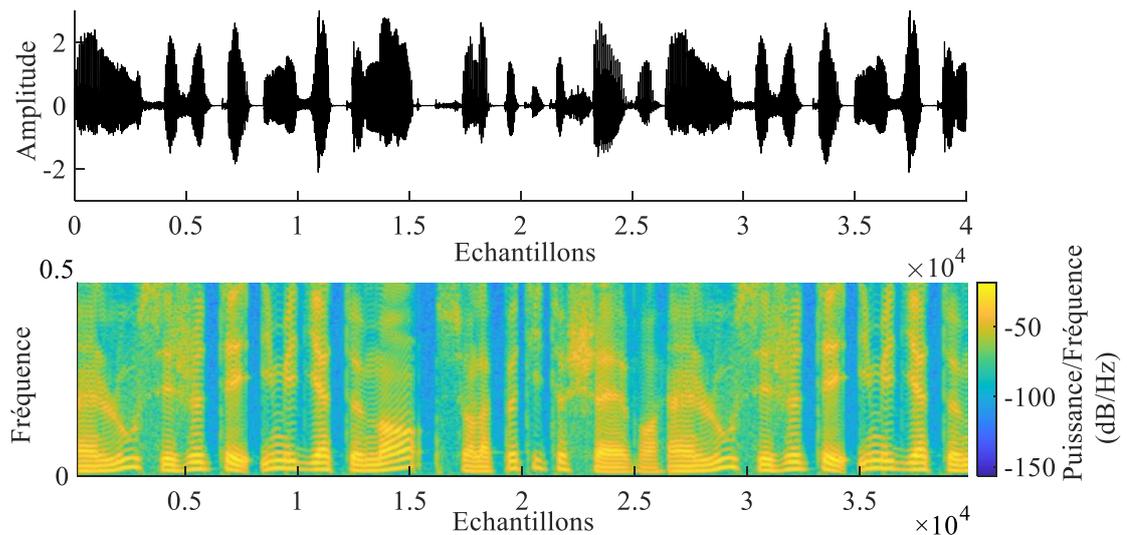


**Figure 2.4.** Séparateur des sources aveugle, structure direct (FBSS).

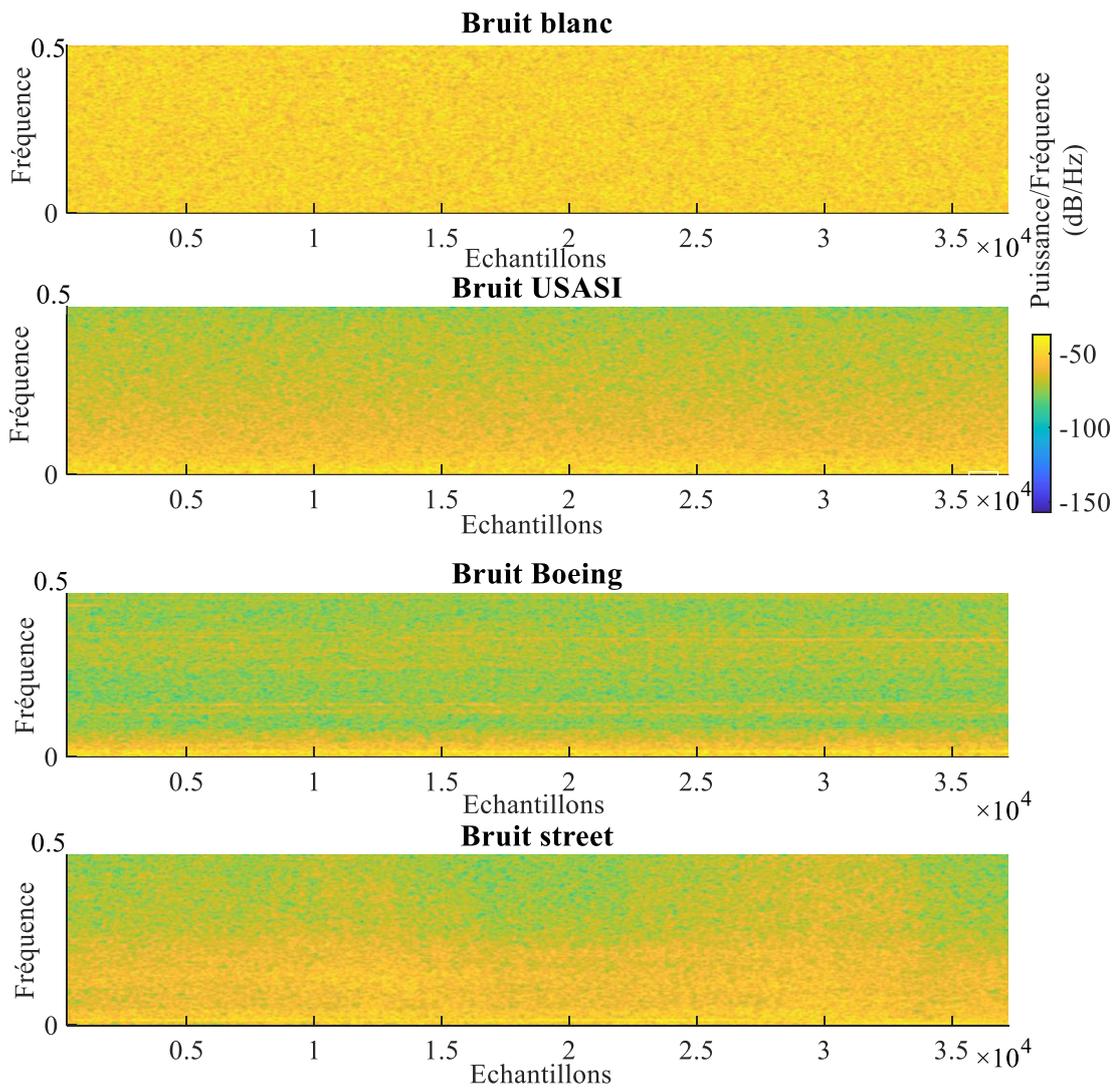
La **Figure 2.5** montre les réponses impulsionnelles (RIs)  $h_1(n)$  et  $h_2(n)$  utilisés dans cette simulation, la longueur des deux réponses est égale à 128 échantillons, ces RIs sont générées en utilisant le modèle donné dans la référence [8]. Dans la **Figure 2.6** on montre l'évolution temporelle du signal parole pur utilisé dans cette simulation, ainsi que son spectrogramme. Le signal de bruit  $b(n)$  il peut être un bruit blanc, USASI (United States of America Standard Institute, maintenant est ANSI), bruit d'avion Boeing mesuré à l'intérieur de la cabine pilote ou un bruit de rue (Street). Ces bruits sont échantillonnés à une fréquence d'échantillonnage  $f_s = 8 \text{ kHz}$ , le spectrogramme de chacun de ces bruits est présenté dans **Figure 2.7**.



**Figure 2.5.** Les réponses impulsionnelles  $h_1(n)$  et  $h_2(n)$ .

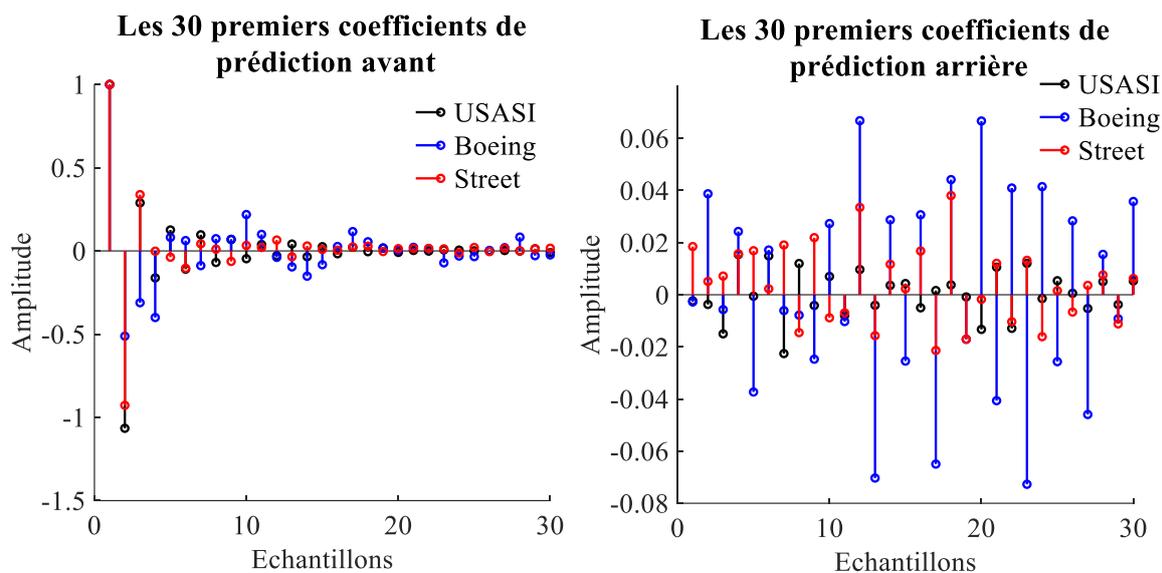


**Figure 2.6.** Évolution temporelle de signal parole pur et son spectrogramme.



**Figure 2.7.** Les spectrogrammes des bruits

Pour comparer les coefficients du modèle AR des bruits, nous représentons dans la **Figure 2.8** les 30 premiers coefficients des prédicteurs avant et arrière des bruits utilisés dans notre simulation. Le bruit blanc évidemment possède un seul coefficient de prédiction non nul (le premier coefficient).



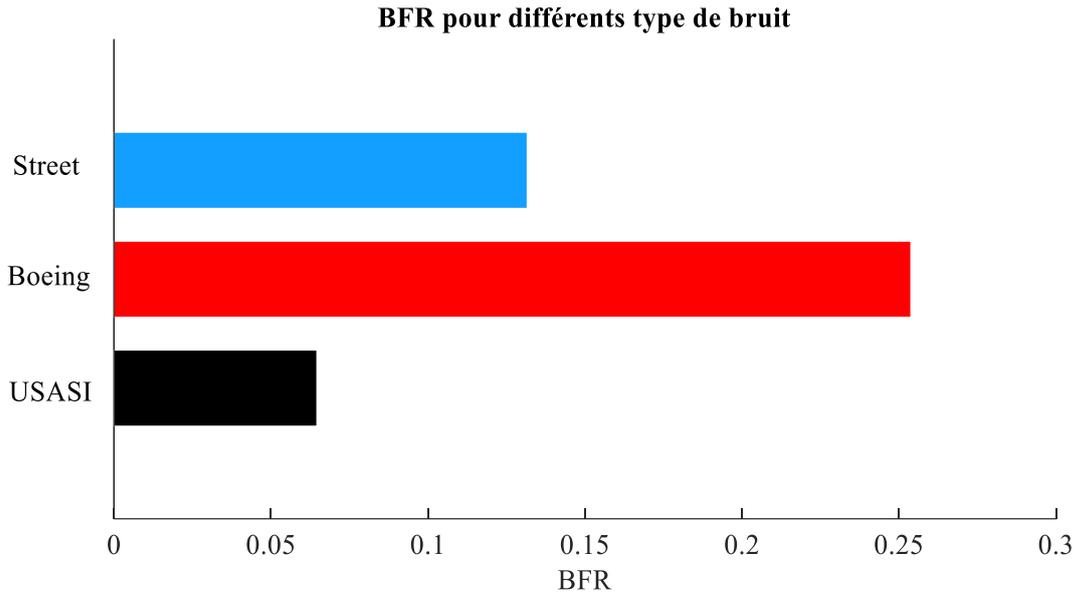
**Figure 2.8.** Les 30 premiers coefficients de prédiction avant et arrière.

Pour être plus objectif, nous proposons de définir l'indicateur appelé "rapport des prédicteurs arrière-avant (Backward to Forward Ratio)" ( $BFR$ ) donné par l'expression suivante :

$$BFR = \frac{\sum_{i=0}^{P-1} |b_i|}{\sum_{i=0}^{P-1} |a_i|} \quad (2.22)$$

Cet indicateur va nous donner une idée sur la puissance des  $P$  premiers prédicteurs arrière (c-à-d  $P$  derniers prédicteurs avant) par rapport aux  $P$  premiers prédicteurs avant. Plus cet indicateur est élevé, plus le prédicteur arrière contribue au processus d'adaptation. Par conséquent, pour un  $BFR$  élevé, l'algorithme TC-SFTF s'influe fortement par l'annulation du filtre prédicteur arrière, tandis que l'algorithme proposé peut apporter une amélioration dans la convergence si l'adaptation est uniquement effectuée sur les coefficients de grande puissance qui conduisent à  $\varepsilon_i^2(n) \approx 0$ .

Pour chaque bruit, on montre dans la **Figure 2.9** la valeur de l'indicateur  $BFR$ . À partir de cette figure, nous pouvons déduire que le bruit Boeing utilisé dans cette simulation possède une valeur  $BFR$  élevé (0.2538), ce qui nécessite un modèle AR d'ordre élevé. Pour le bruit de rue (Street), le  $BFR$  est plus faible (0.1316), et pour le bruit USASI, il est encore plus faible (0.0646).



**Figure 2.9.** L'indicateur BFR pour différents bruits.

Afin d'évaluer les performances de l'algorithme proposé, On utilise les critères suivants [35] :

- **L'évolution temporelle** du signal rehaussé à la sortie  $e_1(n)$ .
- **Le désajustement (système mismatch  $SM$ )** pour évaluer la vitesse de convergence du filtre adaptatives  $w_1$  vers la réponse  $h_1$ , il est calculé à partir de l'expression suivante :

$$SM(n)_{dB} = 10 \log \left[ \frac{\|h_1(n) - w_1(n)\|^2}{\|h_1(n)\|^2} \right] \quad (2.23)$$

Avec  $\|\cdot\|$  dénote la norme Euclidienne.

- **L'erreur quadratique moyenne segmentée ( $EQMSeg$ )**: Permet de quantifier la puissance de bruit à la sortie de séparateur pour chaque algorithme, le  $EQMSeg$  est calculé à la sorite  $e_1(n)$  par l'expression suivante :

$$EQMSeg_{dB} = 10 \log [\sum_{i=0}^{U-1} |e_1(i)|^2] \quad (2.24)$$

Avec  $U$  est la longueur de segment. On note que ce critère est estimé uniquement pendant les périodes de silence.

- **Le rapport signal sur bruit segmenté ( $RSB Seg$ )**: Il quantifie les performances de réduction du bruit pour chaque algorithme. Le  $RSB Seg$  est quantifié comme suit :

$$RSB Seg_{dB} = 10 \log \left[ \frac{\sum_{i=0}^{U-1} |s(i)|^2}{\sum_{i=0}^{U-1} |s(i) - e_1(i)|^2} \right] \quad (2.25)$$

Où  $U$  représente le nombre d'échantillons inclus dans le calcul de  $RSB Seg$ . Le  $RSB Seg$  est évalué uniquement pendant les périodes d'activité vocale.

- **La Complexité de calcul (CC) :** La CC est évaluée pour chaque algorithme, elle est définie par le nombre de multiplications, additions et divisions à chaque itération d'adaptation. Dans cette simulation, nous nous intéressons uniquement au nombre de multiplications, car la complexité de calcul due aux opérations d'additions et de divisions est négligeable. Les paramètres utilisés dans cette simulation sont présentés dans le **Tableau 2.5**.

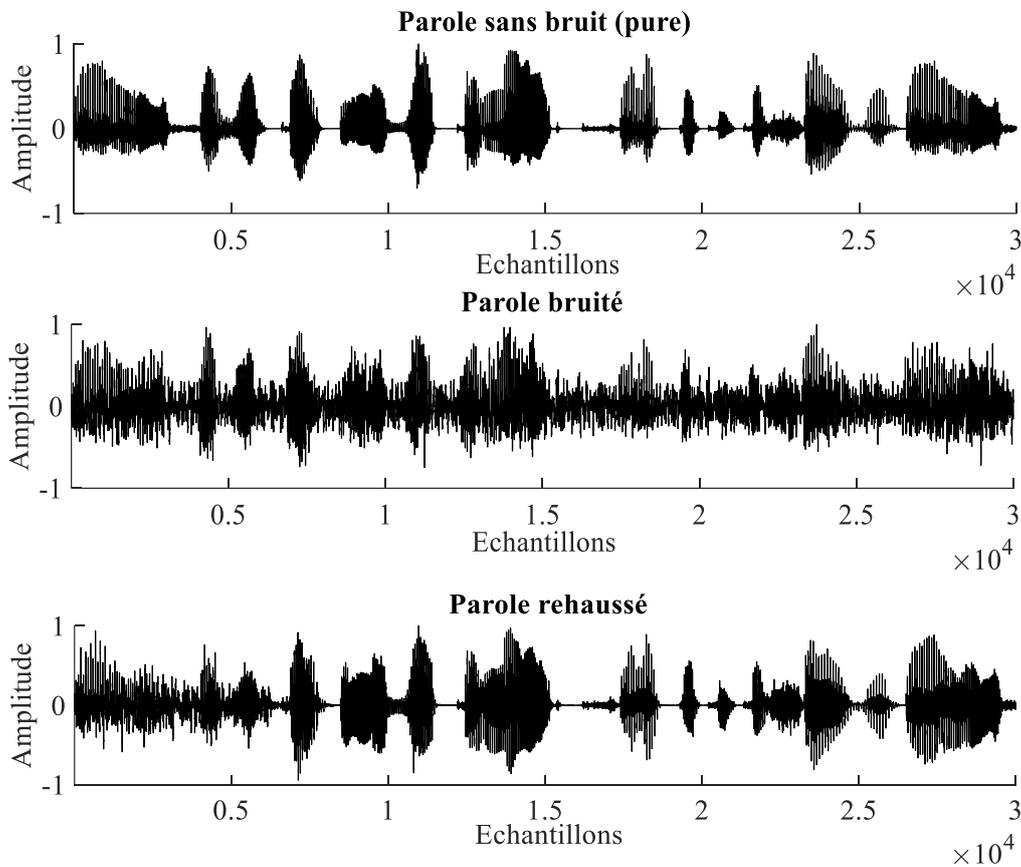
**Tableau 2.5.** Paramètres de simulation.

	Signal parole pur (sans bruit) : $s(n)$ .
Signaux	Signal de bruit $b(n)$ : Bruit blanc, USASI, Boeing et Street.  Fréquence d'échantillonnage : $f_s = 8 \text{ kHz}$ .
Paramètres des filtres adaptative	Deux filtres adaptative $w_1(n), w_2(n)$ de longueur : $M_1 = M_2 = 128$ .
Paramètres d'algorithme TC-NLMS	Le pas d'adaptation : $\mu_1 = 1, \mu_2 = 1$ . (Meilleure Vitesse de convergence)
Paramètres d'algorithme TC-SFTF	Facteurs d'oubli : $\lambda_1 = \lambda_2 = 0.95$ .  Facteurs de fuite : $\eta_1 = \eta_2 = 0.95$  Petite constante : $c = 0.001$
Paramètres d'algorithme 1 (proposé)	Facteurs d'oubli : $\lambda_1 = \lambda_2 = 0.95$ .  Facteurs de fuite : $\eta_1 = \eta_2 = 0.95$ .  Petite constante : $c = 0.001$ .

- **L'évolution temporelle**

En utilisant les paramètres de simulation donnés dans le **Tableau 2.5**. La **Figure 2.10** montre l'évolution temporelle du signal parole sans bruit, signal parole bruité et rehaussé obtenu par l'algorithme proposé (c-à-d Algorithme 1). Le signal de bruit est de type USASI, le rapport signal sur bruit à l'entrée égal à 0 dB, et le bloque d'adaptation partielle  $N = 80$ . A partir de cette figure nous pouvons dire que l'algorithme TC-PU-SFTF proposé conduit à une réduction

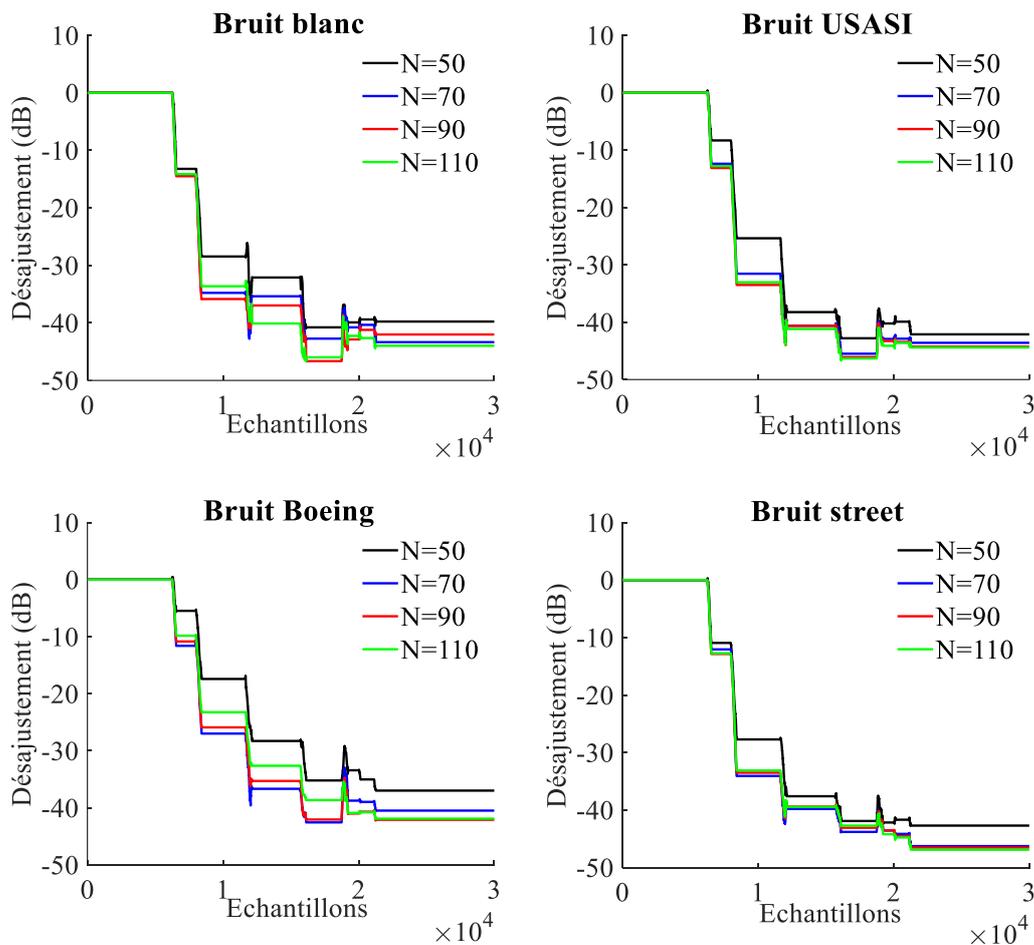
du bruit acoustique élevé même dans des conditions très bruyantes, cela indique l'efficacité de l'algorithme proposé de restaurer le signal de parole dégradé par le bruit.



**Figure 2.10.** Evolution temporelle obtenu par l'algorithme proposé (TC-PU-SFTF). Le bruit est de type : USAS. Bloc d'adaptation  $N = 80$  et  $RSB$  d'entrée =  $0$  dB.

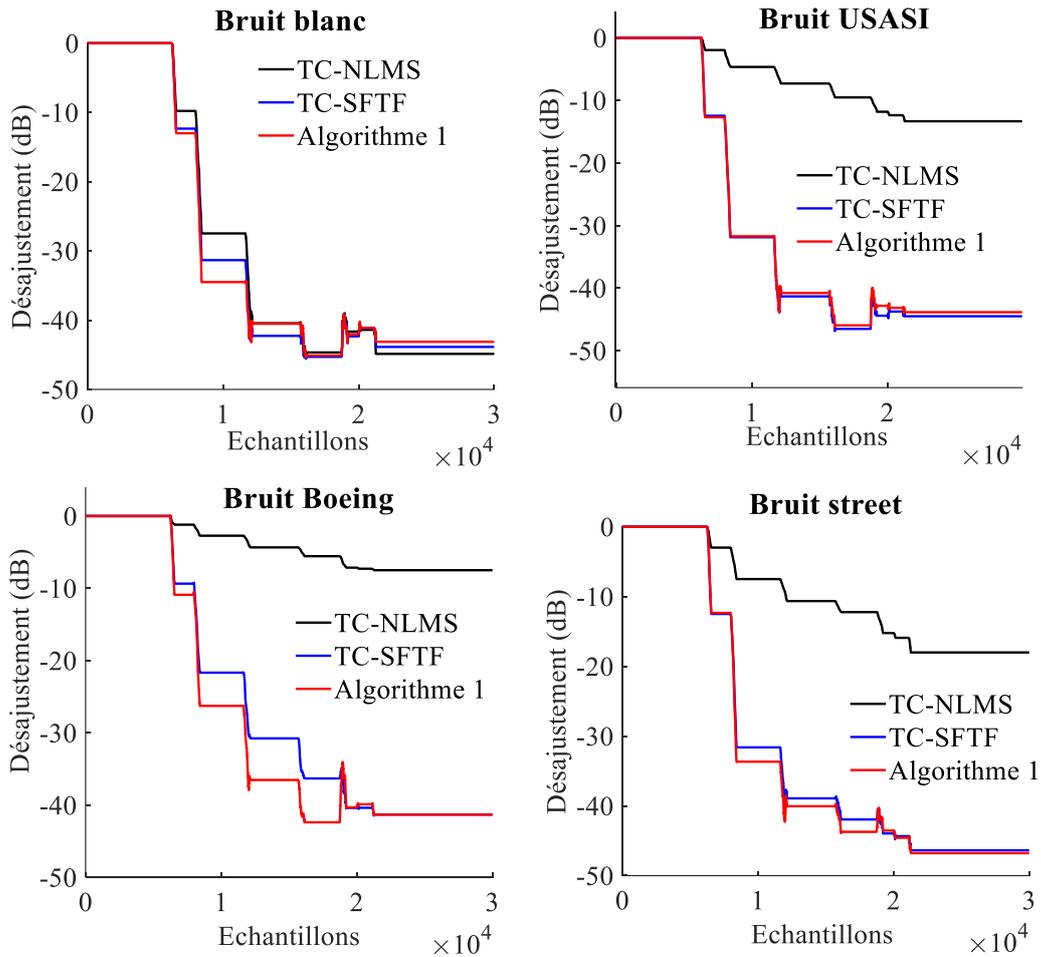
- **Le désajustement** (système mismatch)

Dans cette section, nous évaluons le système mismatch ( $SM$ ), nous avons testé l'algorithme 1 proposé (c-à-d TC-PU-SFTF) dans plusieurs environnements bruyants, à savoir le bruit blanc, USASI, Boeing et un bruit de rue (Street). Les rapports signal/bruit d'entrée des deux mélanges  $m_1(n)$  et  $m_2(n)$  sont égal à  $0$  dB. Dans la **Figure 2.11**, nous rapportons les valeurs de  $SM$  obtenues par l'algorithme proposé pour différentes valeurs de blocs d'adaptation  $N$ . Nous avons choisi différentes valeurs de  $N$  ( $N = 50, 70, 90$  et  $110$ ). On constate qu'une grande valeur de bloque d'adaptation  $N$  n'implique pas nécessairement une vitesse de convergence importante, surtout avec un BFR élevé (Bruit Boeing). Le choix  $N = 70$  donne de meilleures performances que le choix de  $N = 110$ , ce résultat confirme l'analyse faite dans la section 2.3 (voir l'interprétation basée sur la **Figure 2.1** et la **Figure 2.2**).



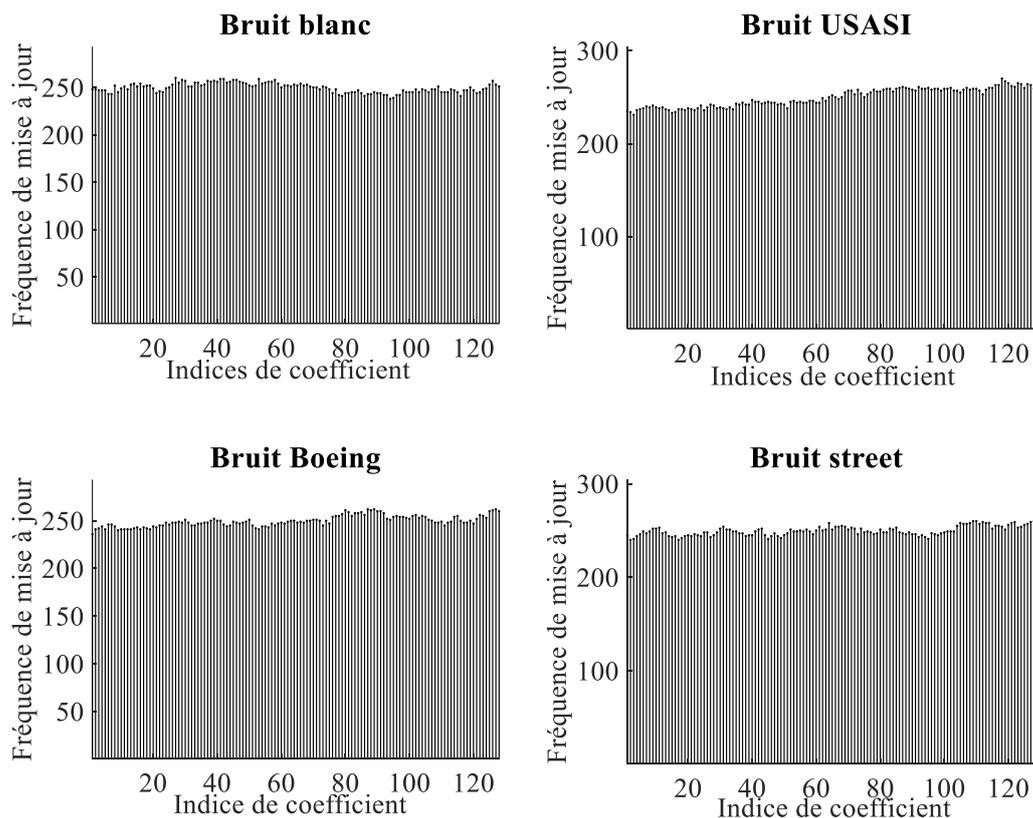
**Figure 2.11.** L'évaluation du désajustement d'algorithme proposé (Algorithme 1) pour différents choix de  $N$ , le  $RSB$  d'entrée = 0dB

Dans la **Figure 2.12**, nous comparons les désajustements obtenus par l'algorithme proposé avec les autres algorithmes (c-à-d TC-NLMS et TC-SFTF), pour le choix de  $N = 80$  et pour de rapport signal sur bruit égal à 0 dB, nous observons que le taux de convergence de l'algorithme classique TC-NLMS est le plus faible, tandis que les autres algorithmes possèdent pour un taux de convergence important. En comparant l'algorithme 1 proposé (c-à-d TC-PU-SFTF) avec l'algorithme TC-SFTF classique, nous constatons une légère amélioration en termes de taux de convergence, notamment avec le bruit Boeing qui présente un BFR élevé (voir la **Figure 2.9**). Ce résultat est hautement attendu selon notre analyse effectuée dans la section 2.3.

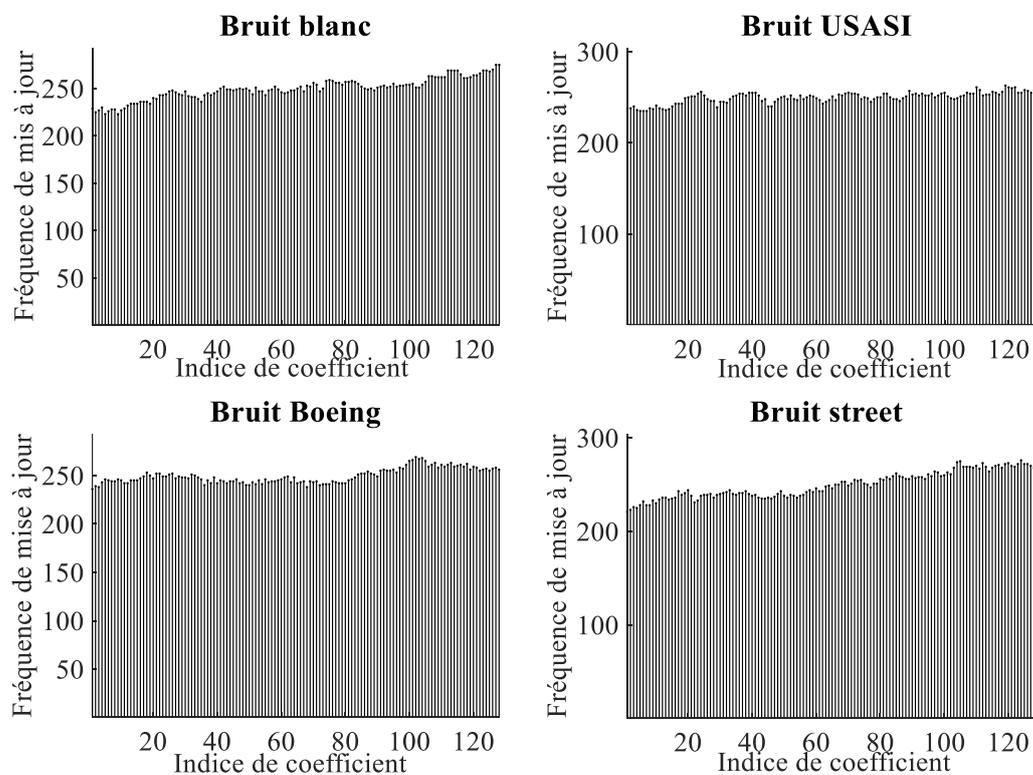


**Figure 2.12.** Comparaison de désajustement pour les algorithmes TC-NLMS, TC-SFTF et l’algorithme proposé (Algorithme 1) pour différents types de bruit,  $N = 80$ ,  $RSB$  d’entrée=0 dB.

Il est important de montrer dans les deux expériences ci-dessus (c-à-d **Figure 2.12**) la fréquence de mise à jour pour chaque coefficient des filtres adaptatifs pendant le processus d'adaptation. Pour cette raison, on présente dans les **Figure 2.13** et **Figure 2.14** pour chaque coefficient des filtres adaptatifs, la fréquence de mise à jour. Cela est donné pour deux phases, pour les 500 premières itérations (phase d'apprentissage avant la convergence) et pour les 500 dernières itérations (après la convergence). À partir des **Figure 2.13** et **Figure 2.14**, nous pouvons voir que pour les deux phases de mise à jour, les coefficients des filtres adaptatifs ont des fréquences d'adaptation très proches. Cela nous donne une idée de la distribution des fréquences de mise à jour des coefficients des filtres adaptatifs, qui ressemble à une distribution uniforme.



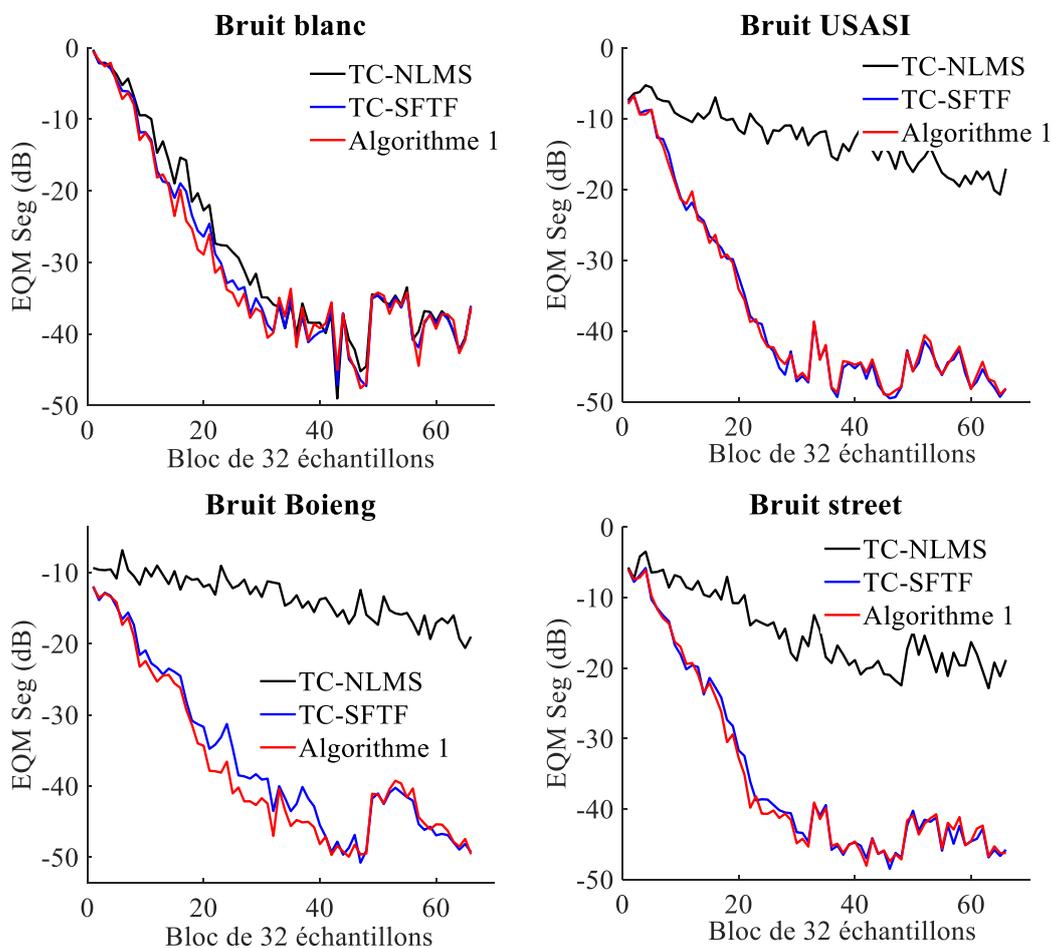
**Figure 2.13.** Fréquence d'adaptation pour chaque coefficient du filtre adaptative  $w_1(n)$  pour les 500 première itération avec  $N = 80$ ,  $RSB$  d'entrée=0 dB.



**Figure 2.14.** Fréquence d'adaptation pour chaque coefficient du filtre adaptative  $w_1(n)$  pour les 500 dernière itération, avec  $N=M/2$ ,  $RSB$  d'entrée=0dB.

- **Erreur quadratique moyenne segmentée (*EQM Seg*)**

Dans cette section, nous allons évaluer l'erreur quadratique moyenne segmentée (*EQM Seg*), en comparant l'algorithme 1 proposé avec les algorithmes TC-NLMS et TC-SFTF. Les résultats obtenus sont donnés dans la **Figure 2.15**, cette figure montre de bonnes performances de l'algorithme proposé pour tous les types de bruits utilisés dans la simulation. On peut conclure à partir de la **Figure 2.15** que l'algorithme proposé possède un bon taux de convergence et un faible bruit résiduel en sortie avec une complexité de calcul plus faible que l'algorithme TC-SFTF.

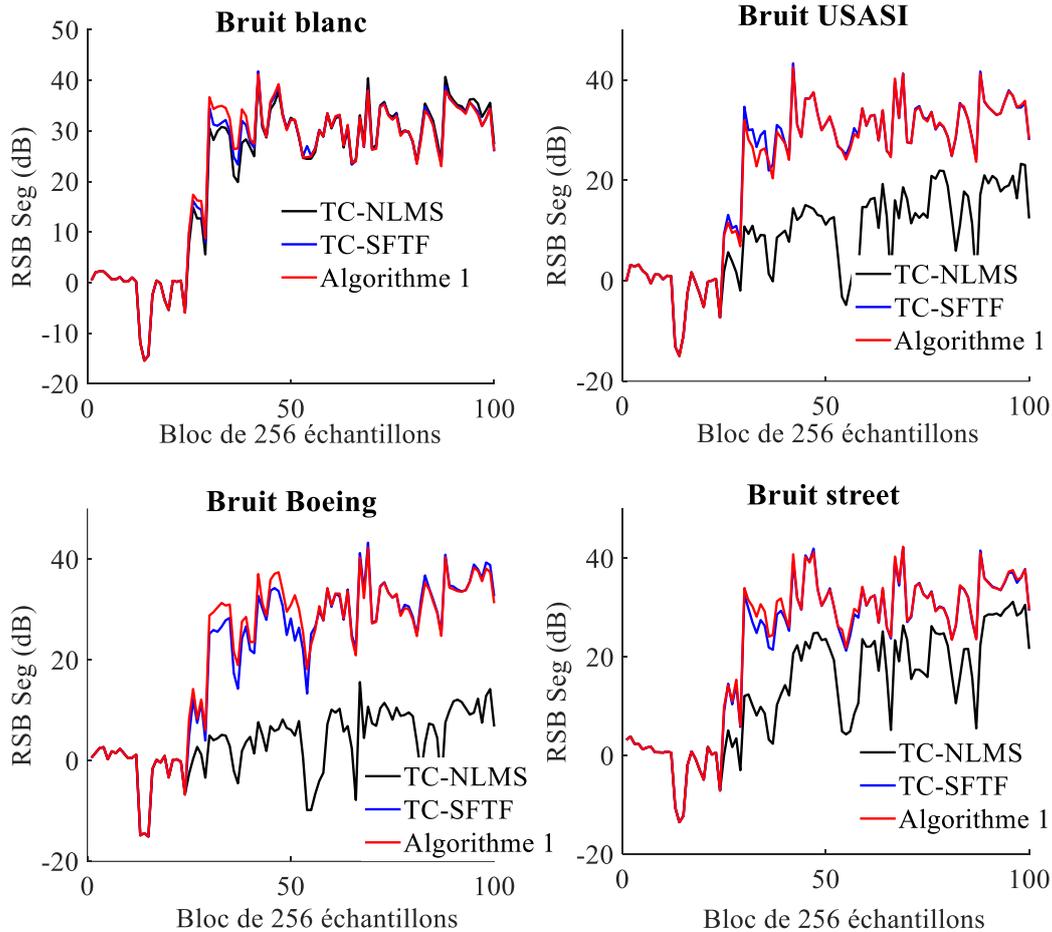


**Figure 2.15.** Evaluation de l'*EQM Seg* pour les algorithmes TC-NLMS, TC-SFTF et l'algorithme 1 proposé pour  $N = 80$  et  $RSB$  d'entrée=0dB.

- **Rapport signal sur bruit segmenté (*RSB Seg*) :**

Nous évaluons le rapport signal sur bruit segmenté (*RSB Seg*) de l'algorithme proposé par rapport aux deux algorithmes TC-NLMS et TC-SFTF. Sur la **Figure 2.16**, nous présentons les résultats obtenus du *RSB Seg* en sortie pour chaque algorithme. D'après ces résultats, on

constater que pour tous les types de bruit, les deux algorithmes TC-SFTF et TC-PU-SFTF donne un meilleur résultat par rapport à l'algorithme TC-NLMS. Nous observons aussi un comportement très similaire de l'algorithme 1 proposé par rapport à l'algorithme TC-SFTF. Ces résultats prouvent l'efficacité de l'algorithme proposé pour la réduction du bruit acoustique dans des situations très bruyant, et pour différents types de bruit, tout avec une complexité plus réduite par rapport à l'algorithme original TC-SFTF.



**Figure 2.16.** L'évaluation du *RSB Seg* de sortie pour les algorithmes TC-NLMS, TC-SFTF et Algorithme 1 proposé,  $N = 80$  et *RSB* d'entrée = 0 dB.

- **Complexité de calcul**

Dans le **Tableau 2.6** et **Tableau 2.7** nous donnons la complexité de calcul, défini par le nombre de multiplications par itération. Pour les trois algorithmes, (c-à-d TC-NLMS, TC-SFTF et l'algorithme 1 proposé), nous notons que l'algorithme 1 proposé possède une complexité de calcul plus faible par rapport à sa version originale (c-à-d TC-SFTF). Par exemple, pour  $N = M/4$ , l'algorithme proposé ne nécessite que  $8M$  multiplications par itération, ce qui est

significativement inférieur à  $14M$  multiplications par itération requis pour la version original TC-SFTF. L'algorithme TC-NLMS a toujours une complexité plus faible car il n'a pas une partie de prédiction qui augmente sa complexité de calcul, à l'opposé aux algorithmes des filtres transversaux rapide (FTF).

**Tableau 2.6.** Complexité de calcul.

Algorithme		Complexité de calcul
TC-NLMS		4M
TC-SFTF		14M
Algorithme 1	N=M/2	10M
	N=M/3	26M/3
	N=M/4	8M

**Tableau 2.7.** Nombre des multiplications à chaque itération.

Algorithme		Nombre des multiplications à chaque itération pour différentes tailles des filtres adaptatifs				
		M=64	M=128	M=256	M=512	M=1024
TC-NLMS		256	512	1024	2048	4096
TC-SFTF		896	1792	3584	7168	14336
Algorithme 1	N=M/2	<b>640</b>	<b>1280</b>	<b>2560</b>	<b>10240</b>	<b>10240</b>
	N=M/3	<b>555</b>	<b>1110</b>	<b>2219</b>	<b>8530</b>	<b>8530</b>
	N=M/4	<b>512</b>	<b>1024</b>	<b>2048</b>	<b>8192</b>	<b>8192</b>

## 2.5. Conclusion

Dans ce chapitre, nous avons présenté un nouvel algorithme dérivé de l'algorithme SFTF [32, 19], pour les applications de réduction du bruit acoustique et rehaussement du signal parole. L'algorithme 1 proposé repose sur la compensation de l'erreur d'estimation de gain de Kalman dual généré par élimination de prédicteur arrière, par une réduction de la complexité de calcul en utilisant la technique de filtrage partielle, en mettant à jour des coefficients des filtres transversaux  $\mathbf{a}(n)$  et  $\mathbf{w}(n)$  qui sont moins affectés par l'élimination du prédicteur arrière. Cela

a permet d'avoir un algorithme avec une vitesse de convergence comparable à la version original SFTF, voire meilleure, et avec une complexité des calculs réduite.

La dérivation et l'analyse de l'algorithme proposé, montrent la stratégie de sélection des coefficients pour la mise à jour partielle, afin obtenir la meilleure direction de mis à jour des filtres  $a(n)$  et  $w(n)$ . La mise à jour partielle est basée sur la sélection des coefficients d'indices  $i$  correspondant aux coefficients de gain de Kalman dual de plus grande puissance. Cette manière de sélection a permis d'améliorer légèrement la vitesse de convergence de l'algorithme par rapport à sa version originale, pour le type des bruits autorégressif d'ordre élevé. Ceci est bien prouvé dans l'étude comparative de chaque algorithme présenté dans la section 2.4. L'algorithme proposé est combiné avec le séparateur des sources aveugle pour obtenir une version à deux canaux (TC-PU-SFTF) adaptée aux mélanges convolutifs pour réduction de bruit par la séparation des sources.

Les résultats de simulation ont montrés les meilleures performances de l'algorithme 1 proposé par rapport à celles des algorithmes TC-NLMS et TC-SFTF. Ces performances sont mesurées en termes du rapport signal sur bruit segmenté, l'erreur quadratique moyenne segmentée, le désajustement (système mismatch) et la complexité de calcul. A partir de ces résultats, l'algorithme 1 proposé peut être considéré comme une solution alternative pour la réduction du bruit et le rehaussement du signal de parole bi-capteurs.

# CHAPITRE 3 : NOUVEAU ALGORITHME MULTI-CANAUX POUR LA REDUCTION DE BRUIT

## 3.1.Introduction

Durant les dernières années, de nombreuses nouvelles applications de filtrage adaptatif multi-canaux ont suscité l'intérêt des chercheurs, telles que la réduction du bruit acoustique multi-canaux, l'annulation d'écho acoustique multi-canaux et la formation de faisceau adaptatif (Beamforming). Les techniques de filtrage adaptatif sont généralement basées sur l'optimisation d'une fonction coût, avec ou sans contraintes, en utilisant un algorithme approprié [16, 17].

Dans la réduction du bruit multi-canaux, on supprime les composantes du bruit superposées au signal parole. Le signal parole bruité est capté en utilisant un microphone primaire, ce qui constitue le signal désiré  $d(n)$  pour les filtres adaptatifs multi-canaux. Plusieurs signaux de référence  $x_i(n)$  ( $i = 1, 2, \dots$ ) sont considérés comme des entrées des filtres adaptatifs  $w_i(n)$ . Ces signaux de référence  $x_i(n)$  peuvent être captés en utilisant un réseau de microphones (capteurs) disposés de manière à ce que la composante du signal désiré  $d(n)$  soit très faible ou complètement indétectable [16, 36]. Dans la littérature et les recherches liées au filtrage adaptatif multi-canaux, de nombreux algorithmes adaptatifs ont été proposés, tels que l'algorithme des moindres carrés normalisés (MC-NLMS) [37, 38, 39], cet algorithme est connu par sa faible complexité de calcul. Cependant, cet algorithme souffre d'une lenteur de sa vitesse de convergence en raison de sa forte dépendance aux caractéristiques statistiques des signaux d'entrée de filtre. Cet inconvénient peut poser un problème dans les applications qui nécessitent un court temps d'apprentissage, en particulier lors de l'identification en temps réel des réponses impulsionnelles longues et dispersées [40, 37, 39, 16]. L'algorithme de projection affine multi-canaux (MC-APA), peut résoudre le problème de vitesse de convergence auquel l'algorithme MC-NLMS est subi [16, 17]. Cependant, le principal inconvénient de l'algorithme MC-APA réside dans l'augmentation de la complexité de calcul (CC), dès que la dimension de projection dépasse deux dimensions.

Dans ce chapitre, nous allons développer un nouvel algorithme des moindres carrés normalisés rapide multi-canaux (MC-FNLMS). L'algorithme proposé utilise la décorrélation des signaux pour augmenter la vitesse de convergence, tout en maintenant une faible complexité de calcul. L'algorithme proposé est une version multi-canaux de l'algorithme FNLMS, il introduit un processus de prédiction du premier ordre pour chaque canal. L'avantage de l'algorithme

proposé par rapport aux algorithmes MC-NLMS et MC-APA réside dans sa facilité d'implémentation due à sa faible complexité de calcul, ainsi que de son efficacité pour l'annulation du bruit acoustique. Un autre avantage de l'algorithme proposé est son taux de convergence important, même avec des signaux d'entrée fortement colorés. Une version moins complexe de l'algorithme proposé est développée pour les situations où les signaux de bruit  $x_i(n)$  partagent le même modèle statistique.

Ce chapitre est organisé comme suit : dans la section 3.2, nous présentons l'algorithme FNLMS monocanal, ainsi que les algorithmes classiques MC-APA et MC-NLMS. L'algorithme MC-FNLMS proposé est présenté dans la section 3.3. L'étude comparative de l'algorithme proposé pour la réduction de bruit adaptatif multi-canaux avec les autres algorithmes (c-à-d MC-APA et MC-NLMS) est présentée dans la section 3.4. La conclusion de ce chapitre est donnée dans la section 3.5.

### 3.2. Les algorithmes FNLMS, MC-APA et MC-NLMS

Dans cette section, nous présentons brièvement les trois algorithmes, à savoir les algorithmes FNLMS, MC-APA et l'algorithme MC-NLMS. L'algorithme FNLMS va nous servir à développer l'algorithme proposé dans la section 3.3, les algorithmes MC-APA et MC-NLMS seront comparés avec l'algorithme proposé dans la section 3.4.

#### 3.2.1. L'algorithme NLMS rapide (FNLMS)

L'algorithme FNLMS est reconnu par sa vitesse de convergence avec une faible complexité de calcul. Il est dérivé de l'algorithme FTF (**Tableau 2.1** dans le chapitre précédent) en supprimant les prédicteurs  $\mathbf{a}(n)$  et  $\mathbf{b}(n)$  de l'équation de mise à jour du gain de Kalman dual  $\tilde{\mathbf{c}}(n)$  [21, 22]. Dans l'algorithme FNLMS, le processus de mise à jour du gain de Kalman dual  $\tilde{\mathbf{c}}(n)$ , est basé sur l'erreur  $e(n)$  et la variance  $\alpha(n-1)$  de prédiction du premier ordre du signal  $x(n)$ . L'erreur de filtrage est donnée par la relation ci-dessous :

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

Avec :  $\mathbf{x}^T(n) = [x(n), x(n-1), \dots, x(n-M+1)]$  est le vecteur d'entrée, il contient à l'instant  $n$  les  $M$  dernières valeurs de signal d'entrée  $x(n)$ , et  $\mathbf{w}^T(n) = [w_0(n), w_1(n), \dots, w_{M-1}(n)]$  est le vecteur des  $M$  poids de filtre adaptatif à l'instant  $n$ . L'équation de mise à jour est donnée comme suit :

$$\mathbf{w}(n) = \mathbf{w}(n-1) - \mu_{FLLMS}\gamma(n)\tilde{\mathbf{C}}(n)e(n) \quad (3.2)$$

La quantité  $\gamma(n)$  est le facteur de conversion et  $\tilde{\mathbf{c}}(n)$  est le vecteur gain de Kalman dual d'ordre  $M$ , ce dernier est mis à jour dans l'algorithme FTF en utilisant les vecteurs de prédiction avant et arrière (c-à-d  $\mathbf{a}(n-1)$  et  $\mathbf{b}(n-1)$ ) [32, 30]:

$$\begin{bmatrix} \tilde{\mathbf{c}}(n) \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \tilde{\mathbf{c}}(n-1) \end{bmatrix} - \frac{\varepsilon(n)}{\lambda\alpha(n-1)} \begin{bmatrix} 1 \\ -\mathbf{a}(n-1) \end{bmatrix} + \frac{r(n)}{\lambda\beta(n-1)} \begin{bmatrix} -\mathbf{b}(n-1) \\ 1 \end{bmatrix} \quad (3.3)$$

Où  $\varepsilon(n)$  et  $r(n)$  sont respectivement les erreurs de prédiction avant et arrière,  $\alpha(n)$  et  $\beta(n)$  sont leurs variances respectivement. Dans l'algorithme FNLMS, les prédicteurs  $\mathbf{a}(n-1)$  et  $\mathbf{b}(n-1)$  sont complètement supprimés, le gain de Kalman dual est mis à jour en utilisant l'équation suivante :

$$\begin{bmatrix} \tilde{\mathbf{c}}(n) \\ c(n) \end{bmatrix} = \begin{bmatrix} -\frac{\varepsilon(n)}{\lambda\alpha(n-1) + c_0} \\ \tilde{\mathbf{c}}(n-1) \end{bmatrix} \quad (3.4)$$

Une petite constante positive  $c_0$  est ajoutée afin d'éviter la division par zéro [21, 22, 32]. Les paramètres de prédiction (c-à-d l'erreur de prédiction et sa variance) sont calculés récursivement en utilisant les deux relations ci-dessous :

$$\varepsilon(n) = x(n) + a(n)x(n-1) \quad (3.5)$$

$$\alpha(n) = \lambda\alpha(n-1) + \varepsilon^2(n) \quad (3.6)$$

Afin de calculer la première composante de prédiction  $a(n)$ , un simple algorithme récursif basé sur le calcul des coefficients d'autocorrélation  $r_0(n)$  et  $r_1(n)$  peut être utilisé [21]:

$$r_0(n) = \lambda_a r_0(n-1) + x(n)x(n) \quad (3.7)$$

$$r_1(n) = \lambda_a r_1(n-1) + x(n)x(n-1) \quad (3.8)$$

$$a(n) = \frac{r_1(n)}{r_0(n) + c_a} \quad (3.9)$$

Où  $\lambda_a$  est le facteur d'oubli [16] et  $c_a$  est une petite constante. Le facteur de conversion  $\gamma(n)$  est calculé en utilisant l'équation ci-dessous [21, 22]:

$$\gamma(n) = \frac{1}{1 - \mathbf{x}^T(n)\tilde{\mathbf{c}}(n)} \quad (3.10)$$

En utilisant la propriété de décalage temporel du signal d'entrée étendu à  $M + 1$ , nous obtenons les deux expressions de vecteur  $\mathbf{x}_{M+1}^T$  :

$$\mathbf{x}_{M+1}^T = [\mathbf{x}^T(n), x(n - M)] \quad (3.11)$$

$$\mathbf{x}_{M+1}^T = [x(n), \mathbf{x}^T(n - 1)] \quad (3.12)$$

En multipliant les deux côtés de l'équation (3.4) par le côté droit des équations (3.11) et (3.12) respectivement, nous obtenons l'équation suivante [30, 32, 21]:

$$\mathbf{x}^T(n)\tilde{\mathbf{c}}(n) + \tilde{c}(n)x(n - M) = \mathbf{x}^T(n - 1)\tilde{\mathbf{c}}(n - 1) - \frac{x(n)\varepsilon(n)}{\lambda\alpha(n - 1) + c_0} \quad (3.13)$$

En se basant sur la relation (3.10), On peut écrire :

$$\mathbf{x}^T(n)\tilde{\mathbf{c}}(n) = 1 - \gamma^{-1}(n) \quad (3.14)$$

$$\mathbf{x}^T(n - 1)\tilde{\mathbf{c}}(n - 1) = 1 - \gamma^{-1}(n - 1) \quad (3.15)$$

En substituant ces dernières quantités (c-à-d  $\mathbf{x}^T(n)\tilde{\mathbf{c}}(n)$  et  $\mathbf{x}^T(n - 1)\tilde{\mathbf{c}}(n - 1)$ ) dans l'équation (13), nous obtenons la nouvelle équation de mise à jour de  $\gamma(n)$  :

$$\gamma^{-1}(n) = \gamma^{-1}(n - 1) + \tilde{c}(n)x(n - M) + \frac{x(n)\varepsilon(n)}{\lambda\alpha(n - 1) + c_0} \quad (3.16)$$

Ce qui peut être réécrit comme suit :

$$\gamma(n) = \frac{\gamma(n - 1)}{1 + \gamma(n - 1) + \delta(n)} \quad (3.17)$$

Où :

$$\delta(n) = \tilde{c}(n)x(n - M) + \frac{x(n)e(n)}{1 + \lambda\alpha(n - 1) + c_0} \quad (3.18)$$

La complexité de calcul totale de cet algorithme égal à  $2M$  multiplications par itération. On résume l'algorithme NLMS rapide (FNLMS) dans le **Tableau 3.1**:

**Tableau 3.1.** Algorithme NLMS rapide (FNLMS) [21].

---

**Initialisation**

$$\tilde{\mathbf{c}}(0) = \mathbf{0}_{M \times 1}, \gamma(0) = 1, r_1(0) = 0, r_0(0) = 1, \alpha(0) = 1, 0.9 < \lambda < 1,$$

$$0.9 < \lambda_a < 1, c_a = c_0 = 0.0001$$

---

**Prédiction**

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

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

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

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

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

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

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

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

**Filtrage**

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

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

---

**3.2.2. Algorithmes (MC-APA) et (MC-NLMS)**

Dans le cas du filtrage adaptatif monocanal, on dérive l'algorithme de projection affine (APA) en utilisant une optimisation avec multiples contraintes imposés sur l'erreur de filtrage à la sortie du filtre adaptatif, tout en minimisant la norme euclidienne du changement  $\delta \mathbf{w}(n)$  des poids de filtre adaptatif entre l'instant  $n$  et  $n-1$ , voici la méthode [16, 17]:

Le changement des poids de filtre adaptatif entre les instants  $n$  et  $n - 1$  est donné par l'équation ci-dessous :

$$\delta \mathbf{w}(n) = \mathbf{w}(n) - \mathbf{w}(n - 1) \quad (3.19)$$

Les  $P$  contraintes imposées sur l'erreur à l'instant  $n - 1, n - 2, \dots, n - P$ , s'écrivent :

$$e(n - k) = 0 \Leftrightarrow d(n - k) = \mathbf{w}^T(n) \mathbf{x}(n - k) \quad k = 1, 2, \dots, P \quad (3.20)$$

Où  $P < M$ .

Afin de trouver la solution du  $\mathbf{w}(n)$ , nous pouvons résoudre le problème d'optimisation précédent en utilisant la méthode des multiplicateurs de Lagrange avec multiples contraintes, ce qui donne l'équation de mis à jour suivante [16] :

$$\mathbf{w}(n) = \mathbf{w}(n - 1) + \mathbf{A}^T(n) (\mathbf{A}(n) \mathbf{A}^T(n))^{-1} \mathbf{e}(n) \quad (3.21)$$

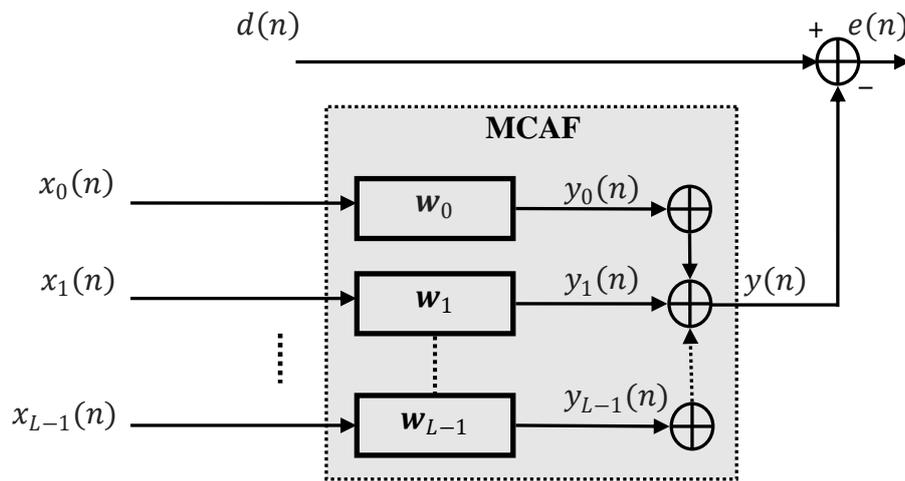
Où :

$$\mathbf{e}(n) = \mathbf{d}(n) - \mathbf{A}(n) \mathbf{w}(n - 1) \quad (3.22)$$

$$\mathbf{A}^T(n) = [\mathbf{x}(n), \mathbf{x}(n - 1), \dots, \mathbf{x}(n - P)] \quad (3.23)$$

$$\mathbf{d}^T(n) = [d(n), d(n - 1), \dots, d(n - P)] \quad (3.24)$$

Dans le cas de filtrage adaptatif multi-canaux (**Figure 3.1**), nous avons  $L$  signal d'entrée  $x_i(n)$ , ce qui nécessite à mettre à jour  $L$  filtres adaptatifs  $\mathbf{w}_i(n)$ .



**Figure 3.1.** Filtrage adaptatif multi-canaux.

L'algorithme de projection affine multi-canaux (MC-APA) est décrit dans le **Tableau 3.2**.

**Tableau 3.2.** Algorithme de projection affine multi-canaux (MC-APA).

	Mult/Div
$\mathbf{e}(n) = \mathbf{d}(n) - \sum_{i=1}^L \mathbf{A}_i(n) \mathbf{w}_i(n-1)$	<i>PML</i>
$\mathbf{w}_i(n) = \mathbf{w}_i(n-1) + \mu_{MCAPA} \mathbf{A}_i^T(n) \left( \sum_{i=1}^L \mathbf{A}_i(n) \mathbf{A}_i^T(n) \right)^{-1} \mathbf{e}(n)$	$O(L^3) + LMP^2 + LMP + P^2 + P$
Total Mult/Div	$O(L^3) + LMP^2 + 2LMP + P^2 + P$

Où  $O(L^3)$  représente la proportionnalité à  $L^3$  de la complexité de calcul pour l'inversion de la matrice de dimension  $L \times L$  en utilisant la méthode d'élimination de Gauss-Jordan. La stabilité de l'algorithme MC-APA est vérifiée lorsque  $0 < \mu_{MCAPA} < 2$ .

Dans le cas particulier où  $P = 1$ , l'algorithme APA multi-canaux se réduit à l'algorithme classique des moindres carrés normalisés multi-canaux (MC-NLMS), où  $0 < \mu_{MCNLMS} < 2$ .

**Tableau 3.3.** Algorithme MC-NLMS.

	Mul/Div
$\mathbf{e}(n) = \mathbf{d}(n) - \sum_{i=1}^N \mathbf{x}_i^T(n) \mathbf{w}_i(n-1)$	<i>LM</i>
$\mathbf{w}_i(n) = \mathbf{w}_i(n-1) + \mu_{MCNLMS} \frac{\mathbf{x}_i(n)}{\sum_{i=1}^N \mathbf{x}_i^T(n) \mathbf{x}_i(n)} \mathbf{e}(n)$	$2LM + 2$
Total Mult/Div	$3LM + 2$

### 3.3. Algorithme FNLMS multi-canaux proposé (MC-FNLMS)

Dans cette section, nous présentons la dérivation de l'algorithme multi-canaux proposé, dérivé de l'algorithme FNLMS. L'algorithme proposé est appelé algorithme des moindres carrés normalisé rapide multi-canaux (MC-FNLMS). Pour cela, pour chaque signal d'entrée  $\mathbf{x}_i(n)$ , nous générons à la fois les paramètres de prédiction, à savoir l'erreur de prédiction et sa variance, ainsi que le facteur de conversion associé. De plus, nous développons une version à faible complexité (complexité réduite), en estimant les paramètres de prédiction en basant sur un seul canal. Pour démontrer les conditions pour lesquelles l'algorithme proposé converge vers les solutions optimales, nous présentons également une étude de stabilité.

### 3.3.1. Dérivation de l'algorithme proposé

Dans notre développement, nous considérons la mise à jour de  $L$  filtres adaptatifs en utilisant un seul signal d'erreur  $e(n)$ . L'erreur du filtrage est définie selon les mêmes relations que celle de l'algorithme MC-NLMS décrit dans le **Tableau 3.3**. Cependant, l'équation de mise à jour des filtres est donnée par :

$$\mathbf{w}_i(n) = \mathbf{w}_i(n-1) - \mu_{MCFNLMS} e(n) \gamma_i(n) \tilde{\mathbf{c}}_i(n) \quad i = 1, 2, \dots, L \quad (3.25)$$

Où  $\mu_{MCFNLMS}$  est le pas d'adaptation qui contrôle le de convergence de l'algorithme. Les gains de Kalman dual  $\tilde{\mathbf{c}}_i(n)$  sont mets à jour donc par l'équation ci-dessous :

$$\begin{bmatrix} \tilde{\mathbf{c}}_i(n) \\ c_i(n) \end{bmatrix} = \begin{bmatrix} -\frac{\varepsilon_i(n)}{\lambda \alpha_i(n-1) + c_0} \\ \tilde{\mathbf{c}}_i(n-1) \end{bmatrix} \quad i = 1, 2, \dots, L \quad (3.26)$$

Les erreurs de prédiction des signaux d'entrée  $\varepsilon_i(n)$ , et leurs variances correspondantes  $\alpha_i(n)$ , se calculent par les relations suivantes :

$$\varepsilon_i(n) = x_i(n) - a_i(n)x_i(n-1) \quad i = 1, 2, \dots, L \quad (3.27)$$

$$\alpha_i(n) = \lambda \alpha_i(n-1) + \varepsilon_i^2(n) \quad i = 1, 2, \dots, L \quad (3.28)$$

Où  $a_i(n)$  représente les coefficients de prédiction du premier ordre des signaux  $x_i(n)$  (c'est-à-dire correspondant au canal  $i$ ). Les équations (3.29), (3.30) et (3.31) ci-dessous permettent d'estimer les quantités  $a_i(n)$  :

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

$$r_{1,i}(n) = \lambda_a r_{1,i}(n-1) + x_i(n)x_i(n-1) \quad i = 1, 2, \dots, L \quad (3.30)$$

$$a_i(n) = \frac{r_{1,i}(n)}{r_{0,i}(n) + c_a} \quad i = 1, 2, \dots, L \quad (3.31)$$

Les facteurs de conversions,  $\gamma_i(n)$ , sont calculés pour chaque signal  $x_i(n)$  à l'aide des relations suivantes :

$$\gamma_i(n) = \frac{\gamma_i(n-1)}{1 + \gamma_i(n-1) + \delta_i(n)} \quad i = 1, 2, \dots, L \quad (3.32)$$

$$\delta_i(n) = c_i(n)x_i(n-M) + \frac{x_i(n)\varepsilon_i(n)}{1 + \lambda a_i(n-1) + c_0} \quad i = 1, 2, \dots, L \quad (3.33)$$

Dans la plupart des applications du filtrage adaptatif multi-canaux, les signaux d'entrée  $x_i(n)$  présentent des propriétés statistiques très proche. Ainsi, il est possible d'estimer tous les paramètres de prédiction en n'utilisant qu'un seul canal. Par conséquent, la complexité de calcul peut être réduit. Le **Tableau 3.4** résume l'algorithme MC-FNLMS proposé (Algorithme 2).

**Tableau 3.4.** Algorithme MC-FNLMS proposé (Algorithme 2).

<b>Initialisation</b>	Paramètres de	Paramètres de
$\tilde{\mathbf{c}}_i(0) = \mathbf{0}, \gamma_i(0) = 1, r_{1,i}(0) = 0, r_{0,i}(0) = 1$	prédictions	prédictions estimé à
$\alpha_i(0) = 1$	estimé à partir	partir d'un seul canal
$0.9 < \lambda < 1, 0.9 < \lambda_a < 1, c_a = c_0 = 0.0001$	de $L$ canal	(version réduite)
	Mult/Div	Mult/Div
<b>Prédiction</b>		
$r_{0,i}(n) = \lambda_a r_{0,i}(n-1) + x_i(n)x_i(n)$	$2L$	$2$
$r_{1,i}(n) = \lambda_a r_{1,i}(n-1) + x_i(n)x_i(n-1)$	$2L$	$2$
$a_i(n) = \frac{r_{1,i}(n)}{r_{0,i}(n) + c_a}$	$L$	$1$
$\varepsilon_i(n) = x_i(n) - a_i(n)x_i(n-1)$	$L$	$1$
$\alpha_i(n) = \lambda \alpha_i(n-1) + \varepsilon_i^2(n)$	$2L$	$2$
$\begin{bmatrix} \tilde{\mathbf{c}}_i(n) \\ c_i(n) \end{bmatrix} = \begin{bmatrix} -\frac{\varepsilon_i(n)}{\lambda \alpha_i(n-1) + c_0} \\ \tilde{\mathbf{c}}_i(n-1) \end{bmatrix}$	$3L$	$3$
$\delta_i(n) = c_i(n)x_i(n-M)$	$4L$	$4$
$+ \frac{x_i(n)\varepsilon_i(n)}{1 + \lambda a_i(n-1) + c_0}$		
$\gamma_i(n) = \frac{\gamma_i(n-1)}{1 + \gamma_i(n-1) + \delta_i(n)}$	$L$	$1$
$\mathbf{e}(n) = \mathbf{d}(n) - \sum_{i=1}^N \mathbf{x}_i^T(n) \mathbf{w}_i(n-1)$	$LM$	$LM$
<b>Filtrage</b>		
$\mathbf{w}_i(n) = \mathbf{w}_i(n-1)$		
$- \mu_{MCFNLMS} e(n) \gamma_i(n) \tilde{\mathbf{c}}_i(n)$	$LM + 2L$	$LM + 2L$
Total Mult/Div	$2LM + 18L$	$2LM + 2L + 16$

### 3.3.2. Analyse de la convergence

L'analyse théorique de la convergence de l'algorithme proposé (MC-FNLMS) peut être compliquée pour le cas général des signaux. Cependant, on peut simplifier l'analyse en considérant un cas particulier des signaux d'entrée de type bruits blancs gaussiens, dans lesquels

les coefficients prédicteurs  $a_i$  peuvent être remplacés par des zéros ( $(a_i(n) = 0)$ ). Par conséquent, les erreurs de prédiction, sont égales aux signaux d'entrée, c'est-à-dire  $\varepsilon_i(n) = x_i(n)$ . De plus, on suppose que toutes les variables de l'algorithme sont proches de leurs valeurs optimales, en particulier  $\alpha_i(n+1) \approx \alpha_i(n)$ . En utilisant ces hypothèses, on peut considérer l'Équation (4.28) comme une équation décrivant une suite inconnue  $\alpha_i(n)$  plutôt qu'une équation de récurrence de mise à jour. Cette approche a une très grande importance, car elle nous permet d'exprimer les quantités  $\alpha_i(n)$ , en fonction des signaux d'entrée  $x_i^2(n)$ , qui va faciliter ainsi leur manipulation mathématique de l'analyse de la stabilité comme nous le verrons par la suite. En vertu de la relation (4.28) et les hypothèses mentionnées, on peut écrire l'approximation suivante :

$$\alpha_i(n) \approx \lambda \alpha_i(n) + x_i^2(n) \quad i = 1, 2, \dots, L \quad (3.34)$$

$$\alpha_i(n) \approx \frac{x_i^2(n)}{(1 - \lambda)} \quad i = 1, 2, \dots, L \quad (3.35)$$

Les signaux  $x_i(n)$  sont considérés des bruits blancs gaussiens de variance  $\sigma_i^2 = E[x_i^2(n)]$ . En utilisant l'approximation stochastique des variances  $\sigma_i^2 = E[x_i^2(n)] \approx x_i^2(n)$ , la relation (3.35) peut être réécrite comme suit :

$$\alpha_i(n) \approx \frac{\sigma_i^2}{(1 - \lambda)} \quad i = 1, 2, \dots, L \quad (3.36)$$

Les équations de mise à jour des gains de Kalman dual pour différents instants sont données par ci-dessous :

$$\begin{bmatrix} \tilde{c}_i(n) \\ c_i(n) \end{bmatrix} = \begin{bmatrix} -\frac{\varepsilon_i(n)}{\lambda \alpha_i(n-1) + c_0} \\ \tilde{c}_i(n-1) \end{bmatrix} \approx \begin{bmatrix} -\frac{x_i(n)}{\lambda \left( \frac{\sigma_i^2}{1-\lambda} \right) + c_0} \\ \tilde{c}_i(n-1) \end{bmatrix} \quad i = 1, 2, \dots, L \quad (3.37)$$

.....

$$\begin{bmatrix} \tilde{c}_i(n-M) \\ c_i(n-M) \end{bmatrix} \approx \begin{bmatrix} -\frac{x_i(n-M)}{\lambda \left( \frac{\sigma_i^2}{1-\lambda} \right) + c_0} \\ \tilde{c}_i(n-M-1) \end{bmatrix} \quad i = 1, 2, \dots, L \quad (3.38)$$

À partir de ces dernières équations, on peut déduire l'approximation de  $\tilde{c}_i(n)$  :

$$\tilde{\mathbf{c}}_i(n) \approx -\frac{[x(n), x(n-1), \dots, x(n-M+1)]^T}{\frac{\lambda\sigma_i^2}{1-\lambda} + c_0} = -\frac{\mathbf{x}_i(n)}{\frac{\lambda\sigma_i^2}{1-\lambda} + c_0} \quad i = 1, 2, \dots, L \quad (3.39)$$

En remplace ce dernier dans l'équation de mise à jour de facteur  $\gamma_i(n)$ :

$$\gamma_i(n) = \frac{1}{1 - \mathbf{x}_i^T(n)\tilde{\mathbf{c}}_i(n)} \quad i = 1, 2, \dots, L \quad (3.40)$$

$$\gamma_i(n) \approx \frac{1}{1 + \mathbf{x}_i^T(n)\frac{\mathbf{x}_i(n)}{\frac{\lambda\sigma_i^2}{1-\lambda} + c_0}} \quad i = 1, 2, \dots, L \quad (3.41)$$

Sachant que :

$$\mathbf{x}_i^T(n)\mathbf{x}_i(n) \approx M\sigma_i^2 \quad i = 1, 2, \dots, L \quad (3.42)$$

Les facteurs  $\gamma_i(n)$  s'approximent donc comme ci-dessous :

$$\gamma_i(n) \approx \frac{1}{1 + \frac{M\sigma_i^2}{\frac{\lambda\sigma_i^2}{1-\lambda} + c_0}} \quad i = 1, 2, \dots, L \quad (3.43)$$

On écrit l'équation de mise à jour avec ces deux dernières approximations de  $\tilde{\mathbf{c}}_i(n)$  et  $\gamma_i(n)$  (c-à-d (3.39) et (3.43)) :

$$\begin{aligned} \mathbf{w}_i(n) &= \mathbf{w}_i(n-1) - \mu_{MCFNLMS}e(n) \frac{1}{1 + \frac{M\sigma_i^2}{\frac{\lambda\sigma_i^2}{1-\lambda} + c_0}} \frac{-\mathbf{x}_i(n)}{\frac{\lambda\sigma_i^2}{1-\lambda} + c_0} \\ &= \mathbf{w}_i(n-1) + \mu_{MCFNLMS}e(n) \frac{\mathbf{x}_i(n)}{\frac{\lambda\sigma_i^2}{1-\lambda} + c_0 + M\sigma_i^2} \quad i = 1, 2, \dots, L \quad (3.44) \\ &= \mathbf{w}_i(n-1) + \frac{\mu_{MCFNLMS}}{\left(\frac{\lambda}{(1-\lambda)M} + \frac{c_0}{M\sigma_i^2} + 1\right)} \frac{e(n)\mathbf{x}_i(n)}{M\sigma_i^2} \end{aligned}$$

On obtient en final, l'équation de mis à jour des filtres  $w_i(n)$  écrites ci-dessous :

$$\mathbf{w}_i(n) = \mathbf{w}_i(n-1) + \frac{N\mu_{MCFNLMS}}{\left(\frac{\lambda}{(1-\lambda)M} + \frac{c_0}{M\sigma_i^2} + 1\right)NM\sigma_i^2} \mathbf{x}_i(n) e(n) \quad i = 1, 2, \dots, L \quad (3.45)$$

En tenant compte les mêmes propriétés des signaux d'entrée (signaux gaussiens blancs), l'équation de mise à jour des filtres  $\mathbf{w}_i(n)$  pour l'algorithme MC-NLMS (**Tableau 3.3**) peut être réécrite comme suit :

$$\mathbf{w}_i(n) = \mathbf{w}_i(n-1) + \mu_{MCNLMS} \frac{\mathbf{x}_i(n)}{\sum_{i=1}^N M\sigma_i^2} e(n) \quad i = 1, 2, \dots, L \quad (3.46)$$

Ou encore :

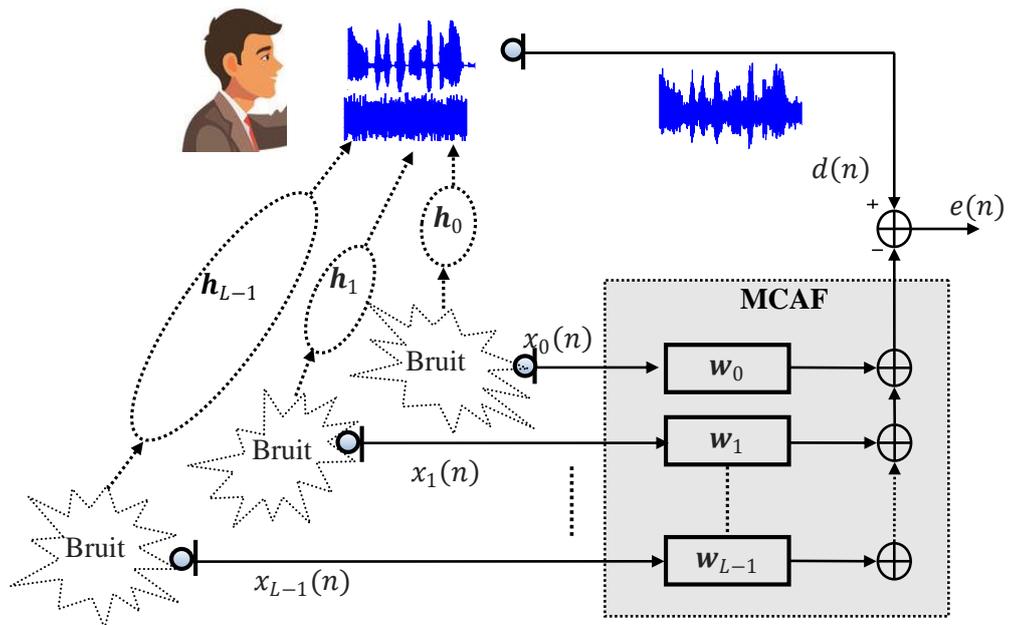
$$\mathbf{w}_i(n) = \mathbf{w}_i(n-1) + \mu_{MCNLMS} \frac{\mathbf{x}_i(n)}{NM\sigma_i^2} e(n) \quad i = 1, 2, \dots, L \quad (3.47)$$

En comparant l'équation (3.45) (algorithme MC-FNLMS) avec l'équation (3.47) (algorithme MC-NLMS), et sachant que l'algorithme MC-NLMS converge pour  $0 < \mu_{MCNLMS} < 2$ . On peut ainsi écrire la condition de convergence de l'algorithme MC-FNLMS comme suit :

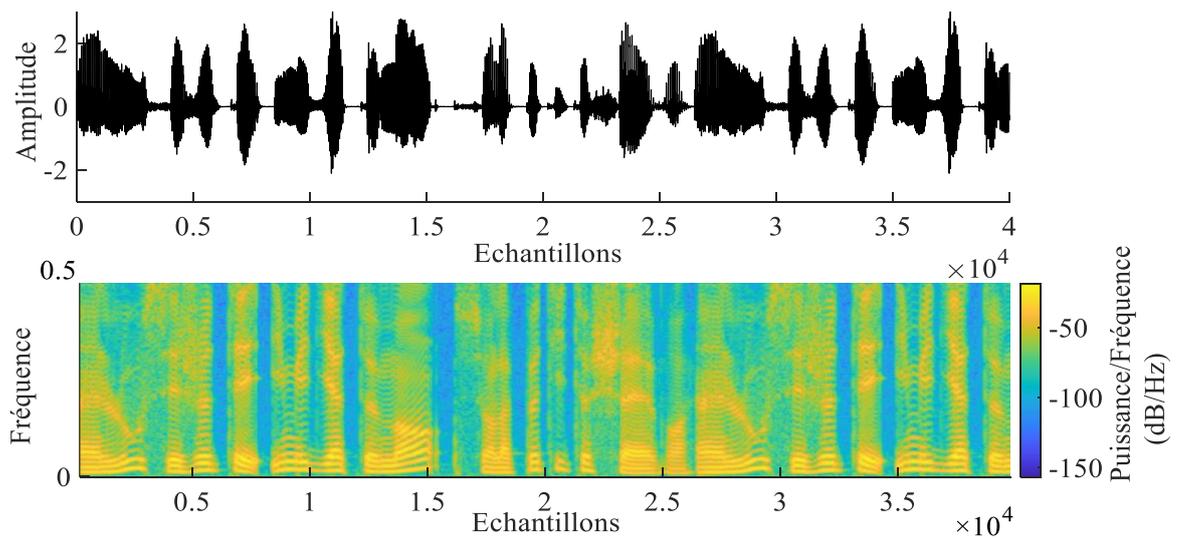
$$0 < \frac{N\mu_{MCFNLMS}}{\left(\frac{\lambda}{(1-\lambda)M} + \frac{c_0}{M\sigma_i^2} + 1\right)} < 2 \Leftrightarrow 0 < \mu_{MCFNLMS} < \frac{2\left(\frac{\lambda}{(1-\lambda)M} + \frac{c_0}{M\sigma_i^2} + 1\right)}{N} \quad (3.48)$$

### 3.4. Simulation et résultats

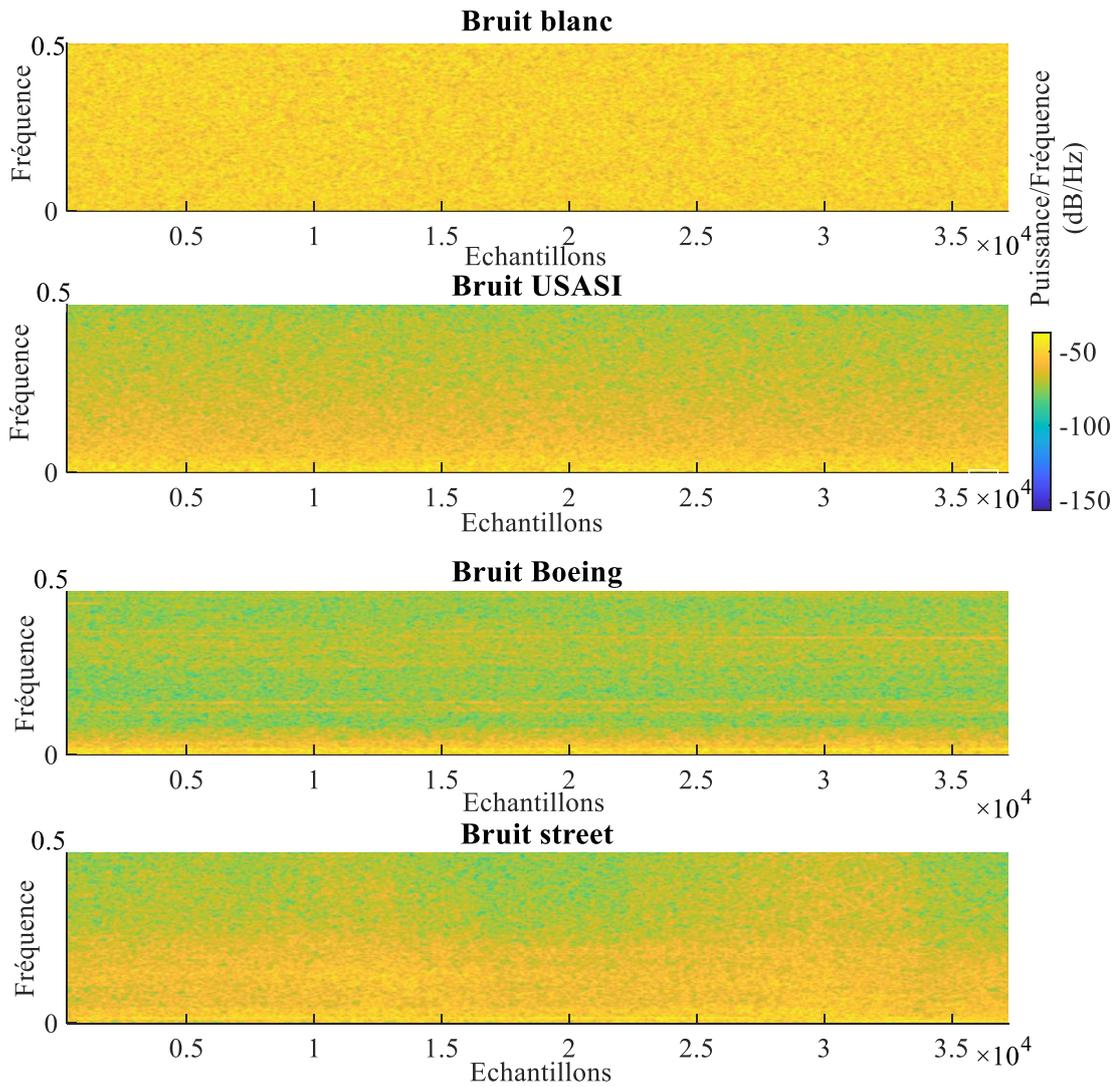
Dans cette simulation, nous allons évaluer les performances de l'algorithme MC-FNLMS proposé (Algorithme 2) pour la réduction de bruit acoustique multi-canaux (**Figure 3.2**). Le signal de parole original et son spectrogramme est présentés dans la **Figure 3.3**. C'est le même signal utilisé dans la simulation du chapitre 2. L'algorithme proposé sera comparé aux algorithmes MC-NLMS et MC-APA. Nous considérons quatre sources de bruit stationnaire et ponctuel (bruits USASI, de rue, rose et intérieur d'avion Boeing), le spectrogramme de chacun de ces bruits est donné à la **Figure 3.3**. Tous les signaux dans cette simulation, sont échantillonnés avec une fréquence d'échantillonnage égal à 8 kHz.



**Figure 3.2.** Réduction de bruit adaptatif multi-canaux.



**Figure 3.3.** Signal de parole nette et son spectrogramme.



**Figure 3.4.** Spectrogrammes des bruits.

Afin de générer les quatre bruits de référence, on dispose de quatre microphones placés à proximité de chacun de ces sources de bruits, de manière à ce que  $x_i(n) = b_i(n)$ . Le signal désiré  $d(n)$  est un mélange du signal de parole original  $s(n)$  et d'une superposition des bruits convolutifs par les réponses acoustiques  $h_i(n)$ . Le rapport signal à bruit d'entrée (SNR) est égal à 0 dB, l'évaluation temporelle et le spectrogramme du signal bruité sont présentés à la **Figure 3.7**. L'expression du signal bruité est donnée par la relation ci-dessous :

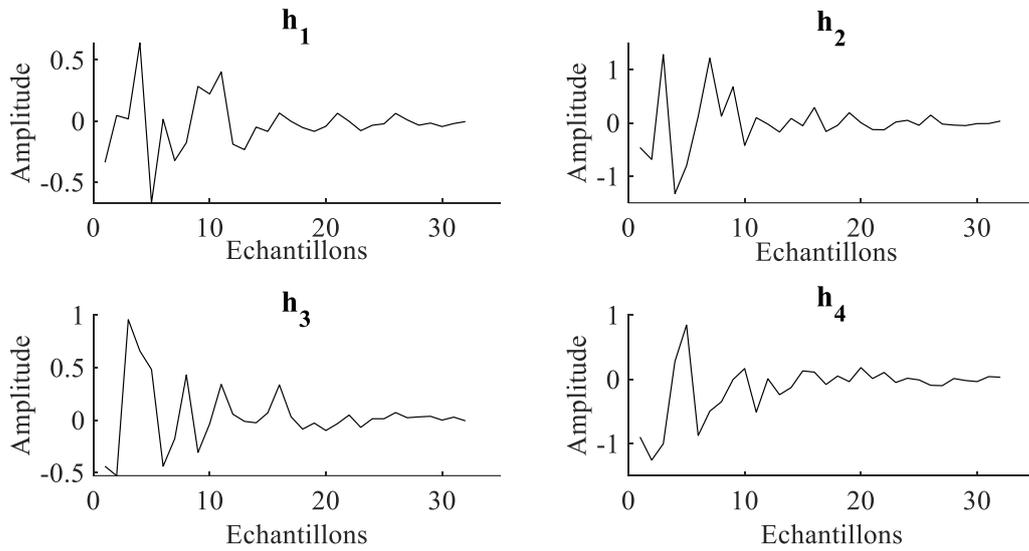
$$d(n) = s(n) + b(n) \quad (3.49)$$

avec :

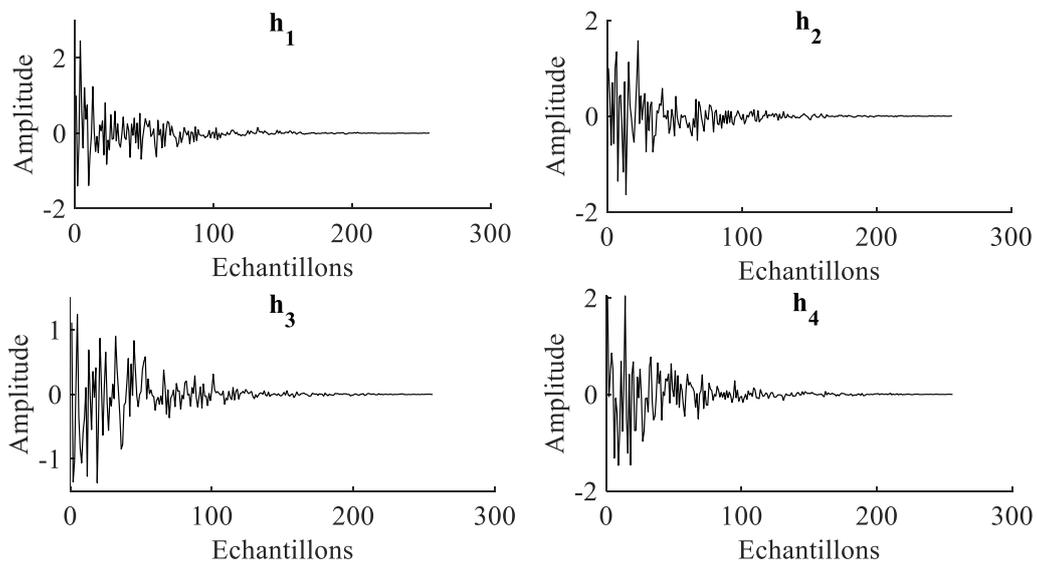
$$b(n) = \sum_{k=1}^4 x_k(n) * h_k(n) \quad (3.50)$$

Où "\*" représente l'opération de convolution discrète.

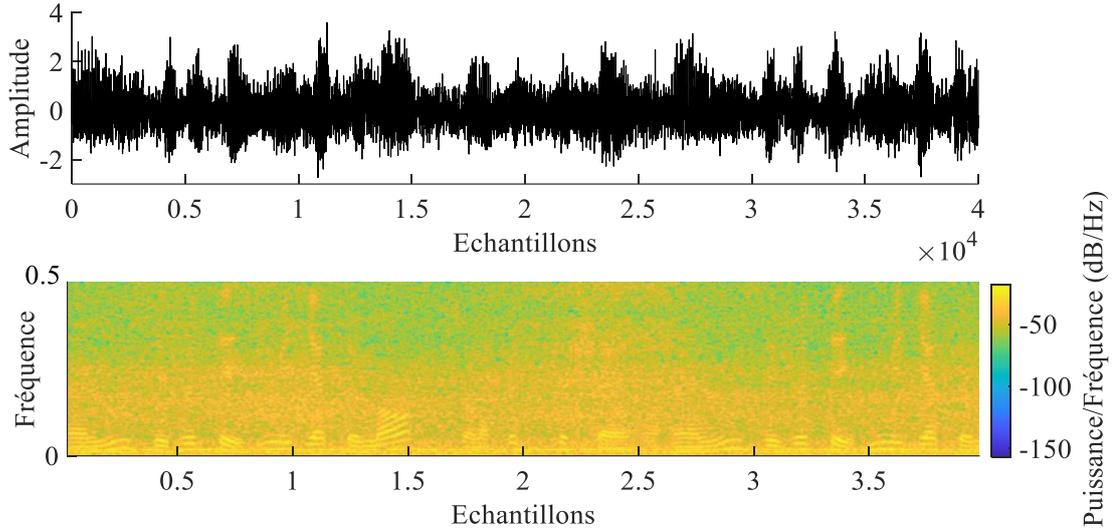
Dans cette simulation, on va considérer deux situations, dans la première situation, on considère quatre réponses impulsionnelles courtes de longueur égale à 32 échantillons pour chacune. Dans la deuxième situation, on considère quatre réponses impulsionnelles longues et dispersées de longueur égale à 256 échantillons pour chacune. Les réponses acoustiques sont représentées respectivement dans la **Figure 3.5** et la **Figure 3.6**.



**Figure 3.5.** Réponses impulsionnelles acoustiques courtes.



**Figure 3.6.** Réponses impulsionnelles acoustiques longues et dispersées.



**Figure 3.7.** Signal de parole bruité ( $RSB = 0dB$ ).

Les critères considérés dans cette simulation sont les mêmes critères utilisés dans la simulation du chapitre 2, en les rappelant ci-dessous :

- **L'évolution temporelle** de signal rehaussé à la sortie  $e(n)$ .
- **Désajustement (système mismatch  $SM$ )** utilisé pour évaluer la vitesse de convergence pour chaque filtre adaptatif  $w_i$  vers la réponse acoustique  $h_i$ , le désajustement se calcule en utilisant l'expression suivante :

$$SM(n)_{dB} = 10 \log \left[ \frac{\|h_i(n) - w_i(n)\|^2}{\|h_i(n)\|^2} \right] \quad (3.51)$$

Avec  $\| \cdot \|$  dénote la norme Euclidienne.

- **L'erreur quadratique moyenne segmentée ( $EQMSeg$ )**: elle permet de quantifier la puissance du bruit à la sortie pour chaque algorithme, le  $EQMSeg$  est calculé à la sortie  $e(n)$  par l'expression suivante :

$$EQMSeg_{dB} = 10 \log \left[ \sum_{i=0}^{U-1} |e(i)|^2 \right] \quad (3.52)$$

Où  $U$  est la longueur de segment. Comme la partie de simulation en chapitre 2, ce critère est estimé uniquement pendant les périodes de silence.

- **Le rapport signal sur bruit segmenté ( $RSB Seg$ )**: il permet de quantifier les performances de réduction du bruit. Le  $RSB Seg$  est quantifié comme suit :

$$RSB Seg_{dB} = 10 \log \left[ \frac{\sum_{i=0}^{U-1} |s(i)|^2}{\sum_{i=0}^{U-1} |s(i) - e(i)|^2} \right] \quad (3.53)$$

Où  $U$  représente le nombre d'échantillons inclus dans le calcul de  $RSB Seg$ . Le  $RSB Seg$  est évalué uniquement pendant les périodes d'activité vocale.

- **La complexité de calcul (CC) :** La CC est évaluée pour chaque algorithme. Dans cette simulation, nous nous intéressons uniquement au nombre de multiplications et de divisions. La complexité de calcul due aux opérations d'addition est largement négligeable en comparaison avec celle des multiplications et des divisions.

Les paramètres utilisés dans cette simulation sont présentés dans le **Tableau 3.5**.

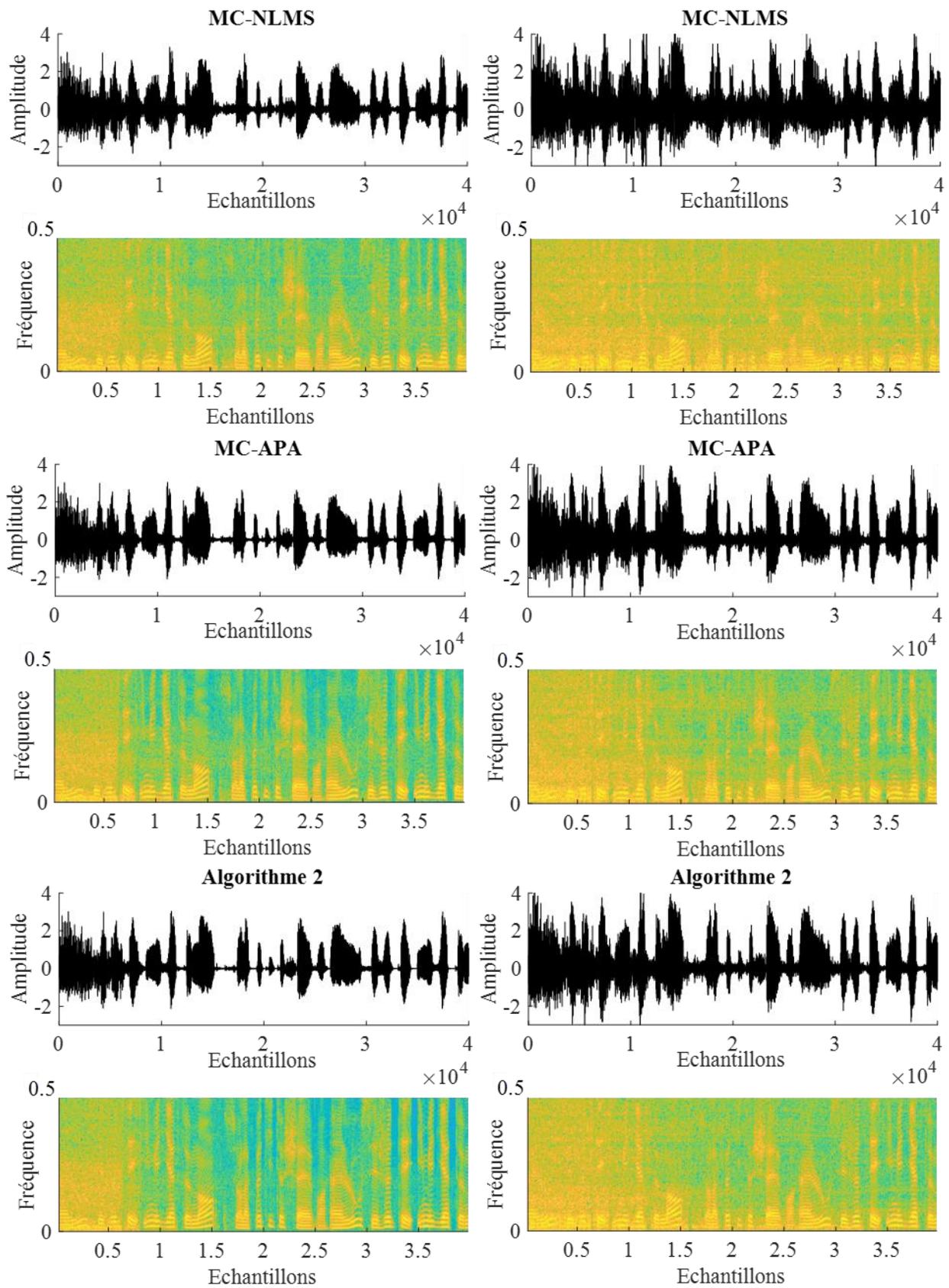
**Tableau 3.5.** Paramètres de simulation.

	Signal parole original (sans bruit) : $s(n)$ .
Signaux	Signal de bruit $b(n)$ : Bruit blanc, USASI, Boeing et Street. Fréquence d'échantillonnage : $f_s = 8 \text{ kHz}$ .
Paramètres des filtres adaptative	Quatre filtres adaptatifs de durée finie, $w_1(n)$ , $w_2(n)$ , $w_3(n)$ et $w_4(n)$ avec : $M_1 = M_2 = M_3 = M_4 = 32$ (Pour les RIs courtes). $M_1 = M_2 = M_3 = M_4 = 256$ (Pour les RIs longues).
Paramètres d'algorithme MC-NLMS	Le pas d'adaptation : $\mu = 1$ . (Meilleure vitesse de convergence)
Paramètres d'algorithme MC-APA	Dimension $P = 2$ . (Complexité la plus faible) Le pas d'adaptation : $\mu_{MC-APA} = 1$ . (Meilleure vitesse de convergence).
Paramètres d'algorithme 2 (proposé)	Facteurs d'oubli : $\lambda = 0.94$ , $\lambda_a = 0.94$ . Petites constantes : $c_a = 0.01$ , $c_0 = 0.00001$ . Le pas d'adaptation : $\mu_{MC-FNLMS} = 1/N = 1/4$ . (Meilleure vitesse de convergence)

- **L'évolution temporelle**

Les paramètres de simulation de cette expérience sont décrits dans le **Tableau 3.5** ci-dessus. Dans la **Figure 3.8**, nous présentons l'évolution temporelle du signal de parole rehaussé à la sortie  $e(n)$ , obtenue par chaque algorithme. De haut vers le bas de la **Figure 3.8**, on présente pour chaque algorithme, l'évolution temporelle et son spectrogramme obtenu à la sortie.

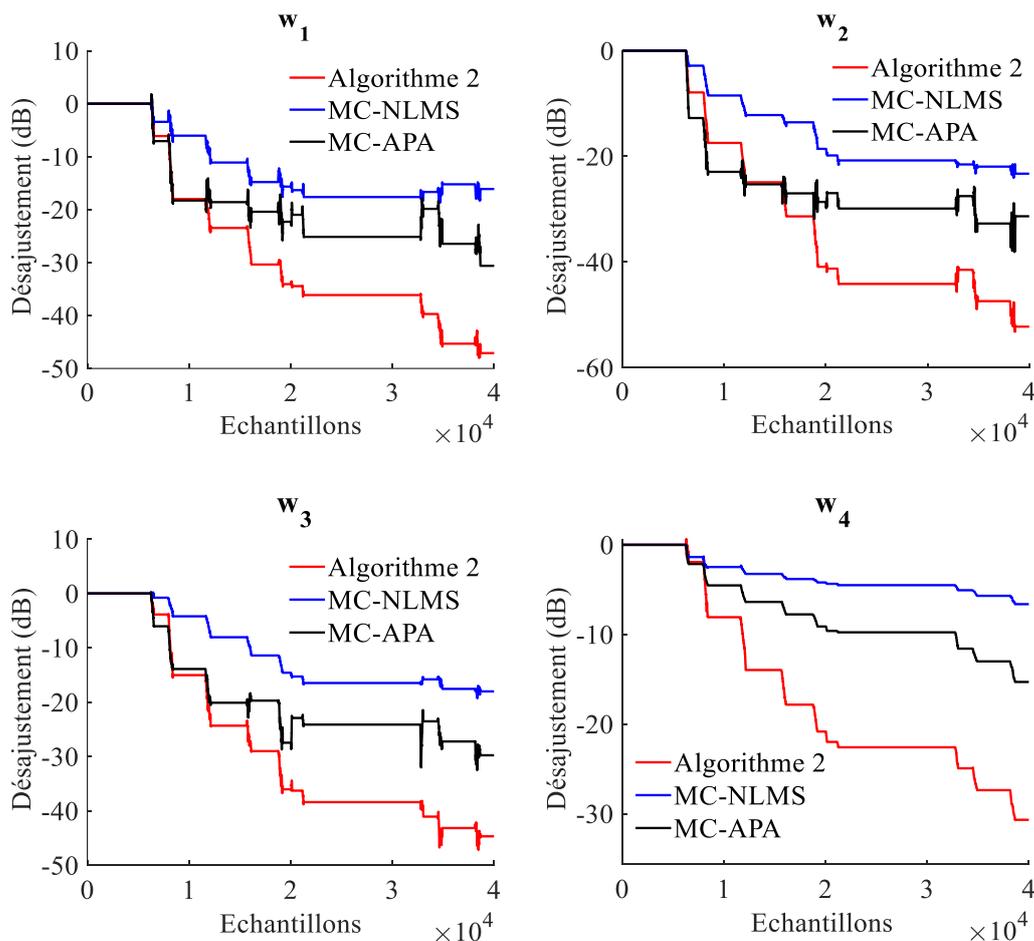
Selon les résultats de la **Figure 3.8**, on voit bien la supériorité de la réduction du bruit, de l'algorithme proposé ainsi que de l'algorithme MC-APA par rapport à l'algorithme MC-NLMS. Toutefois, l'algorithme proposé et l'algorithme MC-APA donnent des résultats qui semblent très similaires si on se base sur la forme du signal dans le domaine temporel. Cependant, les spectrogrammes des signaux obtenus montrent que l'algorithme proposé possède l'avantage de réduire une large bande fréquentielle du bruit, par rapport aux autres algorithmes. Les autres critères suivent, vont permettre d'évaluer objectivement les performances de chacun des algorithmes (c-à-d Algorithme 2, MC-APA, MC-NLMS).



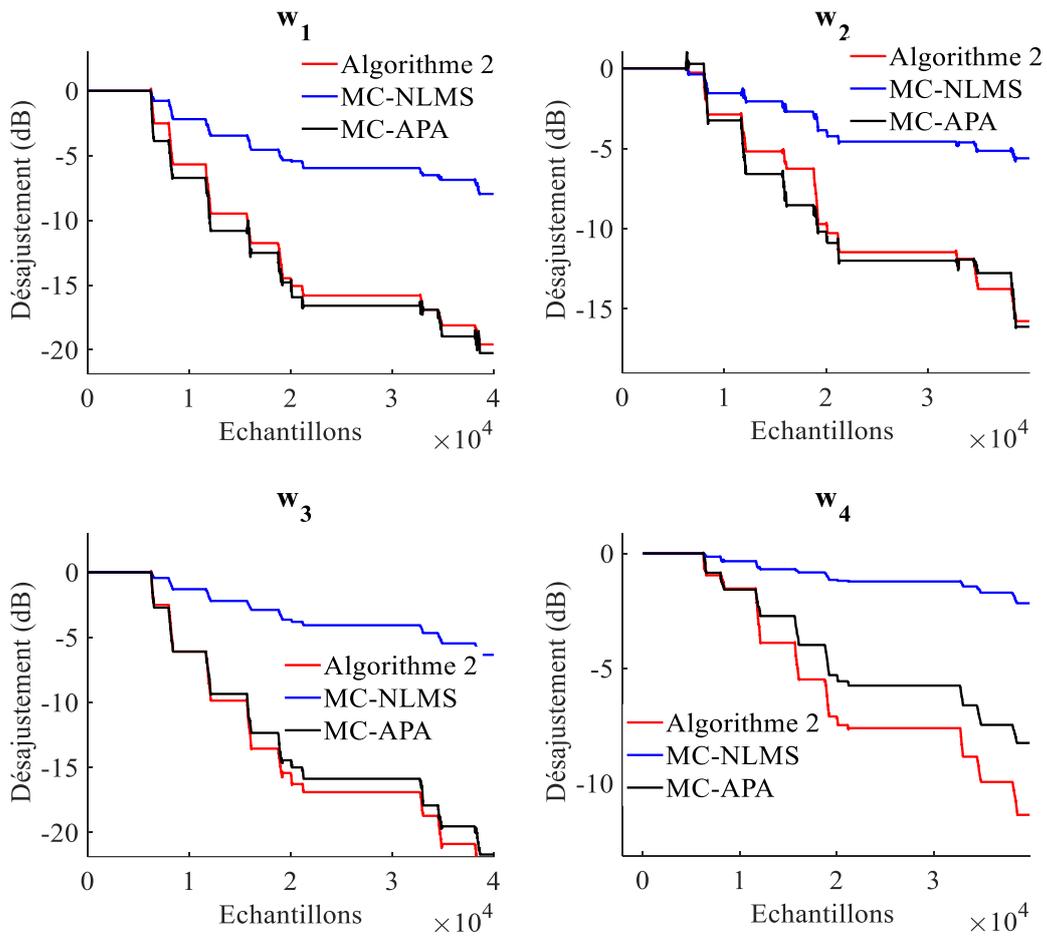
**Figure 3.8.** Signaux rehaussés obtenus en sortie des différents algorithmes (à gauche : Simulation des RIs courtes ; à droite : Simulation des RIs longues et dispersées).

- **Désajustement (système mismatch  $SM$ )**

Dans cette section, nous évaluons le critère de désajustement. Pour ce faire, nous avons testé l'algorithme MC-FNLMS proposé avec les bruits décrits précédemment (c-à-d USASI, Street, rose et bruit intérieur d'avion Boeing). Le rapport signal à bruit d'entrée (RSB) est fixé à 0 dB. Nous avons effectué deux parties de simulations : la partie (i) où on utilise des réponses impulsionnelles courtes (**Figure 3.9**), et la partie (ii) où les réponses impulsionnelles sont choisies longues et dispersées (**Figure 3.10**). En se basant sur ces résultats, on peut constater que le taux de convergence de l'algorithme classique MC-NLMS est le plus faible, tandis que les autres algorithmes possèdent un taux de convergence important dans les deux expériences (c-à-d RIs courtes et longues). On note également que l'algorithme MC-FNLMS proposé possède le taux de convergence le plus important par rapport à l'algorithme MC-APA, notamment dans la première expérience (c-à-d RIs courtes). Ces résultats mettent en évidence la supériorité de l'algorithme proposé pour l'identification des réponses acoustiques de différents nature (RIs courtes ou longue dispersées).



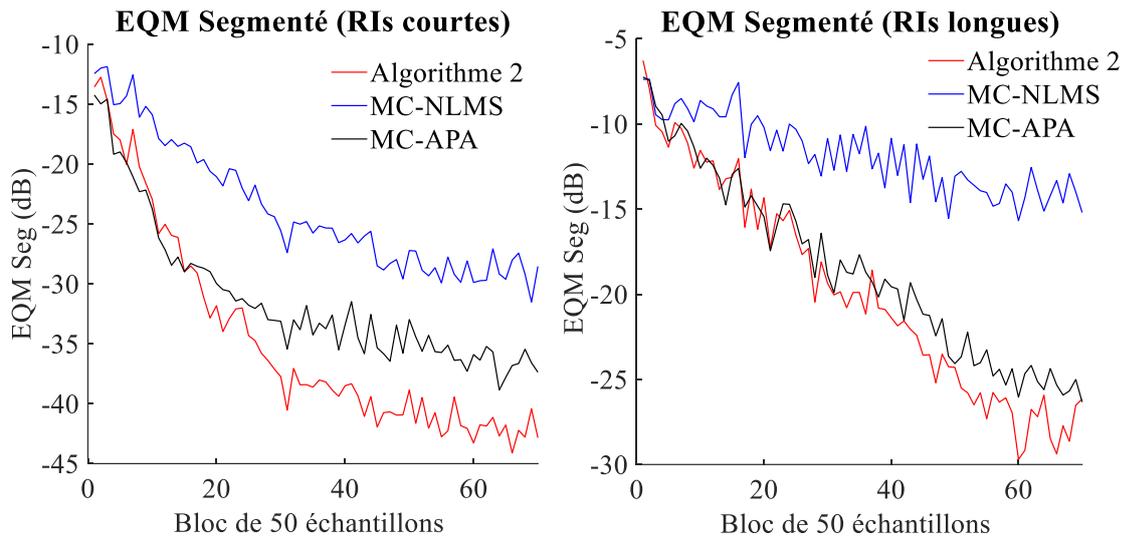
**Figure 3.9.** Désajustement obtenu par les différents algorithmes (Simulation des RIs courtes).



**Figure 3.10.** Désajustement obtenu par les différents algorithmes (Simulation des RIs longues et dispersées).

- **L'erreur quadratique moyenne segmentée (*EQM<sub>Seg</sub>*)**

Dans cette section, nous évaluons le critère d'erreur quadratique moyenne segmentée (*EQM<sub>Seg</sub>*). Les signaux et les paramètres utilisés pour chaque algorithme sont indiqués dans le **Tableau 3.5**. Les résultats de l'*EQM<sub>Seg</sub>* sont présentés dans la **Figure 3.11**. D'après la figure, on constate que les performances de l'algorithme proposé sont satisfaisantes pour toutes les simulations. En comparant l'algorithme proposé par rapport aux autres algorithmes, on peut conclure que l'algorithme MC-FNLMS proposé possède le meilleur taux de convergence tout avec un faible bruit résiduel en sortie.

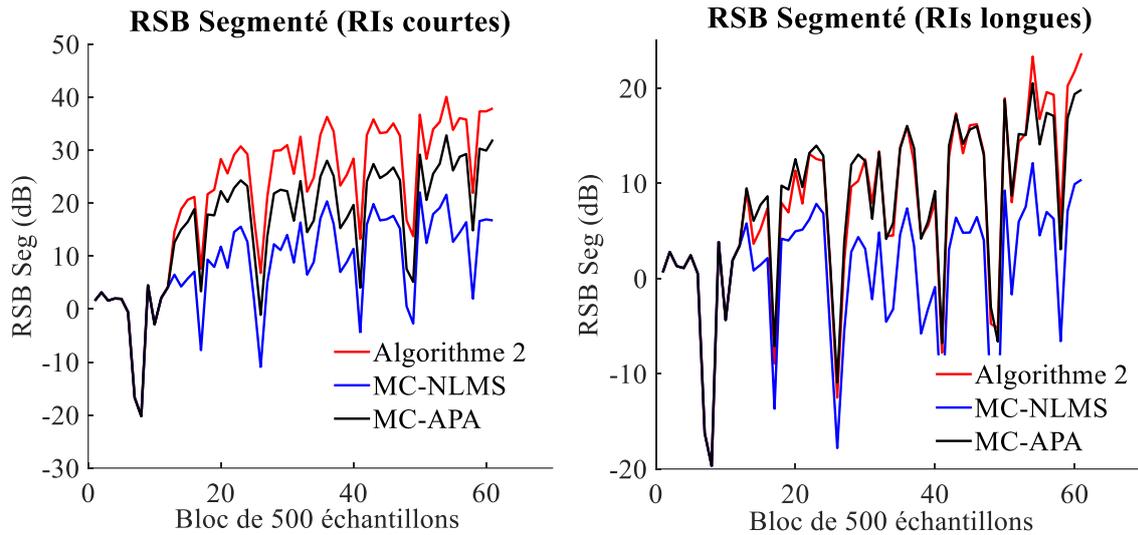


**Figure 3.11.** EQM Segmentée obtenue à la sortie par les différents algorithmes.

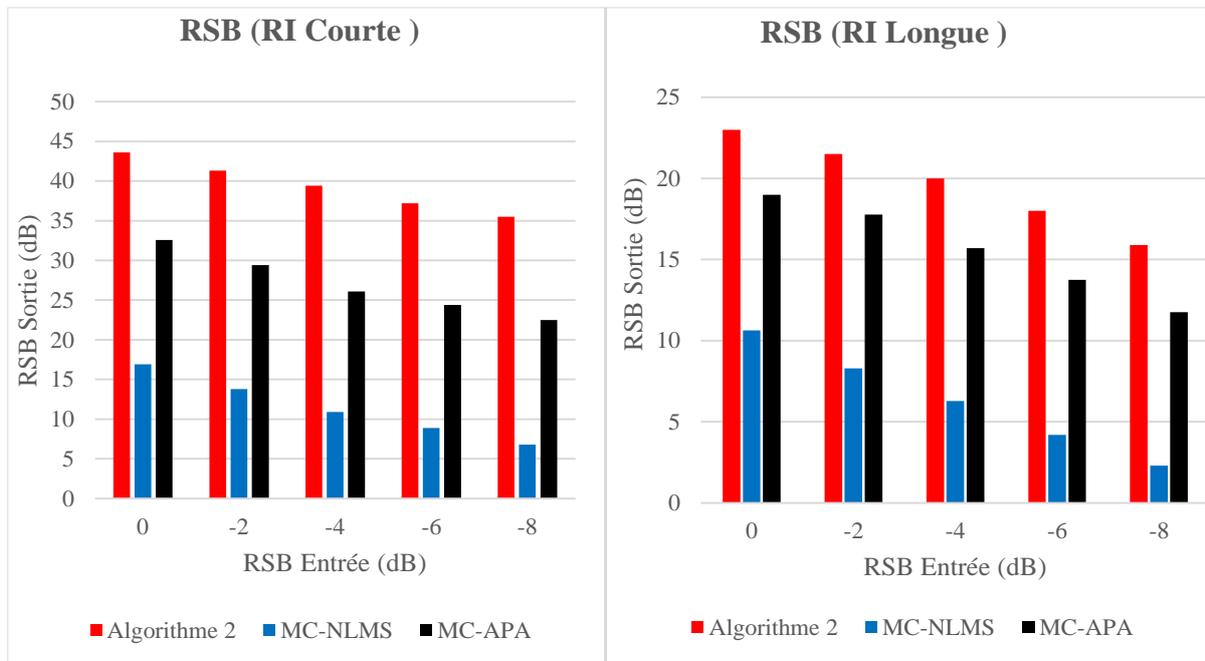
- **Rapport signal sur bruit segmenté (*RSB Seg*)**

Dans cette section, nous évaluons le critère de *RSB Seg* de l’algorithme MC-FNLMS, en le comparant aux algorithmes MC-NLMS et MC-APA. Les paramètres de simulation restent les mêmes comme ils sont détaillés dans le **Tableau 3.5**. Les résultats du critère de *RSB Seg* pour chaque algorithme sont donnés dans la **Figure 3.12**. D’après cette figure et pour le cas des réponses impulsionnelles acoustiques courtes, on peut constater un excellent résultat de rapport signal sur bruit obtenu par l’algorithme MC-FNLMS proposé. D’autre part, pour le cas des réponses acoustiques longues dispersées, les performances de réduction du bruit de l’algorithme MC-FNLMS sont comparables à celles du MC-APA. Cette expérience met en évidence l’efficacité de l’algorithme proposé dans la réduction du bruit acoustique, tout en conservant une complexité nettement inférieure à celle de l’algorithme MC-APA.

Dans la **Figure 3.13**, on présente le rapport signal à bruit obtenu à la sortie par chaque algorithme et pour différents rapports signal sur bruit en entrée (c-à-d 0 dB, -2 dB, -4 dB, -6 dB et -8 dB). Nous pouvons constater les performances les plus élevées de l’algorithme proposé dans tous les cas de réponse impulsionnelle acoustique. Cette expérience démontre l’efficacité de l’algorithme MC-FNLMS proposé dans l’annulation du bruit acoustique et l’amélioration de la qualité du signal parole.



**Figure 3.12.** RSB Segmenté obtenu en sortie des différents algorithmes.

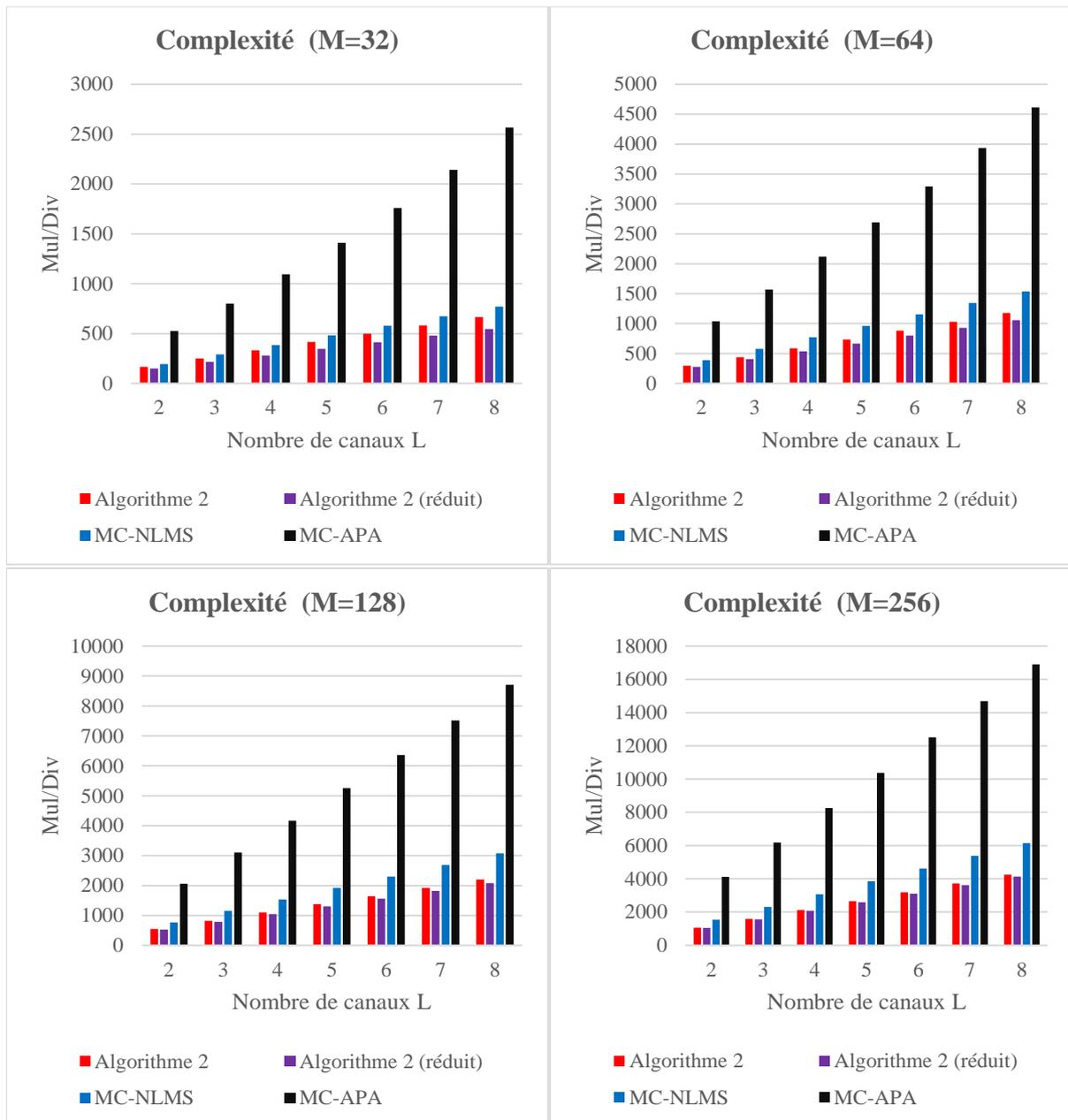


**Figure 3.13.** RSB obtenu à la sortie par les différents algorithmes (Simulation des RI longues et dispersées).

- **La complexité de calcul (CC)**

Dans la **Figure 3.14**, nous évaluons la complexité de calcul (CC) des trois algorithmes, à savoir MC-NLMS, MC-APA et les deux versions de l'algorithme MC-FNLMS proposés (version non réduite et réduite). À partir des résultats présentés, on remarque que l'algorithme proposé possède la plus faible CC par rapport aux algorithmes MC-APA et MC-NLMS, l'algorithme MC-APA requiert un nombre considérable d'opérations de multiplication et de

division, ce qui rend cet algorithme difficile à implémenter dans un calculateur de faible puissance calculatoire. À l'opposé, l'algorithme proposé (versions non réduite et réduite) se distingue par sa faible complexité avec sa bonne vitesse de convergence. Si on compare la version réduite de l'algorithme MC-FNLMS avec sa version non réduite, la version réduite demeure plus avantageuse en termes de complexité, car elle n'utilise qu'un seul canal pour estimer les paramètres de prédiction.



**Figure 3.14.** Évaluation la complexité de calcul (CC) pour les algorithmes MC-NLMS, MC-APA et MC-FNLMS (version non réduite ainsi que la version réduite).

### 3.5. Conclusion

Dans ce chapitre, nous avons présenté un nouvel algorithme multi-canaux (Algorithme 2), à convergence rapide et à faible complexité de calcul (CC), pour améliorer la qualité du signal de parole et réduire le bruit. Cet algorithme est dérivé de l'algorithme FNLMS [20] qui possède l'avantage de ses performances en termes de complexité et de vitesse de convergence.

La section 3.3 présente l'algorithme proposé, qui est une version multi-canaux dérivé de l'algorithme FNLMS par l'estimation des paramètres de prédiction pour chaque canal. Cet algorithme est nommé algorithme des moindres carrés normalisé rapide multi-canaux (MC-FNLMS). Une version de complexité réduite est également développée pour le cas des multi-signaux qui partage les mêmes propriétés statistiques. Une étude de convergence de l'algorithme proposé pour les signaux de type bruit blanc gaussien est aussi présentée, elle nous a permis d'établir les conditions sous lesquelles l'algorithme proposé converge vers les solutions optimales.

La section de simulation de ce chapitre présente une étude comparative entre l'algorithme proposé et les algorithmes MC-APA et MC-NLMS. Nous avons évalué les performances de l'algorithme proposé pour différents types de bruit en utilisant plusieurs critères, à savoir l'évaluation temporelle, le désajustement, l'erreur quadratique moyenne segmentée (*EQM Seg*), le rapport signal sur bruit (*RSB Seg*) et la complexité de calcul (CC). Nous avons démontré à partir de ces critères la supériorité de l'algorithme proposé pour le rehaussement du signal avec une faible complexité de calcul. Ce qui permet d'employer cet algorithme dans des environnements très bruyants, avec un coût d'implémentation raisonnable.

# CHAPITRE 4 : DEUX NOUVAUX ALGORITHMES MULTI-CANAUX EN SOUS BANDE

## 4.1.Introduction

Comme nous l'avons vu dans le chapitre précédent, le filtrage adaptatif multi-canaux est l'un des outils intéressants pour la réduction adaptative du bruit. Cependant, le filtrage multi-canaux souffre du nombre important de poids à mettre à jour, ce qui devient plus délicat si un grand nombre de filtres sont employé. En conséquence, les performances de la vitesse de convergence et de la complexité de calcul se dégradent encore davantage. Diverses techniques ont été développées pour améliorer les performances des algorithmes adaptatif multi-canaux. Pour augmenter la vitesse de convergence, la technique de décorrélation du signal peut être utilisée avant le processus de filtrage, mais au prix de l'augmentation de la complexité de calcul [20, 41]. La technique de mise à jour partielle peut réduire efficacement la complexité de calcul, cependant cette technique entraîne une dégradation dans la vitesse de convergence [28, 27]. L'approche du filtrage adaptatif en sous-bandes largement utilisée dans la littérature [42, 43, 44, 45], permet d'améliorer la vitesse de convergence et de réduire la complexité de calcul.

Dans ce chapitre, nous présentons deux algorithmes en sous-bande, basés sur l'algorithme développé en chapitre 3 MC-FNLMS (algorithme 2). L'idée consiste à développer des variantes en sous-bande (MC-SB-FNLMS) afin d'optimiser les performances en termes de vitesse de convergence et de complexité de calcul. Le filtrage en sous-bande permet d'utiliser des filtres adaptatifs de taille réduite et de les mettre à jour à une fréquence moins élevée que celle en pleine-bande, ce qui réduit significativement la complexité de calcul [46, 47, 48, 45]. De plus, grâce à la décorrélation résultant de la décomposition en sous-bandes, les algorithmes que nous allons proposer peuvent atteindre un taux de convergence optimal. Le premier algorithme présenté est appelé multi-bandes MC-SB-FNLMS référencé par "algorithme 3". Cet algorithme repose sur une réalisation multi-bandes du filtrage adaptatif en sous-bande, cette réalisation permet d'avoir une version plus rapide en termes de taux de convergence par rapport à la version en pleine-bande. Le second algorithme référencé par "algorithme 4" est basé sur la réalisation en boucle ouverte (MC-SB-FNLMS en boucle ouverte). Cette réalisation permet de réduire la complexité de calcul tout en améliorant légèrement la vitesse de convergence. Les algorithmes que nous présentons peuvent être appliqués à toutes les applications reposant sur le filtrage adaptatif multi-canaux.

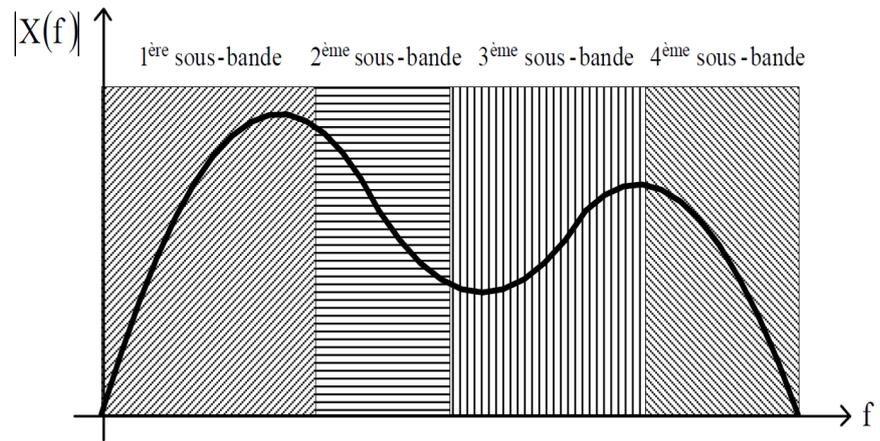
Ce chapitre est structuré comme suit : dans la Section 4.2, nous présentons le principe du filtrage adaptatif en sous-bande. Les deux algorithmes sous-bande MC-SB-FNLMS proposés (réalisés en multi-bandes et en boucle ouverte) sont présentés dans la Section **Erreur ! Source du renvoi introuvable.** Enfin, dans la Section **Erreur ! Source du renvoi introuvable.**, nous évaluons les performances des algorithmes proposés pour l'identification adaptative multi-canaux et pour le filtrage adaptatif spatiaux-fréquentielle.

## 4.2. Principe du filtrage adaptatif en sous-bande

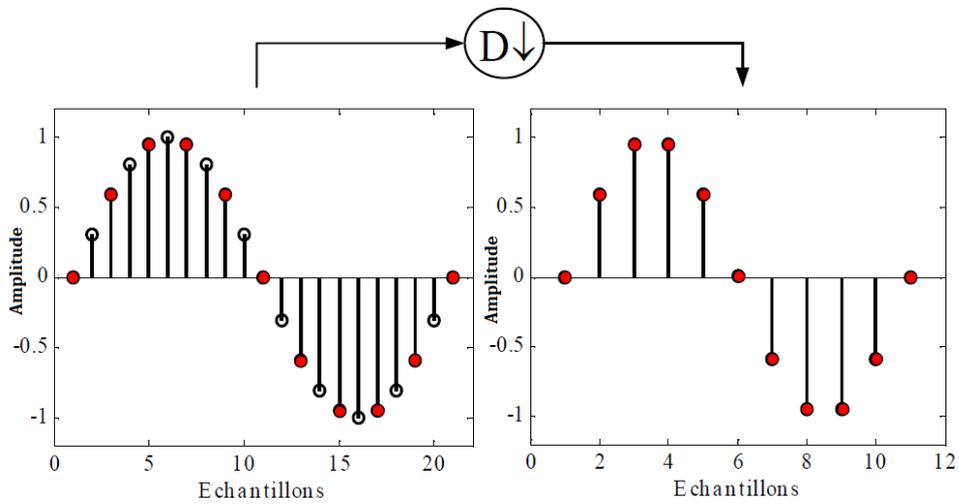
Les techniques de filtrage adaptatif en sous-bandes sont généralement employées pour le but de réduire la complexité de calcul, en particulier pour l'identification des longues réponses impulsionnelles [46]. Parfois, ce traitement améliore également la vitesse de convergence des algorithmes adaptatifs, notamment pour les signaux colorés. Cette approche repose sur la décomposition de spectre de signal en sous-bandes fréquentielles et le sur-échantillonnage en temps en utilisant des banques de filtres.

Avant le traitement en sous-bandes, le signal d'entrée se décompose en  $N_B$  sous-bandes fréquentielles à l'aide des filtres d'analyses (**AF**). Ces filtres se caractérisent par des réponses fréquentielles complémentaires entre elles, afin de générer un ensemble de sous-bandes spectrales représentatives du spectre du signal original [44]. Dans le cas le plus fréquent, le signal est analysé (filtré) par  $N_B$  filtres passe-bande. Cette approche permet d'obtenir  $N_B$  sous-spectres correspondant chacun à une partie du spectre du signal original (**Figure 4.1**). Ces signaux se traitent par le traitement en sous-bande qui opère généralement avec une fréquence plus faible que la fréquence en pleine-bande.

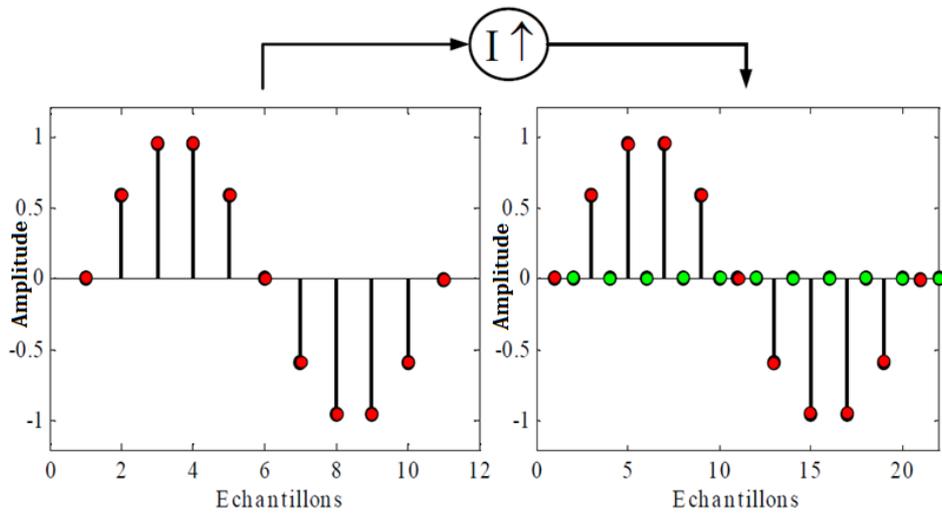
Après le traitement en sous-bandes, les signaux traités correspondent à chaque sous-bande, sont interpolés (sur-échantillonnés) avec le même facteur de décimation  $D$  comme illustré par la **Figure 4.2**. L'interpolateur augmente le taux d'échantillonnage du signal d'entrée en insérant d'échantillons nuls entre chaque paire d'échantillons du signal décimé (**Figure 4.3**). Après l'interpolation, les signaux se filtrent une autre fois en utilisant les filtres de synthèse (ils sont généralement identiques à ceux du filtre d'analyse). Les signaux résultant de ce filtrage sont sommés pour reconstituer les signaux en pleine-bande [46].



**Figure 4.1.** Exemple de découpage d'un signal en quatre sous-bandes.

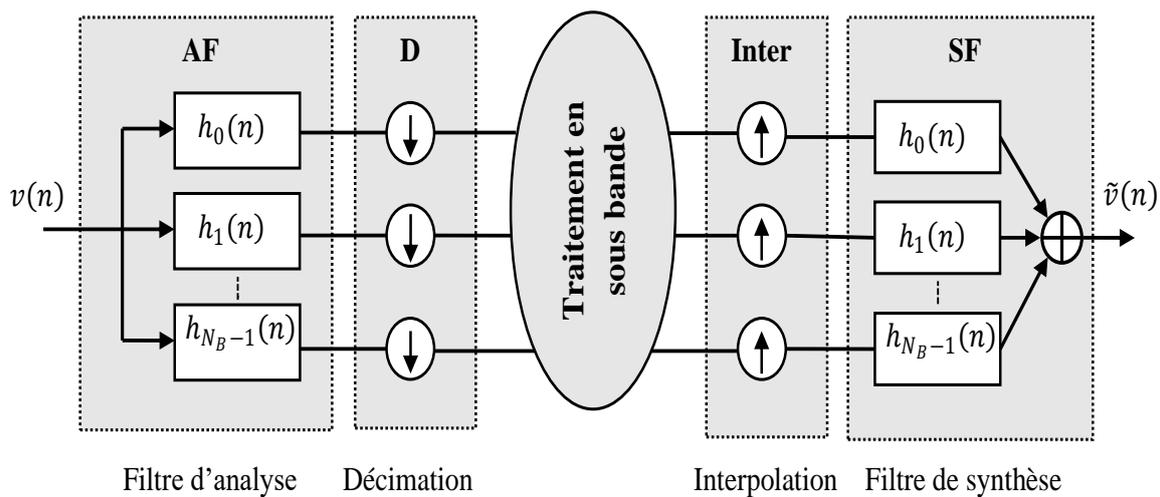


**Figure 4.2.** Opération de décimation par un facteur  $D$ .



**Figure 4.3.** Opération d'interpolation par un facteur  $I$ .

Dans le filtrage adaptatif en sous-bande, la fréquence de mis à jour des filtres adaptatifs est plus faible que celle en bande pleine, ce qui permet d'utiliser un nombre réduit de coefficients (poids) de filtre pour représenter une réponse impulsionnelle (RI) de certaine longueur. La mise en œuvre en sous-bandes d'un algorithme adaptatif de complexité  $O(M)$ , nécessite  $M/D^2$  opérations arithmétiques au lieu de  $M$  (où  $M$  est la longueur de la RI). De plus, l'approche de filtrage adaptatif en sous-bandes réduit les dynamiques spectrales des signaux d'entrée des filtres adaptatifs, ce qui peut améliorer la vitesse de convergence par rapport au traitement en pleine-bande. De nombreuses structures de filtrage adaptatif en sous-bandes ont été développées : les structures en boucle ouverte, en boucle fermée et multi-bandes [46, 49]. Ces structures peuvent être appliquées à des problèmes divers (l'annulation d'écho acoustique, réduction de bruit adaptatif, etc.) pour le but de réduire la complexité de calcul et l'identification rapide des réponses impulsionnelles dispersées. Le schéma global d'un traitement en sous-bandes est représenté dans la **Figure 4.4**.



**Figure 4.4.** Schéma global d'un traitement en sous-bande.

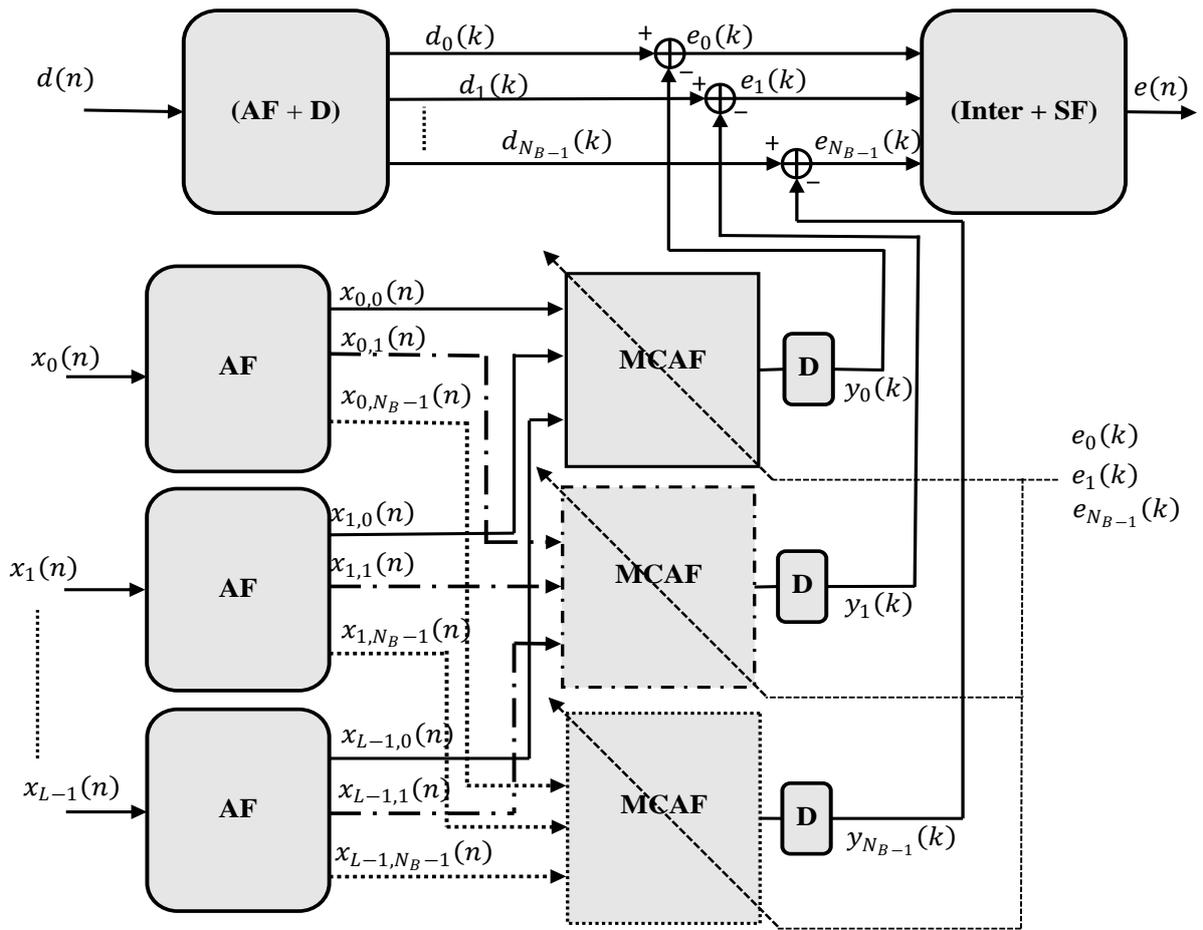
### 4.3. Les algorithmes en sous-bandes proposés

Dans cette section, nous développons deux nouveaux algorithmes multi-canaux rapide en sous-bandes (MC-SB-FNLMS). Nous proposons deux structures en sous-bandes différentes, la structure multi-bandes et la structure en boucle ouverte. La première version proposée en sous-bande (Algorithme 3) repose sur une réalisation multi-bandes, cet algorithme possède un taux de convergence optimale. La deuxième version repose sur la réalisation en boucle ouverte ; cette réalisation permet de réduire considérablement la complexité de calcul tout en améliorant

légèrement la vitesse de convergence. Pour tout ce qui suivra, nous notons le signal désiré par  $d(n)$  et les  $L$  signal d'entrée par  $x_i(n)$  ( $i = 0, 1, 2, \dots, L - 1$ ). Le nombre des filtres d'analyses et de synthèses est indiqué par  $N_B$  ( $N_{Bande}$ ), et  $D$  et  $I$  pour le facteur de décimation et d'interpolation. Pour chaque canal et chaque sous-bande, nous faisons référence respectivement au signal correspondant par les indices  $i$  et  $j$ .

#### 4.3.1. Algorithme MC-SB-FNLMS basé sur la réalisation multi-bandes (Algorithme 3 proposé)

L'implémentation multi-bandes de l'algorithme MC-SB-FNLMS proposé est présenté dans la **Figure 4.5**. Dans cette structure, nous utilisons des filtres adaptatifs en pleine-bande pour produire l'estimation des signaux sous bande désirés  $y_i(k)$ . Les filtres adaptatifs multi-canaux (MCAF) sont identiques de longueur  $M$  pour chacun, ils sont placés avant le décimateur, contrairement au filtrage adaptatif en sou-bande basé sur la structure en boucle ouverte où les filtres adaptatifs multi-canaux de longue  $M_s = M/D$  sont placés après le décimateur. Dans notre algorithme proposé, les filtres adaptatifs en pleine-bande  $w_i(k)$  de la partie **MCAF**, sont mets à jour à un temps discret  $k$  de fréquence plus faible de celle de bande plaine. Pour chaque canal et pour chaque sous-bande, nous calculons les premiers coefficients de prédiction  $a_{i,j}(n)$ , qui servent à calculer le vecteur d'adaptation  $\gamma_{i,j}(n)\tilde{\mathcal{C}}_{i,j}(n)$ . Le nombre total de multiplications et de divisions est égal à  $LM(N_B/D + 1) + L(16N_B + 2N_B/D)$ , la complexité de calcul avec une décimation critique ( $D = N_B$ ) devient de  $2ML + 16LN_B + 2L$  multiplications et divisions par itération. Si les signaux d'entrée possèdent des propriétés statistiques communes (ce qui est le cas courant), alors un canal pour chaque sous-bande peut suffire pour estimer les coefficients de prédiction, nous avons donc  $a_{i,j}(k) = a_{0,j}(k)$ . Dans ce cas, la complexité de cet algorithme avec une décimation critique ( $D = N_B$ ) sera égale à  $2ML + 16N_B + 2L$  multiplications et divisions par itération. L'algorithme proposé peut donner une bonne identification des réponses impulsionnelles inconnues si les filtres d'analyse sont de réponse pratiquement complémentaire. L'algorithme MC-SB-FNLMS proposé basé sur la réalisation multi-bandes est décrit dans le **Tableau 4.1**.



**Figure 4.5.** Structure multi-bandes de filtrage adaptatif multi-canaux en sous-bande.

**Tableau 4.1.** L'algorithme MC-SB-FNLMS multi-bandes (Algorithme 3).

	Estimation des paramètres de prédictions en utilisant $L$ canal.	Estimation des paramètres de prédictions en utilisant un seul canal.
	Mult/Div	Mult/Div
<b>Initialisation</b> $\tilde{\mathbf{C}}_{i,j}(0) = \mathbf{w}_i(0) = \mathbf{0}_{M \times 1}; \gamma_{i,j}(0) = 1; r_{1,i,j}(0) = 0; p_{i,j}(n) = r_{0,i,j}(0) = E_0$		
<b>Prédiction</b> $r_{0,i,j}(n) = \lambda_a r_{0,i,j}(n-1) + x_{i,j}(n)x_{i,j}(n)$ $r_{1,i,j}(n) = \lambda_a r_{1,i,j}(n-1) + x_{i,j}(n)x_{i,j}(n-1)$ $a_{i,j}(n) = \frac{r_{1,i,j}(n)}{(r_{0,i,j}(n) + c_a)}$ $\varepsilon_{i,j}(n) = x_{i,j}(n) - a_{i,j}(n)x_{i,j}(n-1)$ $p_{i,j}(n) = \lambda p_{i,j}(n-1) + \varepsilon_{i,j}^2(n)$ $\begin{bmatrix} \tilde{\mathbf{C}}_{i,j}(n) \\ c_{i,j}(n) \end{bmatrix} = \begin{bmatrix} -\frac{\varepsilon_{i,j}(n)}{\lambda p_{i,j}(n-1) + c_0} \\ \tilde{\mathbf{C}}_{i,j}(n) \end{bmatrix}$ $\delta(n) = \tilde{c}_{i,j}(n)x_{i,j}(n-M) + \frac{x_{i,j}(n)\varepsilon_{i,j}(n)}{\lambda p_{i,j}(n-1) + c_0}$ $\gamma_{i,j}(n) = \frac{\gamma_{i,j}(n-1)}{1 + \gamma_{i,j}(n-1) + \delta_{i,j}(n)}$	$2LN_B$ $2LN_B$ $LN_B$ $LN_B$ $2LN_B$ $3LN_B$ $4LN_B$ $LN_B$ $LM$	$2N_B$ $2N_B$ $N_B$ $N_B$ $2N_B$ $3N_B$ $4N_B$ $N_B$ $LM$
<b>Filtrage</b> $y(n) = \sum_{i=0}^{L-1} \mathbf{X}_i^T(n) \mathbf{w}_i(n-1)$		
<b>Mise à jour en temps d'échantillonnage faible <math>k</math></b> $e_j(k) = d_j(k) - y_j(k)$ $\mathbf{w}_i(k) = \mathbf{w}_i(k-1) - \mu \sum_{j=0}^{N_B-1} e_j(k) \gamma_{i,j}(k) \tilde{\mathbf{C}}_{i,j}(k)$	$L(M+2)(N_B/D)$	$L(M+2)(N_B/D)$
	<b>Total Mult/Div:</b>	<b>Total Mult/Div:</b>

$LLM(N_B/D + 1)$ $+ L(16N_B$ $+ 2N_B/D)$	$LM(N_B/D + 1)$ $+ 16N_B + 2LN_B/D$
--	--

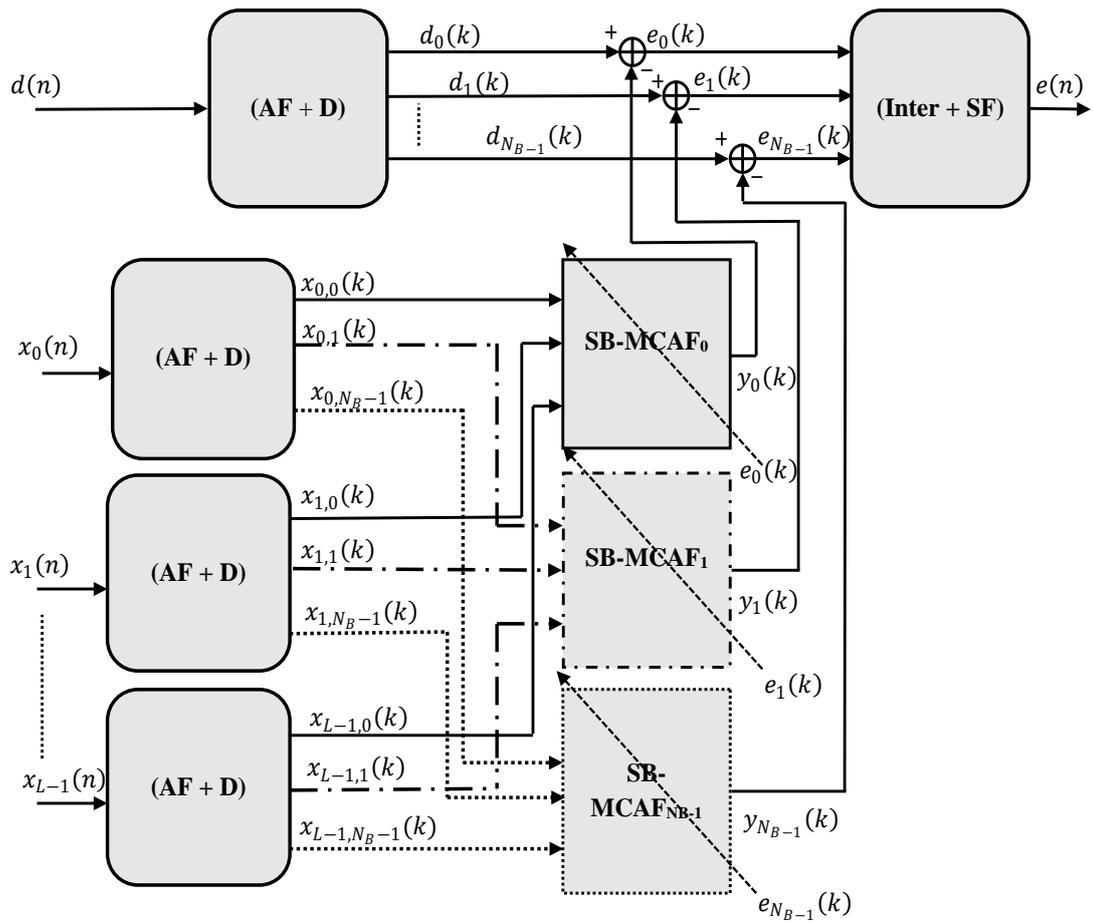
### 4.3.2. Algorithme MC-SB-FNLMS basé sur la réalisation en boucle ouverte (Algorithme 4 proposé)

La réalisation en boucle ouverte de l'algorithme MC-SB-FNLMS, est présentée à la **Figure 4.6**. Chaque signal  $x_i(n)$  et  $d(n)$  est filtré puis décimé, ce qui produit  $N_B$  signaux en sous-bandes avec un faible taux d'échantillonnage. Les signaux en sous-bandes  $x_{i,j}(k)$  servent d'entrée à  $N_B$  filtres adaptatifs multi-canaux. L'erreur  $e_j(k)$ , correspond à la différence entre la sortie  $y_j(k)$  et le signal désiré en sous-bande  $d_j(k)$  est utilisée pour la mise à jour des sous-filtres adaptatifs multi-canaux (**SB-MCAF<sub>j</sub>**) à un taux d'échantillonnage réduit  $k = n/D$ . Ces filtres possèdent une longueur ( $M_S = M/D$ ) qui peuvent être beaucoup inférieures à celles requises en pleine-bande ( $M$ ). L'erreur en pleine-bande est synthétisée en utilisant le filtre de synthèse (**SF**) après l'interpolation des signaux sous-échantillonnés (**Inter**).

L'algorithme MC-SB-FNLMS en boucle ouverte proposé minimise les fonctions objectifs  $\vartheta_j(n) = E[e_j^2(n)]$ , qui reposent sur les signaux d'erreurs en sous-bandes [46]. En raison des chevauchements et de la non-complémentarité de la décomposition en sous-bandes, qui est générées par le banc de filtres non idéale, l'algorithme proposé pourrait ne pas identifier avec précision les réponses inconnus. Cependant, cet inconvénient n'est pas très critique, car pour la plupart des applications, les erreurs engendrées par la décomposition en sous-bandes peuvent être bien plus faible que l'EQM théorique en pleine-bande lui-même (par exemple dans le cas de GSC).

La complexité de calcul de l'algorithme 4 est égale à  $2LN_B M_S/D + 18LN_B/D$  multiplications et divisions par itération. Avec une décimation critique ( $D = N_B$ ), cette complexité se réduit à  $2LM_S + 18L$ , où  $M_S = M/D$ . Comme pour l'algorithme 3 proposé, si les signaux d'entrée proviennent du même processus, les paramètres de prédictions peuvent être estimés en utilisant un seul canal. Par conséquent, la complexité de calcul de l'algorithme devienne  $2LN_B M_S/D + 2LN_B/D + 16N_B/D$  multiplications et divisions par itération.

En comparant la réalisation en boucle ouverte de l'algorithme proposé avec celle du multi-bandes, la réalisation en boucle ouverte présente l'avantage d'une complexité de calcul réduite. L'algorithme MC-SB-FNLMS proposé en boucle ouverte est résumé dans le **Tableau 4.2**.



**Figure 4.6.** Structure en boucle ouverte de filtrage adaptatif multi-canaux en sous-bandes.

**Tableau 4.2.** Algorithme MC-SB-FNLMS en boucle ouverte (Algorithme 4)

	Estimation des paramètres de prédictions en utilisant $L$ canal.	Estimation des paramètres de prédictions en utilisant un seul canal.
	Mult/Div	Mult/Div
<p><b>Initialisation</b></p> $\tilde{\mathbf{C}}_{i,j}(0) = \mathbf{w}_{i,j}(0) = \mathbf{0}_{M_S \times 1}, \quad \gamma_{i,j}(0) = 1,$ $r_{1,i,j}(0) = 0, p_{i,j}(0) = r_{0,i,j}(0) = E_0$		
<p><b>Mise à jour de tous les paramètres en temps d'échantillonnage faible <math>k</math></b></p> <p><b>Prédiction</b></p> $r_{0,i,j}(k) = \lambda_a r_{0,i,j}(k-1) + x_{i,j}(k)x_{i,j}(k)$ $r_{1,i,j}(k) = \lambda_a r_{1,i,j}(k-1) + x_{i,j}(k)x_{i,j}(k-1)$ $a_{i,j}(k) = \frac{r_{1,i,j}(k)}{(r_{0,i,j}(k) + c_a)}$ $\varepsilon_{i,j}(k) = x_{i,j}(k) - a_{i,j}(k)x_{i,j}(k-1)$ $p_{i,j}(k) = \lambda p_{i,j}(k-1) + \varepsilon_{i,j}^2(k)$ $\begin{bmatrix} \tilde{\mathbf{C}}_{i,j}(k) \\ c_{i,j}(k) \end{bmatrix} = \begin{bmatrix} -\frac{\varepsilon_{i,j}(k)}{\lambda p_{i,j}(k-1) + c_0} \\ \tilde{\mathbf{C}}_{i,j}(k) \end{bmatrix}$ $\delta(k) = \tilde{c}_{i,j}(k)x_{i,j}(k-M) + \frac{x_{i,j}(k)\varepsilon_{i,j}(k)}{\lambda p_{i,j}(k-1) + c_0}$ $\gamma_{i,j}(k) = \frac{\gamma_{i,j}(k-1)}{1 + \gamma_{i,j}(k-1) + \delta_{i,j}(k)}$ <p><b>Filtrage</b></p> $e_j(k) = d_j(k) - \sum_{i=0}^{L-1} \mathbf{X}_{i,j}^T(k) \mathbf{w}_{i,j}(k-1)$ $\mathbf{w}_{i,j}(k) = \mathbf{w}_{i,j}(k-1) - \mu e_j(k) \gamma_{i,j}(k) \tilde{\mathbf{C}}_{i,j}(k)$	$2LN_B/D$ $2LN_B/D$ $LN_B/D$ $LN_B/D$ $2LN_B/D$ $3LN_B/D$ $4LN_B/D$ $LN_B/D$ $LN_B M_S/D$ $(LN_B M_S + 2LN_B)/D$	$2N_B/D$ $2N_B/D$ $N_B/D$ $N_B/D$ $2N_B/D$ $3N_B/D$ $4N_B/D$ $N_B/D$ $LN_B M_S/D$ $(LN_B M_S + 2LN_B)/D$
	<b>Total Mult/Div</b>	<b>Total Mult/Div</b>
	$2LN_B M_S/D$ $+ 18LN_B/D$	$2LN_B M_S/D$ $+ 2LN_B/D$ $+ 16N_B/D$

#### 4.4. Simulation et résultats

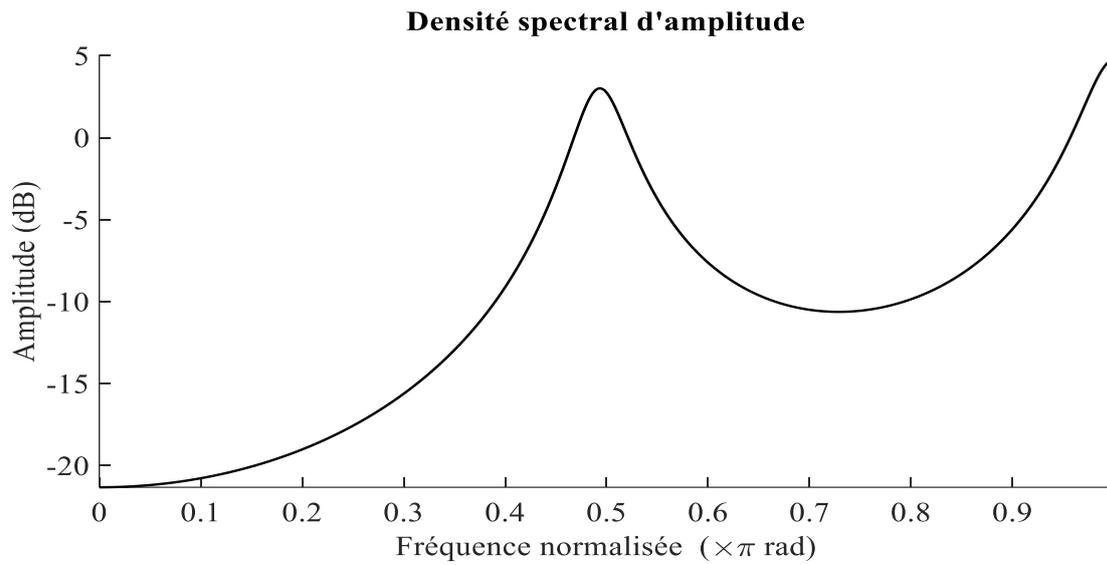
Dans cette section, nous analysons les performances des algorithmes développés ; à savoir l'algorithme MC-SB-FNLMS multi-bandes (algorithme 3) et l'algorithme MC-SB-FNLMS en boucle ouverte (algorithme 4). L'évaluation des performances est donnée en termes de taux de convergence et de complexité de calcul. La simulation est divisée en deux parties de problème différent. Dans la première partie, nous analysons le problème d'identification multi-canaux des réponses impulsionnelles aléatoires. Cette partie de la simulation repose sur le tracé des courbes de l'erreur quadratique moyen segmentée (*EQM Seg*). Dans la deuxième partie de la simulation, nous considérons une application des algorithmes proposés dans le Beamforming adaptatif GSC. Comme dans la première partie de simulation, l'évaluation dans cette deuxième partie est basée sur le tracé des courbes d'*EQM Seg*. Dans les deux parties de la simulation, on fait une étude comparative des deux algorithmes proposés avec les algorithmes MC-FNLMS (proposé au chapitre 4) et MC-NLMS en sous-bandes. Nous utilisons des signaux d'entrée  $x_i(n)$  colorés, chacun est généré en filtrant un bruit blanc gaussien à travers d'un filtre générateur qui donne à sa sortie les signaux  $x_i(n)$ , les signaux  $x_i(n)$  possèdent la densité spectrale d'amplitude donnée dans la **Figure 4.7**. Les signaux  $x_i(n)$  sont décrits par le modèle autorégressif (AR) suivant :

$$x_i(n) = a_1 x_i(n-1) + a_2 x_i(n-2) + a_3 x_i(n-3) + v(n) \quad (4.1)$$

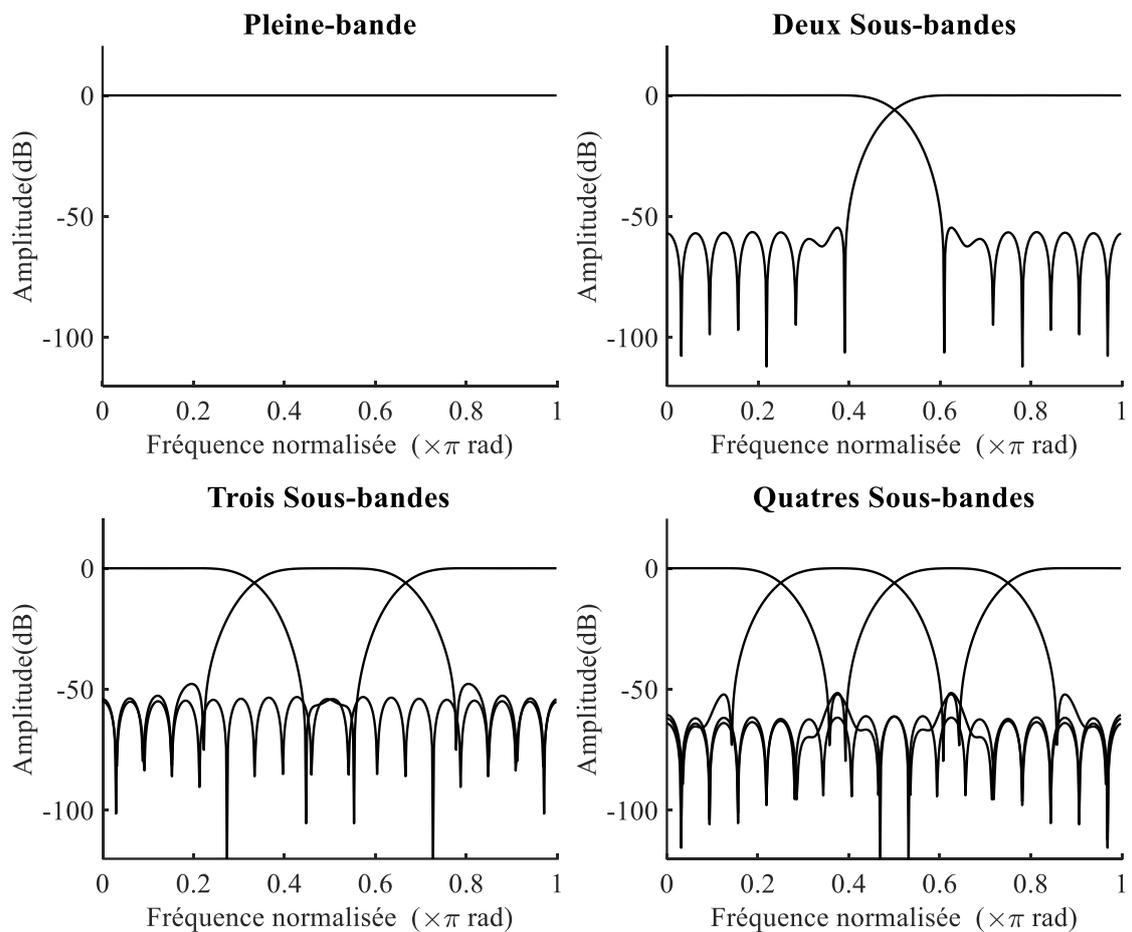
Où :

$a_1 = -0.865$ ,  $a_2 = -0.8066$ ,  $a_3 = -0.7703$  et  $v(n)$  est un bruit blanc gaussien de variance  $\sigma_v^2 = 1$ .

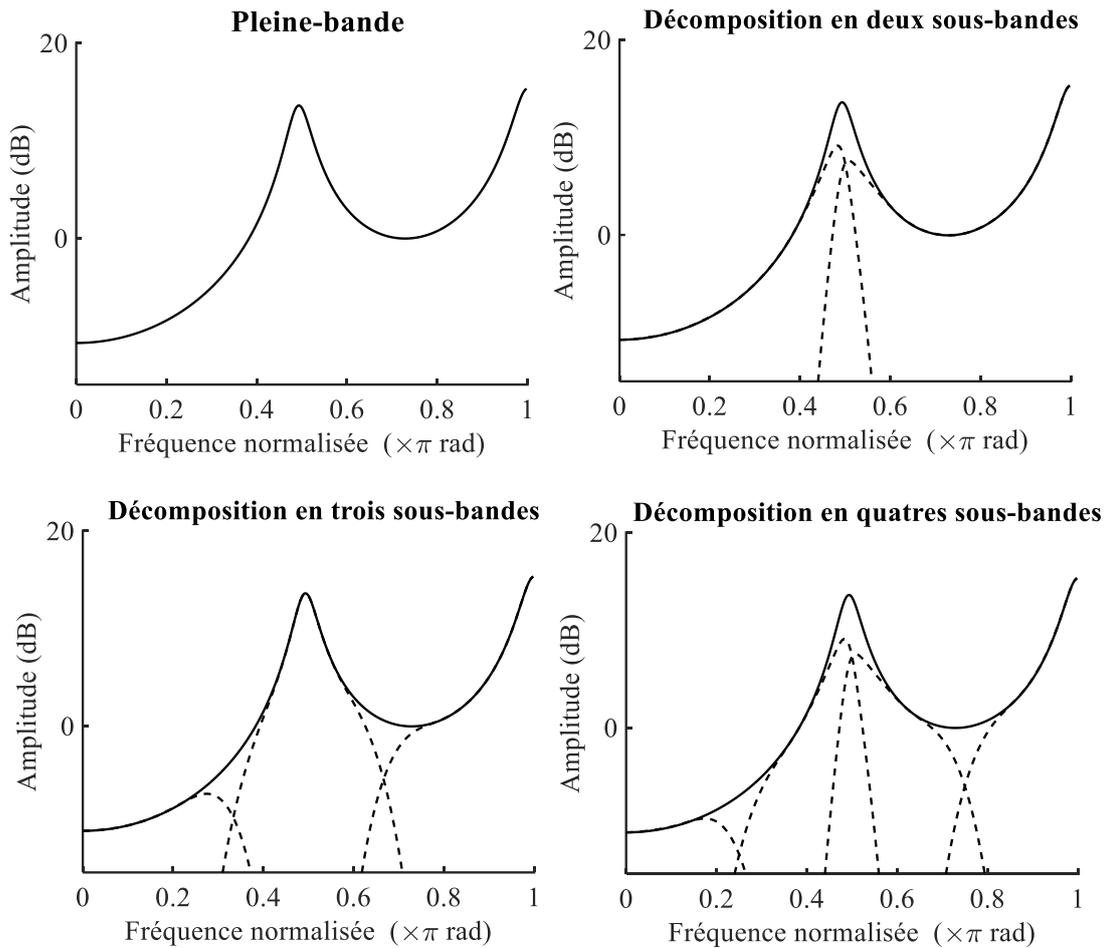
Nous utilisons  $N_B$  filtres d'analyse et de synthèse de réponse impulsionnelle fini (RIF), ces filtres de longueur de 32 échantillons (32 poids) pour chacun. La réponse fréquentielle d'amplitude de ces filtres pour le cas  $N_B = 4$  est représentée sur la **Figure 4.8**. Après le filtrage à travers les filtres d'analyses, la densité spectrale d'amplitude ( $DSA = \sqrt{DSP}$ ) des signaux en sous-bande est représenté dans la **Figure 4.9**. Les paramètres de chaque algorithme utilisé dans cette simulation sont donnés dans le **Tableau 4.3**.



**Figure 4.7.** Densité spectrale d'amplitude (*DSA*) des signaux d'entrée  $x_i(n)$ .



**Figure 4.8.** Réponses fréquentielles d'amplitude des filtres d'analyses.



**Figure 4.9.** Densité spectrale d'amplitude (*DSA*) des signaux sous-bande après l'analyse.

**Tableau 4.3.** Paramètres de simulation.

Signaux	Signaux autorégressifs $x_i(n)$
Paramètres des filtres adaptatifs	Quatre filtres RIF adaptatifs pleine-bande, $w_1(n)$ , $w_2(n)$ , $w_3(n)$ and $w_4(n)$ avec $M_1 = M_2 = M_3 = M_4 = 100$
Paramètres d'algorithme SB-MC-NLMS (multi-bandes et en boucle ouverte)	Le pas d'adaptation : $\mu = 1$ . (Vitesse de convergence optimale)
Paramètres des algorithmes MC-SB-FNLMS proposés (multi-bandes et en boucle ouverte)	Facteurs d'oubli : $\lambda_a = 0.97$ ; $\lambda = 0.9985$ ; Petites constantes : $c_a = 0.001$ ; $c_0 = 0.1$ . Le pas d'adaptation : $\mu_{MC-SB-FNLMS} = 1/4$ .

#### 4.4.1. Identification multi-canaux

Dans cette section, nous nous intéressons à l'identification de quatre réponses impulsionnelles aléatoires (RIs). Chaque RI possède une longueur de 100 échantillons. Le signal désiré  $d(n)$  est obtenu en filtrant chaque signal coloré  $x_i(n)$  à travers de RI correspondante (c-à-d  $h_i(n)$ ). Le signal  $d(n)$  est donné par :  $d(n) = \sum_{i=0}^3 x_i(n) * h_i(n)$ . Les signaux  $x_i(n)$  possèdent les mêmes propriétés statistiques. Pour chaque algorithme, nous traçons l'erreur quadratique moyenne segmentée (*EQM Seg*) en utilisant des segments de blocs ( $N_{block}$ ) de 150 échantillons. L'*EQM Seg* est calculée par l'équation suivante :

$$EQM Seg_{dB} (block) = \sum_{n=100(block-1)+1}^{100(block)} 10 \log_{10}[e^2(n)] \quad (4.2)$$

#### A) Résultats obtenus par l'algorithme MC-SB-FNLMS multi-bandes (Algorithme 3) :

- *EQM Seg*

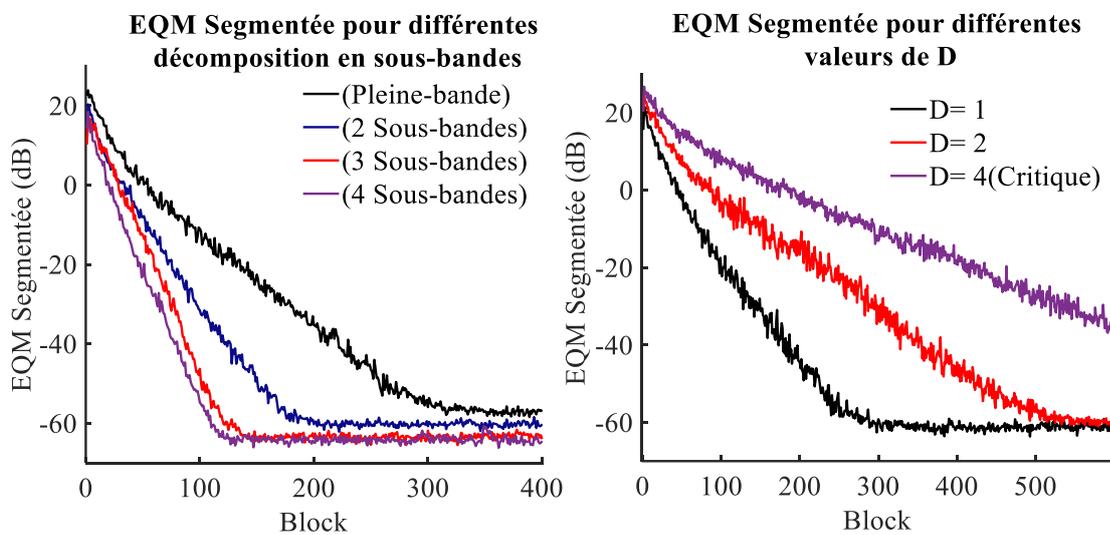
Nous évaluons l'erreur quadratique moyenne segmentée (*EQM Seg* avec 100 échantillons par segment) de l'algorithme 3 (c-à-d MC-SB-FNLMS réalisé en multi-bandes), par rapport à sa version en pleine-bande développé au chapitre 4 et par rapport à l'algorithme MC-SB-NLMS. Les résultats obtenus pour chaque algorithme sont présentés dans la **Figure 4.10** et la **Figure 4.11**.

D'après les résultats de l'évaluation de l'erreur quadratique moyenne segmentée (*EQM Seg*) présentés dans la **Figure 4.10** (à gauche), est il est remarqué que le taux de convergence de l'algorithme proposé en sous-bande est plus important que sa version en pleine-bande (Algorithme 2), en particulier pour une décomposition en sous-bandes élevée. Ceci est dû de la décorrélation introduite par la décomposition en sous-bandes, qui diminue la dynamique spectrale du signal. On rappelle que l'algorithme en pleine-bande (c-à-d algorithme 2), nous utilisons uniquement les premiers coefficients de prédictions AR pour décorréler les signaux  $x_i(n)$ . En revanche, la décomposition en sous-bandes introduit une décorrélation supplémentaire et par conséquent, une vitesse de convergence optimale peut être atteinte.

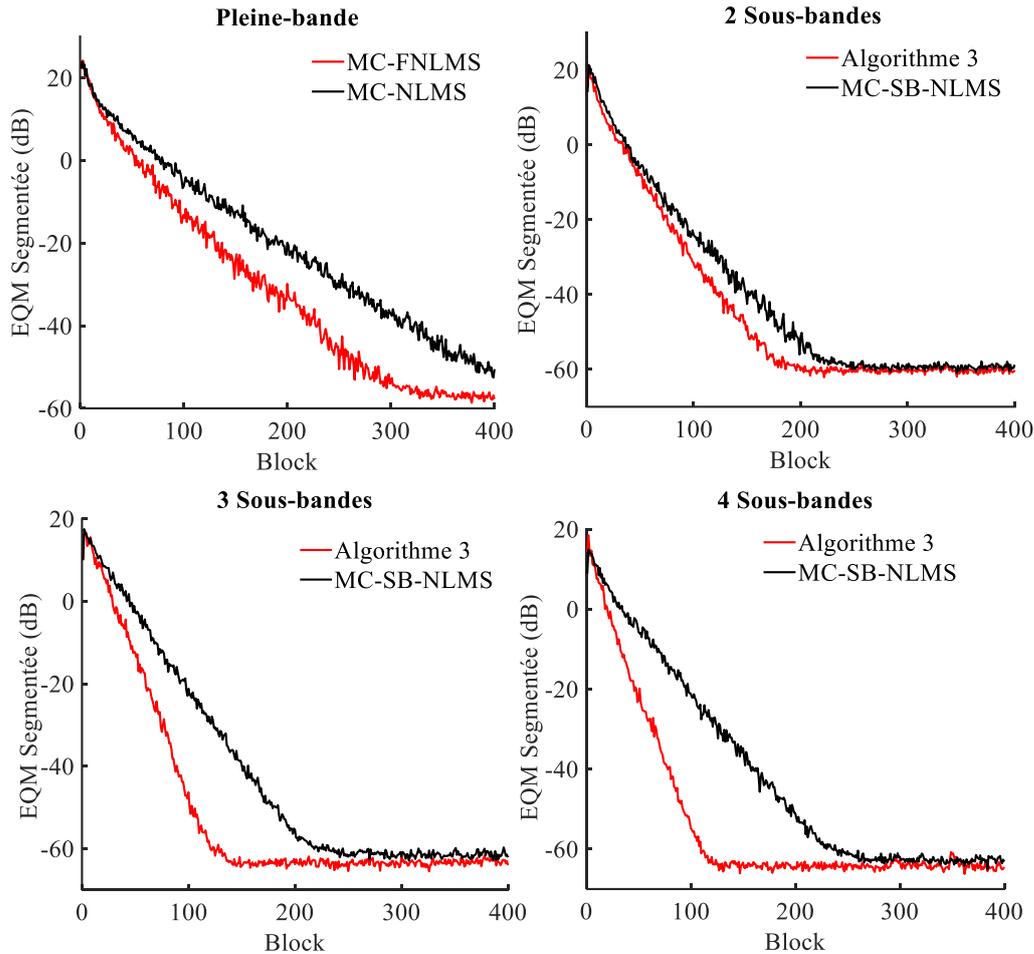
Afin d'examiner l'effet du choix du facteur de décimation  $D$  et pour la décomposition en quatre sous-bandes ( $N_B = 4$ ), nous traçons dans la **Figure 4.10** (à droite) pour différents facteurs de décimation, l'erreur quadratique moyenne segmentée (*EQM Seg*). Comme le montre cette

figure, les valeurs élevées du facteur de décimation réduisent le taux de convergence, ce qui n'est pas surprenant, tandis que la taille des filtres adaptatifs dans cet algorithme reste constante.

Dans la **Figure 4.11**, nous présentons les évolutions de l'erreur quadratique moyenne segmentée (*EQM Seg*), en comparant l'algorithme proposé (algorithme 3) avec l'algorithme classique MC-SB-NLMS, le facteur de décimation est  $D = 1$ . Pour 2, 3 et 4 décompositions en sous-bandes, on observe que l'algorithme proposé possède un taux de convergence plus important que l'agorithme classique MC-SB-NLMS, en particulier lorsque le nombre de sous-bandes est élevé, ce résultat montre la superiorité de l'algorithme proposé par rapport à l'algorithme classique MC-SB-NLMS tout avec une faible complexité de calcul.



**Figure 4.10.** *EQM Seg* (100 échantillons par segment) obtenu par l'algorithme 3, à gauche : *EQM Seg* différentes décomosition en sous-bandes ( $D = 1$ ), à droite : 4 sous-bandes avec différentes valeurs de  $D$  ( $\mu = 0.05$ ).



**Figure 4.11.** Comparaison de l'*EQM Seg* (100 échantillons par segment) obtenue par l'algorithme 3 et l'algorithme MC-SB-NLMS (multi-bandes) pour différentes décompositions en sous-bandes.

- **Complexité computationnel (CC)**

Dans la **Figure 4.12**, nous présentons la complexité de calcul (CC) des algorithmes MC-SB-NLMS (multi-bandes), l'algorithme 2, ainsi que l'algorithme 3 proposé (version non réduite et réduite). A partir des résultats de la **Figure 4.12**, nous pouvons constater que l'algorithme proposés possède une complexité de calcul plus faible par rapport à l'algorithme MC-SB-NLMS pour tous les choix de  $L$ ,  $N_B$  et  $D$ . Cependant, en comparant l'algorithmes proposés avec l'algorithme MC-FNLMS (Algorithme 2), l'algorithme 3 requiert plus d'opérations arithmétiques pour un facteur de décimation important, sauf pour le cas ( $D = N_B$ ) où l'algorithme proposé atteint une complexité optimale, en particulier de celle réduite. On peut dire que les résultats donnés par cette comparaison, montrent la supériorité en termes de CC de l'algorithme proposé par rapport à l'algorithme classique MC-SB-NLMS, tout en gardant un bon taux de convergence.

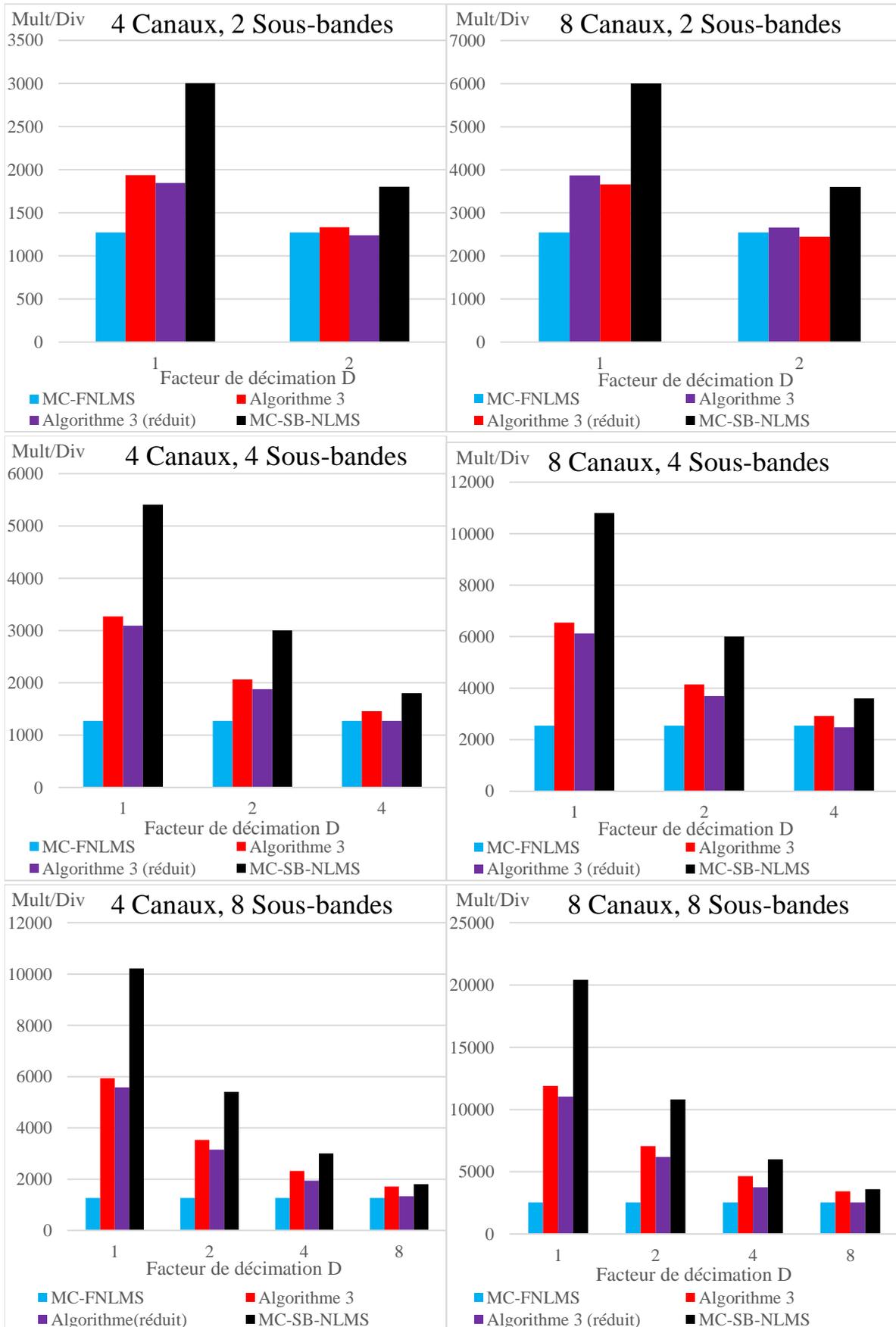


Figure 4.12. Comparaison de la complexité de calcul de l'algorithme 3.

## B) Résultats obtenus par l'algorithme MC-SB-FNLMS en boucle ouverte

(Algorithme 4) :

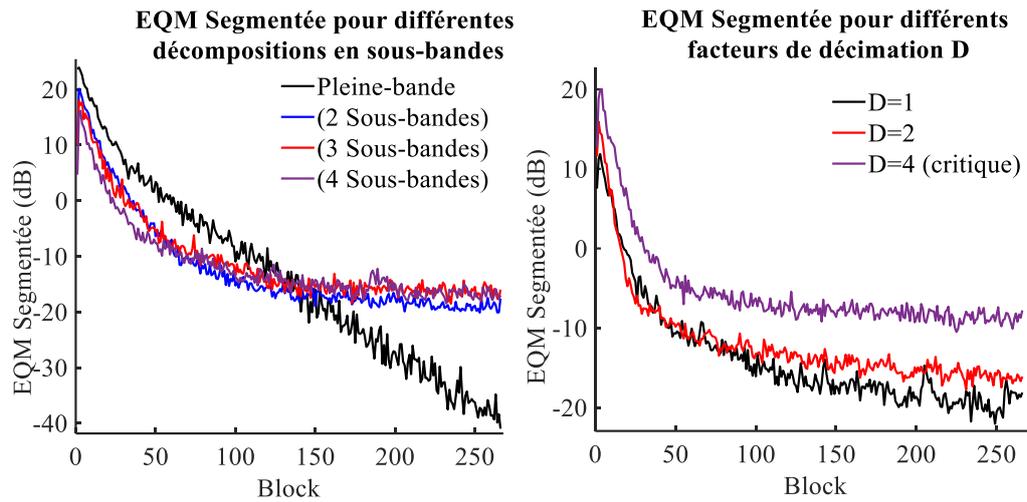
- *EQM Seg*

Pour évaluer l'EQM Seg (avec 100 échantillons par segment) de l'algorithme 4 proposé (c-à-d MC-SB-FNLMS réalisé en boucle ouverte), nous avons répété les mêmes expériences décrites dans la section A. Les résultats obtenus sont rapportés dans les **Figure 4.13** et **Figure 4.14**.

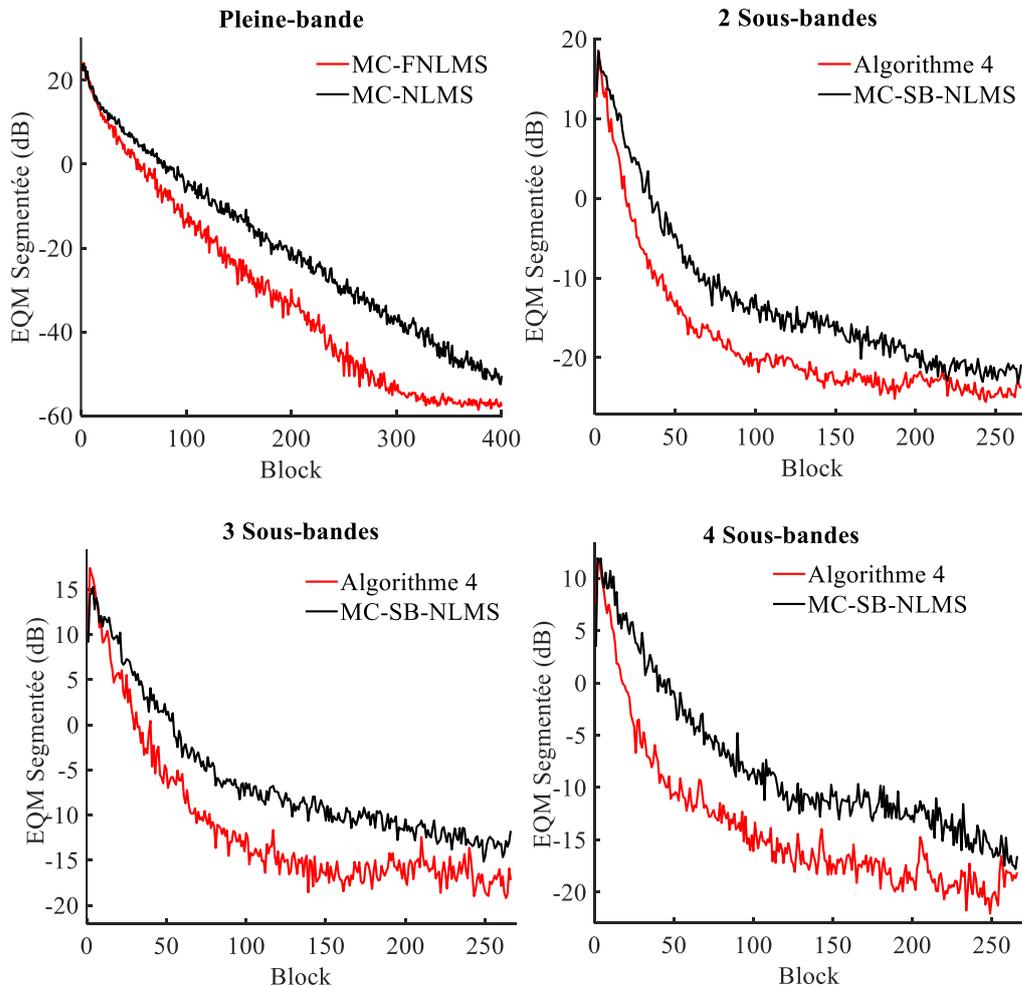
Dans la **Figure 4.13** (à gauche), l'algorithme 4 proposé est comparé à sa version en plein-bande (c-à-d algorithme 2). D'après cette figure, l'algorithme proposé présente un taux de convergence légèrement meilleur dans le régime transitoire, cependant l'erreur quadratique moyenne (EQM) en régime permanent est un peu importante. Ceci est dû aux chevauchements des sous-bandes fréquentielles et à la non-complémentarité de la décomposition en sous-bandes pour identifier les réponses impulsionnelles en bande pleine. Cependant, ce problème, n'est pas critique lorsque l'EQM est théoriquement élevé. De plus, la complexité de calcul de l'algorithme proposé peut être considérablement inférieure à celle de la version en plein-bande (c-à-d algorithme 2). Par conséquent, ce coût en complexité de calcul peut être considéré comme une compensation pour la dégradation des performances de l'EQM en régime permanent.

Nous traçons dans la **Figure 4.13** (à droite) la courbe de l'erreur quadratique moyenne segmentée (*EQM Seg*) pour différents facteurs de décimation. Comme le montre cette figure, le taux de convergence de l'algorithme n'est pas affecté par le facteur de décimation  $D$  en comparaison avec la réalisation multi-bandes, étant donné que la décimation dans cet algorithme réduit la taille des filtres adaptatifs ( $M_s = M/D$ ), cependant une valeur élevée de  $D$  entraîne une EQM élevée dans le régime permanent.

Pour différentes décompositions en sous-bandes, nous comparons les performances de l'algorithme 4 proposé et de l'algorithme MC-SB-NLMS en termes de l'erreur quadratique moyenne segmentée (*EQM Seg*) (facteur de décimation  $D = 1$ ). Les résultats obtenus sont rapportés dans la **Figure 4.14**. Les résultats de *EQM Seg*, nous montre la supériorité de l'algorithme 4 proposé en termes de taux de convergence par rapport à l'algorithme MC-SB-NLMS, en particulier pour une décomposition en 4 sous-bandes.



**Figure 4.13.** *EQM Seg* (100 échantillons par segment) obtenue par l’algorithme 4, à gauche : *EQM Seg* différentes décompositions en sous-bandes avec  $D = 1$ , à droite : 4 sous-bandes avec différentes valeurs de  $D$ .



**Figure 4.14.** Comparaison de l’*EQM Seg* (100 échantillons par segment) obtenue par l’algorithme 4 et l’algorithme MC-SB-NLMS (multi-bandes) pour différentes décompositions en sous-bandes.

- **Complexité de calcul (CC)**

Comme dans la partie A), nous présentons à la

**Figure 4.15** une étude comparative de la complexité de calcul (CC) de chaque algorithme, exprimée en termes du nombre de multiplications et de divisions par itération, examinée pour différentes valeurs de  $L$ ,  $N_B$  et  $D$ . Nous pouvons voir d'après la

**Figure 4.15**, que les algorithmes proposés (versions non réduites et réduites) nécessitent moins d'opérations arithmétiques par itération. Pour  $L=8$ ,  $N_B = 8$  et  $D = 4$ , l'algorithme proposé a besoin seulement 1488 Mult/Div par itération (version non réduite) et 1264 Mult/Div par itération (version réduite). D'autre part, l'algorithme MC-FNLMS et l'algorithme MC-SB-NLMS nécessitent 2544 et 1804 Mult/Div par itération. On peut conclure à partir de ces résultats, la supériorité de l'algorithme 4 proposé par rapport à l'algorithme classique MC-SB-NLMS, et sa version en plein-bande (c-à-d Algorithme 2), tout en ayant une vitesse de convergence très acceptable. En comparant l'algorithme 4 avec celui de l'algorithme 3, l'algorithme 4 est un bon choix pour atteindre une complexité de calcul optimale avec un bon taux de convergence dans le cas où la valeur de l'EQM au régime permanent n'est pas le critère principal visé.

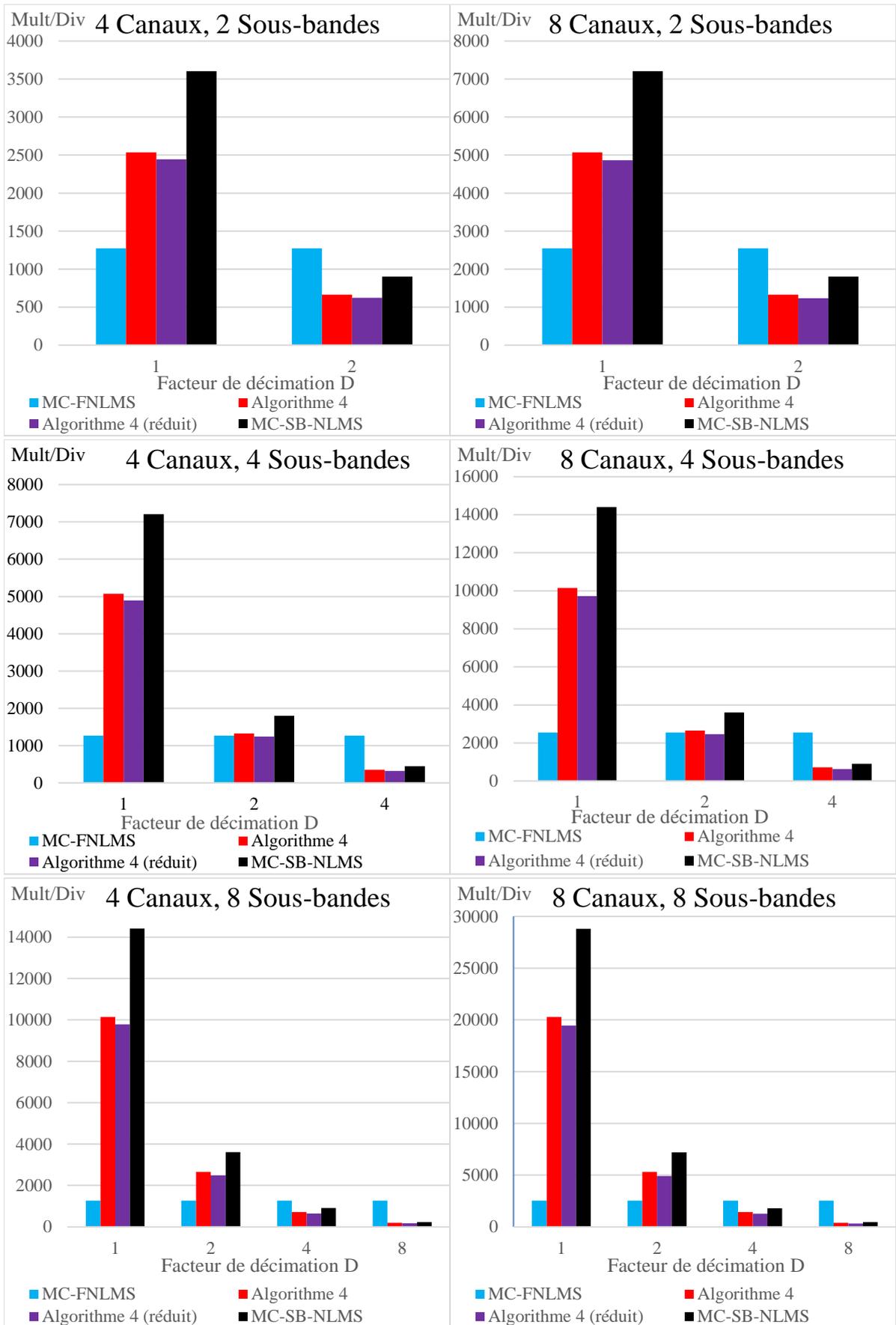
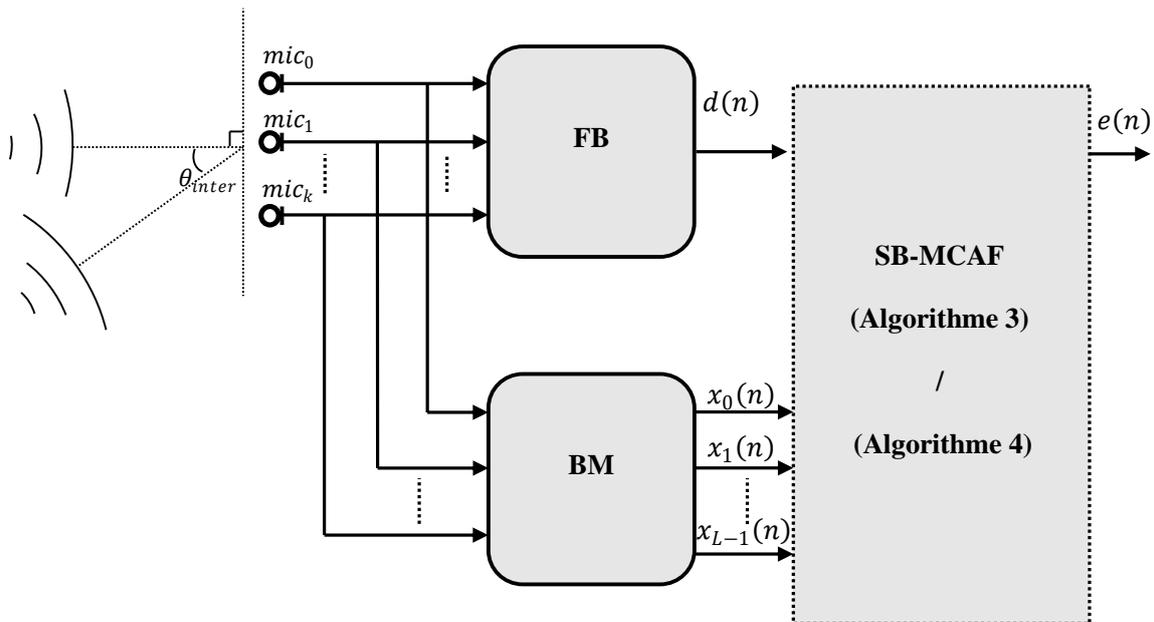


Figure 4.15. Comparaison de la complexité de calcul de l'algorithme 4.

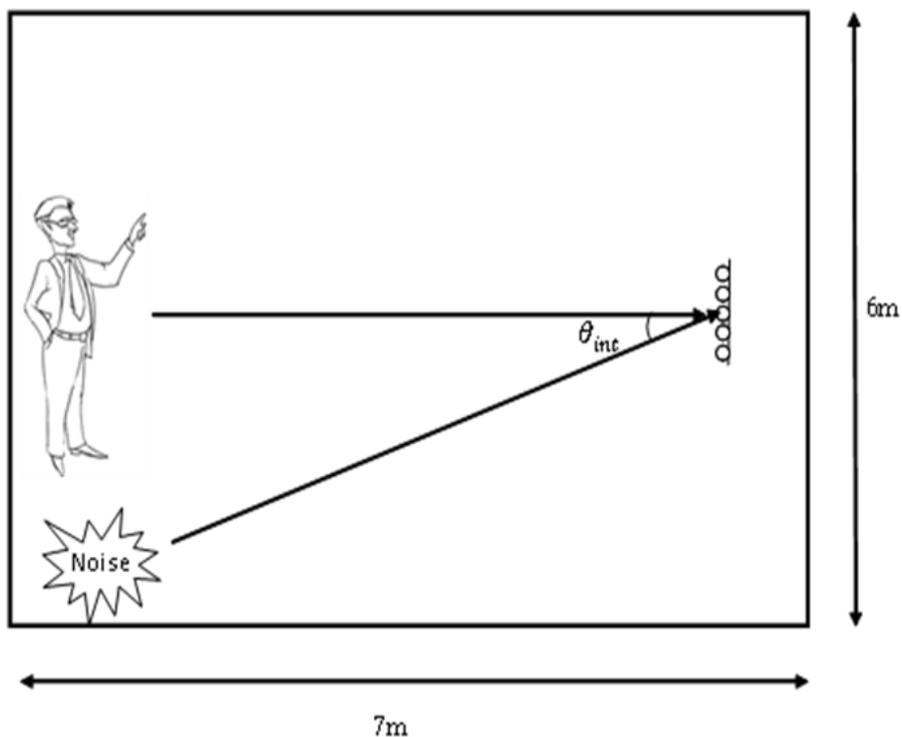
#### 4.4.2. Application au Beamforming adaptatif GSC

Dans cette section, on considère un formateur adaptatif GSC (Generalized Sidelobe Canceller) basé sur les deux algorithmes proposés (c'est-à-dire algorithme 3 et algorithme 4), comme illustré dans **Figure 4.16**. Nous considérons une salle rectangulaire dans laquelle nous plaçons 5 microphones ( $N_{mic} = 5$ ) espacés les uns des autres d'une distance  $d = 3 \text{ cm}$  (**Figure 4.17**). La fréquence d'échantillonnage est égale à 8 kHz avec laquelle, la condition d'échantillonnage spatial de Nyquist-Shannon est largement satisfaisante. La longueur  $M$  de chaque filtre adaptatif en pleine-bande est 100. La partie de Beamformer fixe (**FB**) du GSC est définie par la matrice :  $\mathbf{FB}^T = \frac{1}{5}[1 \ 1 \ 1 \ 1 \ 1]$ .

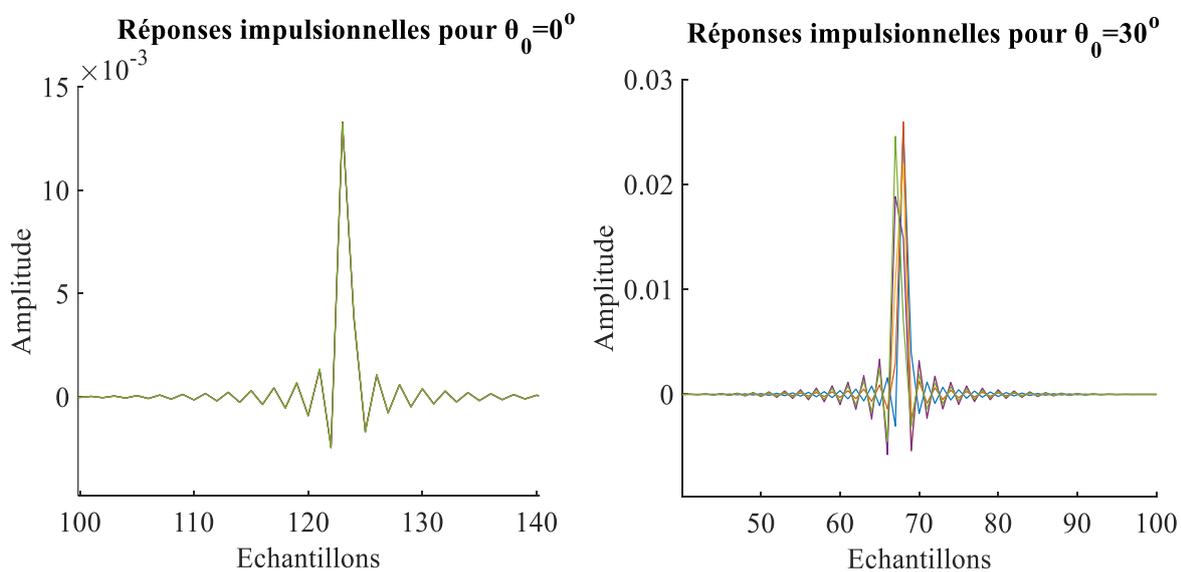
Nous considérons deux signaux issus de deux sources acoustiques différentes. Le premier signal est le signal USASI (United States of America Standard Institute, maintenant appelé ANSI). Ce signal représente le signal informatif avec un angle  $\theta_0 = 0^\circ$  par rapport à la normale. Le deuxième signal simule un signal d'interférence à rejeter par le GSC. Il s'agit du même signal autorégressif décrit précédemment (partie A). Ce signal provient de l'angle  $\theta_{inter} = -30^\circ$  par rapport à la normale. Pour générer les réponses impulsionnelles acoustique entre la source et chaque microphone, nous avons utilisé la méthode d'image décrite dans la référence [50] sans tenir en compte les réverbérations, ce qui signifie que les ondes acoustiques sont totalement absorbées par les murs. Les réponses impulsionnelles acoustiques sont représentées sur la **Figure 4.18**. La vitesse du son est 340 m/s.



**Figure 4.16.** GSC basé sur les algorithmes proposés.



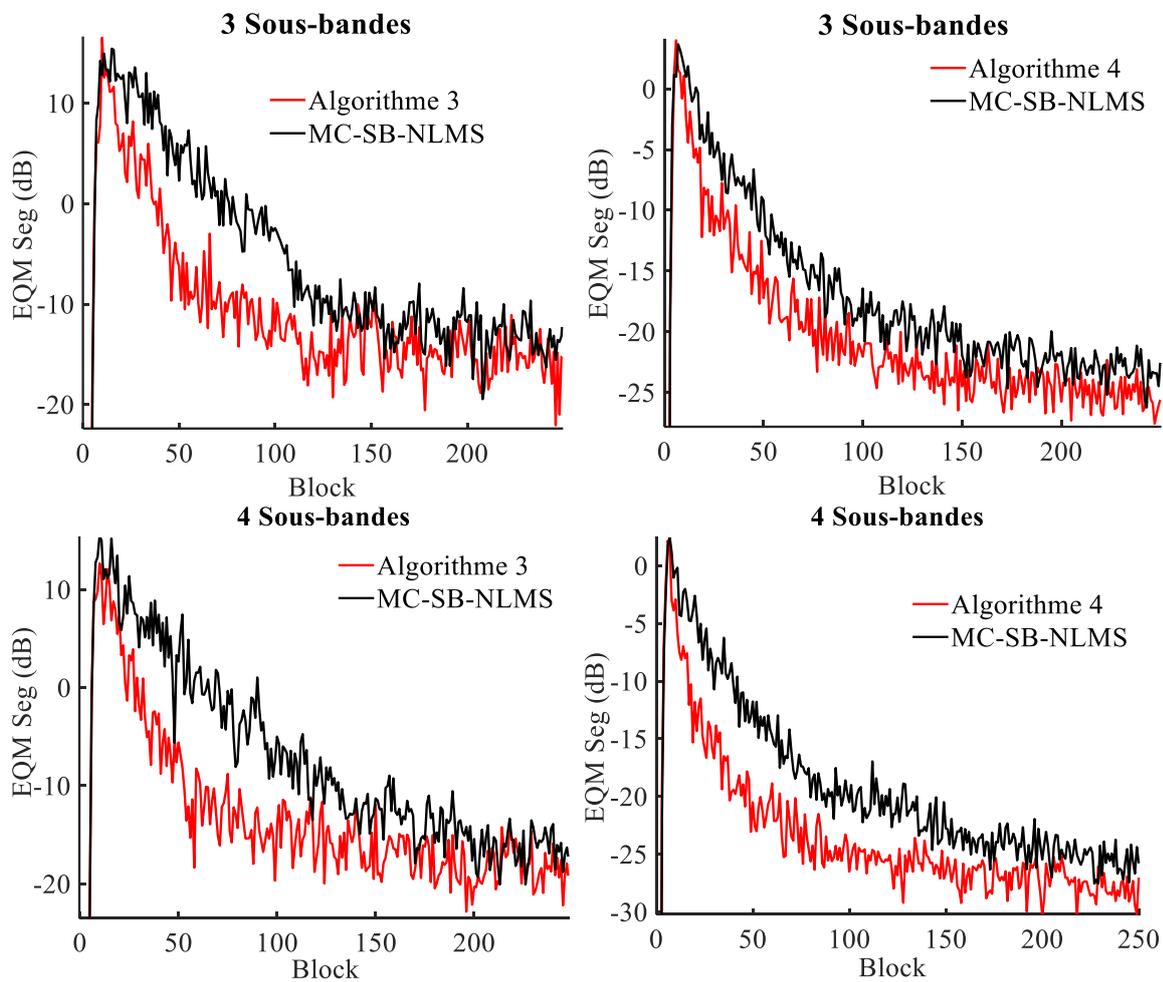
**Figure 4.17.** Dimensions de la salle.



**Figure 4.18.** Réponses impulsionnelles acoustiques entre les sources et les microphones.

Pour examiner la courbe de l'erreur quadratique moyenne segmentée (*EQM Seg*), avec un segment de bloc de 25 échantillons, nous traçons à la **Figure 4.19** les courbes de l'*EQM Seg* pour chaque algorithme en absence du signal d'intérêt (seul le bruit provient de l'angle  $\theta_{inter}$ ). Ces courbes sont le résultat de 10 tests indépendants avec les mêmes conditions de simulation.

En comparant les algorithmes proposés avec l'algorithme MC-SB-NLMS, nous constatons clairement l'amélioration de la vitesse de convergence, en particulier avec l'algorithme 3 (réalisation multi-bandes), cela traduit la capacité du GSC à rejeter les interférences et de bruit.



**Figure 4.19.** Comparaison de l'*EQM Seg* obtenue à la sortie du GSC pour les algorithmes proposés et l'algorithme MC-SB-NLMS.

D'après les expériences de la partie 4.4.1 et la partie 4.4.2, nous avons confirmé les bonnes performances et la supériorité des algorithmes MC-SB-FNLMS proposés (réalisations à bandes multiples et en boucle ouverte) par rapport à sa version en pleine-bande (c-à-d algorithme 2) et par rapport à l'algorithme MC-SB-NLMS, dans le contexte de l'identification multi-canaux des réponses impulsionnelles et pour la réduction spatiaux-fréquentielle en utilisant le GSC.

## 4.5. Conclusion

Dans ce chapitre, nous avons présenté deux algorithmes multi-canaux rapides et à faible complexités. Les algorithmes proposés reposent sur une implémentation en sous-bandes de l'algorithme 2 (proposé au chapitre 3), Le premier algorithme en sous-bandes proposé dans ce chapitre est fondé sur une réalisation multi-bandes, dénoté sous le nom "algorithme 3". Le deuxième algorithme dénoté par "algorithme 4", repose sur une réalisation sous-bande en boucle ouverte. L'avantage de l'approche en sous-bande est d'avoir une version de l'algorithme 2 (MC-FNLMS) à convergence rapide, tout en réduisant la complexité de calcul.

Le premier algorithme basé sur la réalisation multi-bandes, présente l'avantage d'un taux de convergence élevé. Cependant, la complexité de calcul de cette réalisation est plus élevée pour un faible facteur décimation. Le deuxième algorithme repose sur la réalisation en boucle ouverte. Contrairement à la réalisation multi-bandes, cette réalisation est caractérisée par sa faible complexité de calcul. Néanmoins, cette réalisation a pour effet d'augmenter l'erreur quadratique moyenne en régime permanent.

L'étude comparative entre les deux réalisations de l'algorithme proposé et les algorithmes MC-FNLMS (algorithme 2) ainsi que MC-NLMS, basée sur les courbes de l'erreur quadratique moyenne segmentée et la complexité de calcul, met en évidence les performances des deux algorithmes par rapport aux autres algorithmes. De plus, la simulation du GSC basé sur les algorithmes proposés démontre la capacité de ces algorithmes de réduire les interférences et le bruit acoustique. En conclusion, les deux algorithmes en sous-bandes proposés se profilent comme des choix d'algorithmes prometteurs pour le filtrage adaptatif multi-canaux.

## CONCLUSION GENERALE ET PERSPECTIVES

L'étude que nous avons présentée dans cette thèse concerne le problème de la réduction de bruit acoustique et le rehaussement du signal de parole en développant de nouveaux algorithmes de filtrage adaptatif multi-canaux.

La première contribution de cette thèse est présentée dans le deuxième chapitre qui a été consacré au développement d'un algorithme bi-capteurs pour la séparation aveugle des sources (SAS). L'algorithme proposé est basé sur la mise à jour des coefficients des filtres transversaux  $\mathbf{a}(n)$  et  $\mathbf{w}(n)$  qui ne sont pas affectés par l'élimination des prédicteurs arrière dans l'algorithme TC-SFTF, ce qui permet de garder la même vitesse de convergence par rapport à la version originale de TC-SFTF avec un bénéfice dans la complexité des calculs. Les performances de l'algorithme proposé ont été évaluées à l'aide de mesures objectives, démontrant son efficacité significative dans divers tests, en termes de convergence, rapport signal-sur-bruit, erreur quadratique moyenne et complexité de calcul. Ces résultats de tests démontrent la capacité de l'algorithme proposé à réduire les composantes du bruit avec une faible complexité de calcul.

La seconde contribution de cette thèse est donnée dans le chapitre 3, dans lequel nous avons proposé un nouvel algorithme adaptatif multi-canaux (MC-FNLMS) rapide et à faible complexité pour la réduction de bruit. L'algorithme proposé est une version multi-canaux de l'algorithme FNLMS ; il introduit un processus de prédiction du premier ordre pour décorrélérer les signaux afin d'augmenter le taux de convergence tout en maintenant une faible complexité de calcul. Une version moins complexe de l'algorithme proposé est aussi développée. Une étude détaillée de la stabilité de l'algorithme proposé est aussi donnée. Dans la simulation, l'algorithme proposé a été testé et comparé avec les algorithmes MC-NLMS et MC-APA. Les résultats de simulation ont démontré la supériorité de l'algorithme proposé par rapport aux autres algorithmes.

Dans le dernier travail présenté dans cette thèse, nous avons proposé dans le chapitre 4 deux algorithmes multi-canaux en sous-bandes dérivés de l'algorithme développé dans le chapitre 3. Le premier algorithme (multi-bandes MC-SB-FNLMS) est basé sur une réalisation multi-bandes de filtrage adaptatif en sous-bandes. Cet algorithme possède l'avantage de sa vitesse de convergence par rapport à la version en pleine bande (MC-FNLMS). Le second algorithme (Open-loop MC-SB-FNLMS) est basé sur la réalisation en boucle ouverte de filtrage adaptatif en sous-bandes. Cette réalisation a permis de réduire la complexité de calcul tout en gardant la

même vitesse de convergence. Nous avons également dérivé, pour chacun de ces deux algorithmes proposés, une version moins complexe, en estimant les paramètres de prédiction sur un seul canal. Les algorithmes en sous-bandes proposés sont testés dans deux problèmes ; le problème de l'identification des réponses acoustiques et le problème de la réduction de bruit spatio-temporelle par le système GSC. Tous les résultats de simulation ont montré la supériorité des algorithmes proposés en comparaison avec la version pleine bande et l'algorithme classique MC-NLMS.

Comme perspective de ce travail, nous envisagerons :

- L'implémentation pratique des algorithmes proposés sur des cartes DSP ou FPGA.

## Liste des notations, symboles et des abréviations

### Liste des notations et des symboles

$x(n)$	: Signal d'entrée de filtre adaptatif
$y(n)$	: Signal de sortie de filtre adaptatif
$d(n)$	: Signal désiré de filtre adaptatif
$e(n)$	: Signal d'erreur de filtre adaptatif
$s(n)$	: Signal de parole
$b(n)$	: Signal de bruit
$m(n)$	: Signal de mélange
$\delta(n)$	: Impulsion de Kronecker
$h(n)$	: Réponse impulsionnelle du canal acoustique
$v(n)$	: Erreur de prédiction du modèle autorégressif
$\bar{e}(n)$	: Estimation de l'erreur de prédiction avant l'algorithme FTF et l'algorithme SFTF
$\bar{r}(n)$	: Estimation de l'erreur de prédiction arrière de l'algorithme FTF
$\alpha(n)$	: Estimation de la variance de l'erreur de prédiction avant l'algorithme FTF et l'algorithme SFTF
$\beta(n)$	: Estimation de la variance de l'erreur de prédiction arrière de l'algorithme FTF
$\varepsilon(n)$	: Estimation de l'erreur de prédiction avant l'algorithme FNLMS
$\mathbf{x}(n)$	: Vecteur des derniers échantillons du signal d'entrée de filtre adaptatif
$\mathbf{X}(n)$	: Vecteur de taille $ML$ , contenant $L$ vecteurs $\mathbf{x}_i$ de taille $M$ pour chacun
$\mathbf{w}(n)$	: Vecteur de filtre adaptatif à l'instant $n$
$w_i$	: Poids de filtre adaptatif
$\mathbf{k}(n)$	: Gain de Kalman
$\mathbf{a}(n)$	: Estimation du vecteur de prédiction avant
$\mathbf{b}(n)$	: Estimation du vecteur de prédiction arrière
$\tilde{\mathbf{c}}_{M+1}(n)$	: Gain de Kalman dual d'ordre $M + 1$
$\tilde{\mathbf{c}}_M(n)$	: Gain de Kalman dual d'ordre $M$
$\gamma(n)$	: Facteur de conversion
$e_o$	: Erreur optimale
$y_o$	: Estimation optimale du signal désiré
$\mu_{LMS}$	: Pas d'adaptation de l'algorithme LMS
$\mu_{NLMS}$	: Pas d'adaptation de l'algorithme NLMS
$\mu_{FNLMS}$	: Pas d'adaptation de l'algorithme FNLMS
$\mu_{MCNLMS}$	: Pas d'adaptation de l'algorithme MC-NLMS
$\mu_{MCFNLMS}$	: Pas d'adaptation de l'algorithme MC-FNLMS
$\mu_{MCAPA}$	: Pas d'adaptation de l'algorithme MC-APA

$\lambda$	: Facteur d'oubli
$\delta_{NLMS}$	: Petite constante empêchant la division par zéro de l'algorithme NLMS
$c$	: Petite constante empêchant la division par zéro
$\eta$	: Facteur d'oubli de mise à jour du prédicteur avant
$\mathbf{R}$	: Matrice d'autocorrélation
$\mathbf{p}$	: Matrice d'intercorrélation
$\tilde{\mathbf{R}}(n)$	: Estimation de la matrice d'autocorrélation
$\tilde{\mathbf{r}}(n)$	: Estimation de la matrice d'intercorrélation
$J$	: Fonction coût
$\hat{J}_{min}$	: Valeur minimale de la fonction coût
$\lambda_{min}$	: Plus petite valeur propre de la matrice d'autocorrélation
$\lambda_{max}$	: Plus grande valeur propre de la matrice d'autocorrélation
$\sigma_x^2$	: Variance de $x(n)$
$\sigma_d^2$	: Variance de $d(n)$
$tr[.]$	: Trace d'une matrice
$[.]^T$	: Transposée d'une matrice
$\ .\ $	: Norme euclidienne d'un vecteur
$\sum$	: Somme
$M$	: Taille du filtre adaptatif
$M_S$	: Taille du filtre adaptatif en sous-bande
$N_B$	: Nombre de sous-bandes
$L$	: Nombre de filtres adaptatifs
$P$	: Dimension de l'algorithme APA
$N_{mic}$	: Nombre de microphones
$D$	: Indice de décimation
$I$	: Indice d'interpolation
$O(M^3)$	: Complexité proportionnelle à $M^3$
$O(M^2)$	: Complexité proportionnelle à $M^2$
$fs$	: Fréquence d'échantillonnage

### Liste des abréviations

AF	: Adaptive Filter- Analysis Filter
ANC	: Adaptive Noise Canceller
ANR	: Adaptive Noise Reduction
APA	: Affine Projection Algorithm
AR	: Autoregressive
BBSS	: Backward Blind Source Separation
BM	: Blocking Matrix
BFR	: Backward to Forward predictor Ratio

BSS	: Blind Source Separation
CC	: Complexité de Calcul (Computational Complexity)
D	: Decimator
DSA	: Densité Spectrale d'Amplitude (Amplitude Spectral Density)
DSP	: Densité Spectrale de Puissance (Power Spectral Density)
EQM	: Erreur Quadratique Moyenne (Mean Squared Error)
FB	: Fixed Beamformer
FBSS	: Forward Blind Source Separation
FNLMS	: Fast Normalized Least-Mean Square
GSC	: Generalized Sidelobe Canceler
Inter	: Interpolator
IR	: Impulse Response
LMS	: Least-Mean Square
MCAF	: Multi-Channel Adaptive Filter
MC-APA	: Multi-Channel Affine Projection Algorithm
MC-FNLMS	: Multi-Channel Fast Normalized Least-Mean Square Algorithm
MC-SB-FNLMS	: Multi-Channel Subband Fast Normalized Least-Mean Square
MSE	: Mean Square Error
NLMS	: Normalized Least-Mean Square
NR	: Noise Reduction
RI	: Réponse Impulsionnelle (Impulse Response)
RIF	: Réponse Impulsionnelle Finie (Finite Impulse Response)
RLS	: Recursive Least Squares Algorithm
RSB	: Rapport Signal à Bruit (Signal-to-Noise Ratio)
SAS	: Séparation Aveugle des Sources (Blind Source Separation)
SB	: Sous-Bande (Subband)
SF	: Synthesis Filter
SFTF	: Simplified Fast Transversal Filter
SM	: Système Mismatch
SNR	: Signal-to-Noise Ratio
SSA	: Soustraction Spectrale d'Amplitude (Spectral Amplitude Subtraction)
SSP	: Soustraction Spectrale de Puissance (Spectral Power Subtraction)
TC-NLMS	: Two-Channel Normalized least-mean square
TC-PU-SFTF	: Two-Channel Partial Update Simplified Fast Transversal Filter
TC-SFTF	: Two-Channel Simplified Fast Transversal Filter
USASI	: United States of America Standard Institute

## REFERENCES

- [1] I. Cohen, Y. Huang, J. Chen and J. Benesty, Noise Reduction in Speech Processing, vol. 2, S. T. i. S. Processing, Ed., Springer, 2009.
- [2] S. Boll, «Suppression of acoustic noise in speech using spectral subtraction,» *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 27, pp. 113-120, 1979.
- [3] B. Widrow, J. Glover, J. McCool, J. Kaunitz, C. Williams, R. Hearn, J. Zeidler, J. Eugen and R. Goodlin, "Adaptive noise cancelling: Principles and applications," *Proceedings of the IEEE*, vol. 63, no. 12, pp. 1692 - 1716, December 1975.
- [4] E. Weinstein, M. Feder and A. Oppenheim, "Multi-channel signal separation by decorrelation," *IEEE Transactions on Speech and Audio Processing*, vol. 1, no. 4, pp. 405 - 413, October 1993.
- [5] D. Yellin and E. Weinstein, "Multichannel signal separation: methods and analysis," *IEEE Transactions on Signal Processing*, vol. 44, no. 1, pp. 106 - 118, January 1996.
- [6] S. V. Gerven and D. V. Compernelle, "Signal separation by symmetric adaptive decorrelation: stability, convergence, and uniqueness," *IEEE Transactions on Signal Processing*, vol. 43, no. 7, pp. 1602 - 1612, July 1995.
- [7] H.-L. N. Th and C. Jutten, "Blind source separation for convolutive mixtures," *Signal Processing*, vol. 45, no. 2, pp. 209-229, August 1995.
- [8] M. Djendi, A. Gilloire and P. Scalart, "Noise Cancellation using Two Closely Spaced Microphones: Experimental Study with a Specific Model and Two Adaptive Algorithms," in *IEEE. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Toulouse, France, 2006.
- [9] M. Djendi and P. Scalart, "Reducing over- and under-estimation of the a priori SNR in speech enhancement techniques," *Digital Signal Processing*, vol. 32, pp. 124-136, September 2014.
- [10] A. Cheffi, M. Djendi and A. Guessoum, "New efficient two channel forward set-membership partial-update NLMS algorithms for blind speech enhancement and acoustic noise reduction," *Applied Acoustics, Elsevier*, vol. 141, pp. 322-332, December 2018.
- [11] M. Djendi, "An efficient frequency-domain adaptive forward BSS algorithm for acoustic noise reduction and speech quality enhancement," *Computers & Electrical Engineering*, vol. 52, pp. 12-27, May 2016.

- [12] K. Ghribi, M. Djendi and D. Berkani, "A wavelet-based forward BSS algorithm for acoustic noise reduction and speech enhancement," *Applied Acoustics*, vol. 105, pp. 55-66, April 2016.
- [13] A. Sayed and T. Kailath, "A state-space approach to adaptive RLS filtering," *IEEE Signal Processing Magazine*, vol. 11, no. 3, pp. 18 - 60, July 1994.
- [14] B. Hassibi, T. Kailath and A. Sayed, "Array algorithms for H/sup /spl infin// estimation," *IEEE Transactions on Automatic Control*, vol. 45, no. 4, pp. 702 - 706, April 2000.
- [15] O. Frost, "An algorithm for linearly constrained adaptive array processing," *Proceedings of the IEEE*, vol. 60, no. 8, pp. 926 - 935, August 1972.
- [16] Haykin.S, Adaptive Filter Theory, 5th ed., Prentice-Hall Information and System Sciences Series, 2008.
- [17] A. Sayed, Fundamentals of Adaptive Filtering, New York: Wiley, 13 juin 2003.
- [18] M. Zerouali and M. Djendi, "An Efficient Two-Channel Partial Update Simplified Fast Transversal Filter (PUSMFTF) algorithm for Acoustic Noise Reduction and Speech Enhancement," in *1st National Conference on Applied Science and Advanced Materials.NCASAM-2021*, Skikda, Algeria, 2021.
- [19] R. Henni, M. Djendi and M. Djebari, "A new efficient two-channel fast transversal adaptive filtering algorithm for blind speech enhancement and acoustic noise reduction," *Computers & Electrical Engineering*, vol. 73, pp. 349-368, January 2019.
- [20] M. Zerouali and M. Djendi, "A New Efficient Multi-Channel Fast NLMS (MC-FNLMS) Adaptive Algorithm for Audio Teleconferencing systems," *Physical Communication*, vol. 62, 2024.
- [21] A. Benallal and M. Arezki, "A fast convergence normalized least-mean-square type algorithm for adaptive filtering," *International Journal of Adaptive Control and Signal Processing*, vol. 28, no. 10, pp. 1073-1080, October 2014.
- [22] M. Messini and M. Djendi, "A new adaptive filtering algorithm for stereophonic acoustic echo cancellation," *Applied Acoustics, Elsevier*, vol. 146, pp. 345-354, March 2019.
- [23] N. Wiener, Extrapolation, Interpolation, and Smoothing of Stationary Time Series: With Engineering Applications, The MIT Press, 1949.
- [24] M. François and B. Maurice, Filtrage adaptatif, théorie et algorithmes, Lavoisier, 09-2005.

- [25] M. Djendi, *Thèse de doctorat en Traitement du signal et télécommunications : Méthodes améliorées de débruitage bi-capteurs dans un contexte automobile*, École doctorale Mathématiques, télécommunications, informatique, signal, systèmes, électronique (Rennes), 2010.
- [26] S. Werner, M. d. Campos and P. Diniz, "Partial-update NLMS algorithms with data-selective updating," *IEEE Transactions on Signal Processing*, vol. 52, pp. 938 - 949, April 2004.
- [27] K. Dogancay and O. Tanrikulu, "Adaptive filtering algorithms with selective partial updates," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 48, pp. 762 - 769, 8, August 2001.
- [28] S. Douglas, «Adaptive filters employing partial updates,» *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 44, pp. 209 - 216, March 1997.
- [29] M. S. E. Abadi and J. H. Husøy, "Mean-square performance of the family of adaptive filters with selective partial updates," *Signal Processing*, vol. 88, pp. 2008-2018, August 2008.
- [30] J. Cioffi and T. Kailath, "Fast recursive least squares transversal," *IEEE Trans. Acoust. Speech*, vol. 32, pp. 304 - 337, April 1984.
- [31] G. Carayannis, D. Manolakis and N. Kalouptsidis, "A fast sequential algorithm for least-squares filtering and prediction," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 31, pp. 1394 - 1402, December 1983.
- [32] A. Benallal and A. Benkrid, "A simplified FTF-type algorithm for adaptive filtering," *Signal processing, Elsevier*, pp. 904-917, 2007.
- [33] R. HENNI, "Thèse de doctorat en Télécommunications: Méthodes améliorée pour la réduction du bruit et echo acoustique dans les systèmes de communications à main - libres.," Blida, Algérie, 2021.
- [34] P. Regalia, "Numerical stability issues in fast least-squares," *Opt. Eng.*, vol. 31, p. 1144–1152, June 1992.
- [35] Y. Hu and P. C. Loizou, "Evaluation of objective quality measures for speech enhancement," *Transactions on Audio, Speech and Language Processing*, vol. 16, pp. 229–238,, 2008.
- [36] E. Ferrara and B. Widrow, "Multichannel adaptive filtering for signal enhancement," *IEEE Transactions on Circuits and Systems*, vol. 28, pp. 606-610, 1981.

- [37] F. Lindstrom, C. Schüldt et I. Claesson, «Efficient multichannel NLMS implementation for acoustic echo cancellation,» *Journal on Audio, Speech, and Music Processing*, vol. 2007, 2006.
- [38] J. Benesty, P. Duhamel et Y. Grenier, «A multichannel affine projection algorithm with applications to multichannel acoustic echo cancellation,» *IEEE Signal Processing Letters*, vol. 3, pp. 35-37, 1996.
- [39] K. K. Das et J. K. Satapathy, «Frequency-Domain Block Filtered-x NLMS Algorithm for Multichannel ANC,» chez *2008 First International Conference on Emerging Trends in Engineering and Technology*, 2008.
- [40] B. W. E Ferrara, «Multichannel adaptive filtering for signal enhancement,» *IEEE Transactions on Circuits and Systems*, 1981.
- [41] S. Zhang, H. C. So, W. Mi and H. Han, "A family of adaptive decorrelation NLMS algorithms," *IEEE Transactions on Circuits and Systems I*, vol. 65, pp. 638 - 649, 21 August 2017.
- [42] M. Djendi and R. Bendoumia, "A new efficient two-channel backward algorithm for speech intelligibility enhancement: A subband approach," *Applied acoustics, Elsevier*, vol. 76, pp. 209-222, February 2014.
- [43] A. Sayoud, M. Djendi and A. Guessoum, "A new speech enhancement adaptive algorithm based on fullband–subband MSE switching," *International Journal of Speech Technology, Springer*, vol. 22, p. 993–1005, 2019.
- [44] R. BENDOUMIA, "Thèse de doctorat en Génie Electrique: Annulation du bruit par les méthodes de séparation de sources," Université de Blida, Algérie, 2014.
- [45] K. Lee et W. Gan, «Improving convergence of the NLMS algorithm using constrained subband updates,» *IEEE Signal Processing Letters*, vol. 11, pp. 736 - 739, September 2004.
- [46] K. Lee, W. Gan et S. Kuo, *Subband Adaptive Filtering Theory and Implementation*, Wiley, 2009.
- [47] A. Gilloire and M. Vetterli, "Adaptive filtering in sub-bands with critical sampling: analysis, experiments, and application to acoustic echo cancellation," *IEEE transactions on signal processing*, vol. 40, pp. 1862-1875, 1992.
- [48] N. J. Fliege, *Multirate Digital Signal Processing*, New York: John Wiley & Sons, 1994.
- [49] F. Yang, M. Wu, P. Ji et J. Yang, «An improved multiband-structured subband adaptive filter algorithm,» *IEEE Signal Processing Letters*, vol. 19, pp. 647 - 650, 2012.

- [50] J. Allen et D. Berkley, «Image method for efficiently simulating small-room acoustics,» *The Journal of the Acoustical Society of America*, vol. 65, p. 943, 1979.
- [51] M. Zoulikha et M. Djendi, «A new regularized forward blind source separation algorithm for automatic speech quality enhancement,» *Applied Acoustics*, vol. 112, pp. 192-200, November 2016.
- [52] M. Akil, *Thèse de Doctorat : Séparation de sources convolutives*, Université Joseph Fourier – Grenoble 1, 2008.
- [53] A. Bessekri, M. Djendi and A. Guessoum, "A new simplified fast transversal filter algorithm based on subband approach (SSFTF) for acoustic echo cancellation," *Applied Acoustics, Elsevier*, vol. Volume 161, no. 107178, April 2020.
- [54] A. BOUNIF, *Thèse de DOCTORAT en Electronique: Etude de l'annulation d'echo acoustique stéréophonique par les algorithmes de filtrage adaptatif transverse*, Université de Blida, Algérie, 2014.
- [55] B. Breed and J. Strauss, "A short proof of the equivalence of LCMV and GSC beamforming," in *IEEE Signal Processing Letters*, 2002.
- [56] H. Buchner, J. Benesty and W. Kellermann, "Generalized multichannel frequency-domain adaptive filtering: efficient realization and application to hands-free speech communication," *Signal Processing, Elsevier*, vol. 85, pp. 549-570, March 2005.
- [57] K. Buckley and L. Griffiths, "An adaptive Generalized Sidelobe Canceller with derivative constraints," *IEEE Transactions on Antennas Propagat*, vol. 34, pp. 311 - 319, 1986.
- [58] C. Plapous, *Thèse de Doctorat: Traitements pour la réduction de bruit. Application à la communication parlée*, Université de Rennes, France, 2005.
- [59] S. Cecchi, A. Terenzi, V. Bruschi, A. Carini and S. Orcioni, "A subband implementation of a multichannel and multiple position adaptive room response equalizer," *Applied Acoustics, Elsevier*, vol. 173, no. 107702, February 2021.
- [60] H. Chen, Q. Wan, X. Zhang and R. Fan, "Robust beamforming with inter-atom-interference mitigation approach for uniform circular arrays," *AEU - International Journal of Electronics and Communications*, Vols. 69, Issue 1, pp. 236-241, January 2015.
- [61] M. Djendi, P. Scalart and A. Gilloire, "Analysis of two-sensors forward BSS structure with post-filters in the presence of coherent and incoherent noise," *Speech Communication- Elsevier*, vol. 55, pp. 975-987, November–December 2013.

- [62] K. Doğançay, *Partial-Update Adaptive Signal Processing: Design Analysis and Implementation*, Academic Press, Elsevier, 17 sept. 2008.
- [63] S. Douglas, «A family of normalized LMS algorithms.,» *IEEE Signal Processing Letters*, vol. 1, pp. 49 - 51, 1994.
- [64] C. Fox, *Thèse de Doctorat: Débruitage multicapteur appliqué à la téléphonie mains-libres en automobile*, École de l'institut Mines-Télécom, France., 2013.
- [65] L. Griffiths and C. Jim, "An alternative approach to linearly constrained adaptive beamforming," *IEEE Transaction on Antennas Propagat*, vol. 30, pp. 27-34, January 1982.
- [66] K. Han, H. Qian, S. Hu and K. Kang, "Performance analysis of hybrid beamforming systems with analog mismatches," *Physical Communication*, vol. 53, no. 101770, 2022.
- [67] E. Heydari, M. S. E. Abadi and S. M. Khademiyan, "Improved multiband structured subband adaptive filter algorithm with L0-norm regularization for sparse system identification," *Digital Signal Processing, Elsevier*, vol. 122, no. 103348, 15 April 2022.
- [68] Y. A. Huang and J. Benesty, "Adaptive multi-channel least mean square and Newton algorithms for blind channel identification," *Signal Processing, Elsevier*, vol. 82, pp. 1127-1138, August 2002.
- [69] K.-C. Huarng and C.-C. Yeh, "Performance analysis of derivative constraint adaptive arrays with pointing errors," *IEEE Transactions on Antennas and Propagation*, vol. 40, pp. 975 - 981, August 1992.
- [70] J. Li and M. Akagi, "A hybrid microphone array post-filter in a diffuse noise field," *Applied Acoustics, Elsevier*, vol. 69, pp. 546-557, June 2008.
- [71] Y. Li, Y. Wang and T. Jiang, "Sparse-aware set-membership NLMS algorithms and their application for sparse channel estimation and echo cancelation," *AEU - International Journal of Electronics and Communications*, Vols. 70, Issue 7, pp. 895-902, July 2016.
- [72] A. Maroosi and S. A. Tabatabaee, "A novel efficient time and frequency waveform design for filter bank multicarrier communication systems by using Hybrid gray wolf optimization algorithm," *Physical Communication*, vol. 49, no. 101468, 2021.
- [73] J. R. Mohammed and K. H. Sayidmarie, "Performance evaluation of the adaptive sidelobe canceller system with various auxiliary configurations," *AEU - International Journal of Electronics and Communications*, vol. 80, pp. 179-185, October 2017.

- [74] V. Patel and J. Cheer, "A hybrid multi-reference subband control strategy for active noise control headphones," *Applied Acoustics, Elsevier*, vol. 197, no. 103348, August 2022.
- [75] L. Qin, M. Wu and Z. Dong, "Robust adaptive beamforming using multi-snapshot direct data domain approach," *AEU - International Journal of Electronics and Communications*, vol. 75, pp. 124-129, May 2017.
- [76] C. R. Rao, M. R. Murthy and K. S. Rao, "Speech enhancement using sub-band cross-correlation compensated Wiener filter combined with harmonic regeneration," *AEU - International Journal of Electronics and Communications*, Vols. 66, Issue 6, pp. 459-464, June 2012.
- [77] B. Rivet, *Thèse de Doctorat: La bimodalité de la parole au secours de la séparation de sources*, Institut National Polytechnique de Grenoble, France, 2006.
- [78] A. Sayoud and M. Djendi, "Efficient subband fast adaptive algorithm based-backward blind source separation for speech intelligibility enhancement," *International Journal of Speech Technology*, vol. 23, p. 471–479, 2020.
- [79] L. C. Siyuan Song a b, S. Luan, D. Yao, J. Li et Y. Yan, «An integrated multi-channel approach for joint noise reduction and dereverberation,» *Applied Acoustics, Elsevier*, vol. 171, n° % 1107526, January 2021.
- [80] Y. Yu and H. Zhao, "Proportionate NSAF algorithms with sparseness-measured for acoustic echo cancellation," *AEU - International Journal of Electronics and Communications, Elsevier*, vol. 75, pp. 53-62, May 2017.
- [81] Z. Zecevic, B. Krstajic and M. Radulovic, "A new adaptive algorithm for improving the ANC system performance," *AEU - International Journal of Electronics and Communications*, Vols. 69, Issue 1, pp. 442-448, January 2015.
- [82] J. Benesty, J. Chen and Y. Huang, *Microphone Array Signal Processing*, Springer, 2008.
- [83] M. Djendi, «New efficient adaptive fast transversal filtering (FTF)-type algorithms for mono and stereophonic acoustic echo cancelation,» *International Journal of Adaptive Control and Signal Processing, Wiley Online Library.*, vol. 29, pp. 273-301, February 2014.
- [84] A. Sayoud, M. Djendi, S. Medahi et A. Guessoum., «A dual fast NLMS adaptive filtering algorithm for blind speech quality enhancement,» *Applied Acoustics*, vol. 135, pp. 101-110, 2018.
- [85] W. Liu et S. Weiss, *Wideband beamforming: concepts and techniques*, A John Wiley and Sons, Ltd., Publication, 2010.

- [86] R. Bendoumia and M. Djendi, "Variable step-size subband backward BSS algorithms for speech quality enhancement," *Applied Acoustics*, vol. 86, pp. 25-41, December 2014.
- [87] A. CHEFFI, «Thèse de doctorat: Proposition de nouvelles méthodes pour la réduction de bruit acoustique et la rehaussement du signal parole dans environnement très bruite,» Université de Blida 1, 2021.
- [88] A. SAYOUD, «Thèse de doctorat en Télécommunications, Nouvelle structures adaptatifs pour le rehaussement automatique du signal de parole,» Université de Blida, Algérie, 2021.
- [89] M. ZOULIKHA, «Thèse de doctorat en électronique: Méthode automatiques et robustes pour le rehaussement du signal de parole appliquées à la téléphonie mains-libres,» Université Blida, 2021.