

الجمهورية الجزائرية الديمقراطية الشعبية  
République Algérienne démocratique et populaire

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

جامعة سعد دحلب البليدة  
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا  
Faculté de Technologie

قسم الإلكترونيك  
Département d'Électronique



## Mémoire de Master

Filière Télécommunication  
Spécialité Réseaux et télécommunications

présenté par

OMARI Medina

&

ABDAT Soumaya

---

# Nouvelle version proportionnelle de l'algorithme Backward bi-capteurs pour la réduction du bruit acoustique

---

Proposé par : Mr BENDOUMIA Rédha

Année Universitaire 2018-2019

## *Remerciements*

---

*Nous tenons à remercier en premier lieu le Dieu Le tout puissant qui nous a accordé la volonté, la santé et le courage d'accomplir ce travail.*

*Nous adressons toute notre reconnaissance et nos plus vifs  
remerciements à notre promoteur*

*Mr. BENDOUMIA REDHA pour avoir accepté de nous encadrer, nous le remercions énormément pour ses précieux conseils qu'il nous a prodigué tout au long de ce projet, pour sa disponibilité sa patience et son soutien, et surtout pour ses qualités professionnelles et humaines.*

*Un grand remerciement aux membres du jury chacun par son propre nom pour l'honneur et l'intérêt qu'ils nous ont accordé en acceptant d'examiner et d'évaluer notre mémoire.*

*Nous exprimons également notre gratitude à tous les professeurs qui ont collaboré à notre formation depuis notre premier cycle d'étude jusqu'à notre cycle universitaire, les enseignants de la faculté d'électronique de manière générale et à tous ceux qui ont contribué de près ou de loin à l'élaboration de ce travail.*

*On ne saurait oublier de remercier nos chères parents, qui étaient toujours à nos cotés et nous a aidé et soutenu, Nous leurs souhaitons la bonne santé et la longue vie, Incha Allah*

*On vous dit tous, MERCI*

## *Dédicace*

---

*Je dédie ce travail : a DIEU le tout puissant de m'avoir donné le courage et la volonté pour accomplir ce modeste travail.*

*A Mon très cher Père Djamel Edine: Aucune dédicace ne saurait exprimer l'amour et le respect que j'ai toujours pour vous. Rien au monde ne vaut les efforts fournis jour et nuit pour mon éducation et mon bien être. Ce travail est le fruit de tes sacrifices que tu as consentis pour mon éducation le long de ces années.*

*A Ma chère Mère Samia : Tu représente pour moi la source de tendresse et l'exemple de dévouement qui n'a pas cessé de m'encourager. Tu as fait plus qu'une mère puisse faire pour que ses enfants suivent le bon chemin dans leur vie et leurs études.*

*A ma grand mère Qui m'a accompagné par ses prières, sa douceur, puisse Dieu lui prêter longue vie et beaucoup de santé et de bonheur.*

*A toute la famille ABDAT est spécialement Soumaya*

*A tous mes enseignants et spécialement mon encadreur*

*MR BENDOUMIA Rédha*

*A tous mes amis qui m'ont aidé et encouragé,*

*Medina*

## *Dédicace*

---

*Je dédie ce modeste travail aux étoiles de ma vie*

*Ma mère CHERIFA et mon père MOHAMMED*

*Mon mari YOUSSEF*

*À tous les membres de la famille ABDAT, MESSAOUDINE et  
MIRAOU*

*À toute la famille OMARI et spécialement MEDINA*

*À tous mes enseignants*

*Et à tous ceux qui m'aiment et qui croient en moi*

*Soumaya*

---

**ملخص:** الحد من الضوضاء الصوتية وتحسين إشارة الكلام هي المفتاح لتحسين الجودة المدركة للصوت ، وقد تم معالجة هذه المشكلات من خلال العديد من التقنيات ؛ من بين هذه التقنيات ، الفصل الأعمى لمصادر المستشعر الثنائي مع خوارزميات التصفية التكيفية ، مشكلة هذه الخوارزميات هو في حالة الاستجابات النبضية المتفرقة. لإيجاد حل فعال لهذه المشكلة ، سنقترح خوارزمية تكيفية متناسبة يتم تنفيذها على البنية الخلفية للفصل الأعمى للمصادر ثنائية الاستشعار. برهنت الخوارزمية المقترحة (IPBLNMS) القدرة على تخفيف الضوضاء، وتقديم أداء جيد.

**كلمات المفاتيح:** الحد من الضوضاء الصوتية. أجهزة استشعار ثنائية إلى الخلف. ردود الاندفاع المتفرق.

---

**Résumé :** La réduction du bruit acoustique et le rehaussement du signal de parole sont la clé pour améliorer la qualité perçue de la voix, ces problèmes ont été traités par plusieurs techniques ; parmi ces techniques la séparation aveugle des sources bi-capteurs combinées avec les algorithmes du filtrage adaptatif, l'inconvénient de ces algorithmes est dans le cas des réponses impulsionnelles sparses. Pour trouver une solution efficace a ce problème, on va proposer un algorithme adaptatif proportionnel implémenté sur la structure Backward de la séparation aveugle des sources bi-capteurs. L'algorithme proposé (IPBLNMS) a réussi à atténuer le bruit et donner des bonnes performances.

**Mots clés :** Réduction du Bruit Acoustique ; structure bi-capteurs Backward ; réponses impulsionnelles sparses.

---

**Abstract :** The acoustic noise reduction and the speech enhancement are the key to improving the perceived quality of the voice, these problems have been addressed by several techniques; Among these techniques the blind separation of bi-sensor sources combined with adaptive filtering algorithms, the disadvantage of these algorithms is in the case of sparsal impulse responses. To find an effective solution to this problem, we will propose a proportional adaptive algorithm implemented on the Backward structure of the blind separation of bi-sensor sources. The proposed algorithm (IPBLNMS) was able to attenuate the noise, and give good performances.

**Keywords :** Acoustic Noise Reduction; backward bi-sensors structure; sparse impulse responses.

---

## Liste des abréviations

- IP : Internet Protocol.
- EQM : Erreur Quadratique Moyenne.
- RIF : Réponse impulsionnelle finie.
- LMS : Least Mean Squares.
- NLMS : Normalized Least Mean Square.
- PNLMS : Proportionate Normalized Least Mean Square.
- IPNLMS : Improved PNLMS.
- USASI : USA Standard Institute.
- TFD : Transformée de Fourier discrète.
- DSP : Densité spectrale de puissance.
- SAS : Separation Aveugle de Sources.
- FLMS : Forward LMS.
- FNLMS : Forward NLMS.
- BLMS : Backward LMS.
- 2CBLMS : Two-Channel Backward LMS.
- BNLMS : Backward NLMS.
- SM : System Mismatch.
- SNR : Signal to Noise Ratio.
- SNRSeg : SNR segmental.
- PBNLMS : Proportionate Backward NLMS.
- IPBNLMS : Improved Proportionate Backwad NLMS.
- RI : Réponse Impulsionnelle.
- DAV : Détection d'Activité Vocale.
- dB : Decibel.
- Log : Logarithm.
- ms : Millisecond.

# Table des matières

Introduction générale .....	01
-----------------------------	----

## Chapitre 01 : Généralité sur le filtrage adaptatif.

1.1. Introduction.....	04
1.2. Filtre de Wiener .....	04
1.3. Filtrage adaptatif .....	06
1.3.1. Principe de filtrage adaptatif.....	07
1.3.2. Applications des filtres adaptatifs.....	07
1.4. Réponse impulsionnelle acoustique.....	09
1.5. Réponse impulsionnelle sparse.....	10
1.6. Algorithmes adaptatifs .....	11
1.6.1. Choix de l'algorithme.....	11
1.6.2. Algorithme du gradient stochastique LMS.....	12
1.6.3. Algorithme du gradient stochastique normalisé (NLMS).....	13
1.6.4. Algorithme Proportionné Normalisé LMS (PNLMS).....	14
1.6.5. Algorithme PNLMS amélioré (IPNLMS) .....	15
1.7. Complexité algorithmique.....	16
1.8. Conclusion .....	16

## Chapitre 02 : Séparation aveugle de source par des algorithmes adaptatifs bi-capteur.

2.1. Introduction.....	17
2.2. Généralités sur le bruit .....	17
2.2.1. Définitions du bruit.....	17
2.2.2. Quelques types du bruit .....	17
2.3. Réduction du bruit.....	18
2.3.1. Méthode mono-capteur .....	18
2.3.2. Méthode bi-capteur .....	20
2.4. Séparation aveugle de sources .....	21
2.4.1. Structure directe (Forward).....	22
2.4.2. Structure récursive (Backward) .....	24
2.5. L'algorithme LMS et NLMS avec la structure Direct (Forward) .....	25

2.5.1. L'algorithme FLMS.....	25
2.5.2. L'algorithme FNLMs .....	26
2.6. L'algorithme LMS et NLMS avec la structure Récursive (Backward).....	27
2.6.1. L'algorithme BLMS .....	27
2.6.2. L'algorithme BNLMs.....	28
2.7. Conclusion .....	29

### **Chapitre 03 : Algorithmes proposés Backward NLMS proportionnel bi-capteurs amélioré.**

3.1. Introduction.....	30
3.2. Problème de l'algorithme Backward NLMS classique.....	30
3.2.1. Critère du désajustement.....	30
3.2.2. Critère rapport signal sur bruit (SNR).....	31
3.3. Algorithme Backward proportionnel proposée .....	32
3.3.1. Structure générale des algorithmes Backward NLMS proportionnel.....	32
3.3.2 Algorithme proposé Backward NLMS proportionnel amélioré (IPBNLMs).....	33
3.3.3 Vérification de la règle optimale du nouvel algorithme IPBNLMs proposé.....	36
3.4. Complexité algorithmique.....	39
3.5. Conclusion .....	39

### **Chapitre 04 : Résultats et Simulations**

4.1. Introduction.....	40
4.2. Présentation des signaux de simulation.....	40
4.2.1. Signal de parole.....	40
4.2.2. Bruit USASI.....	41
4.2.3. Réponses impulsionnelles.....	42
4.3. Détection d'activité vocale (DAV).....	43
4.4. Résultats de l'algorithme BNLMs.....	44
4.5. Résultat de l'algorithme IPBNLMs.....	45
4.5.1. Effet du paramètre $\alpha$ sur l'algorithme IPBNLMs.....	46
4.5.2. Signaux de parole estimés.....	48
4.5.3 .Effet de la taille des filtres.....	49
4.5.4. Effet de pas d'adaptation.....	55



4.6. Conclusion.....	59
Conclusion générale.....	60
Bibliographie.....	62

## Liste des figures

<b>Figure 1.1.</b> Représentation schématique d'un filtre de Wiener .....	04
<b>Figure 1.2.</b> Principe d'un filtre adaptatif.....	07
<b>Figure 1.3.</b> Identification de systèmes.....	08
<b>Figure 1.4.</b> Prédiction .....	08
<b>Figure 1.5.</b> Modélisation inverse.....	09
<b>Figure 1.6.</b> Annulation d'interférences.....	09
<b>Figure 1.7.</b> Propagation du son dans une salle acoustique.....	10
<b>Figure 1.8.</b> Réponse impulsionnelle dispersive.....	10
<b>Figure 1.9.</b> Réponse impulsionnelle sparse.....	11
<b>Figure 1.10.</b> Schéma représentatif de l'algorithme LMS.....	12
<b>Figure 2.1.</b> Principe de débruitage mono-capteur.....	18
<b>Figure 2.2.</b> Structure complète d'un mélange convolutif bi-capteurs.....	20
<b>Figure 2.3.</b> Structure simplifié d'un mélange convolutif bi-capteurs.....	21
<b>Figure 2.4.</b> Structure de séparation de source Direct (Forward).....	22
<b>Figure 2.5.</b> Structure de séparation de source Réursive (Backward).....	24
<b>Figure 3.1.</b> Convergence de l'algorithme BNLMS : $\mu_{21n} = 0.9$ , $L = 128$ .....	31
<b>Figure 3.2.</b> SNR de l'algorithme BNLMS pour les deux RI .....	32
<b>Figure 3.3.</b> Structure générale des algorithmes Backward NLMS proportionnel bi-Capteurs.....	33

<b>Figure 3.4.</b> Structure de l’algorithme Backward NLMS proportionnel amélioré bi-capteur (IPBNLMS).....	34
<b>Figure 4.1.</b> Signal de parole.....	41
<b>Figure 4.2.</b> Bruit USASI.....	41
<b>Figure 4.3.</b> Réponses impulsionnelles dispersives.....	42
<b>Figure 4.4.</b> Réponses impulsionnelles sparse.....	43
<b>Figure 4.5.</b> Segmentation de signal de parole.....	43
<b>Figure 4.6.</b> Résultat pour une réponse dispersive : $\mu_{21n} = 0.9, L = 128$ .....	44
<b>Figure 4.7.</b> Résultat pour une réponse sparse : $\mu_{21n} = 0.9, L = 128$ .....	44
<b>Figure 4.8.</b> Résultat pour une réponse dispersive : $\mu_{21n} = 0.9, L = 128$ .....	45
<b>Figure 4.9.</b> Résultat pour une réponse sparse : $\mu_{21n} = 0.9, L = 12845$ .....	45
<b>Figure 4.10.</b> SM pour $L=1024, \mu_{21n} = 0.9, \alpha = 1, (\times 8)$ .....	46
<b>Figure 4.11.</b> SNR pour $L=1024, \mu_{21n} = 0.9, \alpha = 1, (\times 8)$ .....	46
<b>Figure 4.12.</b> SM pour $L=1024, \mu_{21n} = 0.9, \alpha = -0.5, (\times 8)$ .....	47
<b>Figure 4.13.</b> SNR pour $L=1024, \mu_{21n} = 0.9, \alpha = -0.5, (\times 8)$ .....	47
<b>Figure 4.14.</b> SM pour $L=1024, \mu_{21n} = 0.9, \alpha = -1, (\times 8)$ .....	48
<b>Figure 4.15.</b> SNR pour $L=1024, \mu_{21n} = 0.9, \alpha = -1, (\times 8)$ .....	48
<b>Figure 4.16.</b> Signal de parole estimé : $L=128, \mu_{21n} = 0.9, (\times 4)$ .....	49
<b>Figure 4.17.</b> Signal de parole estimé : $L=128, \mu_{21n} = 0.9, (\times 4)$ .....	49
<b>Figure 4.18.</b> SM pour $L=128, \mu_{21n} = 0.9, (\times 4)$ .....	50
<b>Figure 4.19.</b> SNR pour $L=128, \mu_{21n} = 0.9, (\times 4)$ .....	50
<b>Figure 4.20.</b> SM pour $L=256, \mu_{21n} = 0.9, (\times 4)$ .....	51

<b>Figure 4.21.</b> SNR pour $L=256, \mu_{21n} = 0.9, (\times 4)$ .....	51
<b>Figure 4.22.</b> SM pour $L=512, \mu_{21n} = 0.9, (\times 5)$ .....	52
<b>Figure 4.23.</b> SNR pour $L=512, \mu_{21n} = 0.9, (\times 5)$ .....	52
<b>Figure 4.24.</b> SM pour $L=1024, \mu_{21n} = 0.9, (\times 8)$ .....	53
<b>Figure 4.25.</b> SNR pour $L=1024, \mu_{21n} = 0.9, (\times 8)$ .....	53
<b>Figure 4.26.</b> SM pour $L=2048, \mu_{21n} = 0.9, (\times 8)$ .....	54
<b>Figure 4.27.</b> SNR pour $L=2048, \mu_{21n} = 0.9, (\times 8)$ .....	54
<b>Figure 4.28.</b> SM pour $L=1024, \mu_{21n} = 0.1, (\times 8)$ .....	55
<b>Figure 4.29.</b> SNR pour $L=1024, \mu_{21n} = 0.1, (\times 8)$ .....	55
<b>Figure 4.30.</b> SM pour $L=1024, \mu_{21n} = 0.5, (\times 8)$ .....	56
<b>Figure 4.31.</b> SNR pour $L=1024, \mu_{21n} = 0.5, (\times 8)$ .....	56
<b>Figure 4.32.</b> SM pour $L=1024, \mu_{21n} = 0.9, (\times 8)$ .....	57
<b>Figure 4.33.</b> SNR pour $L=1024, \mu_{21n} = 0.9, (\times 8)$ .....	57
<b>Figure 4.34.</b> SM pour $L=1024, \mu_{21n} = 1.5, (\times 8)$ .....	58
<b>Figure 4.35.</b> SNR pour $L=1024, \mu_{21n} = 1.5, (\times 8)$ .....	58

## Liste des tableaux

<b>Tableau 1.1.</b> Complexité algorithmique des trois algorithmes.....	16
<b>Tableau 2.1.</b> L'algorithme FLMS.....	25
<b>Tableau 2.2.</b> L'algorithme FNLMS.....	26
<b>Tableau 2.3.</b> L'algorithme BLMS.....	27
<b>Tableau 2.4.</b> L'algorithme BNLMS.....	28
<b>Tableau 3.1.</b> Pseudo code de l'algorithme PBNLMS capteur 01.....	37
<b>Tableau 3.2.</b> Pseudo code de l'algorithme PBNLMS capteur 02.....	37
<b>Tableau 3.3.</b> Pseudo code de l'algorithme BNLMS capteur 01.....	38
<b>Tableau 3.4.</b> Pseudo code de l'algorithme BNLMS capteur 02.....	38
<b>Tableau 3.5.</b> Complexité algorithmique des algorithmes.....	39

# Introduction générale

---

Le traitement de la parole est aujourd'hui une composante fondamentale dans le domaine de la télécommunication. Cette discipline scientifique a connu depuis les années 60 une expansion fulgurante, liée au développement des moyens et des techniques de télécommunications. L'importance particulière du traitement de la parole dans ce cadre plus général s'explique par la position privilégiée de la parole comme vecteur d'information dans notre société humaine. [01]

Avec le développement des technologies de communication, telles que la voix sur IP et le système de communication mains libres, la demande en qualité de service est considérablement accrue. Une sorte de réponse impulsionnelle clairesemée a été fréquemment rencontrée au cours des dernières années.

De façon générale les filtres adaptatifs sont des systèmes appliqués sur des données bruitées pour obtenir une information utile à un certain instant  $t$ . Un système de filtrage linéaire est fragmenté si sa réponse impulsionnelle contient un petit nombre de prises actives en présence d'un grand nombre de prises non actives. L'annuleur d'écho acoustique est un exemple de tels systèmes [02].

Les algorithmes de filtrage adaptatif peuvent trouver des applications dans de nombreux systèmes du monde réel. De nos jours, avec la demande croissante de communications de meilleure qualité, l'augmentation de la longueur du filtre adaptatif de dizaines à plusieurs centaines des milliers causes que les algorithmes d'adaptation classiques rencontrent de nouveaux défis. Premièrement la vitesse de convergence des algorithmes adaptatifs est fortement dégradée, car leur vitesse de convergence est inversement proportionnelle à la longueur du filtre adaptatif. Deuxièmement, la complexité informatique d'algorithmes

adaptatifs est considérablement augmentée car il est généralement une fonction de la longueur du filtre adaptative. Troisièmement, la qualité de convergence du filtre adaptatif est dégradée. De nombreux efforts ont été déployés pour trouver de nouveaux algorithmes adaptatifs permettant de résoudre ces problèmes.

Récemment, un nouveau paradigme d'adaptation proportionnelle, a été développé dans une nouvelle perspective [03]. Sa philosophie principale vient du fait que la plupart de ces réponses impulsionnelles longues sont de nature clairsemée (sparse). Une tentative avait été faite pour surmonter cette limitation en proposant un algorithme commutant alternativement entre les algorithmes NLMS et PNLMS. Après cela, l'algorithme bien connu PNLMS amélioré (IPNLMS) est proposé, qui utilise un mélange contrôlé d'algorithmes PNLMS et NLMS.

Depuis plusieurs années, la séparation aveugle de sources fait l'objet de nombreux travaux en traitement du signal, en télécommunication par la méthode bi-capteurs qui est une technique récente dans le domaine du traitement du signal. Elle vise à extraire des sources statistiquement indépendantes à partir d'un mélange linéaire de ces sources.

Par la suite, les problèmes de la réduction du bruit acoustique ont été traités par les techniques de séparation aveugle des sources bi-capteurs combinées avec les algorithmes du filtrage adaptatif [04] [05] [06]. Tous les algorithmes bi-capteurs sont proposés dans le cas où le mélange linéaire convolutif est caractérisé par des réponses impulsionnelles dispersives. L'inconvénient major de ces algorithmes est indispensable dans le cas des réponses impulsionnelles sparse.

Pour trouver une solution efficace a ce problème ; dans ce projet de fin d'étude, on va proposer un algorithme du filtrage adaptatif proportionnel implémenté sur la structure récursive (Backward) de séparation aveugle des sources bi-capteurs exactement pour la réduction du bruit acoustique dans un système sparse.

La structure Backward donne une bonne performance par rapport à la structure Forward en termes de qualité de la parole. Elle est souvent utilisée pour réduire les composantes du bruit acoustique à la sortie du traitement, donc elle est très efficace pour le débruitage et le rehaussement de la parole par rapport a la première [07].

Donc l'objectif principal de ce projet de fin étude est de reconstruire un signal aussi proche que possible de l'original, en appliquant la techniques de séparation aveugle des sources bi-capteurs combinées avec les algorithmes du filtrage adaptatif afin d'accélérer leur vitesse de convergence, de réduire le désalignement en régime permanent et de réduire leur complexité de calcul exactement dans un mélange convolutif avec des réponses impulsionnelles sparses. Pour cela notre travail sera présenté comme suit :

- **Le premier chapitre : Généralité sur le filtrage adaptatif.**

Il étudie la notion du filtrage adaptatif, son principe et ses applications ; les algorithmes adaptatifs LMS, NLMS, PNLMS et IPNLMS avec leurs avantages et inconvénients.

- **Le deuxième chapitre : Séparation aveugle de sources par des Algorithmes adaptatifs bi-capteur.**

Il représente les types de bruit, la réduction du bruit, la Séparation aveugle des sources, et enfin l'algorithme LMS et NLMS avec les deux structures Forward et Backward.

- **Le troisième chapitre : Algorithmes proposés Backward NLMS proportionnel bi-capteurs amélioré.**

Il représente la réduction du bruit en utilisant l'algorithme IPBNLMS proposés dans les systèmes sparse et sa structure, Vérification de la règle optimale du nouvel algorithme IPBNLMS proposé

- **Le quatrième chapitre : Résultats et Simulations.**

Il est entièrement consacré aux résultats de simulations de l'algorithme proposé ainsi que la comparaison de ce dernier avec l'algorithme BNLMS classique en termes de plusieurs critères de performances.



# Chapitre 1 Généralité sur le filtrage adaptatif

---

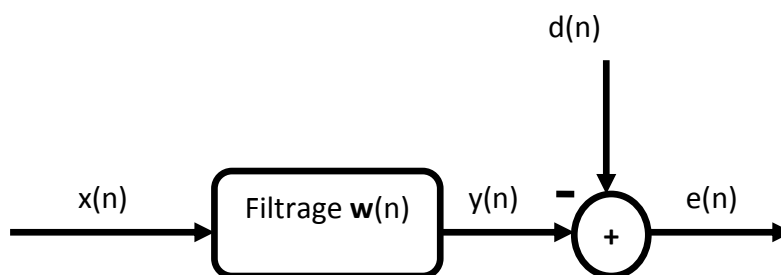
## 1.1. Introduction

Le filtrage adaptatif est très important dans nombreuses applications telles que l'annulation du bruit, la formation de faisceaux adaptative et égalisation aveugle.

Dans ce chapitre ; nous présentons le filtrage adaptatif son principe et ses applications ; ensuite nous allons présenter les algorithmes adaptatifs utilisés pour identifier les réponses impulsionnelles sparse et dispersive.

## 1.2. Filtre de Wiener

Généralement les signaux temporels sont entachés d'une interférence ou d'un bruit non désiré. Il faut alors trouver une solution permettant de supprimer tout ou au moins réduire ces composantes perturbatrices. Le filtrage de Wiener est parmi les méthodes de débruitage classiques les plus utilisées. Il est adéquat pour les situations dans lesquelles le signal ou le bruit sont stationnaires.



**Figure 1.1.** Représentation schématique d'un filtre de Wiener.

$x(n)$  : Signal d'entrée.

$y(n)$  : La sortie du filtre.

$\mathbf{W}(n)$  : Le vecteur des coefficients ajustables du filtre.

$e(n)$  : Le signal d'erreur entre la réponse désirée et la sortie du filtre.

Donc, le principe du filtre de Wiener est de trouver en sortie une réponse  $y(n)$ , la plus proche possible d'une réponse désirée  $d(n)$ , on note l'erreur d'estimation  $e(n) = d(n) - y(n)$ , en minimisant l'erreur quadratique moyenne (EQM) [08].

La sortie  $y(n)$  est donnée par :

$$y(n) = \sum_{l=0}^{L-1} \mathbf{w}_l(n) x(n-l) \quad (1.1)$$

On minimise le critère de l'EQM :

$$\mathbf{J} = E\{e(n)^2\} \quad (1.2)$$

$\mathbf{J}$ : La fonction cout.

$E$  : L'espérance mathématique.

Ou le signal d'erreur est:

$$e(n) = d(n) - y(n) = d(n) - \sum_{l=0}^{L-1} \mathbf{w}_l(n) \mathbf{x}(n-l) \quad (1.3)$$

C'est plus pratique d'utiliser une notation vectorielle pour la sortie du filtre:

$$y(n) = \mathbf{w}(n)^T \mathbf{x}(n) \quad (1.4)$$

Ou:

$$\mathbf{w}(n) = [\mathbf{w}_0(n) \mathbf{w}_1(n) \dots \dots \mathbf{w}_{L-1}(n)]^T \quad (1.5)$$

Est un vecteur de longueur L contenant les coefficients du filtre réponse impulsionnelle finie (RIF) et

$$\mathbf{x}(n) = [\mathbf{x}(n) \mathbf{x}(n-1) \dots \dots \mathbf{x}(n-L+1)]^T \quad (1.6)$$

Est le vecteur des L données d'entrée les plus récentes.

La minimisation de  $\mathbf{J}$  par rapport au vecteur  $\mathbf{w}$  donne à l'optimum:

$$E\{e_{\min}(n) \mathbf{x}(n)\} = \mathbf{0}_{L \times 1} \quad (1.7)$$

C'est le principe d'orthogonalité. On en déduit l'équation de Wiener-Hopf: [09]

$$\mathbf{R}_{xx} \mathbf{w}_{opt} = \mathbf{p}_{xd} \quad (1.8)$$

Et le filtre optimal:

$$\mathbf{w}_{opt} = \mathbf{R}_{xx}^{-1} \mathbf{p}_{xd} \quad (1.9)$$

Avec :

$\mathbf{R}_{xx}$ : C'est la matrice d'autocorrélation de signal d'entrée  $x(n)$ .

$\mathbf{p}_{xd}$  : C'est le vecteur d'inter-corrélation entre la sortie désirée  $d(n)$  et l'entrée  $x(n)$ .

Aussi:

$$\mathbf{w}_{opt} = \arg \min_{\mathbf{h}} J(\mathbf{w}) \quad (1.10)$$

### 1.3. Filtrage adaptatif

Le filtrage adaptatif est un outil puissant en traitement du signal, communications numériques, et contrôle automatique. C'est un système numérique dont les coefficients se modifient eux mêmes en fonction des signaux extérieurs. Il est utilisé chaque fois qu'un environnement est mal connu ou changeant ou pour supprimer des perturbations situées dans le domaine de fréquences du signal utile.

De façon générale les filtres adaptatifs sont des systèmes appliqués sur des données bruitées pour obtenir une information utile à un certain instant "n", ces systèmes étant mis en œuvre dans trois configurations [09] :

- Le filtrage.
- Le lissage.
- La prédiction.

Les signaux utilisés dans le filtrage adaptatif sont :

- $x(n)$  : le signal d'entrée de filtre.
- $d(n)$  : le signal de référence.
- $y(n)$  : le signal de sortie de filtre.
- $e(n)$  : le signal d'erreur.
- $w(n)$  : les coefficients de filtre.

Les filtres adaptatifs peuvent être classés en fonction des choix qui sont faits sur les points suivants [10]:

- Le critère d'optimisation.
- L'algorithme de mise à jour des coefficients.
- La structure du filtre programmable.
- Le type de signal traité, mono ou multidimensionnel.

### 1.3.1. Principe de filtrage adaptatif

La technique du filtrage adaptatif se décompose classiquement en deux étapes [11], une étape de filtrage qui permet d'obtenir une estimation du signal inconnu en convoluant le signal d'entrée  $x(n)$  avec les coefficients du filtre adaptatif  $w(n)$ . L'erreur d'estimation  $e(n)=d(n)-y(n)$  est ensuite utilisée dans la partie adaptation pour mettre à jour les coefficients du filtre, puis une étape d'adaptation qui permet d'ajuster les coefficients du filtre adaptatif  $w$  suivant un algorithme donné.

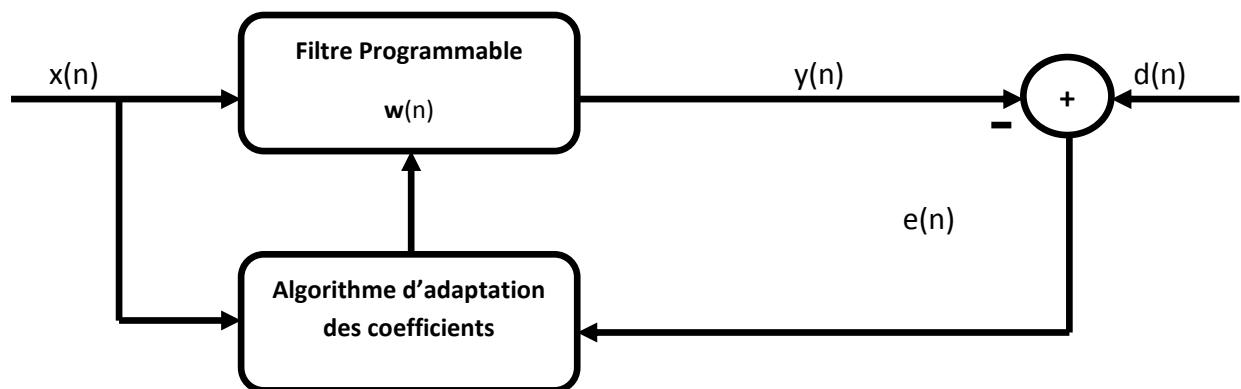


Figure 1.2. Principe d'un filtre adaptatif.

### 1.3.2. Applications des filtres adaptatifs

Les applications sont diverses mais présentent beaucoup des caractéristiques. On dispose une entrée  $x(n)$  ainsi que d'une réponse désirée (référence)  $d(n)$  et l'erreur  $e(n)$ , qui est la différence entre  $d(n)$  et la sortie du filtre  $y(n)$ , sert à contrôler (adapter) les valeurs des coefficients du filtre.

Ce qui différencie essentiellement les applications provient de la façon de définir la réponse désirée  $d(n)$  [09].

On peut distinguer quatre grandes classes d'applications:

- Identification de systèmes.
- Prédiction.
- Modélisation inverse.
- Annulation d'interférences.

### a). Identification de systèmes

La figure 1.3 illustre le contexte du problème d'identification.

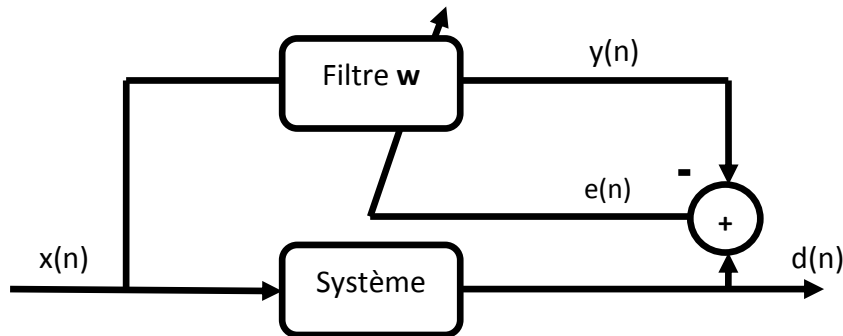


Figure 1.3. Identification de systèmes.

$d(n)$  est la sortie du système que l'on souhaite identifier.

### b). Prédiction

La figure 1.4 illustre le contexte du problème de prédiction.

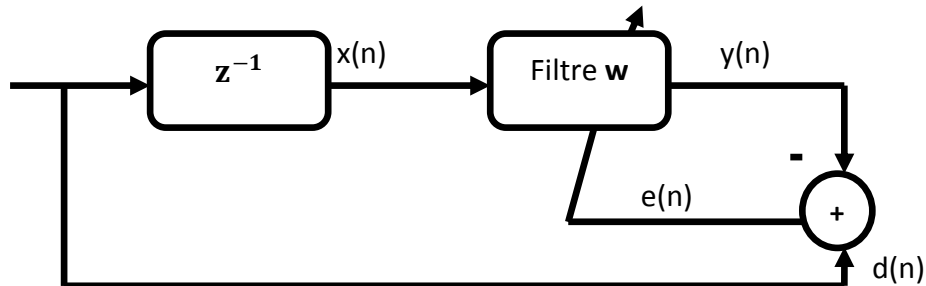


Figure 1.4. Prédiction.

$d(n)$  est le signal à l'instant  $n$  et  $y(n)$  le signal prédit à partir du signal aux instants précédents.

### c).Modélisation inverse

La figure 1.5 illustre le contexte du problème de modélisation inverse.

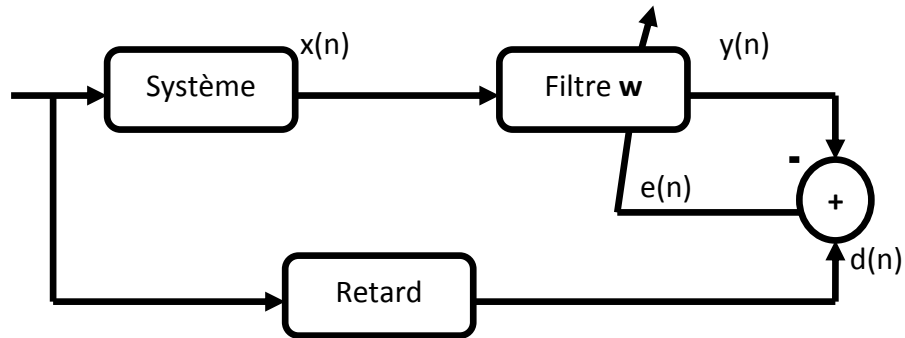


Figure 1.5. Modélisation inverse.

$d(n)$  est l'entrée (retardée) du système que l'on cherche à « inverser ».

### d). Annulation d'interférences

La figure 1.6 illustre le contexte du problème d'annulation d'interférence.

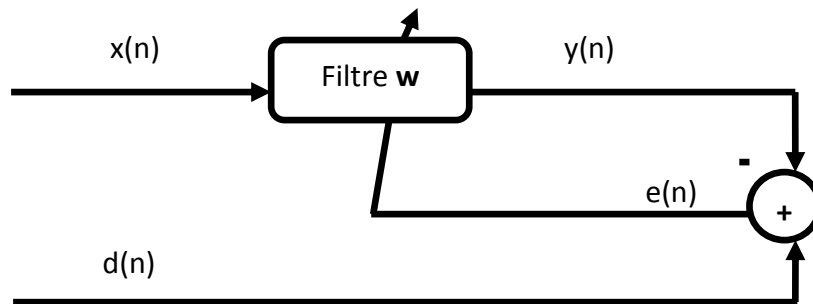
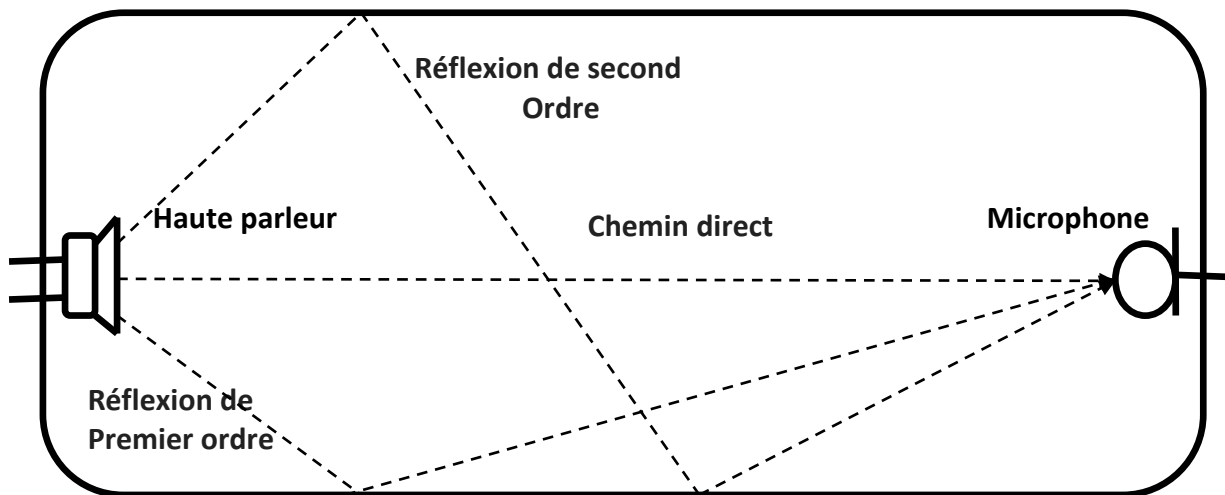


Figure 1.6. Annulation d'interférences.

$d(n)$  est un signal primaire qui comporte les interférences à annuler.  $x(n)$  est le signal de référence dénué (ou presque) d'information et obtenu par un capteur proche de celui qui fournit  $d(n)$ .

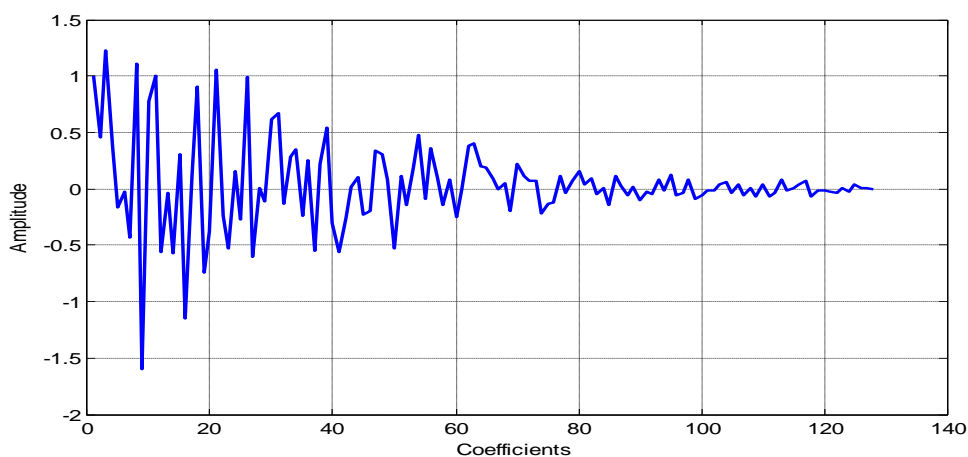
## 1.4. Réponse impulsionnelle acoustique

Dans une salle, le premier son à être entendu est celui venant directement de la source. Puis, après un court instant, l'auditeur entendra les reflets atténués du son (échos) sur les murs, lorsque le son est réfléchi à plusieurs reprises par les murs, chaque réflexion sera encore retardée et atténuée. Un plus grand nombre d'inspections de la réponse impulsionnelle de la pièce a permis de constater que le son décroît de manière exponentielle [12].



**Figure 1.7.** Propagation du son dans une salle acoustique.

Par conséquent, la réponse impulsionnelle dispersive de la salle montrée ci-dessus peut être similaire à la figure 1.8 :



**Figure 1.8.** Réponse impulsionnelle dispersive.

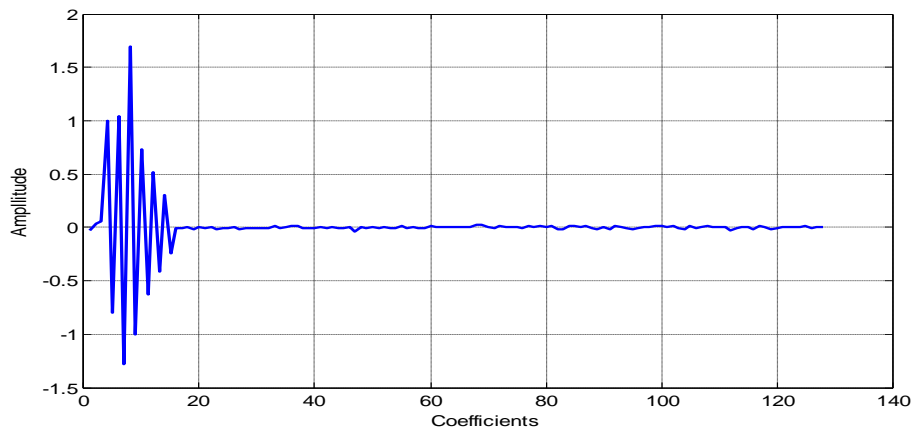
Les effets d'échos peuvent être réduits en ayant des absorbants autour du mur. Dans le cas, la réponse impulsionnelle à moins de coefficients actifs, comme illustré à la figure 1.9.

la réponse impulsionnelle est dite sparse, en raison de la majorité de ses filtres sont inactifs.

### 1.5. Réponse impulsionnelle sparse

Une réponse impulsionnelle parcimonieuse est une réponse impulsionnelle qui n'a qu'un faible pourcentage de ses composantes avec une magnitude significative, le reste étant nul ou très faible. Cela pourrait être défini comme une réponse impulsionnelle rare si une grande partie de son énergie est concentrée dans une petite fraction de sa durée.

Par exemple, dans une réponse impulsionnelle de réseau, seulement environ 8-12 ms sur une durée de 64 ou 128 ms sont actives et les autres sont des zéros (ou inactifs) [03].



**Figure 1.9.** Réponse impulsionnelle sparse.

## 1.6. Algorithmes adaptatifs

Les algorithmes adaptatifs traditionnels attribuent une même taille de pas à tous les coefficients, indépendamment de la structure de la réponse impulsionnelle cible. Pour ces algorithmes, les coefficients élevés ont besoin plus d'itérations qui convergent vers l'état d'équilibre que les petits coefficients.

### 1.6.1. Choix de l'algorithme

Le choix de l'algorithme se fera en fonction des critères suivants [10]:

- La rapidité de convergence qui sera le nombre d'itérations nécessaires pour converger la solution optimale.
- La mesure de cette "proximité" entre cette solution optimale et la solution obtenue,
- La capacité de poursuite des variations (non-stationnarités) du système,
- La robustesse au bruit,
- La complexité,
- La structure (modularité, parallélisme, ...),
- Les propriétés numériques (stabilité et précision) dans le cas d'une précision limitée sur les données et les coefficients du filtre.



### 1.6.2. Algorithme du gradient stochastique LMS

L'algorithme LMS (Least Mean Squares) est un choix populaire dans beaucoup d'applications exigeant le filtrage adaptatif. C'est une approximation de l'algorithme du gradient déterministe [13].

Deux raisons principales de sa popularité, sa simplicité et sa complexité informatique réduite. En outre, il y a plusieurs variantes de l'algorithme qui peuvent être employées spécifiquement afin de résoudre différents types de problèmes qui sont inhérents à certaines applications.

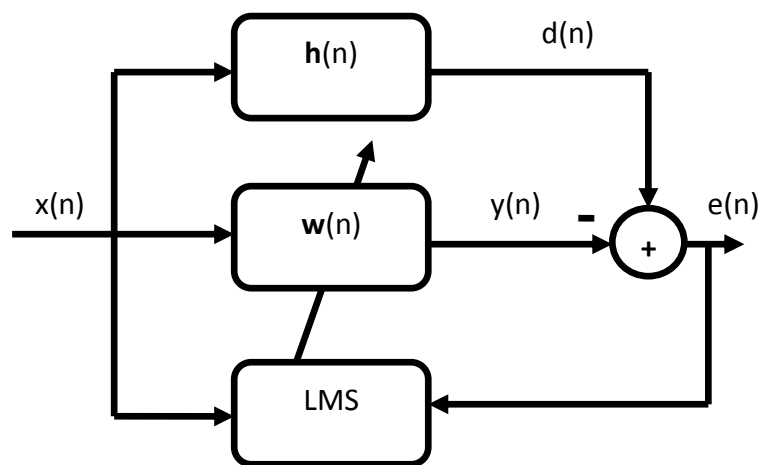


Figure 1.10. Schéma représentatif de l'algorithme LMS.

Calcul de la sortie du filtre:

$$y(n) = \mathbf{w}^T \mathbf{x}(n) \quad (1.11)$$

Calcul du signal d'erreur:

$$e(n) = d(n) - y(n) \quad (1.12)$$

Mise à jour du filtre:

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \mu \mathbf{x}(n)e(n) \quad (1.13)$$

$0 < \mu < \frac{2}{\lambda_{\max}}$ ,  $\mu$  est le pas d'adaptation de l'algorithme qui démarre avec une initialisation quelconque  $\mathbf{w}(0)$ .

L'algorithme LMS est très simple, il nécessite seulement  $(2L + 1)$  multiplications et  $2L$  additions par itération, où  $L$  est le nombre de coefficients du filtre.

## Convergence de l'algorithme LMS

L'analyse de la convergence du LMS se fait en utilisant les deux critères suivants [13] :

- Convergence en moyenne du filtre  $\mathbf{w}(n)$ , càd:

$$\lim_{n \rightarrow \infty} E\{\mathbf{w}(n)\} = \mathbf{w}_{opt} \quad (1.14)$$

- Convergence du critère  $J(n)$  (en moyenne quadratique), càd:

$$\lim_{n \rightarrow \infty} J(n) = J(\infty) = \text{constante} \quad (1.15)$$

## Caractéristiques de l'algorithme LMS

- L'algorithme LMS est très simple.
- Les performances du LMS dépendent de trois facteurs:
  - i.* Le pas d'adaptation  $\mu$ ,
  - ii.* Les valeurs propres  $\lambda_i$  de la matrice  $\mathbf{R}$ ,
  - iii.* La longueur  $L$  du filtre  $\mathbf{h}$ .
    - Avec un pas d'adaptation petit, le LMS converge lentement.
    - Avec un pas d'adaptation grand, le LMS converge rapidement.

### 1.6.3. Algorithme du gradient stochastique normalisé (NLMS)

L'algorithme NLMS (Normalized Least Mean Square) consiste à normaliser le pas d'adaptation  $\mu$  dans l'algorithme LMS par rapport à l'énergie du signal d'entrée pour réduire au minimum l'effet de la variation de la puissance du signal d'entrée et de rendre ainsi la convergence plus au moins uniforme en passant d'une étape d'adaptation à une autre.

Le pas d'adaptation est calculé en utilisant la formule suivante [03]:

$$\mu = \frac{\mu_n}{\mathbf{x}^T(n)\mathbf{x}(n)} \quad (1.16)$$

Pour la stabilité du NLMS il faut que :  $0 < \mu_n < 2$

La mise à jour des coefficients du filtre adaptatif par l'algorithme NLMS est alors donnée par [03]:

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \frac{\mu_n}{\mathbf{x}^T(n)\mathbf{x}(n) + \delta_{\text{NLMS}}} \mathbf{x}(n)e(n) \quad (1.17)$$

Où  $\delta_{\text{NLMS}}$  est un facteur permettant de suivre plus ou moins rapidement les variations d'énergie dans le signal d'entrée  $\mathbf{x}(n)$ .

### Avantages de l'algorithme NLMS

- La faible complexité des calculs.
- La stabilité numérique.

### Inconvénient de l'algorithme NLMS

La transitoire dépend de la dispersion spectrale du signal d'entrée. La convergence peut devenir très lente dans le cas de la parole.

#### 1.6.4. Algorithme Proportionné Normalisé LMS (PNLMS)

L'algorithme PNLMS représente une nouvelle technique de filtrage adaptatif dans le cas d'annulation d'écho acoustique. L'algorithme PNLMS a été proposé pour la première fois en introduisant une matrice diagonale de commande pas à pas  $\mathbf{G}(n)$  pour associer un paramètre pas à chaque coefficient. Cette technique consiste à adapter chaque étape à une valeur désirée avec un taux de convergence qui varie d'une étape d'adaptation à l'autre. Cet algorithme exploite la faible densité des réponses impulsionnelle pour réaliser une adaptation sensiblement plus rapide que celle réalisée par l'algorithme conventionnel NLMS.

L'algorithme PNLMS nécessite plus d'opérations que l'algorithme NLMS mais il présente l'avantage de converger plus rapidement que ce dernier.

Une version normalisée peut être obtenue en normalisant l'équation de mise à jour avec la norme euclidienne du vecteur d'entrée afin d'éliminer l'influence de la puissance du signal d'entrée [03] :

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \frac{\mu_n}{\mathbf{x}^T(n)\mathbf{G}(n)\mathbf{x}(n) + \delta_{\text{PNLMS}}} \mathbf{G}(n)\mathbf{x}(n)e(n) \quad (1.18)$$

Où  $\delta_{\text{PNLMS}}$  est un paramètre de régularisation ( $\delta_{\text{PNLMS}} = \frac{\delta_{\text{NLMS}}}{L}$ ).

Et  $\mathbf{G}(n)=\text{diag} [ g_0(n) \dots \dots g_{L-1}(n) ]$  est la matrice diagonale ( $L \times L$ ). (1.19)

La définition originale de l'élément diagonal de la matrice  $\mathbf{G}(n)$  dans l'algorithme PNLMS est décrite comme suit:

$$g_l(n) = \max \{ \{ |w_l(n)| \}, \rho \max \{ \delta_p, |w_0(n)|, \dots |w_{L-1}(n)| \} \} \quad (1.20)$$

Avec  $l=0,1,\dots, L-1$

$$g_l(n) = \frac{k_l(n)}{\frac{1}{L} \sum_{i=0}^{L-1} |k_i(n)|} \quad (1.21)$$

$\delta_p$  est utilisé pour empêcher le blocage de  $w(n)$  pendant l'étape d'initialisation. Sa valeur typique est  $\delta_p=0.01$ . Le paramètre  $\rho$  est utilisé pour empêcher les coefficients de caler quand ils sont beaucoup plus petits que les plus grands, sa valeur typique est de  $1 / L$  à  $5 / L$

Avantage du PNLMS est qu'il nécessite moins d'informations a priori sur le chemin d'écho, puisque la mise à jour ne dépend que des estimations de filtre actuelles.

### 1.6.5. Algorithme PNLMS amélioré (IPNLMS)

Les algorithmes adaptatifs proportionnels ont été initialement conçus pour une réponse impulsionnelle sparse. Pour une réponse impulsionnelle non clairsemée (non-sparse), la vitesse de convergence de l'algorithme PNLMS se dégrade considérablement, même pire que l'algorithme NLMS. Le PNLMS amélioré (IPNLMS) est proposé pour éviter la dégradation dans le cas où l'impulsion sous-jacente la réponse est non-sparse.

La raison pour laquelle l'algorithme PNLMS converge plus lentement que l'algorithme NLMS pour réponse impulsionnelle dispersive est en raison du choix brutal, l'opérateur maximum. Au lieu de ce choix brutal, la moyenne des coefficients actuels est adoptée pour éliminer partiellement les effets négatifs d'une estimation inexacte de  $w(n)$ . Par conséquent, l'algorithme IPNLMS converge aussi rapidement comme l'algorithme PNLMS pour la réponse impulsionnelle sparse et sa performance n'est pas pire que l'algorithme NLMS pour un dispersif.

L'élément diagonal de  $\mathbf{G}(n)$  dans l'IPNLMS algorithme peut être décrit comme:

$$k_l = \frac{1 - \alpha}{2L} + (1 + \alpha) \frac{|w_l(n)|}{2 \sum_{i=0}^{L-1} |w_i(n)| + \delta_{IPNLMS}} \quad (1.22)$$

$\delta_{IPNLMS}$  est un très petit nombre positif pour éviter de diviser par zéro.

$$\delta_{IPNLMS} = \frac{1 - \alpha}{2L} \delta_{NLMS} \quad (1.23)$$

Le paramètre ajustable,  $\alpha \in [-1, 1]$ , peut modifier le comportement de IPNLMS entre NLMS et PNLMS. On peut le voir IPNLMS est équivalent à NLMS lorsque  $\alpha = -1$  et pour  $\alpha$  proche de 1, il se comporte comme PNLMS.

Cela signifie que la même mise à jour sera faite pour tous les coefficients de filtre indépendamment de leurs grandeurs individuelles. Donc, pour cette valeur  $\alpha$  IPNLMS fonctionne comme NLMS pour  $\alpha = -1$  et pour  $\alpha = 1$  se comporte comme PNLMS [14].

### 1.7. Complexité algorithmique

Le tableau suivant résume les opérations de calcul pour les trois algorithmes. On peut remarquer que la complexité globale est accrue ou resté la même lorsque l'amélioration est faite. Pour compenser ce problème, leurs performances doivent être nettement plus élevées.

Algorithme	NLMS	PNLMS	IPNLMS
Addition	2L+1	3L+1	4L+2
Multiplication	3L+1	6L+4	6L+4
Division	1	2	2

**Tableau 1.1.** Complexité algorithmique des trois algorithmes.

### 1.8. Conclusion :

Dans ce chapitre, nous avons présenté le filtrage adaptatif, son principe et ses applications , et aussi nous avons cité les algorithmes adaptatifs LMS sa convergence et ses caractéristiques et NLMS ses avantages et inconvénients , les algorithmes sparse PNLMS et IPNLMS et leur complexité algorithmique.

# Chapitre 2 Séparation aveugle de sources par des algorithmes adaptatifs bi-capteur

---

## 2.1. Introduction

Le problème de la séparation aveugle de sources consiste à retrouver avec le moins de connaissance a priori, des signaux utiles qui ont été mélangés. Pour résoudre ce problème deux structures Forward et Backward sont utilisées pour extraire les signaux sources à partir de la seule connaissance des signaux de mélange.

Dans ce chapitre, nous avons présenté les méthodes de réduction de bruit (mono-capteur et bi-capteur), et nous avons expliqué les deux structures de séparation aveugle de sources.

## 2.2. Généralités sur le bruit

### 2.2.1. Définitions du bruit

Le bruit est tout un signal nuisible qui se superpose au signal utile. Aussi, le bruit est tout phénomène perturbateur gênant la perception ou l'interprétation d'un signal utile (la parole). En acoustique le bruit soit, par nature, aléatoire, il possède certaines caractéristiques statistiques, spectrales ou spatiales [15].

### 2.2.2. Quelques types du bruit

**Bruit acoustique** : il est généré par les mouvements des sources telles que le vent,... ce dernier peut altérer la qualité de la communication et engendrer une perte de l'information transmise [16].

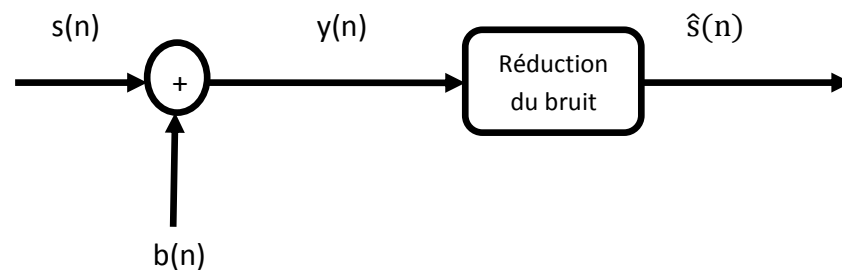
**Bruit blanc** : un bruit blanc est un signal aléatoire stationnaire de variance infinie, il est composé de toutes les fréquences au même niveau statistique, il présente la même énergie pour toute les fréquences.

**Bruit USASI** : (United States of American Standard Institute) c'est un signal aléatoire stationnaire dont ses échantillons sont des variables aléatoires corrélés. Il est utilisé dans le domaine du rehaussement de la parole et la réduction de bruit pour évaluer la vitesse de convergence des algorithmes adaptatifs [17].

## 2.3. Réduction du bruit

### 2.3.1. Méthode mono-capteur

Dans cette partie, nous allons analyser la technique la plus ancienne pour la réduction du bruit, où un seul capteur est utilisé. On cherche à améliorer la qualité de l'écoute à travers cette méthode pour des applications comme la téléphonie mobile et la téléphonie mains libres. Nous pouvons aussi utiliser ces méthodes de débruitage dans plusieurs applications. [15] On considère le problème de débruitage donné par la figure 2.1 :



**Figure 2.1.** Principe de débruitage mono-capteur.

Soit :

- $s(n)$  : signal d'entrée.
- $b(n)$  : signal de bruit.
- $y(n)$  : signal de parole bruité et échantillonné.

Le filtrage dans le domaine fréquentiel du signal microphonique :

$$y(n) = s(n) + b(n) \quad (2.1)$$

$y(n)$  est divisé en trames successives, d'indice  $k$ , qui comporte le même nombre d'échantillons noté  $N$ , qui est choisi de manière à ce que la durée d'une trame soit de l'ordre de 10 à 30ms. On construit ainsi des trames où la parole est supposée stationnaire [17].

$$y_k(n) = s_k(n) + b_k(n) \quad (2.2)$$

On cherche à restituer le signal propre  $s_k(n)$  à partir du signal observé  $y_k(n)$ .

Les transformées de Fourier discrètes (TFD) des signaux  $y_k(n)$ ,  $s_k(n)$  et  $b_k(n)$  sont notées respectivement,  $Y(f,k)$ ,  $B(f,k)$  et  $S(f,k)$  où  $f$  représente la fréquence, donc :

$$Y(f, k) = S(f, k) + B(f, k) \quad (2.3)$$

On peut définir les densités spectrales de puissance des signaux par la relation suivante :

$$\gamma_y(f, k) = \gamma_s(f, k) + \gamma_b(f, k) \quad (2.4)$$

Où  $\gamma_y(f, k)$ ,  $\gamma_s(f, k)$ , et  $\gamma_b(f, k)$  représentent respectivement les densités spectrales de puissance (DSP) des signaux  $y(n)$ ,  $s(n)$  et  $b(n)$ , Cette représentation sous forme de DSP n'est malheureusement pas exploitable en raison de la non-stationnarité du signal de parole.

Chaque signal temporel  $y(n)$  peut donc être représenté dans le domaine fréquentiel par son module  $|Y(f,k)|$  et sa phase  $|\Phi_y(f,k)|$ , Où  $k$  l'indice fréquentiel.

Cette dernière notation est justifiée sous l'hypothèse de la non-corrélation entre le bruit et le signal.

Dans le domaine fréquentiel, l'équation (2.4) peut donc s'exprimer ainsi :

$$|Y(f, k)|e^{-j\phi_y(f,k)} = |S(f, k)|e^{-j\phi_s(f,k)} + |B(f, k)|e^{-j\phi_b(f,k)} \quad (2.5)$$

Alors, le spectre du signal rehaussé, s'exprime par la relation suivante :

$$\hat{S}(f, k) = |\hat{S}(f, k)|e^{-j\phi_s(f,k)} \quad (2.6)$$

Pour résoudre ce problème une méthode de débruitage est utilisé c'est la soustraction spectral, son principe est de soustraire une estimée du bruit à partir du signal observé ; Il existe deux versions de base de la soustraction spectrale l'une utilise la puissance et l'autre utilise l'amplitude [08].

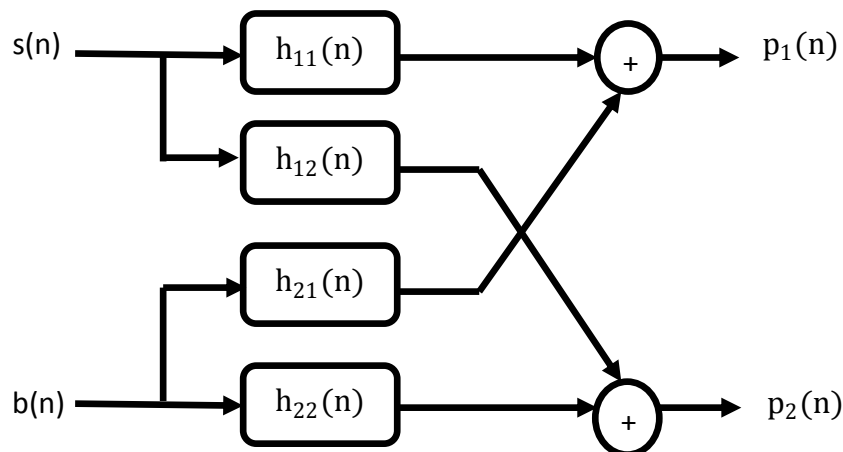


### 2.3.2. Méthode bi-capteur

Contrairement à la méthode présentée dans la partie précédente qui permet de filtrer le bruit d'un signal enregistré avec un seul microphone, la méthode bi-capteur nécessite l'utilisation de deux microphones pour annuler le bruit, une source de parole (locuteur), l'autre source de perturbation (un bruit) [05].

Dans cette partie ; nous allons définir le modèle de mélange convolutif ; La classe générale des mélanges convolutifs tient compte de la déformation du signal propagé et la modélise par un filtrage entre la source et l'observation, ce qui s'écrit mathématiquement sous la forme d'une convolution.

Le mélange convolutif linéaire entre le signal de parole et le bruit peut être représentée par un modèle complet donné par la figure 2.2 :



**Figure 2.2.** Structure complète d'un mélange convolutif bi-capteurs.

Ou :

$s(n)$  : c'est le signal parole.

$b(n)$  : c'est le signal du bruit.

$p_1(n)$  et  $p_2(n)$  : représentent respectivement les deux signaux bruités obtenus à la sortie du mélange convolutif bi-capteurs.

$h_{11}(n)$  et  $h_{22}(n)$  : sont les réponses impulsionnelles directe du chaque canal.

$h_{12}(n)$  et  $h_{21}(n)$  : représente le couplage croisé entre les sources et les microphones.

Les équations des signaux observés à la sortie de ce mélange sont donnés par :

$$p_1(n) = s(n) * h_{11}(n) + b(n) * h_{21}(n) \quad (2.7)$$

$$p_2(n) = b(n) * h_{22}(n) + s(n) * h_{12}(n) \quad (2.8)$$

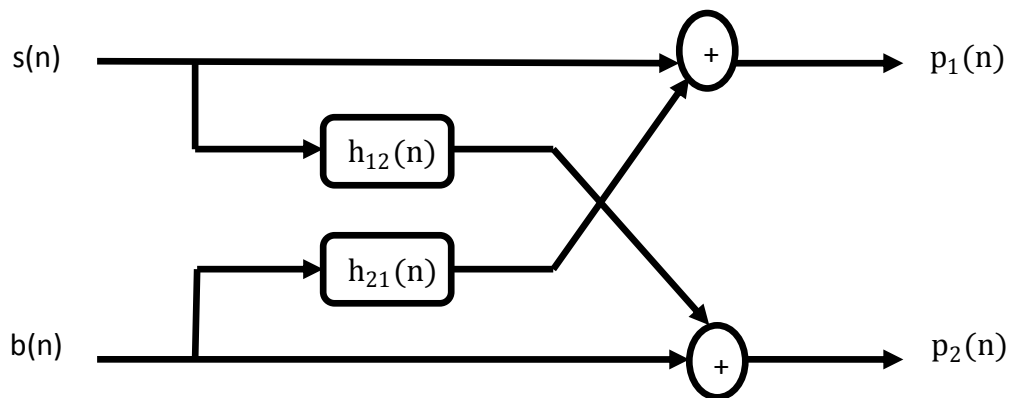
Nous pouvons prendre un modèle du mélange convolutif simplifié dans lequel :

$$h_{11}(n) = h_{22}(n) = \delta(n) \quad (2.9)$$

Où  $\delta(n)$  est l'impulsion de Dirac.

Nous supposons que le premier microphone est placé près du locuteur et le deuxième près de la source de bruit et la distance entre les deux microphones est petite [15].

Donc le modèle du mélange convolutif simplifié est donné par la figure 2.3 :



**Figure 2.3.** Structure simplifiée d'un mélange convolutif bi-capturs.

Les équations 2.7 et 2.8 deviennent :

$$p_1(n) = s(n) + b(n) * h_{21}(n) \quad (2.10)$$

$$p_2(n) = b(n) + s(n) * h_{12}(n) \quad (2.11)$$

## 2.4. Séparation aveugle de sources (SAS)

La plupart des signaux acoustiques résultent du mélange de plusieurs sources, si bien que le traitement du signal acoustique fut l'un des premiers domaines dans lequel on a cherché à appliquer les techniques de SAS [18] [15].

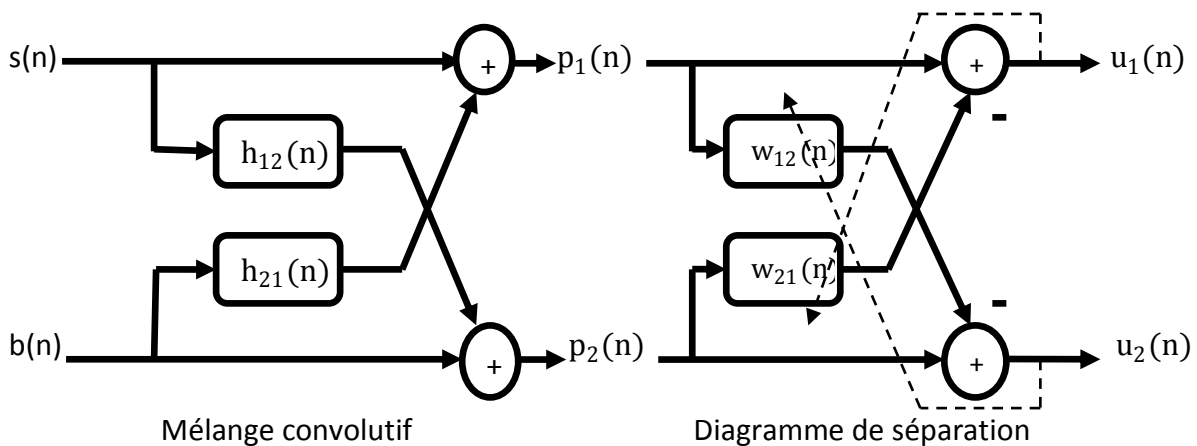
La SAS est une technique qui consiste à restaurer un ensemble des signaux de sources à partir d'observations bruitées, qui sont des mélanges entre les signaux de sources et les

réponses impulsionnelle du milieu. Cette séparation est dite "Aveugle" puisqu'on admet le peu d'information a priori sur les signaux de sources et le système du mélange.

Il existe deux structures appelées Direct (Forward) et Récursive (Backward) qui sont utilisés pour extraire les signaux de sources a partir de la seul connaissance des signaux de mélange.

### 2.4.1. Structure directe (Forward)

La structure la plus utilisée est la structure directe, lorsque chaque sortie du système de séparation est une simple combinaison linéaire des observations, le schéma est illustré par la figure 2.4 :



**Figure 2.4.** Structure de séparation de source Direct (Forward).

Où :  $w_{12}(n)$  et  $w_{21}(n)$  représente les filtres adaptative qu'on cherche à trouver.

Pour ce modèle les signaux de sortie  $u_1(n)$  et  $u_2(n)$  sont données par les formules suivantes :

$$u_1(n) = p_1(n) - p_2(n) * w_{21}(n) \quad (2.12)$$

$$u_2(n) = p_2(n) - p_1(n) * w_{12}(n) \quad (2.13)$$

En remplaçant les expressions  $p_1(n)$  et  $p_2(n)$  dans les expressions de  $u_1(n)$  et  $u_2(n)$  , nous trouvons :

$$u_1(n) = s(n) + b(n) * h_{21}(n) - [b(n) + s(n) * h_{12}(n)] * w_{21}(n) \quad (2.14)$$

$$u_2(n) = b(n) + s(n) * h_{12}(n) - [s(n) + b(n) * h_{21}(n)] * w_{12}(n) \quad (2.15)$$

Donc :

$$u_1(n) = b(n) * [h_{21}(n) - w_{21}(n)] + s(n) * [\delta(n) - h_{12}(n) * w_{21}(n)] \quad (2.16)$$

$$u_2(n) = s(n) * [h_{12}(n) - w_{12}(n)] + b(n) * [\delta(n) - h_{21}(n) * w_{12}(n)] \quad (2.17)$$

Après la convergence :

$$w_{12}^{\text{opt}}(n) = h_{12}(n) \text{ et } w_{21}^{\text{opt}}(n) = h_{21}(n)$$

On obtient les signaux de sortie  $u_1(n)$  et  $u_2(n)$  :

$$u_1(n) = s(n) * [\delta(n) - h_{12}(n) * w_{21}(n)] \quad (2.18)$$

$$u_2(n) = b(n) * [\delta(n) - h_{21}(n) * w_{12}(n)] \quad (2.19)$$

Les deux signaux de sortie  $u_1(n)$  et  $u_2(n)$  convergent respectivement vers les deux signaux originaux  $s(n)$  et  $b(n)$  mais avec une distorsion. Pour éviter cette distorsion, on peut ajouter deux post-filtres  $PF_1(n)$  et  $PF_2(n)$  à la sortie de la structure Forward [06]:

$$PF_1(n) = PF_2(n) = \frac{1}{\delta(n) - w_{12}(n) * w_{21}(n)} \quad (2.20)$$

Donc a la sortie des deux post-filtres  $PF_1(n)$  et  $PF_2(n)$  , les signaux de sortie convergent respectivement vers les deux signaux  $s(n)$  et  $b(n)$ .

### **Avantage de la structure Directe (Forward)**

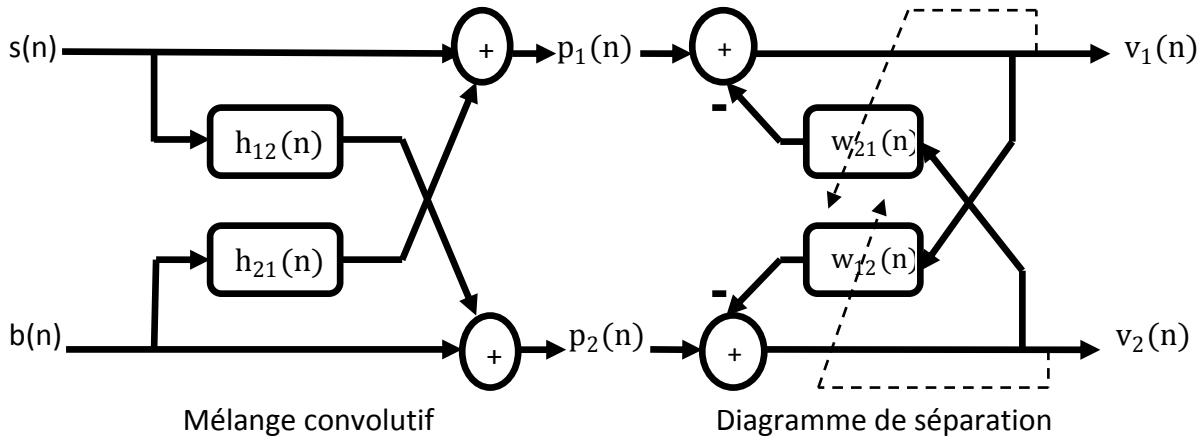
La structure Directe permet d'utiliser des filtres non-causaux de manière naturelle car elle ne fait pas intervenir les différentes sorties du système dans le calcul des sorties des filtres de séparation. [19]

### **Inconvénient de la structure Direct (Forward)**

Nous remarquons que les signaux de sortie  $u_1(n)$  et  $u_2(n)$  sont distordus, c'est l'inconvénient majeur de cette structure, pour éliminer ces distorsions, une solution a été proposée consiste a l'utilisation des post-filtres placés en sortie de la structure Directe. [19]

### 2.4.2. Structure réursive (Backward)

Elle s'applique lorsque chaque sortie du système de séparation est une combinaison linéaire d'une observation  $p_1(n)$  et  $p_2(n)$  et de l'autre sortie  $v_1(n)$  et  $v_2(n)$ . [19]



**Figure 2.5.** Structure de séparation de source Réursive (Backward).

Les deux signaux de sortie  $v_1(n)$  et  $v_2(n)$  de la structure Backward sont donnés par les deux équations suivantes:

$$v_1(n) = p_1(n) - v_2(n) * w_{21}(n) \quad (2.21)$$

$$v_2(n) = p_2(n) - v_1(n) * w_{12}(n) \quad (2.22)$$

En remplaçant les expressions de  $p_1(n)$  et  $v_2(n)$  dans l'expression de  $v_1(n)$ , nous trouvons :

$$v_1(n) = s(n) + b(n) * h_{21}(n) - [p_2(n) - v_1(n) * w_{12}(n)] * w_{21}(n) \quad (2.23)$$

En remplaçant  $p_2(n)$  par son expression dans  $v_1(n)$ , on trouve :

$$v_1(n) = \frac{1}{\delta(n) - w_{12}(n) * w_{21}(n)} [s(n) * (\delta(n) - h_{12}(n) * w_{21}(n)) + b(n) * (h_{21}(n) - w_{21}(n))] \quad (2.24)$$

Et de même pour  $v_2(n)$ , en remplaçant les expressions de  $p_2(n)$  et  $v_1(n)$  dans l'expression de  $v_2(n)$  ensuite de  $p_1(n)$  dans l'expression de  $v_2(n)$ , nous trouvons :

$$v_2(n) = b(n) + s(n) * h_{12}(n) - [p_1(n) - v_2(n) * w_{21}(n)] * w_{12}(n) \quad (2.25)$$

Donc :

$$v_2(n) = \frac{1}{\delta(n) - w_{12}(n) * w_{21}(n)} [b(n) * (\delta(n) - h_{21}(n) * w_{12}(n)) + s(n) * (h_{12}(n) - w_{12}(n))] \quad (2.26)$$

En utilisant l'hypothèse d'optimalité pour les deux filtres adaptatifs  $w_{12}^{opt}(n) = h_{12}(n)$  et  $w_{21}^{opt}(n) = h_{21}(n)$ , les deux signaux de sortie  $v_1(n)$  et  $v_2(n)$  deviennent :

$$v_1(n) = s(n) \quad (2.27)$$

$$v_2(n) = b(n) \quad (2.28)$$

## 2.5. L'algorithme LMS et NLMS avec la structure Direct (Forward)

### 2.5.1. L'algorithme FLMS

Le tableau 2.1 représente l'implémentation de l'algorithme LMS sur la structure directe Forward (FLMS) :

**Initialisation :**

$$\mathbf{w}_{21}(n) = [w_{21,1}(n); w_{21,2}(n); \dots \dots; w_{21,L}(n)]$$

$$\mathbf{w}_{12}(n) = [w_{12,1}(n); w_{12,2}(n); \dots \dots; w_{12,L}(n)]$$

L taille du filtre.

$\mu_{12n}$  et  $\mu_{21n}$  les deux pas d'adaptation.

$$0 < \mu_{12n} < \frac{2}{\lambda_{\max}}, \quad 0 < \mu_{21n} < \frac{2}{\lambda_{\max}} \quad (\lambda_{\max} : \text{l'énergie du signal})$$

**Pour**  $n=0,1, 2,3, \dots$

**Les signaux de sortie :**

$$u_1(n) = p_1(n) - \mathbf{p}_2^T(n) * \mathbf{w}_{21}(n)$$

$$u_2(n) = p_2(n) - \mathbf{p}_1^T(n) * \mathbf{w}_{12}(n)$$

**Les équations de mise à jour :**

$$\mathbf{w}_{21}(n) = \mathbf{w}_{21}(n-1) + \mu_{21n} \mathbf{p}_2^T(n) u_1(n)$$

$$\mathbf{w}_{12}(n) = \mathbf{w}_{12}(n-1) + \mu_{12n} \mathbf{p}_1^T(n) u_2(n)$$

**Fin**

**Tableau 2.1.** L'algorithme FLMS.

## 2.5.2. L'algorithme FNLMS

Le tableau 2.2 représente l'implémentation de l'algorithme NLMS sur la structure directe Forward (FNLMS) :

**Initialisation :**

$$\mathbf{w}_{21}(n) = [w_{21.1}(n); w_{21.2}(n); \dots \dots ; w_{21.L}(n)]$$

$$\mathbf{w}_{12}(n) = [w_{12.1}(n); w_{12.2}(n); \dots \dots ; w_{12.L}(n)]$$

L taille du filtre.

$\mu_{12n}$  et  $\mu_{21n}$  les deux pas d'adaptation.

**Pour**  $n=0,1, 2,3, \dots$

**Les signaux de sortie :**

$$u_1(n) = p_1(n) - \mathbf{p}_2^T(n) * \mathbf{w}_{21}(n)$$

$$u_2(n) = p_2(n) - \mathbf{p}_1^T(n) * \mathbf{w}_{12}(n)$$

**Les équations de mise à jour :**

$$\mathbf{w}_{21}(n) = \mathbf{w}_{21}(n-1) + \mu_{21n} \frac{\mathbf{p}_2^T(n)}{\mathbf{p}_2^T(n)\mathbf{p}_2(n) + \delta_{NLMS}} u_1(n)$$

$$\mathbf{w}_{12}(n) = \mathbf{w}_{12}(n-1) + \mu_{12n} \frac{\mathbf{p}_1^T(n)}{\mathbf{p}_1^T(n)\mathbf{p}_1(n) + \delta_{NLMS}} u_2(n)$$

**Fin**

**Tableau 2.2.** L'algorithme FNLMS.

Où  $\delta_{NLMS}$  est une petite constante positive utilisée pour éviter la division par zéro. La convergence de cet algorithme est garantie pour  $0 < \mu_{12n} < 2$  et  $0 < \mu_{21n} < 2$ .

## 2.6. L'algorithme LMS et NLMS avec la structure Récursive (Backward)

### 2.6.1. L'algorithme BLMS

Le tableau 2.3 représente l'implémentation de l'algorithme LMS sur la structure récursive Backward (BLMS) :

**Initialisation :**

$$\mathbf{w}_{21}(n) = [w_{21.1}(n); w_{21.2}(n); \dots \dots; w_{21.L}(n)]$$

$$\mathbf{w}_{12}(n) = [w_{12.1}(n); w_{12.2}(n); \dots \dots; w_{12.L}(n)]$$

L taille du filtre.

$\mu_{12n}$  et  $\mu_{21n}$  les deux pas d'adaptation.

$$0 < \mu_{12n} < \frac{2}{\lambda_{\max}} \quad , \quad 0 < \mu_{21n} < \frac{2}{\lambda_{\max}}$$

**Pour**  $n=0,1, 2,3, \dots$

**Les signaux de sortie :**

$$v_1(n) = p_1(n) - \mathbf{v}_2^T(n) * \mathbf{w}_{21}(n)$$

$$v_2(n) = p_2(n) - \mathbf{v}_1^T(n) * \mathbf{w}_{12}(n)$$

**Les équations de mise à jour :**

$$\mathbf{w}_{21}(n) = \mathbf{w}_{21}(n-1) + \mu_{21n} \mathbf{v}_2^T(n) v_1(n)$$

$$\mathbf{w}_{12}(n) = \mathbf{w}_{12}(n-1) + \mu_{12n} \mathbf{v}_1^T(n) v_2(n)$$

**Fin**

**Tableau 2.3.**L'algorithme BLMS.

En adaptant l'équation de mise à jour de l'algorithme LMS aux deux filtres adaptatifs de la structure Backward bi-capteurs, nous obtenons l'algorithme Backward de gradient stochastique double (2CBLMS : Two-Channel Backward LMS).

Les équations de mise à jour des coefficients des deux filtres adaptatifs  $w_{12}(n)$  et  $w_{21}(n)$  sont données dans le tableau 2.3,  $\mu_{12n}$  et  $\mu_{21n}$  sont les pas d'adaptations fixes.



$\mathbf{v}_1^T(n)$  et  $\mathbf{v}_2^T(n)$  sont des vecteurs qui contiennent les échantillons les plus récents des signaux de sortie  $v_1(n)$  et  $v_2(n)$  respectivement.

### 2.6.2. L'algorithme BNLMS

Le tableau 2.3 représente l'implémentation de l'algorithme NLMS sur la structure récursive Backward (BNLMS) :

**Initialisation :**

$$\mathbf{w}_{21}(n) = [w_{21.1}(n); w_{21.2}(n); \dots; w_{21.L}(n)]$$

$$\mathbf{w}_{12}(n) = [w_{12.1}(n); w_{12.2}(n); \dots; w_{12.L}(n)]$$

L taille du filtre.

$\mu_{12n}$  et  $\mu_{21n}$  les deux pas d'adaptation.

$$0 < \mu_{12n} < 2 \quad , \quad 0 < \mu_{21n} < 2$$

**Pour**  $n=0,1, 2,3, \dots$

**Les signaux de sortie :**

$$v_1(n) = p_1(n) - \mathbf{v}_2^T(n) * \mathbf{w}_{21}(n)$$

$$v_2(n) = p_2(n) - \mathbf{v}_1^T(n) * \mathbf{w}_{12}(n)$$

**Les équations de mise à jour :**

$$\mathbf{w}_{21}(n) = \mathbf{w}_{21}(n-1) + \mu_{21n} \frac{\mathbf{v}_2^T(n)}{\mathbf{v}_2^T(n)\mathbf{v}_2(n) + \delta_{NLMS}} v_1(n)$$

$$\mathbf{w}_{12}(n) = \mathbf{w}_{12}(n-1) + \mu_{12n} \frac{\mathbf{v}_1^T(n)}{\mathbf{v}_1^T(n)\mathbf{v}_1(n) + \delta_{NLMS}} v_2(n)$$

**Fin**

**Tableau 2.4.** L'algorithme BNLMS.

## **2.7. Conclusion**

Dans ce chapitre, nous avons donné une vue générale sur le bruit, ces différents types. Comme nous avons vu des méthodes pour la réduction du bruit (la méthode mono-capteur et la méthode bi-capteur).

Dans le cadre de la réduction du bruit par les méthodes de séparation aveugle de sources bi-capteurs, nous avons présenté deux structures, la structure directe (Forward) et l'autre récursive (Backward), ainsi que leurs avantages et inconvénients.

# Chapitre 3 Algorithmes proposés Backward NLMS

## proportionnel bi-capteurs amélioré

---

### 3.1. Introduction

La plupart des algorithmes de filtrage adaptatif ont été mis en œuvre sur des structures de séparation de sources aveugles, directe (Forward) et récursive (Backward) pour la réduction du bruit. La structure Backward est souvent utilisée pour séparer le bruit de la parole, nous proposons une nouvelle configuration de l'algorithme adaptative récursive utilisant des paramètres de tailles de pas normalisées. Pour valider les bonnes performances de notre algorithme proposé.

### 3.2. Problème de l'algorithme Backward NLMS classique

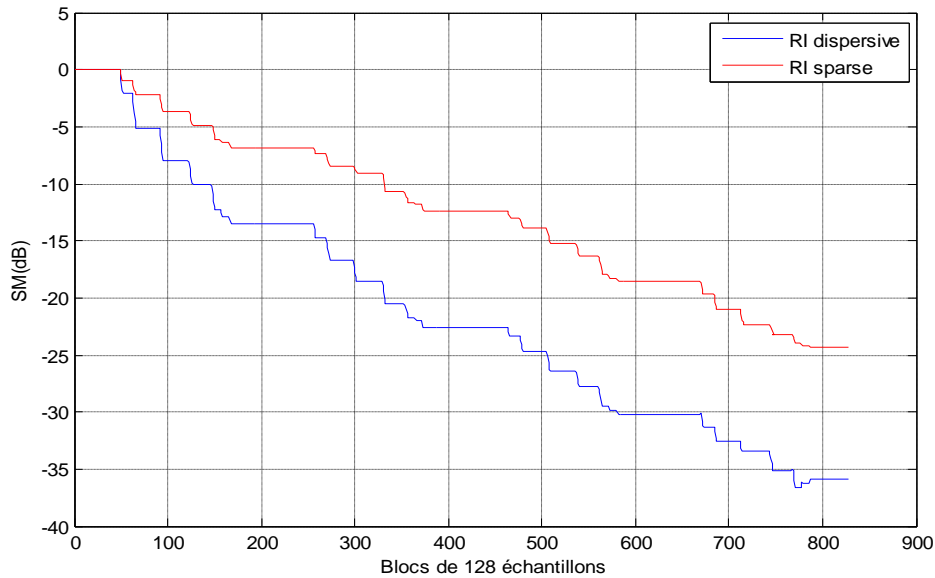
L'identification rentable des systèmes adaptatifs est un problème épineux en traitement de la parole, en particulier lorsque la réponse impulsionnelle acoustique est sparse et longue. Pour résoudre ce problème nous présentons des algorithmes de filtrage adaptatif proportionnels.

#### 3.2.1. Critère du désajustement

Le critère de désajustement est un critère robuste dans l'évaluation des performances. Ce critère se calcule à partir de la distance euclidienne entre les coefficients des réponses impulsionnelles réelles et mesurées (estimées). Ce critère, noté SM pour «System Mismatch», est défini par l'expression suivante [19]:

$$(SM)_{dB} = 10 \log_{10} \left( \frac{\|w_{21} - h_{21}\|^2}{\|h_{21}\|^2} \right) \quad (3.1)$$

Où  $h_{21}$  représente les coefficients du filtre réel, et  $w_{21}$  représente la valeur moyenne des coefficients estimés.



**Figure 3.1.** Convergence de l'algorithme BNLMS :  $\mu_{21n} = 0.9$ ,  $L = 128$

La figure 3.1 représente l'évaluation du système mismatch obtenue par les deux réponses impulsionnelles, on remarque que BNLMS montre une faible vitesse de convergence et ceci est bien prononcé pour les réponses impulsionnelles de type sparse.

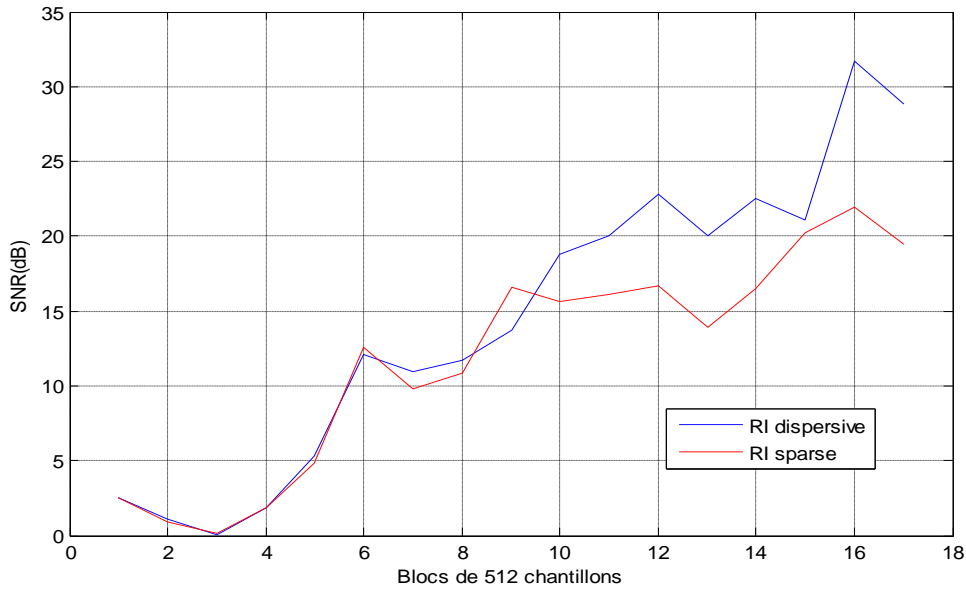
### 3.2.2. Rapport signal sur bruit (SNR)

Il existe plusieurs mesures objectives de la qualité de la parole. Le plus simple est le rapport signal sur bruit (SNR) qui compare les signaux de parole originaux et traités échantillon par échantillon.

Pour quantifier la performance de réduction de bruit des algorithmes proportionnels nous avons évalué SNR segmenté (SNRseg) donné par la formule suivante [17] :

$$SNR_{seg} = 10 \log_{10} \left( \frac{\sum_{n=1}^{N-1} |s(n)|^2}{\sum_{n=1}^{N-1} |s(n) - v(n)|^2} \right) \quad (3.2)$$

Où  $s(n)$ ,  $v(n)$ ,  $N$  sont respectivement le signal de parole originale, le signal de parole amélioré, et le paramètre de la valeur moyenne de la sortie du filtre.



**Figure 3.2.**  $SNR_{seg}$  de l’algorithme BNLMS pour les deux RI.

La figure 3.2 représente les résultats obtenus de l’évaluation du critère  $SNR_{seg}$  des deux réponses impulsionnelles dispersive et sparse.

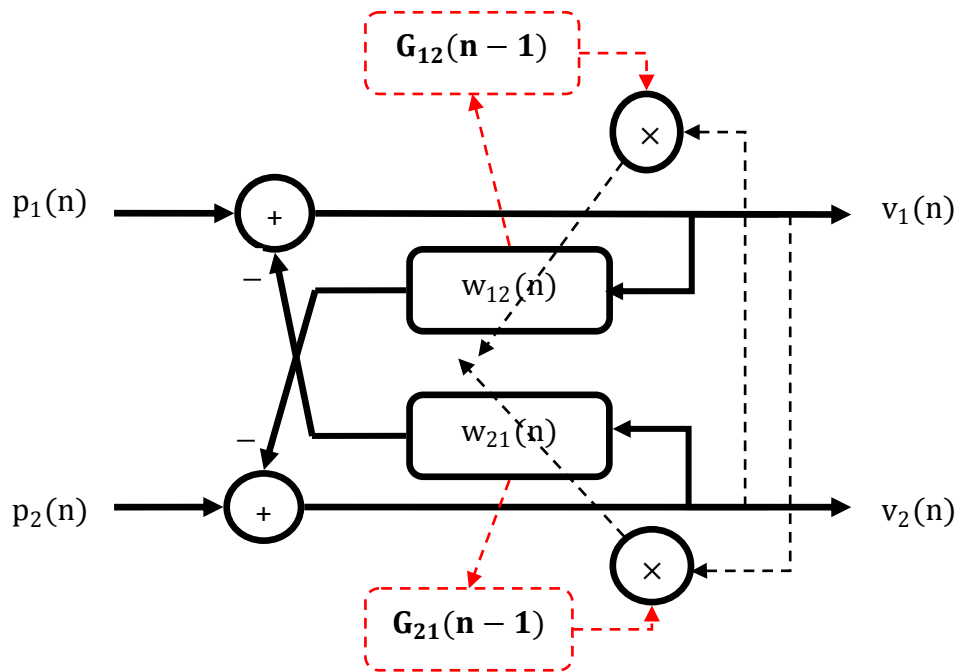
### 3.3. Algorithme Backward proportionnel proposée

La structure Backward est très utilisée pour la séparation de sources, d’après l’étude théorique que nous avons présentée dans le chapitre 02, nous avons remarqué bien que le signal de parole estimé égale exactement le signal de parole original sans distorsion. Pour cela, dans ce chapitre nous utilisons uniquement la structure Backward bi-capteurs employé au rehaussement de la parole.

#### 3.3.1. Structure générale des algorithmes Backward NLMS proportionnel

Plusieurs algorithmes ont été développés pour être utilisés dans le domaine d’identification des systèmes. L’algorithme BNLMS proportionnel présente une convergence très rapide dans un système sparse. Par contre lorsque la réponse impulsionnelle est dispersive, le PBNLMS converge lentement que le BNLMS. Cela implique que la règle proposée dans PBNLMS est loin d’être optimale.

La structure générale d'algorithmes proposés est donnée par la figure 3.3 :



**Figure 3.3.** Structure générale des algorithmes Backward NLMS proportionnel bi-captives.

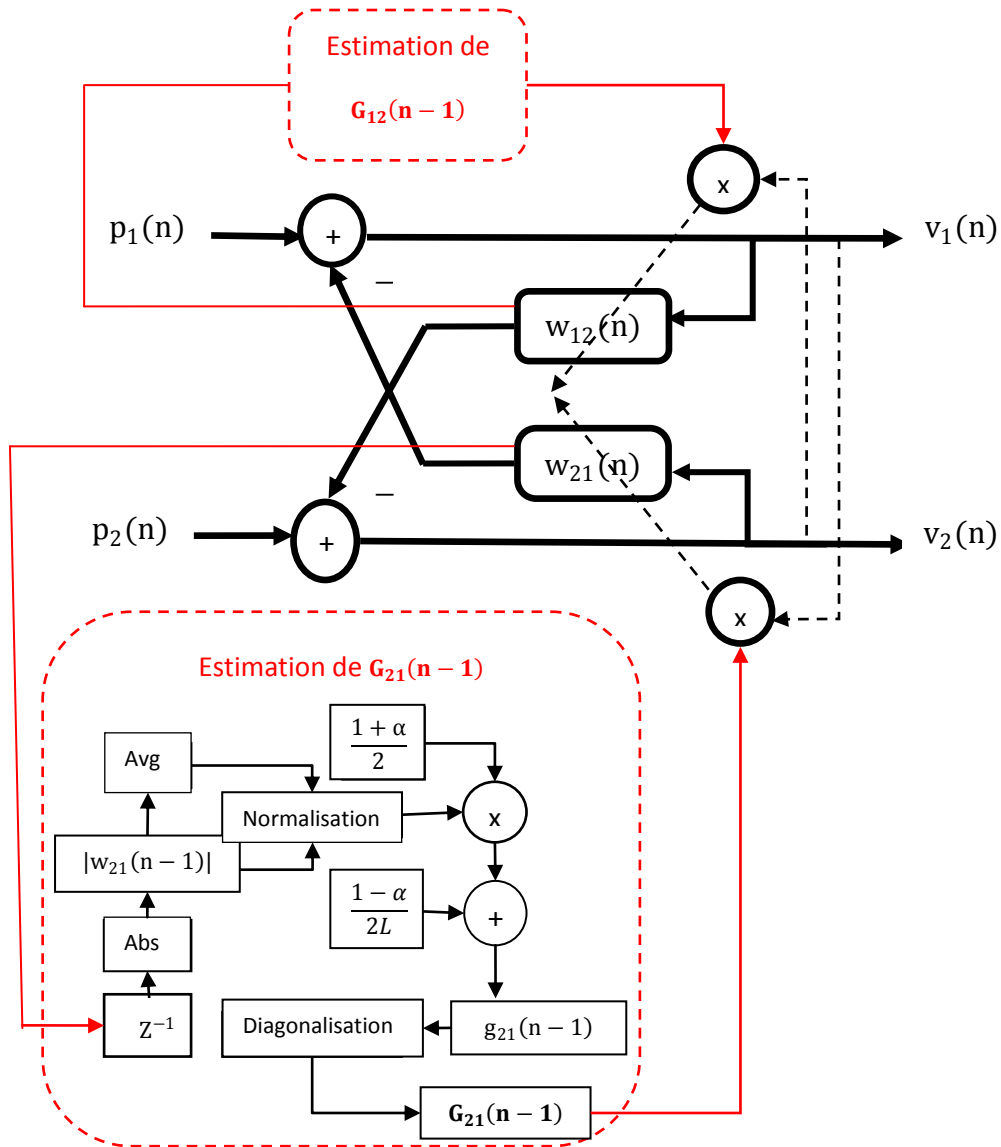
Nous proposons donc une nouvelle règle plus fiable que celle utilisée dans PBNLMS, quelle que soit la nature de la réponse impulsionnelle.

### 3.3.2 Algorithme proposé Backward NLMS proportionnel amélioré (IPBNLMS)

Une amélioration du PBNLMS (en anglais Improved PBNLMS), qui utilise une combinaison de technique de mise à jour proportionnelle (PBNLMS) et non proportionnelle (BNLMS), avec la signification relative de chacun est contrôlé par un facteur  $\alpha$ , ( $\alpha$  c'est le paramètre responsable de la convergence et de la qualité).

Le fait que PBNLMS soit plus lent que BNLMS avec des réponses impulsionnelles dispersives signifie que  $k_1(n)$  à être modifié intuitivement, le choix brutal (maximum) en  $k_1(n)$  entre  $w_1(n)$  et un autre nombre positif peut avoir un désastre effet sur la convergence de l'algorithme lorsque l'estimation des coefficients n'est pas précise [19].

La structure détaillée de cet algorithme IPBNLMS est donnée par la figure 3.4 :



**Figure 3.4.** Structure de l'algorithme Backward NLMS proportionnel amélioré bi-capturer (IPBNLMS).

On déduit l'algorithme IPBNLMS pour les deux capteurs :

$$\mathbf{w}_{12}(n) = [w_{12,1}(n), w_{12,2}(n), \dots, w_{12,L}(n)] \quad (3.3)$$

$$\mathbf{w}_{21}(n) = [w_{21,1}(n), w_{21,2}(n), \dots, w_{21,L}(n)] \quad (3.4)$$

Avec :

$$\mathbf{v}_2(n) = [v_2(n), v_2(n-1), v_2(n-L+1)] \quad (3.5)$$

$$\mathbf{v}_1(n) = [v_1(n), v_2(n-1), v_2(n-L+1)] \quad (3.6)$$

Pour  $n=1,2,\dots$ , (itérations)

$$\mathbf{v}_1(n) = \mathbf{p}_1(n) - \mathbf{w}_{21}(n) * \mathbf{v}_2^T(n) \quad (3.7)$$

$$\mathbf{v}_2(n) = \mathbf{p}_2(n) - \mathbf{w}_{12}(n) * \mathbf{v}_1^T(n) \quad (3.8)$$

La définition originale de l'élément diagonal de la matrice  $\mathbf{G}(n)$  dans l'algorithme IPBNLMS est décrite comme suit [03]:

$$\mathbf{G}_{12}(n) = \text{diag}\{g_{12,0}(n), g_{12,1}(n), \dots, g_{12,L-1}(n)\} \quad (3.9)$$

$$\mathbf{G}_{21}(n) = \text{diag}\{g_{21,0}(n), g_{21,1}(n), \dots, g_{21,L-1}(n)\} \quad (3.10)$$

Pour  $l=0,1,\dots,L-1$

$$g_{12_l} = \frac{k_{12_l}(n)}{\frac{1}{L} \sum_{i=0}^{L-1} k_{12_i}(n)} \quad (3.11)$$

$$g_{21_l} = \frac{k_{21_l}(n)}{\frac{1}{L} \sum_{i=0}^{L-1} k_{21_i}(n)} \quad (3.12)$$

$$k_{12_l}(n) = \frac{1-\alpha}{2L} + (1+\alpha) \frac{|w_{12,l}(n)|}{2 \sum_{i=0}^{L-1} |w_{12,i}(n)| + \delta_{\text{IPNLMS}}} \quad (3.13)$$

$$k_{21_l}(n) = \frac{1-\alpha}{2L} + (1+\alpha) \frac{|w_{21,l}(n)|}{2 \sum_{i=0}^{L-1} |w_{21,i}(n)| + \delta_{\text{IPNLMS}}} \quad (3.14)$$

Où  $\delta_{\text{IPNLMS}}$  est le paramètre de régularisation pour éviter de diviser par zéro, qui doit être pris comme suite [20] :

$$\delta_{\text{IPNLMS}} = \frac{(1-\alpha)}{2L} \delta_{\text{NLMS}}$$

Et :  $\alpha \in [-1, 1]$

Paramètre de pas :  $0 < \mu_{12n} < 2$  et  $0 < \mu_{21n} < 2$

La mise à jour de l'approche IPBNLMS est comme suit [21] :

$$\mathbf{w}_{12}(n) = \mathbf{w}_{12}(n-1) + \mu_{12n} \frac{\mathbf{v}_1^T(n) \mathbf{G}_{12}(n-1) \mathbf{v}_2(n)}{\mathbf{v}_1^T(n) \mathbf{G}_{12}(n-1) \mathbf{v}_1(n) + \delta_{\text{IPNLMS}}} \quad (3.15)$$

$$\mathbf{w}_{21}(n) = \mathbf{w}_{21}(n-1) + \mu_{21n} \frac{\mathbf{v}_2^T(n) \mathbf{G}_{21}(n-1) \mathbf{v}_1(n)}{\mathbf{v}_2^T(n) \mathbf{G}_{21}(n-1) \mathbf{v}_2(n) + \delta_{\text{IPNLMS}}} \quad (3.16)$$

Cela signifie que la même mise à jour sera faite pour tous les coefficients de filtre indépendamment de leurs grandeurs individuelles. Donc, pour cette valeur  $\alpha$  l'algorithme



IPBNLMS fonctionne comme BNLMS pour  $\alpha = -1$  et pour  $\alpha = 1$  se comporte comme PBNLMS [17].

### 3.3.3 Vérification de la règle optimale du nouvel algorithme IPBNLMS proposé

Il est évident qu'une règle optimale devrait mieux exploiter la forme du trajet d'écho estimé et que l'algorithme devrait toujours avoir de meilleures performances (vitesse de convergence) que BNLMS avec des réponses impulsionnelles non dispersives et des performances similaires avec des réponses impulsives très dispersives, En d'autres termes, le chemin d'écho n'a pas besoin d'être clairsemé pour pouvoir accélérer le taux de convergence de l'algorithme.[20]

➤ **1<sup>er</sup> Cas  $\alpha = 1$  :**

$$\delta_{\text{PNLMS}} = \frac{(1 - \alpha)}{2L} \delta_{\text{NLMS}} , \quad \rho = \frac{5}{L} , \delta_p = 0.01 , 0 < \mu_{12n} < 2 , \quad 0 < \mu_{21n} < 2$$

**1<sup>er</sup> Capteur :**

$$k_{21_1}(n) = \frac{1 - \alpha}{2L} + (1 + \alpha) \frac{|w_{21,l}(n)|}{2 \sum_{i=0}^{L-1} |w_{21,i}(n)| + \delta_{\text{IPNLMS}}}$$

$$k_{21_1}(n) = \frac{|w_{21,l}(n)|}{\sum_{i=0}^{L-1} |w_{21,i}(n)| + \delta_{\text{PNLMS}}}$$

$$k_{21_1}(n) = \frac{|w_{21,l}(n)|}{\sum_{i=0}^{L-1} |w_{21,i}(n)|}$$

**Pseudo code de l'algorithme PBNLMS pour 1<sup>er</sup> capteur : [21]**

$$\mathbf{w}_{21}(n) = [w_{21,1}(n), w_{21,2}(n), \dots, w_{21,L}(n)]$$

**Mise à jour**

Pour  $n=1,2,\dots$ , (itérations)

$$\mathbf{v}_1(n) = p_1(n) - \mathbf{w}_{21}(n) * \mathbf{v}_2^T(n)$$

$$\mathbf{G}_{21}(n) = \text{diag}\{g_{21,0}(n), g_{21,1}(n), \dots, g_{21,L-1}(n)\}$$

$$g_{21,l}(n) = \frac{k_{21,l}(n)}{\frac{1}{L} \sum_{i=0}^{L-1} |k_{21,i}(n)|}, l=0,1,\dots,L-1$$

$$k_{21,l}(n) = \max\{|w_{21,l}(n)|, \rho \max\{\delta_p, |w_{21,0}(n)|, \dots, |w_{21,L-1}(n)|\}\}$$

$$\mathbf{w}_{21}(n) = \mathbf{w}_{21}(n-1) + \mu_{21n} \frac{\mathbf{v}_2^T(n) \mathbf{G}_{21}(n-1) \mathbf{v}_1(n)}{\mathbf{v}_2^T(n) \mathbf{G}_{21}(n-1) \mathbf{v}_2(n) + \delta_{\text{PNLMS}}}$$

**Tableau 3.1.** Pseudo code de l'algorithme PBNLMS capteur 01.

**2<sup>eme</sup> Capteur :**

$$k_{12,l}(n) = \frac{1-\alpha}{2L} + (1+\alpha) \frac{|w_{12,l}(n)|}{2 \sum_{i=0}^{L-1} |w_{12,i}(n)| + \delta_{\text{IPNLMS}}}$$

$$k_{12,l}(n) = \frac{|w_{12,l}(n)|}{\sum_{i=0}^{L-1} |w_{12,i}(n)| + \delta_{\text{PNLMS}}}$$

$$k_{12,l}(n) = \frac{|w_{12,l}(n)|}{\sum_{i=0}^{L-1} |w_{12,i}(n)|}$$

**Pseudo code de l'algorithme PBNLMS pour 2<sup>eme</sup> capteur : [21]**

$$\mathbf{w}_{12}(n) = [w_{12,1}(n), w_{12,2}(n), \dots, w_{12,L}(n)]$$

**Mise à jour**

Pour  $n=1,2,\dots$ , (itérations)

$$\mathbf{v}_2(n) = p_2(n) - \mathbf{w}_{12}(n) * \mathbf{v}_1^T(n)$$

$$\mathbf{G}_{12}(n) = \text{diag}\{g_{12,0}(n), g_{12,1}(n), \dots, g_{12,L-1}(n)\}$$

$$g_{12,l}(n) = \frac{k_{12,l}(n)}{\frac{1}{L} \sum_{i=0}^{L-1} |k_{12,i}(n)|}, l=0,1,\dots,L-1$$

$$k_{12,l}(n) = \max\{|w_{12,l}(n)|, \rho \max\{\delta_p, |w_{12,0}(n)|, \dots, |w_{12,L-1}(n)|\}\}$$

$$\mathbf{w}_{12}(n) = \mathbf{w}_{12}(n-1) + \mu_{12n} \frac{\mathbf{v}_1^T(n) \mathbf{G}_{12}(n-1) \mathbf{v}_2(n)}{\mathbf{v}_1^T(n) \mathbf{G}_{12}(n-1) \mathbf{v}_1(n) + \delta_{\text{PNLMS}}}$$

**Tableau 3.2.** Pseudo code de l'algorithme PBNLMS capteur 02.

➤ **2<sup>eme</sup> Cas  $\alpha = -1$  :**

$$0 < \mu_{12n} < 2, 0 < \mu_{21n} < 2$$

**1<sup>er</sup> Capteur :**

$$k_{21_1}(n) = \frac{1 - \alpha}{2L} + (1 + \alpha) \frac{|w_{21,1}(n)|}{2 \sum_{i=0}^{L-1} |w_{21,i}(n)| + \delta_{IPNLMS}}$$

$$k_{21_1}(n) = \frac{1}{L}$$

**Pseudo code de l'algorithme BNLMS pour 1<sup>er</sup> capteur : [20]**

$$\mathbf{w}_{21}(n) = [w_{21,1}(n), w_{21,2}(n), \dots, w_{21,L}(n)]$$

**Mise à jour**

Pour  $n=1,2,\dots$ , (itérations)

$$\mathbf{v}_1(n) = \mathbf{p}_1(n) - \mathbf{w}_{21}(n) * \mathbf{v}_2^T(n)$$

$$\mathbf{w}_{21}(n) = \mathbf{w}_{21}(n-1) + \mu_{21n} \frac{\mathbf{v}_2^T(n) \mathbf{v}_1(n)}{\mathbf{v}_2^T(n) \mathbf{v}_2(n) + \delta_{NLMS}}$$

**Tableau 3.3.** Pseudo code de l'algorithme BNLMS capteur 01.

**2<sup>eme</sup> Capteur :**

$$k_{12_1}(n) = \frac{1 - \alpha}{2L} + (1 + \alpha) \frac{|w_{12,1}(n)|}{2 \sum_{i=0}^{L-1} |w_{12,i}(n)| + \delta_{IPNLMS}}$$

$$k_{12_1}(n) = \frac{1}{L}$$

**Pseudo code de l'algorithme BNLMS pour 2<sup>eme</sup> capteur : [20]**

$$\mathbf{w}_{12}(n) = [w_{12,1}(n), w_{12,2}(n), \dots, w_{12,L}(n)]$$

**Mise à jour**

Pour  $n=1,2,\dots$ , (itérations)

$$\mathbf{v}_2(n) = \mathbf{p}_2(n) - \mathbf{w}_{12}(n) * \mathbf{v}_1^T(n)$$

$$\mathbf{w}_{12}(n) = \mathbf{w}_{12}(n-1) + \mu_{12n} \frac{\mathbf{v}_1^T(n) \mathbf{v}_2(n)}{\mathbf{v}_1^T(n) \mathbf{v}_1(n) + \delta_{NLMS}}$$

**Tableau 3.4.** Pseudo code de l'algorithme BNLMS capteur 02.

Il a été démontré que quelle que soit la réponse impulsionnelle nature, l'algorithme IPBNLMS présente une convergence plus rapide que BNLMS et PBNLMS.

### 3.4. Complexité algorithmique

Des nombreux facteurs contribuent à la complexité d'un algorithme, en termes de nombre total d'additions, multiplications, divisions et logarithmes par itération est évaluée à partir le tableau.

Algorithme	BNLMS	PBNLMS	IPBNLMS
Addition	$2L+1$	$3L+1$	$4L+2$
Multiplication	$3L+1$	$6L+4$	$6L+4$
Division	1	2	2

**Tableau 3.5.** Complexité algorithmique des algorithmes.

### 3.5. Conclusion

Dans ce chapitre nous avons présenté un système de suppression de bruit basé sur une structure de séparation de source aveugle Backward, puis nous avons également proposé un nouveau modèle d'algorithme proportionnel amélioré pour la réduction du bruit dans un système sparse.

Le chapitre suivant va présenter les résultats de simulation du nouvel algorithme IPBNLMS qui surpasse les algorithmes BNLMS, PBNLMS pour le système sparse et le système dispersif.

# Chapitre 4 Résultats et Simulations

---

## 4.1. Introduction

Dans ce chapitre ; nous présentons les résultats de simulations des algorithmes BNLMS et IPBNLMS dans le domaine de rehaussement de la parole.

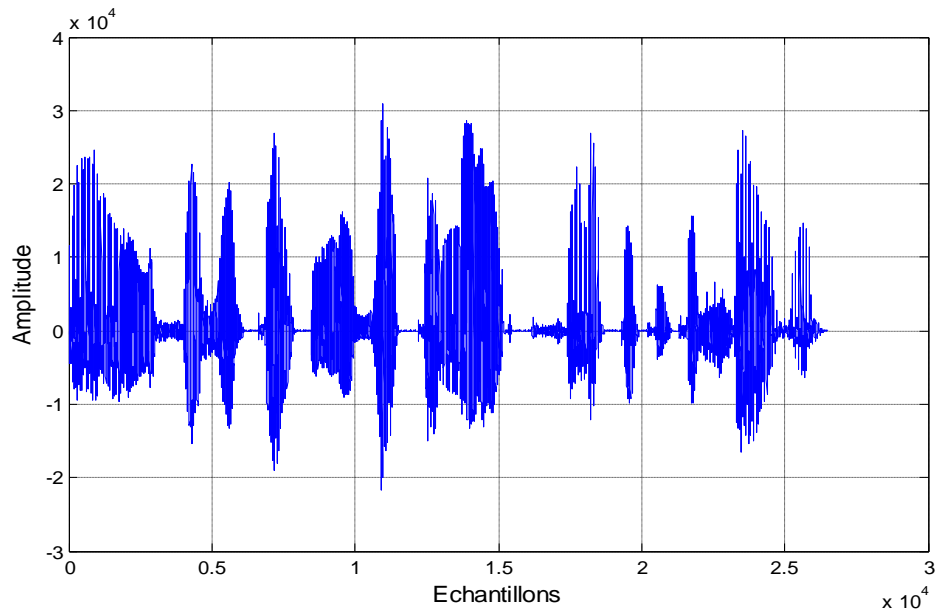
En commençant par la présentation des signaux utilisés tels que le signal de parole ; le bruit et les réponses impulsionnelles. Ensuite , nous représentons les différents résultats de simulations des algorithmes, et en terminant par une comparaison.

Nous présentons également l'influences des paramètres (la taille du filtre  $L$  ; le pas d'adaptation  $\mu$  ; le paramètre  $\alpha$ ) et on fera une comparaison entre les deux algorithmes en utilisant les deux critères rapport signal sur bruit SNR et le critère de désajustement SM.

## 4.2. Présentation des signaux de simulation

### 4.2.1. Signal de parole

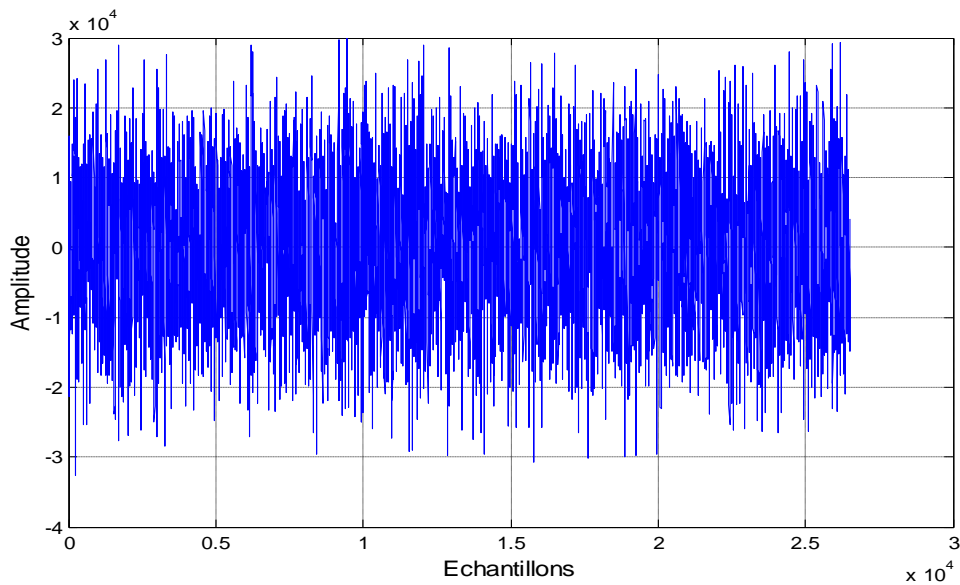
Le signal de parole utilisé dans ce travail est un signal prononcé par un locuteur masculin, échantillonné avec une fréquence d'échantillonnage de 8 kHz et d'environ 4 secondes le scénario de ce signal est : « Un loup s'est jeté immédiatement sur la petite chèvre » ; il est présenté par la figure 4.1 ci-dessous : [16]



**Figure 4.1.** Signal de parole.

#### 4.2.2. Bruit USASI

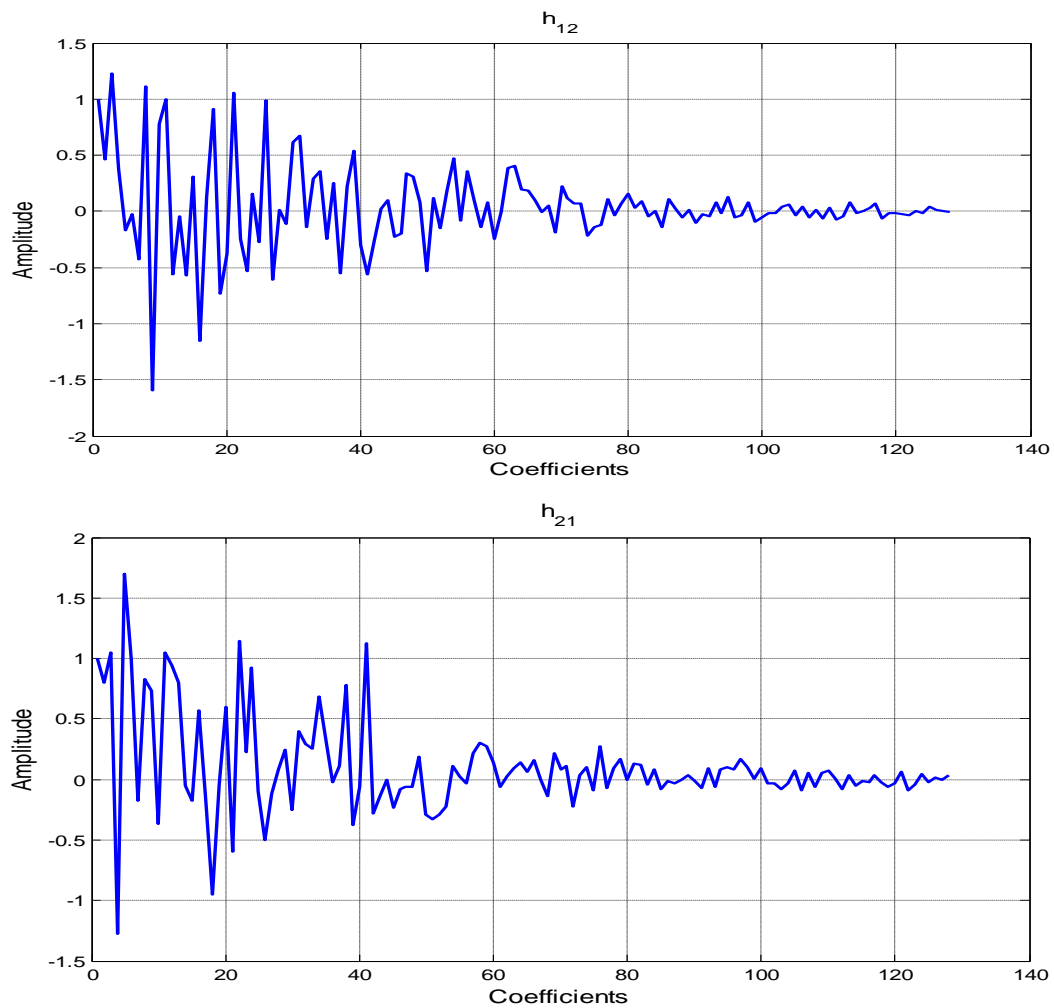
Le bruit USASI est un processus aléatoire stationnaire dont les échantillons sont des Variables aléatoires corrélées. Sa densité spectrale de puissance caractérise le spectre moyen du signal de la parole. il sert à vérifier la convergence de l'algorithme utilisé. Nous avons choisi ce type de bruit car il correspond à un standard reconnu dans les tests acoustiques , Il est présenté dans la figure 4.2 .



**Figure 4.2.** Bruit USASI.

### 4.2.3. Réponses impulsionnelles

Dans ce travail ; nous avons utilisé deux exemples de réponses impulsionnelles sparse et dispersive; Les deux figures 4.3 et 4.4 montre les RI dispersive et sparse respectivement avec une taille de  $L=128$  :



**Figure 4.3.** Réponses impulsionnelles dispersives.

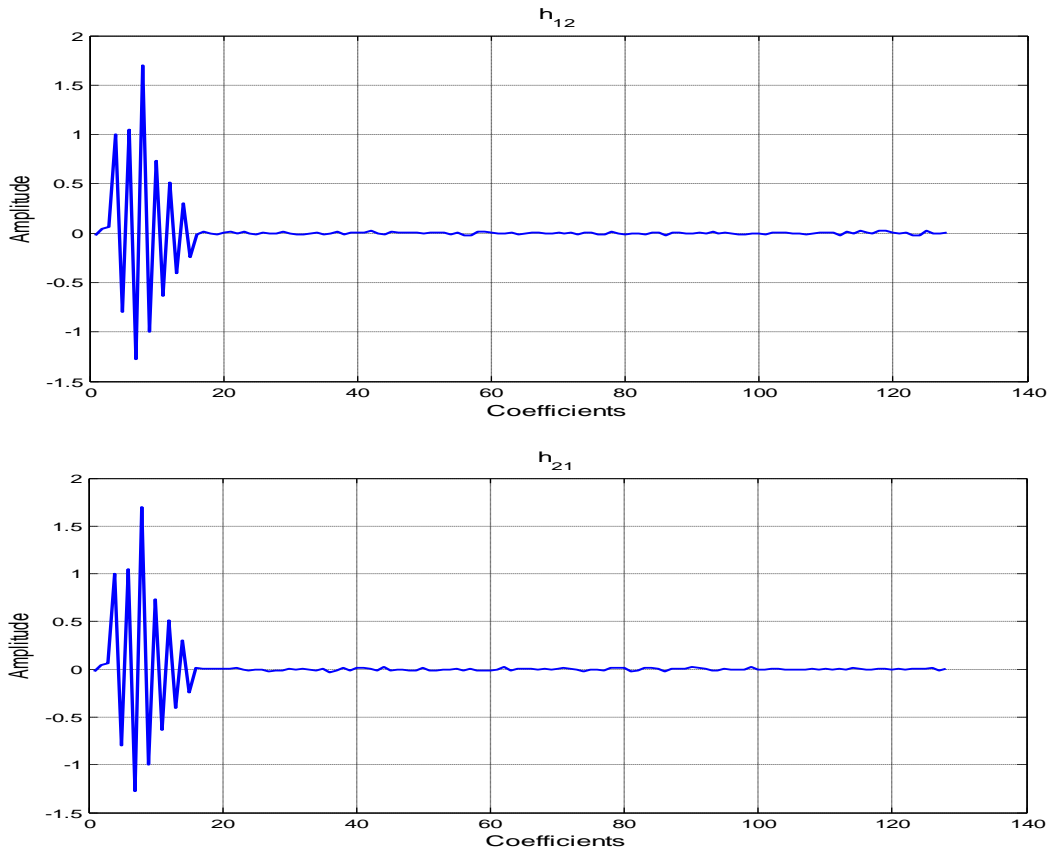


Figure 4.4. Réponses impulsionnelles sparse.

### 4.3. Détection d'activité vocale (DAV)

Dans notre travail, nous avons utilisé une segmentation manuelle, La fonction du DAV (Détection d'Activité Vocale) permet de distinguer entre le bruit en présence de la parole et le bruit sans la parole. La figure 4.5 représente la segmentation de signal de parole :

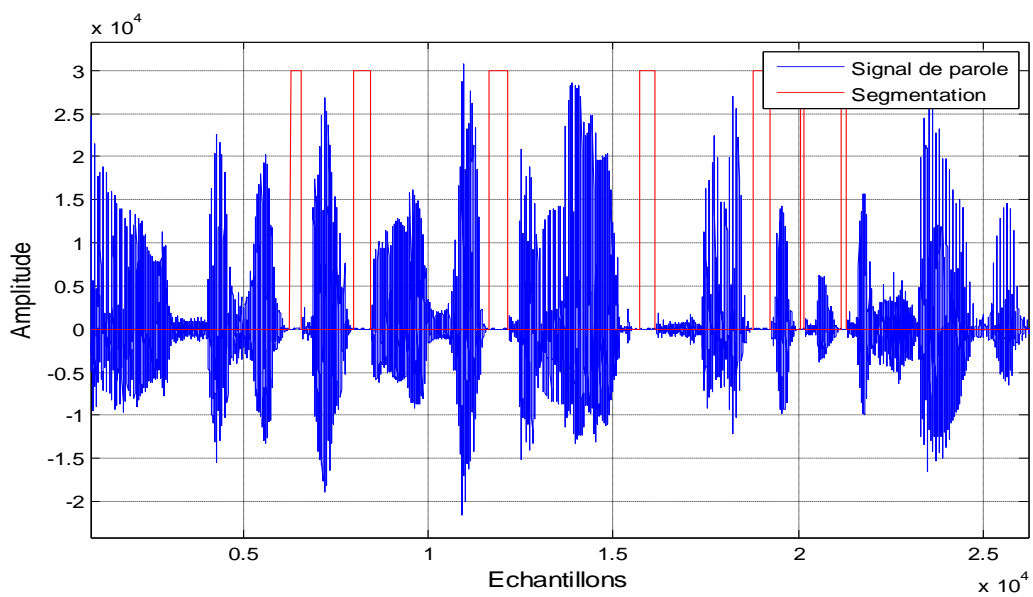


Figure 4.5. Segmentation de signal de parole.



#### 4.4. Résultats de l'algorithme BNLS

Dans cette partie nous avons testé l'algorithme BNLS pour les deux RI et nous avons tracé le signal de mélange  $p_1(n)$  pour la comparaison avec le signal de sortie  $v_1(n)$ .

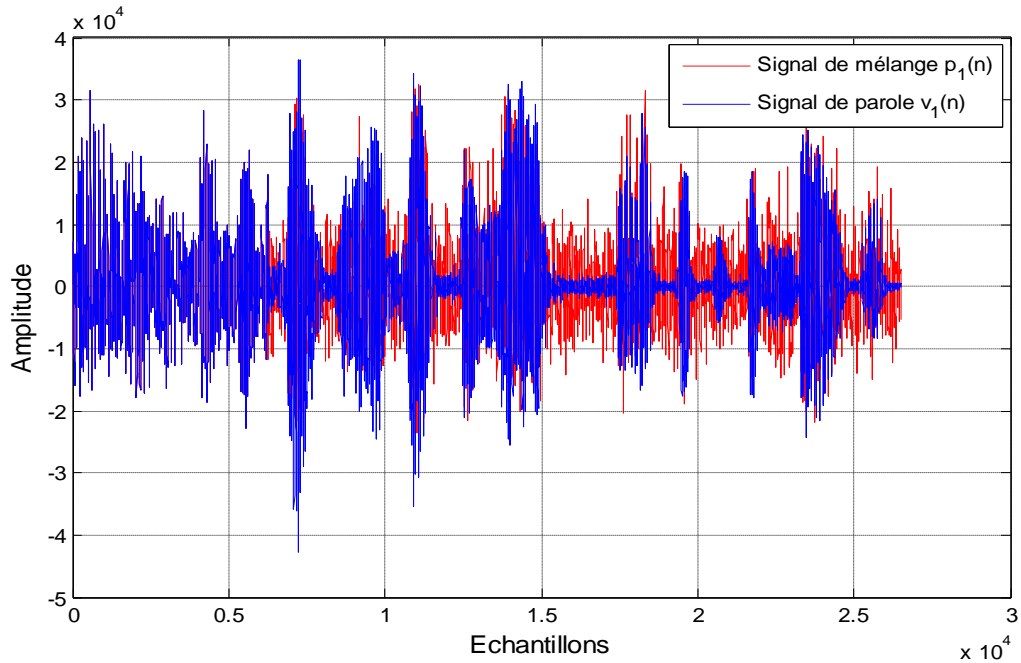


Figure 4.6. Résultat pour une réponse dispersive :  $\mu_{21n} = 0.9$ ,  $L = 128$

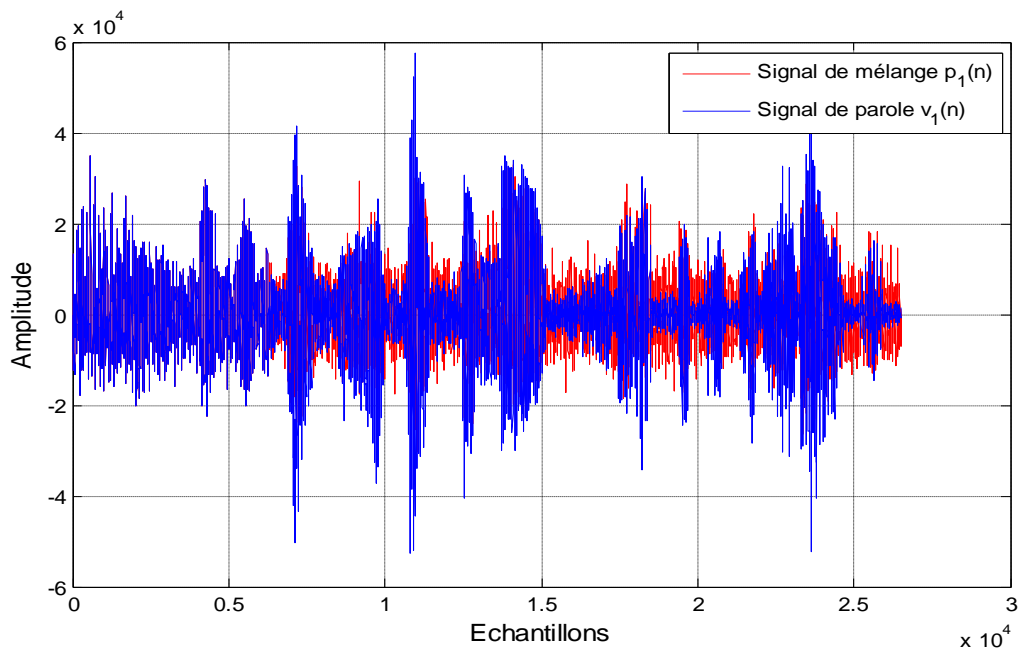
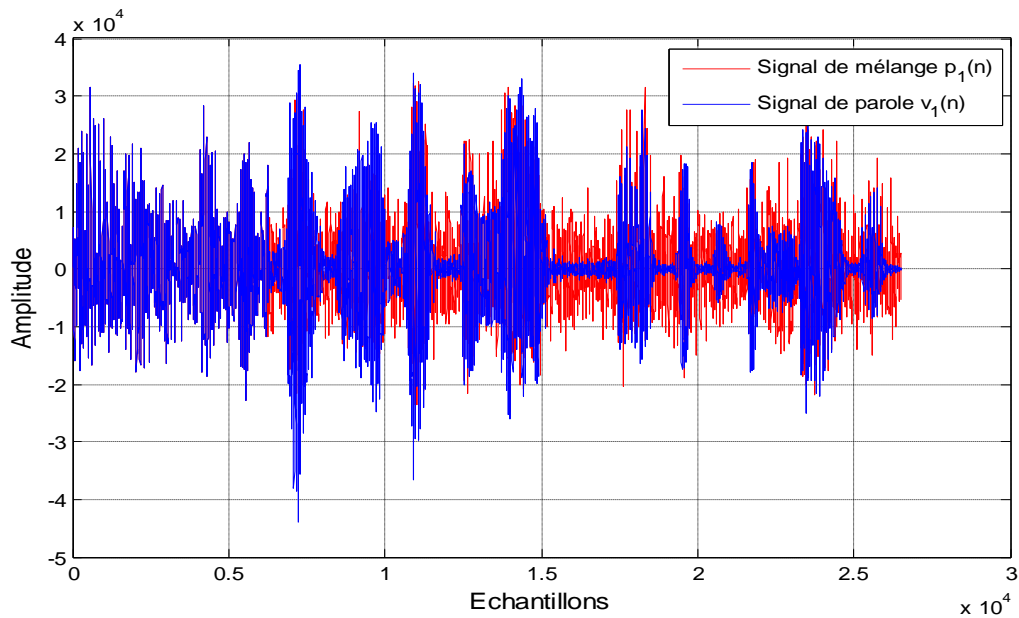


Figure 4.7. Résultat pour une réponse sparse :  $\mu_{21n} = 0.9$ ,  $L = 128$

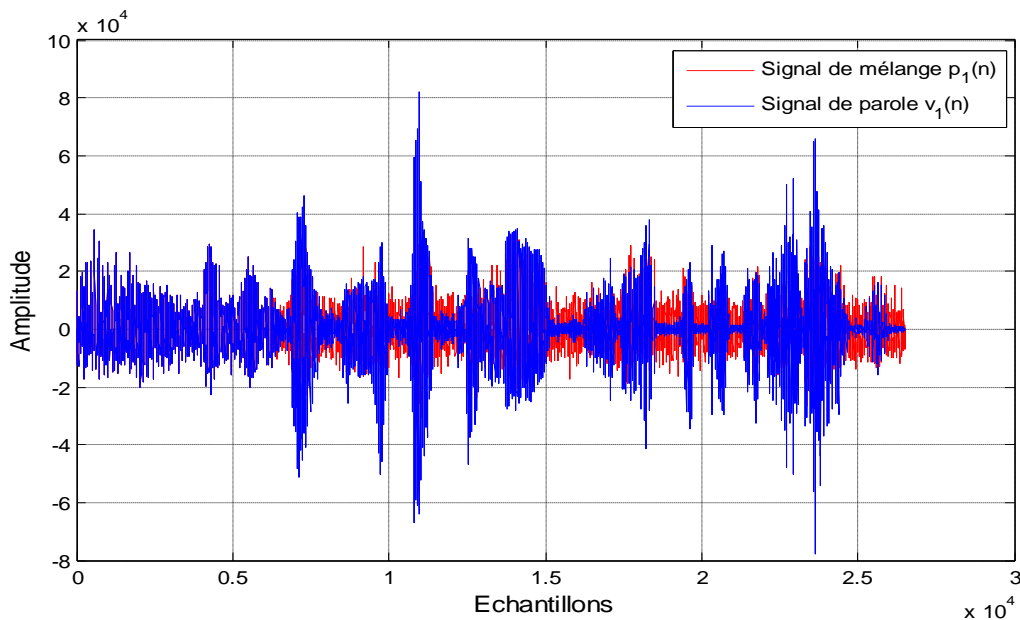
Pour une réponse impulsionnelle sparse ; on voit des périodes d'activités de bruit plus visible.

## 4.5. Résultats de l'algorithme IPBNLMS

Dans cette partie nous avons testé l'algorithme IPBNLMS pour les deux RI et nous avons tracé le signal de mélange  $p_1(n)$  pour la comparaison avec le signal de sortie  $v_1(n)$ .



**Figure 4.8.** Résultat pour une réponse dispersive :  $\mu_{21n} = 0.9$ ,  $L = 128$



**Figure 4.9.** Résultat pour une réponse sparse :  $\mu_{21n} = 0.9$ ,  $L = 128$

Résultats de simulation montrent que l'algorithme IPBNLMS proposé est plus performant que l'algorithme BNLS dans un mélange convolutif sparse. Pour confirmer la supériorité de

l'algorithme proposé, nous avons fait d'autres simulations comparatives basant sur deux critères (SM et  $SNR_{Seg}$ ).

#### 4.5.1. Effet du paramètre $\alpha$ sur l'algorithme IPBNLMS

Nous avons testés l'effet du paramètre  $\alpha$  sur l'algorithme IPBNLMS, on prend  $L=1024$  et

$\mu_{21n} = 0.9$  :

Pour  $\alpha = 1$  :

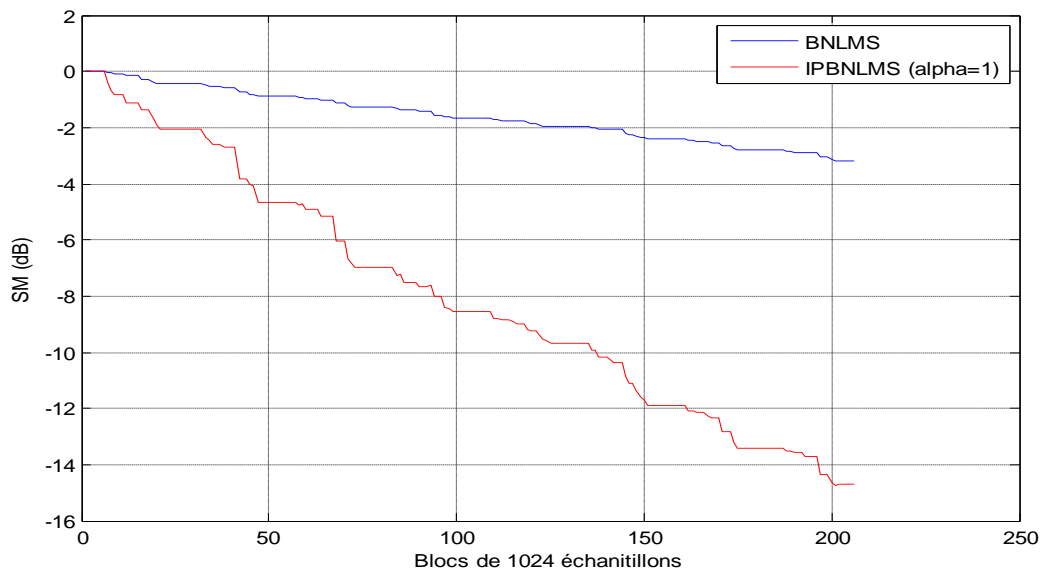


Figure 4.10. SM pour  $L=1024$ ,  $\mu_{21n} = 0.9$ ,  $\alpha = 1$ , ( $\times 8$ )

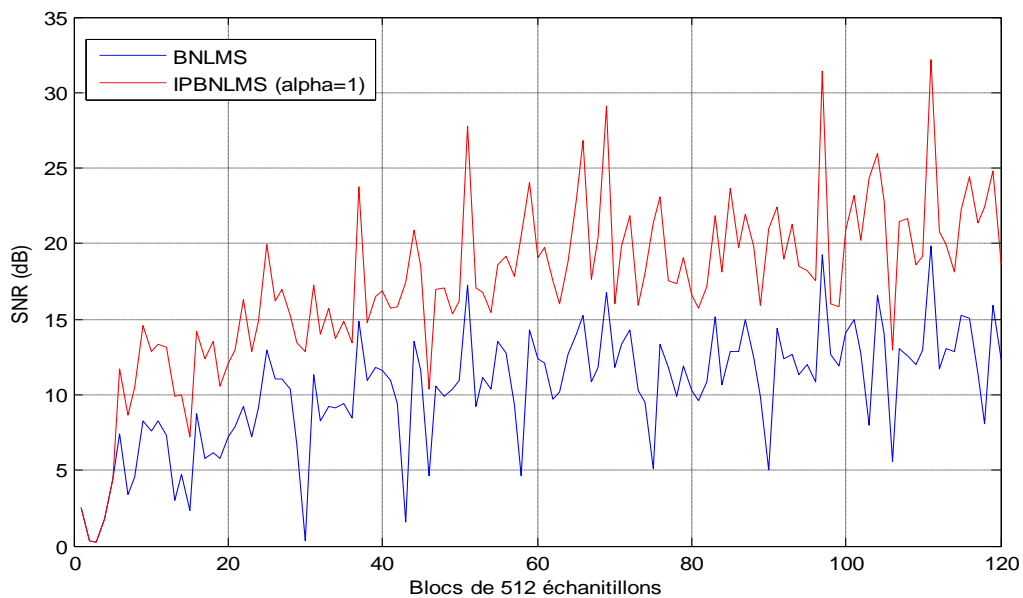
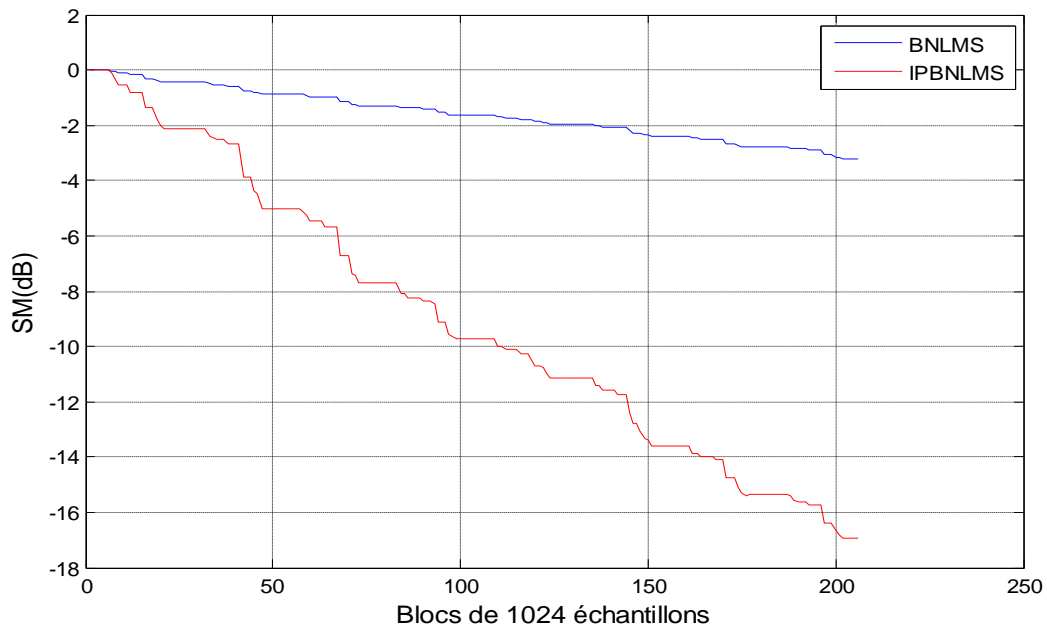


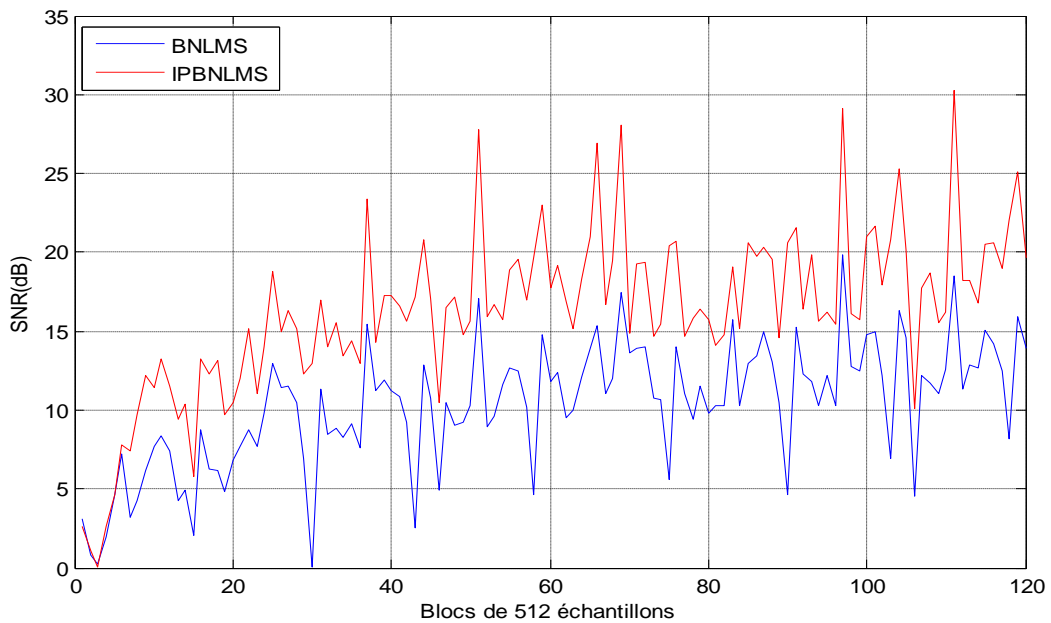
Figure 4.11. SNR pour  $L=1024$ ,  $\mu_{21n} = 0.9$ ,  $\alpha = 1$ , ( $\times 8$ )

D'après la figure 4.10 et la figure 4.11 on remarque que lorsque  $\alpha = 1$  l'algorithme IPBNLMS se comporte comme PBNLMS.

Pour  $\alpha = -0.5$  :



**Figure 4.12.** SM pour  $L=1024$ ,  $\mu_{21n} = 0.9$ ,  $\alpha = -0.5$ , ( $\times 8$ )



**Figure 4.13.** SNR pour  $L=1024$ ,  $\mu_{21n} = 0.9$ ,  $\alpha = -0.5$ , ( $\times 8$ )

D'après la figure 4.12 et la figure 4.13 on remarque que le favorable choix qui donne des meilleurs résultats pour l'algorithme IPBNLMS est  $\alpha = -0.5$ .

Pour  $\alpha = -1$  :

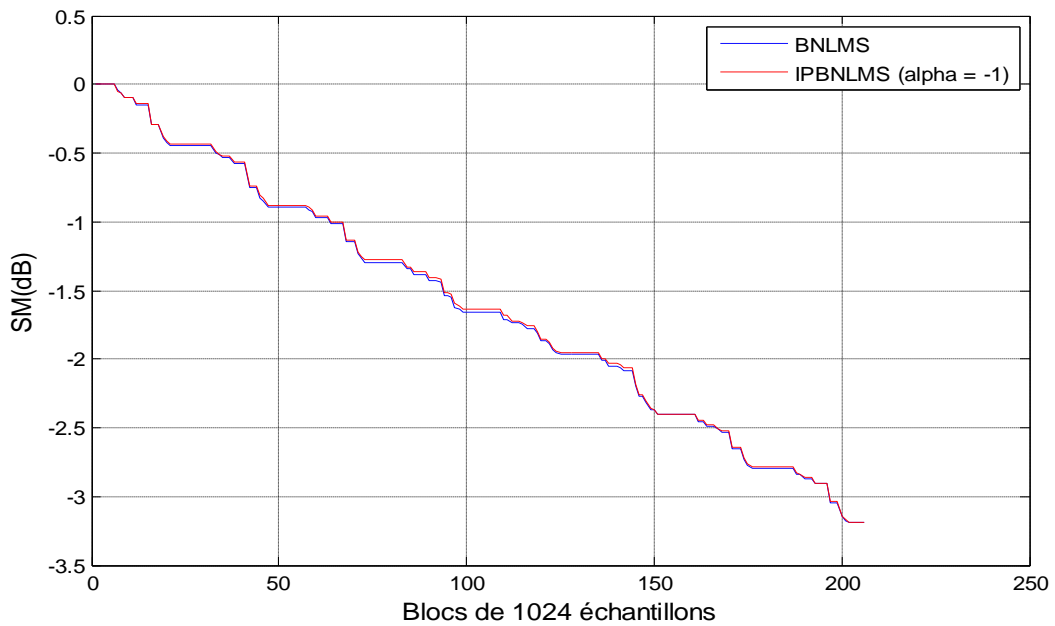


Figure 4.14. SM pour  $L=1024$ ,  $\mu_{21n} = 0.9$ ,  $\alpha = -1$ , ( $\times 8$ )

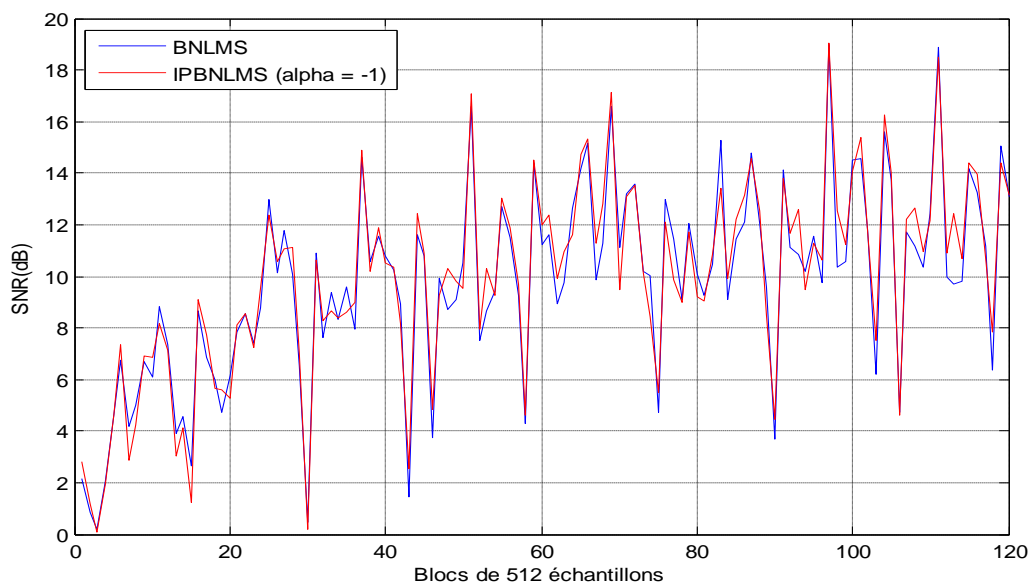


Figure 4.15. SNR pour  $L=1024$ ,  $\mu_{21n} = 0.9$ ,  $\alpha = -1$ , ( $\times 8$ )

D'après la figure 4.14 et la figure 4.15 on remarque que Lorsque  $\alpha = -1$  l'algorithme IPBNLMS est équivalent à BNLMS.

#### 4.5.2. Signaux de parole estimés

Les résultats de simulation obtenus avec un bruit USASI et pour une réponse impulsionnelle sparse avec  $L=128$  et  $\mu_{21n} = 0.9$  :

### Pour l'algorithme BNLMS

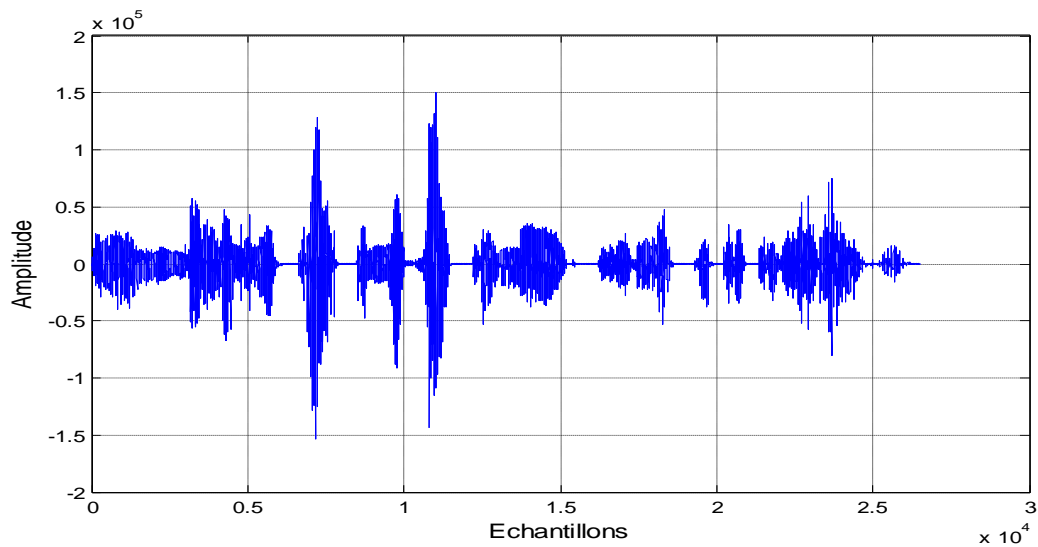


Figure 4.16. Signal de parole estimé :  $L=128$ ,  $\mu_{21n} = 0.9$ , ( $\times 4$ )

### Pour l'algorithme IPBNLMS

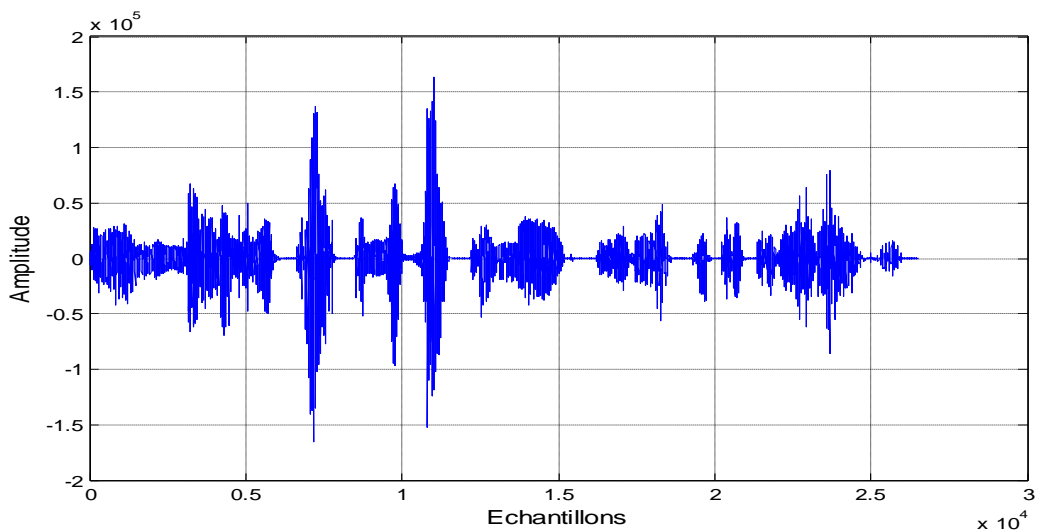


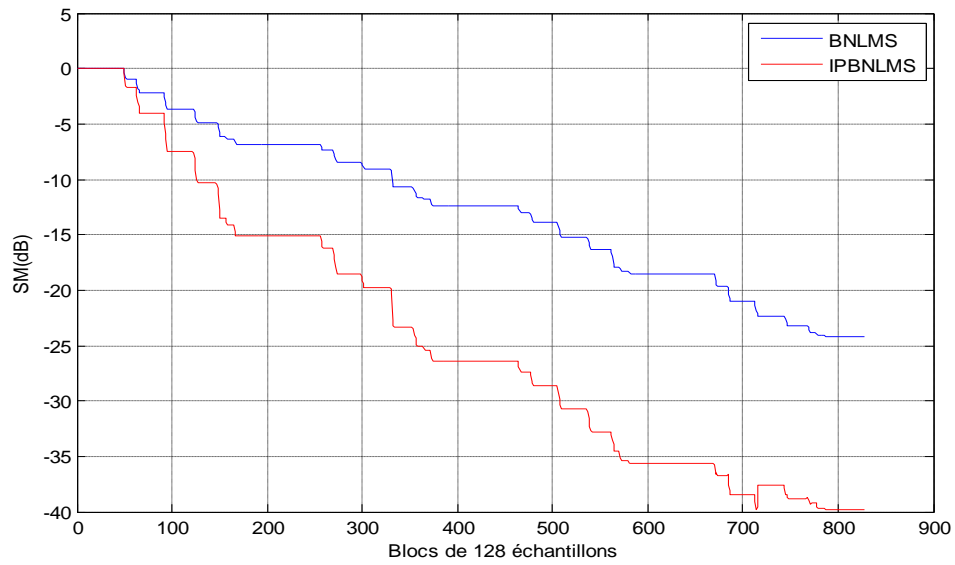
Figure 4.17. Signal de parole estimé :  $L=128$ ,  $\mu_{21n} = 0.9$ , ( $\times 4$ )

On remarque dans la figure 4.17 que l'algorithme proportionnel IPBNLMS donne un bon résultat par rapport à l'algorithme BNLMS.

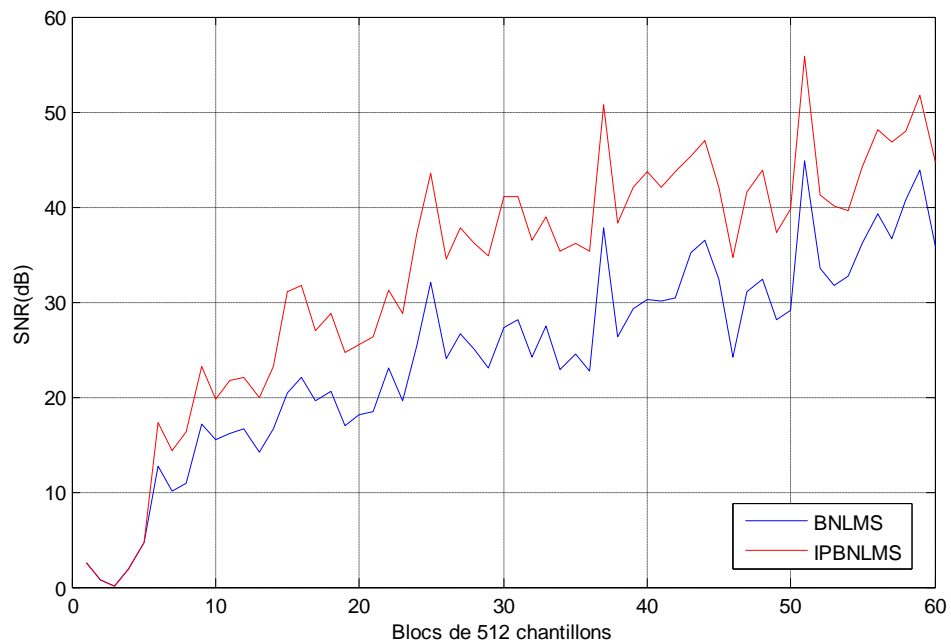
### 4.5.3. Effet de la taille des filtres

Nous avons testés l'effet de la taille des filtres sur les algorithmes BNLMS et IPBNLMS ; on prend  $L=128$ ,  $L=256$ ,  $L=512$ ,  $L=1024$  et  $L=2048$  :

Pour  $L=128$  :



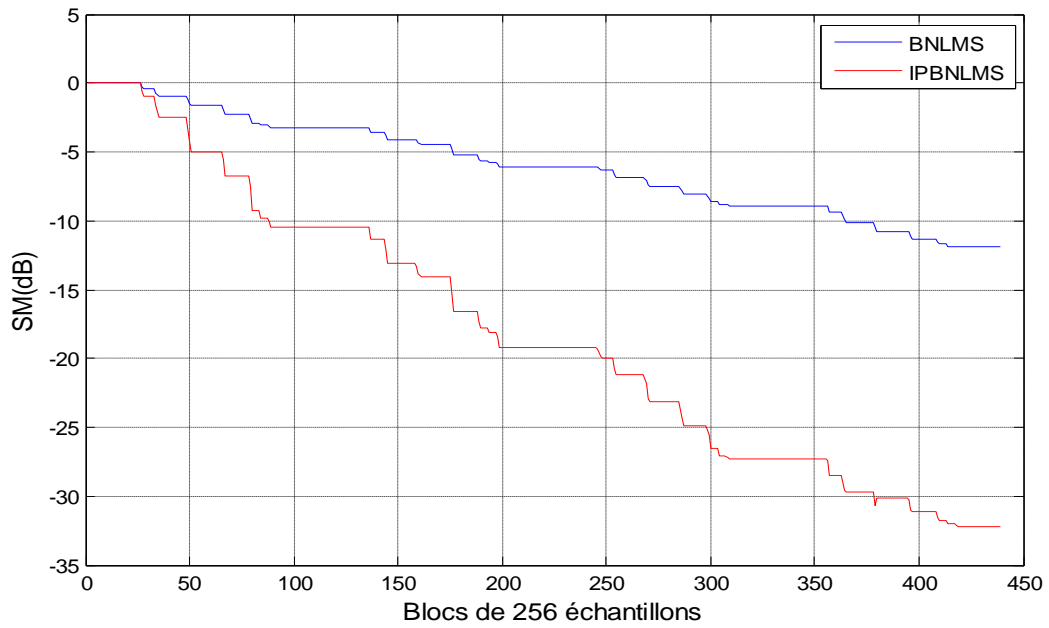
**Figure 4.18.** SM pour  $L=128$ ,  $\mu_{21n} = 0.9$ , ( $\times 4$ )



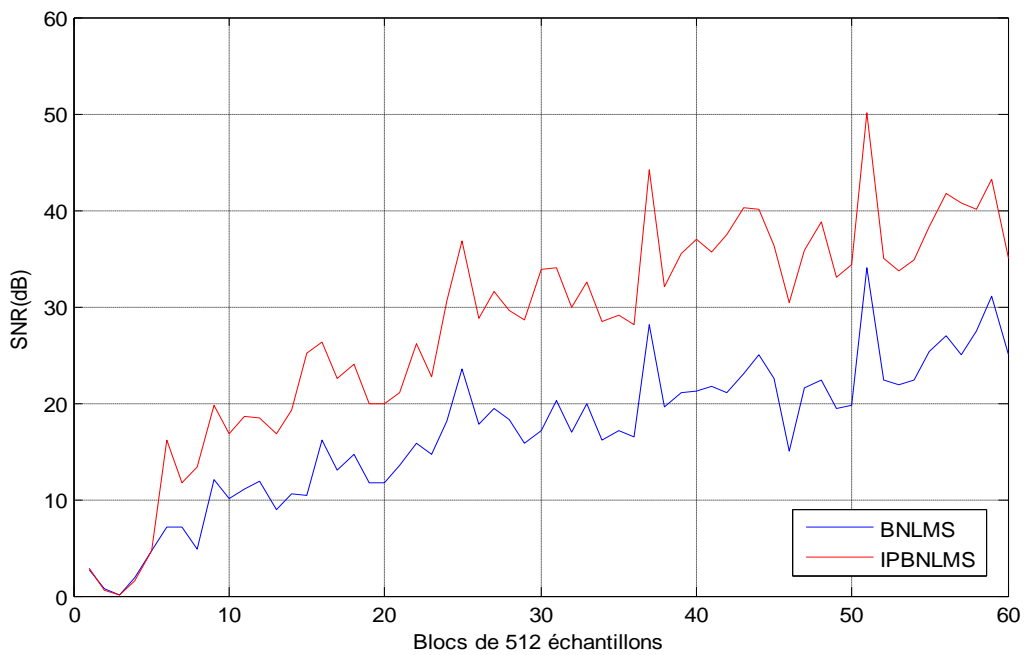
**Figure 4.19.** SNR pour  $L=128$ ,  $\mu_{21n} = 0.9$ , ( $\times 4$ )

D'après les figures 4.18 et 4.19 l'algorithme proportionnel IPBNLMS converge très rapidement par rapport à l'algorithme BNLMS ; et la qualité du signal de la parole estimé est dégradée par rapport à l'algorithme IPBNLMS.

Pour  $L=256$  :



**Figure 4.20.** SM pour  $L=256$ ,  $\mu_{21n} = 0.9$ , ( $\times 4$ )



**Figure 4.21.** SNR pour  $L=256$ ,  $\mu_{21n} = 0.9$ , ( $\times 4$ )

Lorsqu'on augmente la taille des filtres à  $L=256$  ; on remarque que la vitesse de convergence et la précision de l'algorithme BNLMS ont été dégradées.



Pour L=512 :

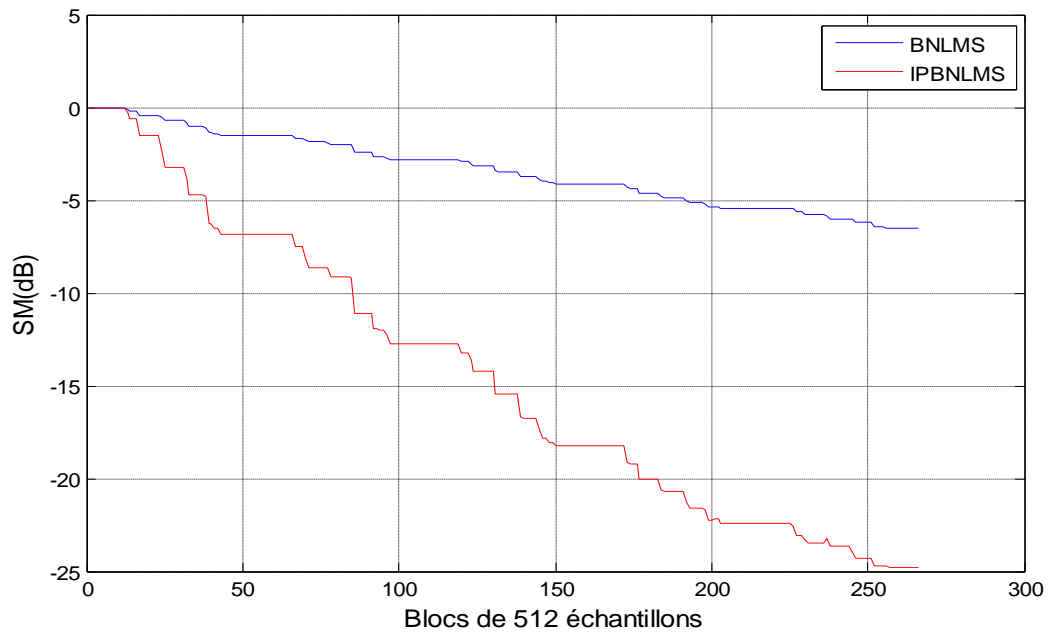


Figure 4.22. SM pour L=512,  $\mu_{21n} = 0.9$ , ( $\times 5$ )

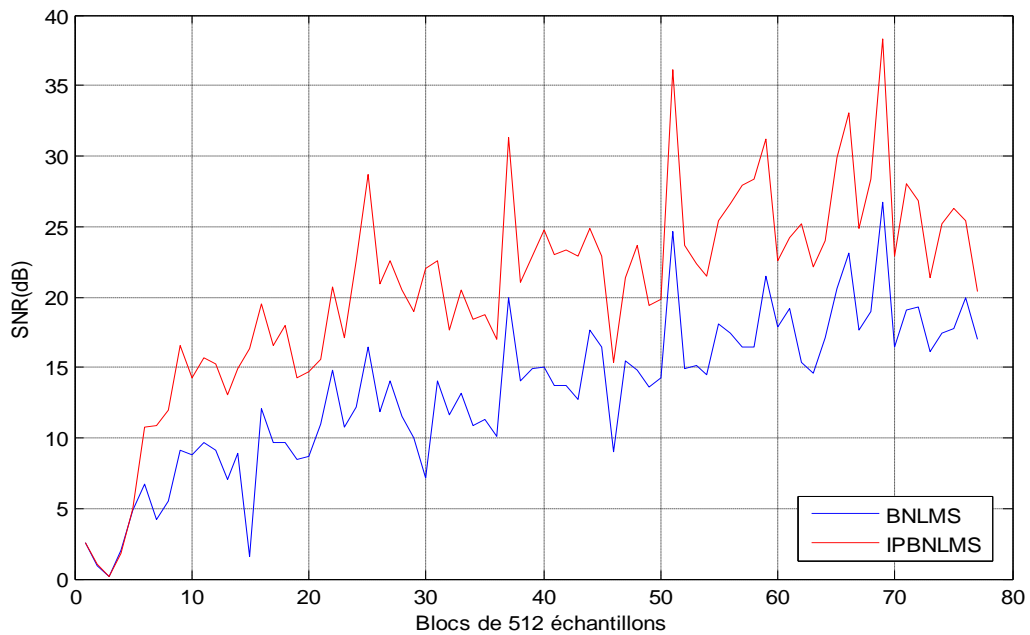
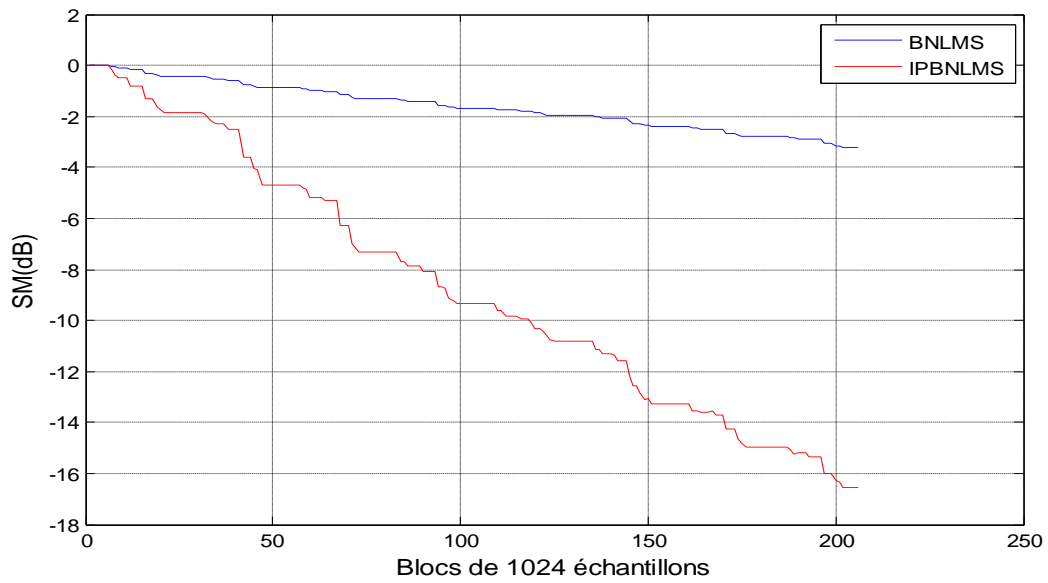


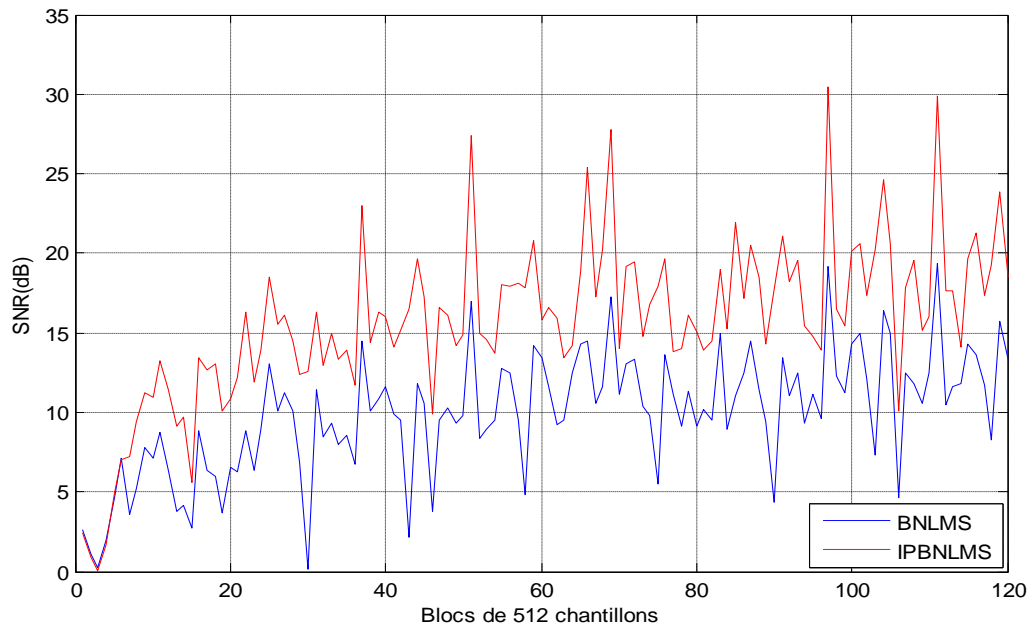
Figure 4.23. SNR pour L=512,  $\mu_{21n} = 0.9$ , ( $\times 5$ )

Lorsque la taille du filtre est L=512 ; la vitesse de convergence des algorithmes BNLMS et IPBNLMS est bien atténuée ; et pour le SNR la qualité des résultats pour L = 512 présente une dégradation légère par rapport à celle de L = 256.

Pour  $L=1024$  :



**Figure 4.24.** SM pour  $L=1024$ ,  $\mu_{21n} = 0.9$ , ( $\times 8$ )



**Figure 4.25.** SNR pour  $L=1024$ ,  $\mu_{21n} = 0.9$ , ( $\times 8$ )

Pour la taille de filtre  $L=1024$  l'algorithme IPBNLMS converge plus rapidement et donne une meilleur qualité de SNR.

Pour  $L=2048$  :

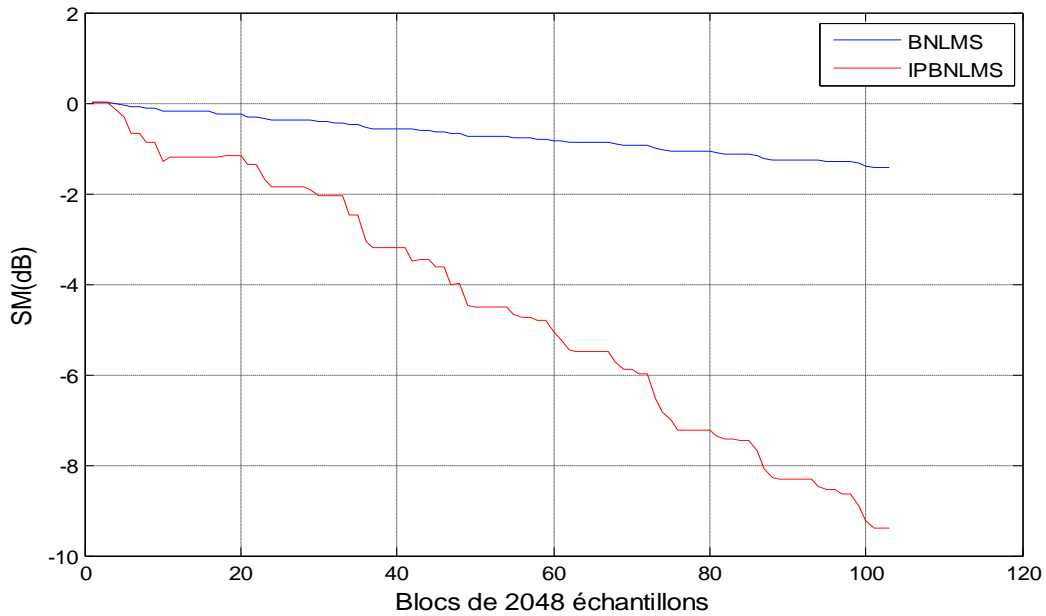


Figure 4.26. SM pour  $L=2048$ ,  $\mu_{21n} = 0.9$ , ( $\times 8$ )

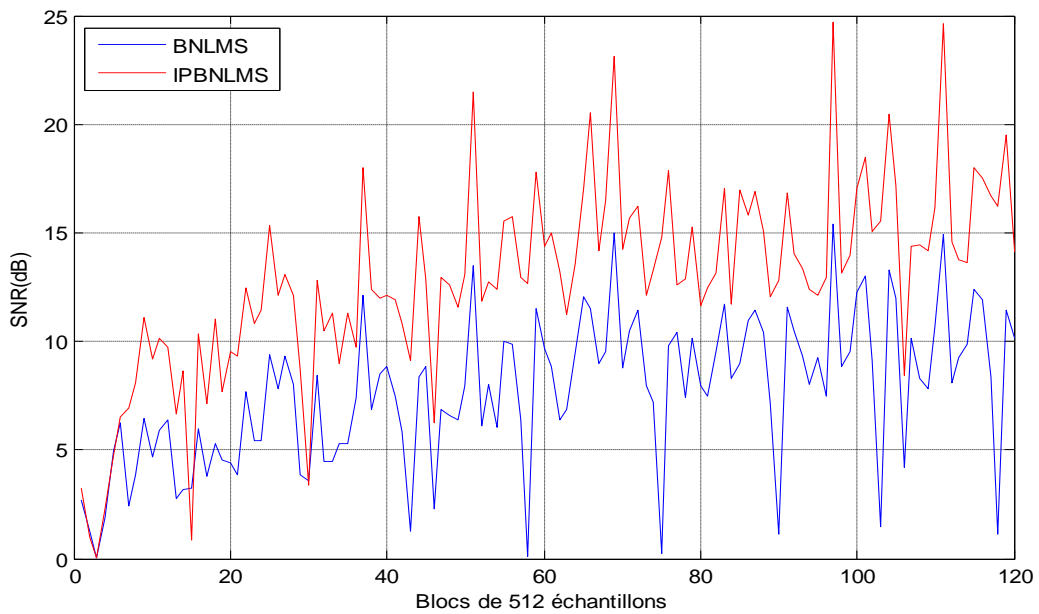


Figure 4.27. SNR pour  $L=2048$ ,  $\mu_{21n} = 0.9$ , ( $\times 8$ )

Nous avons remarqué que l'algorithme proportionnel IPBNLMS est plus efficace que l'algorithme BNLMS quelque soit la taille de réponse impulsionnelle sparse utilisée, ce dernier présente une vitesse de convergence très lente et une qualité dégradée dans les systèmes sparses, donc l'algorithme proposé présente une solution très efficace de ce problème.

#### 4.5.4. Effet de pas d'adaptation

Nous avons testé l'effet de pas d'adaptation sur les algorithmes BNLMS et IPBNLMS ; on prend  $\mu_{21n} = 0.1$ ,  $\mu_{21n} = 0.5$ ,  $\mu_{21n} = 0.9$ ,  $\mu_{21n} = 1.5$  :

Pour  $\mu_{21n} = 0.1$  :

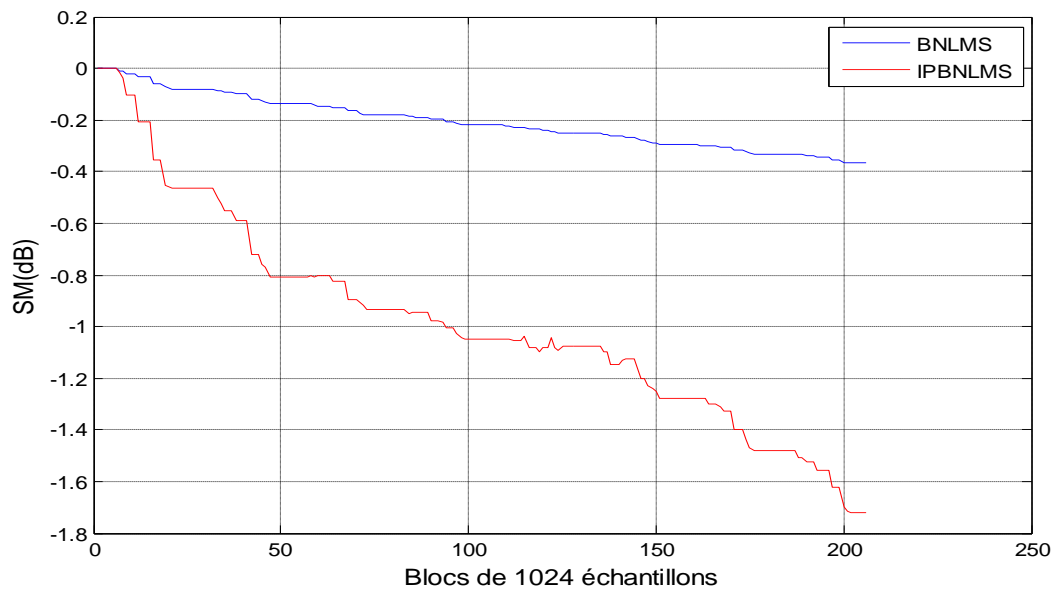


Figure 4.28. SM pour  $L=1024$ ,  $\mu_{21n} = 0.1$ , ( $\times 8$ )

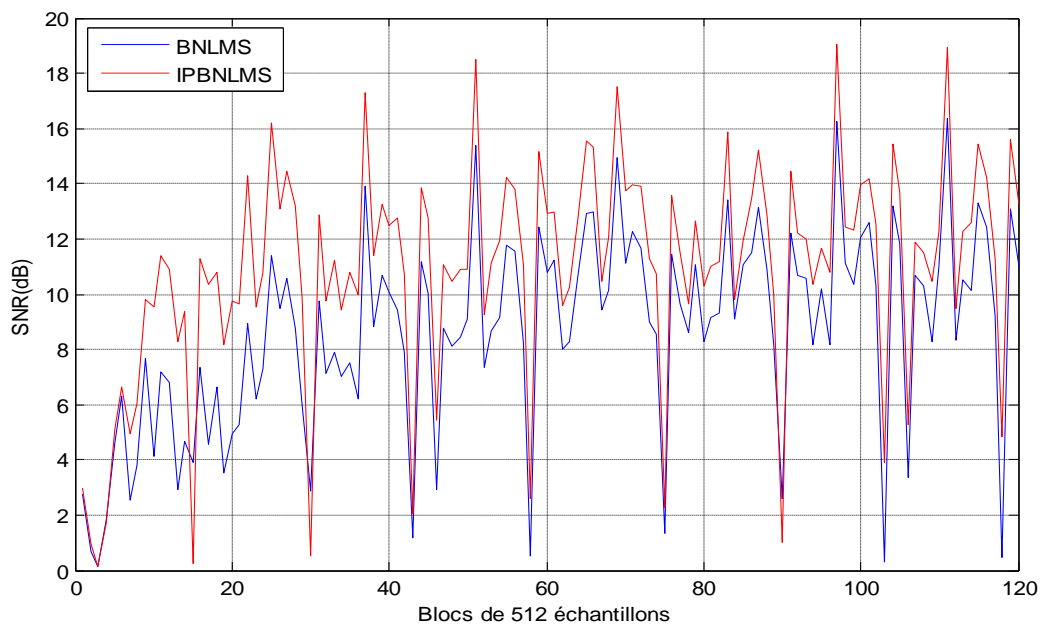


Figure 4.29. SNR pour  $L=1024$ ,  $\mu_{21n} = 0.1$ , ( $\times 8$ )

Pour un pas d'adaptation très petit qui égale à 0.1 la vitesse de convergence est très lente pour l'algorithme proportionnel IPBNLMS.

Pour  $\mu_{21n} = 0.5$  :

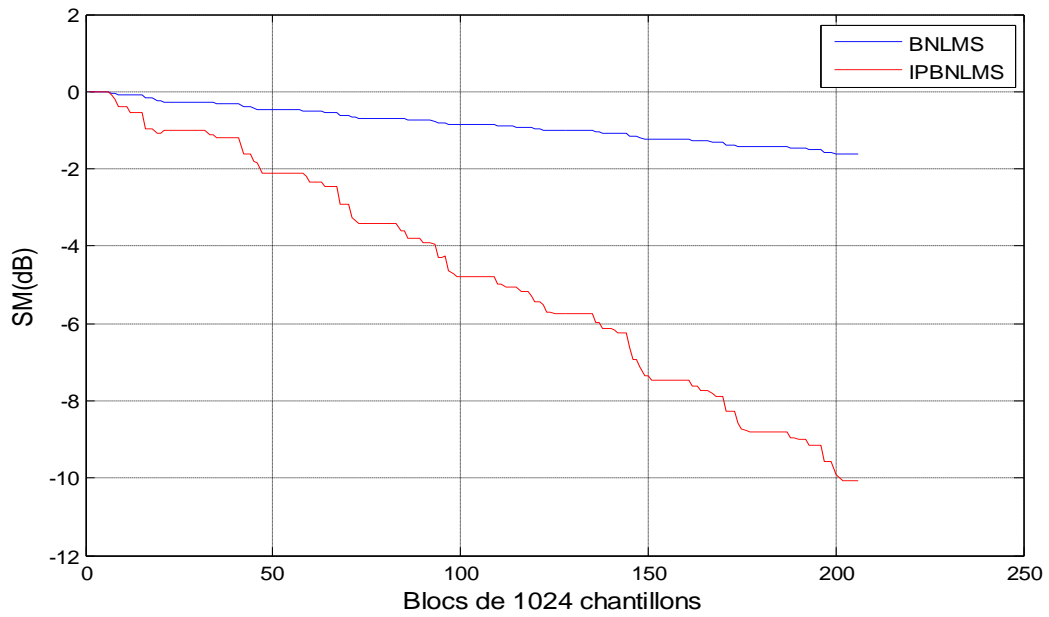


Figure 4.30. SM pour  $L=1024$ ,  $\mu_{21n} = 0.5$ , ( $\times 8$ )

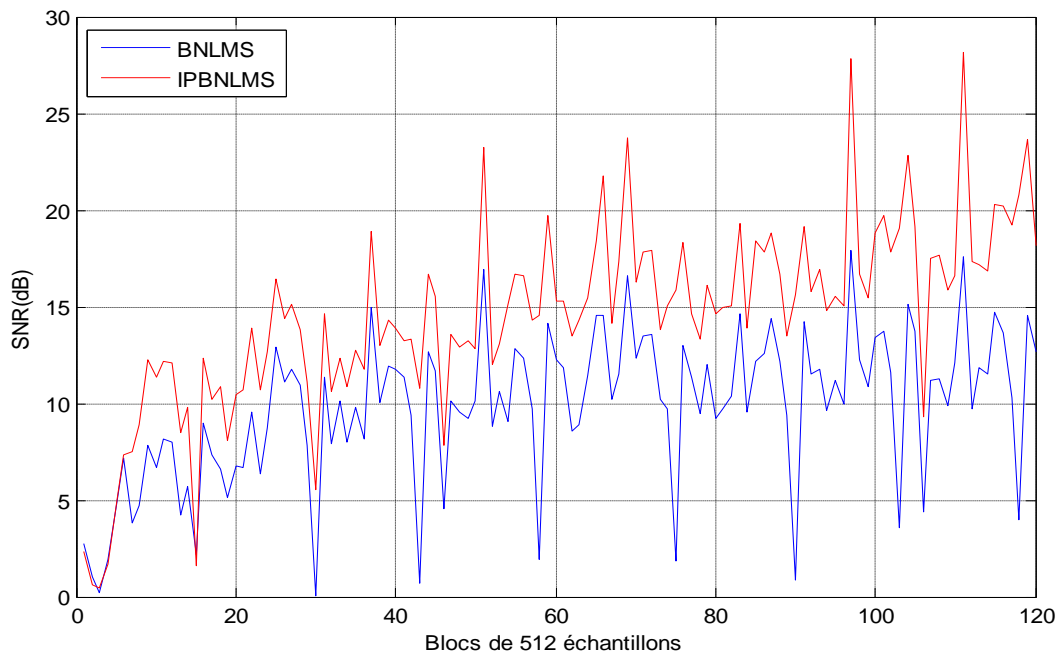
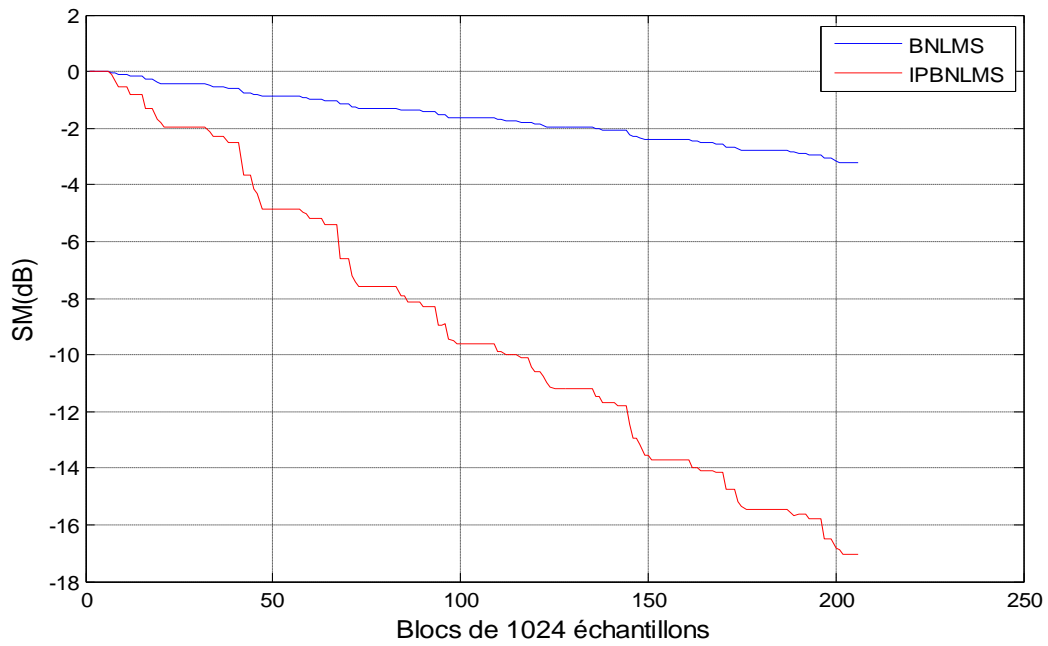


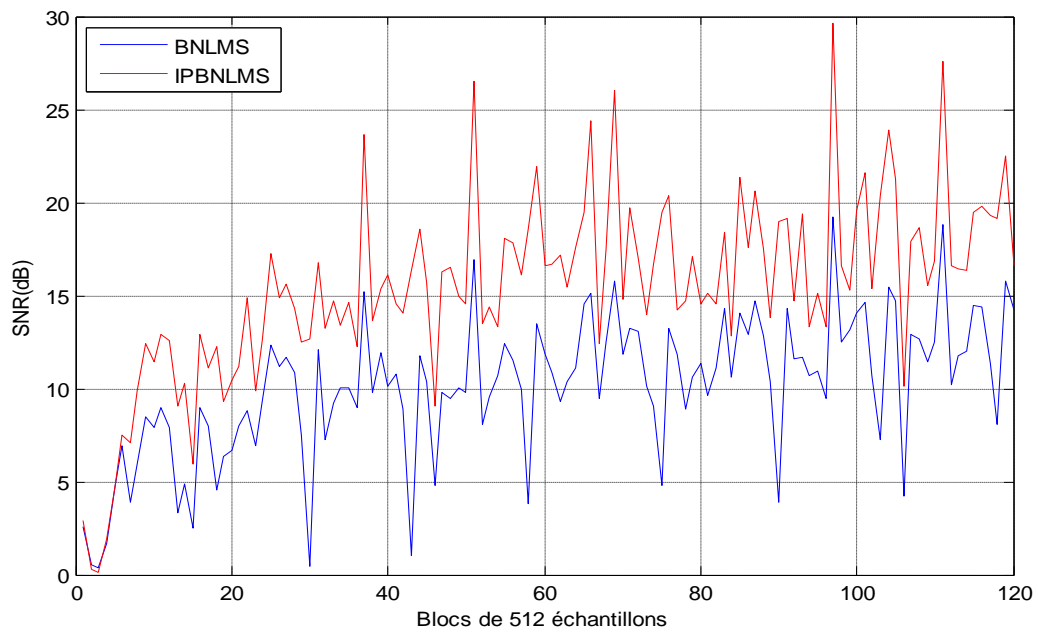
Figure 4.31. SNR pour  $L=1024$ ,  $\mu_{21n} = 0.5$ , ( $\times 8$ )

Lorsqu'on augmente le pas d'adaptation de 0.1 à 0.5 on remarque une amélioration des résultats.

Pour  $\mu_{21n} = 0.9$  :



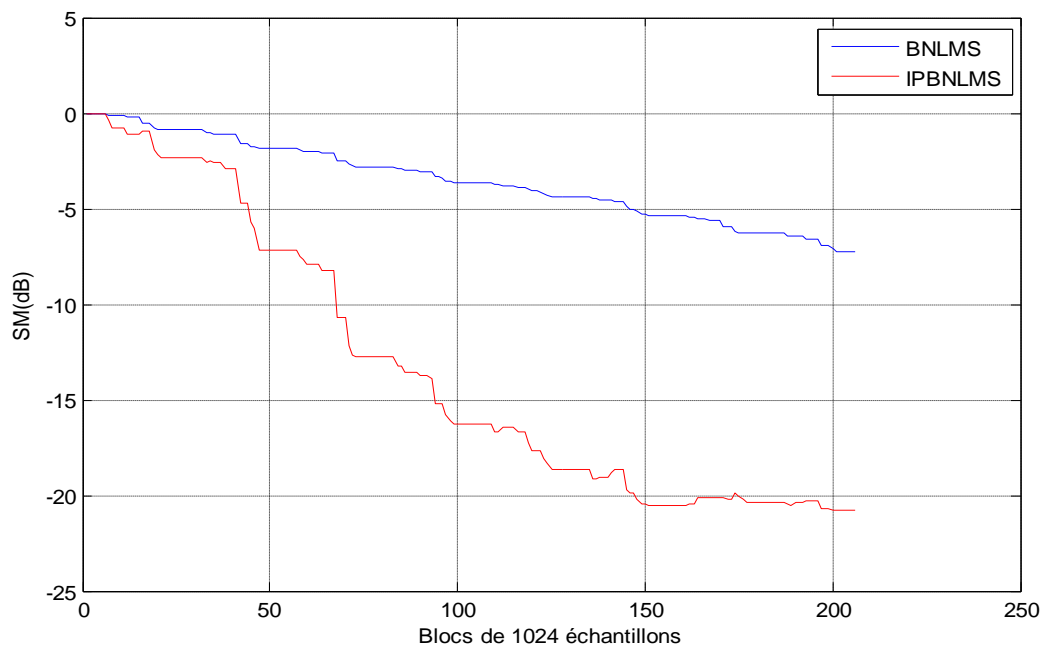
**Figure 4.32.** SM pour  $L=1024$ ,  $\mu_{21n} = 0.9$ , ( $\times 8$ )



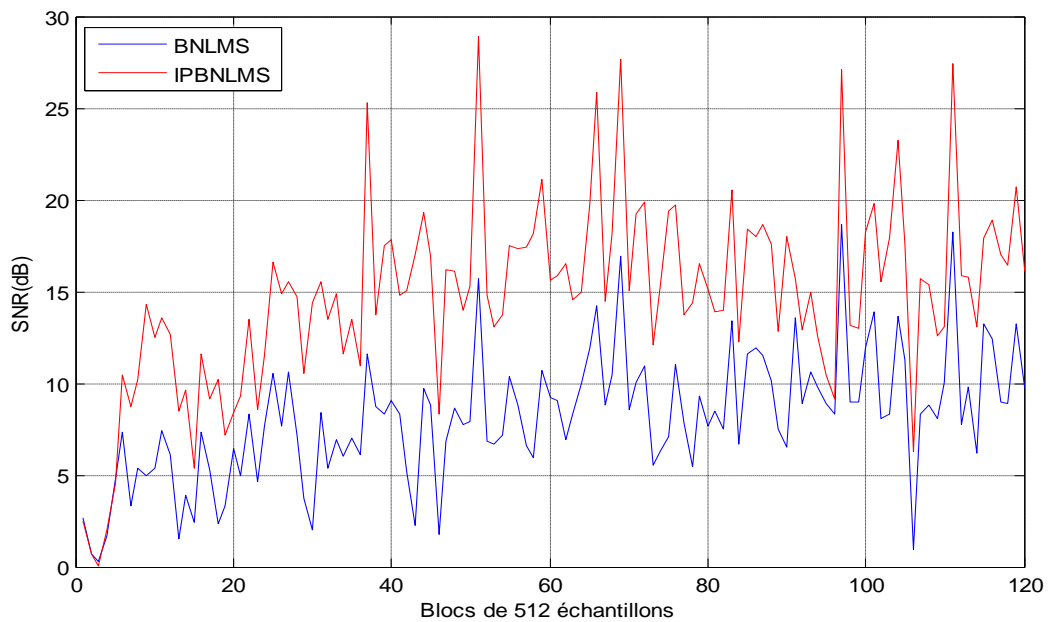
**Figure 4.33.** SNR pour  $L=1024$ ,  $\mu_{21n} = 0.9$ , ( $\times 8$ )

D'après les figures 4.32 et 4.33 on remarque que la vitesse de convergence et la qualité du SNR ont amélioré.

Pour  $\mu_{21n} = 1.5$  :



**Figure 4.34.** SM pour  $L=1024$ ,  $\mu_{21n} = 1.5$ , ( $\times 8$ )



**Figure 4.35.** SNR pour  $L=1024$ ,  $\mu_{21n} = 1.5$ , ( $\times 8$ )

D'après les figures ; lorsque le pas d'adaptation est très petit on remarque une convergence très lente ; à chaque fois qu'on augmente le pas d'adaptation on remarque l'amélioration de la vitesse de convergence et la qualité des signaux estimés.

Ce qui montre que la vitesse de convergence de la structure Backward est proportionnelle au pas d'adaptation.

Nous notons aussi que l'algorithme IPBNLMS donne des bons résultats pour la vitesse de convergence et la qualité des signaux estimés par rapport à l'algorithme BNLMS quelque soit la valeur du pas d'adaptation.

#### **4.6. Conclusion**

D'après les résultats de simulations de l'algorithme BNLMS et l'algorithme proportionnel IPBNLMS on peut conclure que BNLMS donne des résultats moins performants pour les environnements sparse.

Nous avons aussi remarqué qu'à chaque fois qu'on augmente la taille du filtre et le pas d'adaptation l'algorithme proportionnel IPBNLMS donne des meilleurs résultats pour la vitesse de convergence et la qualité du SNR par rapport à l'algorithme BNLMS.



## Conclusion générale

---

Le but principal de ce projet est de faire une étude sur réduction du bruit acoustique dans les systèmes sparse, en utilisant des algorithmes adaptatifs mis en œuvre sur la structure récursive (backward) de la technique de séparation de sources aveugles bi-capteurs.

Dans le premier volet de ce mémoire, nous avons présenté une généralité sur le filtrage adaptatif ainsi les différents algorithmes adaptatifs classiques et proportionnels, et une petite aperçus sur les réponses impulsionnelles sparse et dispersive, Puis on a basés en particulier sur l'étude de la structure de séparation aveugle des sources (Forward et Backward), en employant deux capteurs de prise de son.

Dans la suite de ce document, nous avons fait une étude sur les performances de l'algorithme proposé IPNLMS spécifique adapté a la structure de séparation de sources Backward en comparaison avec les algorithmes NLMS et PNLMS, Qui attribue à chaque coefficient un paramètre de taille de pas proportionnel à sa valeur estimée.

D'après l'étude comparative des performances de ces algorithmes, nous avons remarqué que cette Versions algorithmique proposé donne des bons résultats pour la séparation de sources et le rehaussement du signal de parole. Ceci montre bien que l'algorithme converge vers une solution souhaitée de manière stable.

On a aussi montré, à travers les résultats de simulations réalisés, que la vitesse de convergence de cet algorithme est inversement proportionnelle à la taille du filtre et le rapport signal à bruit sur la voie de référence. Nous avons remarqué également que les performances de la vitesse de convergence sont directement proportionnelles au pas d'adaptation.

Enfin Nous pouvons conclure que l'algorithme proposé (IPBLNMS) a réussi à atténuer le bruit contenant dans le signal utile, et donner des bonnes performances telles que la vitesse rapide de convergence, et le minimum de distorsion du signal utile en sortie du traitement.

Ces expériences ont permis d'analyser et d'étudier les forces et les faiblesses des algorithmes en termes de vitesse de convergence et de convergence, de précision de l'estimation et de complexité de calcul.

## Bibliographie

---

[01] T.Dutoit, « Introduction au traitement Automatique de la parole », Mons ,20 Octobre 2000.

[02] J.Radecki, Z.Zilic, K.Radecka, « Echo cancellation in IP networks », Proceedings of the 5th Midwest Symposium on Circuits and Systems, Tulsa, Okla, USA, vol.2, n°8, 219-222, August 2002.

[03] Ligang LIU, « On Improvement of Proportionate Adaptive Algorithms for Sparse Impulse Response » A dissertation submitted to Kochi University of Technology in partial fulfillment of the requirements for the degree of Doctor of Philosophy, Kochi, Japan, September 2009.

[04] Djendi, M., Gilloire, A., Scalart, P., “Noise cancellation using two closely spaced microphones: experimental study with a specific model and two adaptive algorithms”, IEEE Int. Conf. ICASSP, Toulouse, France, V.3, (May 2006), 744-748.

[05] Djendi, M., “Advanced techniques for two-microphone noise reduction in mobile communications”, Thèse de Doctorat, Université de Rennes 1, France, (Jan. 2010).

[06] Djendi, M., Gilloire, A., Scalart, P., “New frequency domain post-filters for noise cancellation using two closely spaced microphones”, Proc. EUSIPCO, Poznan, V.1, (Sep. 2007), 218-221.

[07] H.L. Nguyen Thi, Ch. Jutten, J. Caelen, « Séparation aveugle de parole et de bruit dans un mélange convolutif ». 13ème colloque GRETSI, Juan-lespins, 16-20 Sep 91.

[08] R.Bendoumia, A.Deba, « Rehaussement du signal parole par l’algorithme de decorrélation symétrique », Mémoire de master, USDB, 2011.

- [09] J.-F. Bercher & P. Jardin « Introduction au filtrage adaptatif », ESIEE Paris, I4-TTS 2003.
- [10] J. Benesty « Traitement des signaux numériques, Filtrage adaptatif et analyse spectrale » INRS-EMT Canada.
- [11] Safi.M, BENSAFIA.S, « Rehaussement du signal de parole par un algorithme fréquentiel de type Backward » Mémoire de master, USDB 2016.
- [12] A.TEDJNI, A.BENALLAL, « Study of sparse adaptive Algorithms for the identification of acoustic impulse responses », Magister report university of Saad Dahleb Blida ,2016.
- [13] J. Benesty « Algorithme du gradient stochastique (least-mean-square – LMS) » INRS-EMT.
- [14] Pradeep Loganathan, Patrick Naylor, Tania Stathaki, « Adaptive Echo Cancellation», Imperial College London, Department of Electrical and Electronic Engineering, Final Year Project Report 2007.
- [15] R.Bendoumia, « annulation du bruit par les methodes de separation de sources aveugles. application aux systemes de telecommunications numeriques », these de doctorat, universite blida 1, 2014.
- [16] F.Bouderbala, O.chabouni, Proposition d'un algorithme rapide à deux canaux pour la réduction du bruit dans les systèmes téléphoniques à mains-libres, Mémoire de master, USDB 2018.
- [17] O.Bouchekouk, M.Bouzidi, « Annulation du bruit par les algorithmes proportionnels implémentés sur la structure Forward de séparation de sources », Mémoire de Master, université Saad Dahleb BLIDA, 2018.
- [18] O.Cherrak, « Séparation aveugle de mélanges linéaires de sources : application à la surveillance maritime », Université de Toulon, Université Sidi Mohamed ben Abdellah (Fès, Maroc), 2016.
- [19] Meriem ZOULIKHA, « Méthodes améliorées pour la réduction du bruit acoustique et le rehaussement du signal de parole », Mémoire de Magister, USDB, Avril 2014.

[20] J. Benesty and Steven L. Gay «An improved pmls algorithm», Bell Laboratories, Lucent Technologies 700 Mountain Avenue Murray Hill, NJ 07974, USA.

[21] Jianming Liu, Steven L. Grant, « Proportionate Adaptive for Block-sparse System Identification » ; IEEE/ACM Transactions on Audio, Speech, and Language Processing , Volume: 24, Issue: 4 , April 2016.