

الجمهورية الجزائرية الديمقراطية الشعبية  
République Algérienne démocratique et populaire

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

جامعة سعد دحلب البليدة  
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا  
Faculté de Technologie

قسم الإلكترونيك  
Département d'Électronique



## Mémoire de Master

Filière Télécommunication  
Spécialité Réseaux et Télécommunication

présenté par

ELAHOUEL Abdelmadjid

&

MANSOURI Miloud

---

# Rehaussement de la parole par les algorithmes Forward et Backward à pas d'adaptation variable et normalisé

---

Proposé par : Mr BENDOUMIA Rédha.

Année Universitaire 2017-2018

## Remerciements

---

Nous tenons à remercier, en tout premier lieu, le Bon Dieu, le Tout-Puissant, de nous avoir donné le courage, la volonté, la patience, et la santé et nous avoir aidés à surmonter toutes les dures épreuves et les moments difficiles pour mener à bien ce travail.

Nous tenons à exprimer nos remerciements à Monsieur BENDOUMIA Rédha, pour avoir accepté de nous encadrer, de nous orienter et de nous conseiller tout au long de la réalisation de ce travail.

Nous exprimons également notre gratitude à tous les professeurs qui ont contribué à notre formation depuis notre premier cycle d'études jusqu'à aujourd'hui.

Nos remerciements les plus vifs s'adressent aussi aux membres de jury d'avoir accepté d'examiner et d'évaluer notre travail.

Nos pensées vont vers nos familles, et surtout vers nos parents qui nous ont permis de poursuivre nos études jusqu'au bout.

## الإهداء

الحمد لله الذي وفقنا لإتمام هذا العمل ونسأله أن يوفقنا ويجعل هذا العمل خالصا لوجهه الكريم.

إلى الوالدة العزيزة التي ربنتي وساعدتني طوال مشواري الدراسي وأدعو الله أن يرحمها كما ربتي صغيرا, و أن يسكنها فسيح جناته.

إلى الوالد العزيز الذي رباني وساعدني في طول هذا المشوار الدراسي أتمنى له طول العمر والصحة والعافية.

إلى كل من ساندني و وقف جنبي وكل أصدقائي وأحبابي في الله الذين كانوا معي في السراء والضراء , وأتمنى للجميع التوفيق إن شاء الله.

## *Dédicace*

Je dédie ce travail et j'exprime ma profonde gratitude à ma MERE (Allah yerhamha) et à mon PERE pour l'éducation qu'ils m'ont prodigués, pour les sacrifices qu'ils ont effectués ainsi que pour le sens du devoir qu'ils m'ont enseigné depuis mon enfance.

Leur bonté, leur constante disponibilité,  
Particulièrement dans les moments les plus difficiles,

Toute ma gratitude pour vous

Mon frère et mes sœurs

Mes meilleurs amis

Et à toute la famille

Et bien sûr aux éléments de la section

Réseaux et Télécommunication

Et à toute la promotion 2018.

*Abdelmadjid*

## الإهداء

الحمد لله الذي وفقنا لإتمام هذا العمل ونسأله أن يتقبله منا ويجعله خالصا لوجهه الكريم  
إلى الوالدة العزيزة التي ربنتني وساعدتني طوال مشواري الدراسي وأتمنى أن يرزقها الله  
ثوب الصحة والعافية.

إلى الوالد العزيز الذي رباني وساعدني في طول هذا المشوار الدراسي أتمنى له طول  
العمر والصحة والعافية

إلى كل من ساندني و وقف جنبي وكل أصدقائي وأحبابي في الله الذين كانوا معي في  
السراء والضراء , وأتمنى للجميع التوفيق إن شاء الله.

## *Dédicaces*

Je dédie ce travail à ma MERE et à mon PERE et je les  
remercie pour l'éducation qu'ils m'ont assurée avec tous les  
moyens déployés, ainsi que pour tous leurs sacrifices, pour le  
sens du devoir qu'ils m'ont enseigné depuis mon enfance

Leur bonté, leur constante disponibilité,

Particulièrement dans les moments les plus difficiles,

Toute ma gratitude pour vous

Mon frère et mes sœurs

Mes meilleurs amis

Et à toute la famille

Et bien sûr aux éléments de la section

Réseaux et Télécommunication

Et à toute la promotion 2018.

*Miloud*

---

## ملخص:

أثناء مكالمة هاتفية من هاتف محمول في سيارة، تتأثر الإشارة الصوتية المرسلّة بتداخل العديد من العوامل المنتجة للضوضاء. هناك العديد من طرق تحسين إشارة الكلام، مثل طريقة بفرقة المصادر التي تعتمد على الإشارات الصوتية والضوضاء المستقلة من حيث المصدر. يهدف مشروعنا إلى تعزيز إشارة الكلام من خلال أساليب التكيف، نحن نعتد في مشروعنا على هياكل نزع الضوضاء ثنائية الاستشعار، فإننا نستخدم خوارزمية الربط المتعكسة المتناظرة المطبقة على كل من الهياكل الأمامية والخلفية.

**كلمات المفاتيح:** تعزيز إشارة، هياكل ثنائية، الاستشعار تكيف متماثل.

---

## Résumé :

Lors d'un appel téléphonique depuis un téléphone portable dans une voiture, le signal vocal transmis est affecté par l'interférence de plusieurs facteurs de production de bruit. Il existe plusieurs méthodes de rehaussement d'un signal parole, comme la méthode de séparation des sources qui repose sur des signaux audio et le bruit indépendants de la source. Notre projet a pour but de rehausser un signal de parole par les méthodes adaptatives. Nous nous baserons sur les structures d'annulation du bruit bi-capteurs, nous utiliserons l'algorithme de décorrélation symétrique adaptatif (SAD) normalisé et à pas d'adaptation variables appliqué aux deux structures dites Forward et Backward.

**Mots clés :** Rehausser un signal; les structures bi-capteurs; décorrélation symétrique adaptatif.

---

## Abstract :

During a phone call from a mobile phone in a car, the transmitted voice signal is affected by the interference of several noise producing factors. There are several methods of speech enhancement, such as the method of distributing sources that relies on audio signals and noise independent of the source. Our project aims to enhance a speech signal through adaptive methods. Based on bi-sensor noise cancellation structures, we use the symmetric adaptive decorrelation (SAD) algorithm standardized with the variable step sizes versions applied to both Forward and Backward structures.

**Keywords :** Enhance a signal ; bi-sensor structures; symmetric adaptive decorrelation.

---

## Listes des acronymes et abréviations

RIF : réponse impulsionnelle finie.

EQM : Erreur Quadratique Moyenne.

MSE: Mean Square Erreur.

LMS : Least-Mean Square.

NLMS : Normalized Least-Mean Square.

SAD : Symmetric adaptive decorrelation.

FSAD : Forward symmetric adaptive decorrelation.

FNSAD : Forward normalise symmetric adaptive decorrelation.

VS-FNSAD : Variable step-size Forward normalise symmetric adaptive decorrelation.

SM : System Mismatch.

RSB : Rapport signal sur bruit.

DAV : Détection d'activité vocale.

# Introduction générale

---

Le bruit est partout et dans la plupart des applications qui sont liées à l'acoustique et à la parole, tel que les interfaces homme-machine, les communications mains libres, la voix à travers IP (VOIP), les prothèses auditives, les systèmes de téléconférence et télécollaboration, et tant d'autres. Le signal d'intérêt (habituellement la parole) qui est capté par un microphone est généralement contaminé par le bruit. Par conséquent, le signal issu du microphone doit être nettoyé par des outils du traitement numérique du signal avant qu'il soit stocké, analysé ou transmis. Ce processus de nettoyage s'appelle souvent le rehaussement de la parole. Ce domaine a et continue d'attirer l'attention des chercheurs pendant plusieurs décennies.

Le bruit peut être défini en tant que n'importe quel signal non désiré perturbant le signal portant l'information utile. Il est présent avec des degrés différents dans presque tous les environnements. Il peut causer des erreurs de transmission et peut même perturber un processus de communication; par conséquent le traitement du bruit est une partie importante et intégrale dans les systèmes de télécommunications modernes et les systèmes de traitement du signal.

Le but d'un algorithme de rehaussement de la parole est d'estimer le signal de parole à partir de sa version corrompue par un bruit.

Généralement, les systèmes intègrent différents algorithmes relativement efficaces pour séparer les signaux, supprimer l'écho et atténuer le bruit. Malheureusement, la performance de ces algorithmes est très dépendante de la façon dont on détecte l'activité vocale entre le locuteur lointain (le signal vocal provenant du haut-parleur) et le locuteur installé en véhicule (le signal à transmettre par le système). Dans les

systèmes actuels, l'activité vocale pour chaque locuteur est évaluée par des algorithmes qui fonctionnent à base d'énergie.

La séparation de sources par les méthodes bi-capteurs est une technique relativement 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. Comme le signal de parole et le bruit de voiture sont indépendants et linéairement mélangés à l'intérieur de la voiture.

Dans notre travail, nous étudions un algorithme de séparation de sources qui s'appelle l'algorithme de décorrélation symétrique adaptatif normalisé à pas d'adaptation variable (VS-NSAD) appliqué à deux structures dites Forward (VS-FNSAD) et Backward (VS-BNSAD). Nous présentons ci-après les différents chapitres de ce mémoire :

**Chapitre 1 :** présente en premier lieu, le filtrage adaptatif et les algorithmes de gradient stochastique (LMS et NLMS). En deuxième lieu, nous présentons la séparation du bruit par la méthode bi-capteur et pour rehausser le signal de la parole et séparer les sources.

**Chapitre 2 :** présente, le principe d'annulation du bruit par le critère de Décorrélation, ainsi l'Algorithme de décorrélation symétrique adaptatif avec pas d'adaptations variables et normalisés pour la version Forward et les différents algorithmes proposés (VS-FNSAD).

**Chapitre 3 :** présente, l'Algorithme de décorrélation symétrique adaptatif Backward (BSAD) avec pas d'adaptations variables et normalisés, et les différents algorithmes proposés (VS-BNSAD).

**Chapitre 4 :** présente les résultats des simulations et les performances d'algorithme SAD et les algorithmes avec pas d'adaptations variables proposés pour les deux versions Forward et Backward.

Ce présent document sera terminé par une conclusion générale.

# Chapitre 1 Rehaussement de la parole par la méthode bi-capteur

---

## 1.1 Introduction

Le traitement du signal est une discipline indispensable de nos jours, il a pour objet l'élaboration ou l'interprétation des signaux porteurs d'informations. Son but est donc de réussir à extraire un maximum d'informations utiles d'un signal perturbé par du bruit en s'appuyant sur les ressources de l'électronique, de l'informatique et de la physique appliquée.

Donc, nous sommes confrontés à la nécessité d'éliminer le bruit et la déformation des signaux. Ces phénomènes sont dus aux processus physiques variables dans le temps, qui sont parfois inconnus. Le milieu (air, fils, fibres, ... etc.), qui est connu comme canal, présente le bruit et la déformation dus aux variations de ses propriétés. Ces variations peuvent être lentes ou rapides. Puisque le plus souvent des variations sont inconnues, c'est l'utilisation du filtrage adaptatif que cela diminue et élimine parfois complètement la déformation du signal.

Les filtres adaptatifs les plus utilisés pendant le procédé d'adaptation, sont des types (RIF : réponse impulsionnelle finie). Ce sont préférables parce qu'ils sont stables, et aucun ajustement spécial n'est nécessaire pour leur fonctionnement [1].

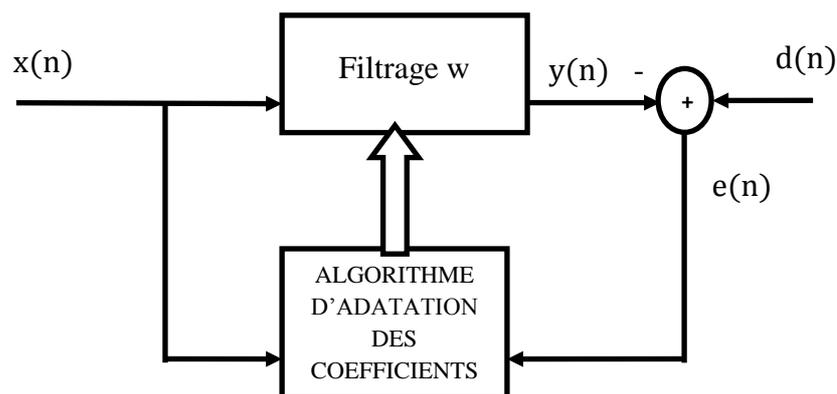
## 1.2 Filtrage adaptatif

Un filtre adaptatif est un filtre 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 est changeant, ou pour supprimer des perturbations situées dans le domaine des fréquences du signal utile, ce que les filtres classiques ne peuvent pas faire [2].

Le filtrage adaptatif est un outil puissant en traitement du signal, communications numériques, et contrôle automatique. Les applications sont diverses mais présentent les caractéristiques suivantes: on dispose d'une entrée  $x(n)$  ainsi que de la 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 [2].

Les différents signaux qui sont utilisés dans le filtre adaptatif qui est représenté à la figure (1.1):

- ❖  $x(n)$  est le signal de l'entrée du filtre,
- ❖  $w(n)$  sont les coefficients du filtre,
- ❖  $y(n)$  est le signal de la sortie du filtre,
- ❖  $d(n)$  est le signal de la référence,
- ❖  $e(n)$  est le signal d'erreur (c'est la différence entre  $d(n)$  et  $y(n)$ ).



**Figure 1.1.** Principe du filtrage adaptatif.

Les filtres adaptatifs peuvent être classés en fonction des choix qui sont faits sur les points suivants [3] :

- ❖ 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.

Il existe deux classes importantes de filtres linéaires optimaux [2]:

- **Filtrage de Wiener** : où les signaux  $d(n)$  et  $x(n)$  sont considérés stationnaires.

- **Filtrage de Kalman**: qui est une généralisation du filtre de Wiener valable aussi dans le cas de processus (ou de signaux) non stationnaires.

### 1.3 Rôle d'un filtre adaptative

Le rôle d'un filtre adaptatif est d'ajuster le paramètre 'w' pour un objectif bien défini c'est à dire la minimisation de l'erreur quadratique moyenne (EQM).

### 1.4 Filtrage de Wiener

Le filtrage de Wiener est adéquat pour les situations dans lesquelles le signal ou le bruit sont stationnaires pour minimiser l'erreur quadratique moyenne (EQM ou MSE en anglais).

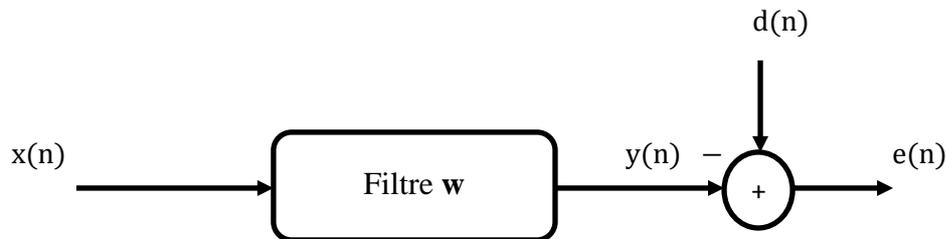


Figure 1.2. Schéma principal du filtrage de Wiener.

#### 1.4.1 Principe

Le problème du filtrage optimal est de trouver le « meilleur » filtre c'est à dire celui permettant d'obtenir en sortie une réponse  $y(n)$  la plus « proche » possible d'une réponse désirée  $d(n)$  lorsque l'entrée est une certaine séquence  $x(n)$ .

On note  $e(n) = d(n) - y(n)$  l'erreur entre la réponse désirée  $d(n)$  et la sortie  $y(n)$ .

On note également  $w(n)$  représente les coefficients du filtre [2].

Le problème consiste donc à rechercher le filtre assurant l'erreur la plus faible  $e(n)$ , au sens d'une certaine fonction de coût, De nombreux choix sont par exemple :

- erreur quadratique moyenne,
- erreur  $L_1$ ,
- erreur  $L_K$ ,
- erreur  $L_\infty$ ,

Parmi celles-ci, l'erreur quadratique moyenne est la plus utilisée, car elle conduit à des développements mathématiques complets et simples, fournit la solution en fonction des caractéristiques au second ordre des variables aléatoires, caractéristiques qui sont les plus simples à estimer, et enfin fournit une solution unique. C'est sur l'estimation linéaire en moyenne quadratique que repose le filtrage de Wiener [2].

On définit la sortie du filtre comme suit:

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

Où « L » est le nombre des coefficients de la réponse impulsionnelle. C'est plus pratique d'utiliser une notation matricielle pour la sortie du filtre:

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

Avec :  $\mathbf{w}(n) = [w_0 \ w_1 \ w_{L-1}]^T$  est un vecteur de longueur L contenant les coefficients du filtre RIF, et  $\mathbf{x}(n) = [x(n) \ x(n-1) \ . \ . \ . \ x(n-L+1)]^T$  est le vecteur des L données d'entrée les plus récentes.

Le signal d'erreur est donné par :

$$e(n) = d(n) - y(n) = d(n) - \mathbf{x}^T(n) \mathbf{w}(n) \quad (1.3)$$

À partir des coefficients du filtre  $\{w_0, w_1, w_2, \dots\}$  on peut minimiser l'erreur quadratique moyenne « EQM »

$$J = E[|e(n)e^*(n)|] = E[e(n)^2] \quad (1.4)$$

D'où :

$$J = E \left[ \left( d(n) - \mathbf{w}^H(n) \mathbf{x}(n) \right) \left( d^*(n) - \mathbf{w}^H(n) \mathbf{x}^*(n) \right) \right] \quad (1.5)$$

$$J = \sigma^2(d) - \mathbf{w}^H(n) \mathbf{r} - \mathbf{w}^T(n) \mathbf{r}^* + \mathbf{w}^H(n) \mathbf{R} \mathbf{w}(n) \quad (1.6)$$

Avec :

$\mathbf{r}$  : Le vecteur d'inter-corrélation entre la sortie désirée  $d(n)$  et l'entrée  $x(n)$ .

$\mathbf{R}$  : La matrice d'auto-corrélation de l'entrée  $x(n)$ . Cette matrice est définie positive, de Toeplitz et à symétrie hermitienne ( $\mathbf{R} = \mathbf{R}^H$ ).

## 1.4.2 Relations d'orthogonalité - Équation de Wiener-Hopf

L'erreur quadratique moyenne « EQM » est donné par l'équation suivante:

$$J = \sigma^2(d) - \mathbf{w}^H(n) - \mathbf{w}^T(n)r^* + \mathbf{w}^H(n)R \mathbf{w}(n)$$

Le vecteur optimum  $\mathbf{w}_{\text{opt}}$  est celui qui annule le gradient du critère :

$$\frac{\partial J}{\partial \mathbf{w}} = 0$$

En écrivant J sous la forme :

$$\frac{\partial J}{\partial \mathbf{w}} = E \left( e(n) \frac{\partial e^*(n)}{\partial \mathbf{w}} \right) + E \left( e^*(n) \frac{\partial e(n)}{\partial \mathbf{w}} \right) \quad (1.7)$$

$$\frac{\partial e^*(n)}{\partial \mathbf{w}} = 0$$

Et

$$\frac{\partial e(n)}{\partial \mathbf{w}} = 2 \mathbf{x}(n)$$

Par conséquent, à l'optimum, on a :

$$E(e^*(n)\mathbf{x}(n)) = 0$$

C'est le **principe d'orthogonalité** signifiant que toutes les entrées  $\mathbf{x}(n)$  sont décorréliées de  $e^*(n)$ . En développant cette dernière équation on obtient [12].

$$E(\mathbf{x}(n)(d^*(n) - \mathbf{x}^H(n)\mathbf{w})) = 0$$

Soit  $\mathbf{R} \cdot \mathbf{w} = \mathbf{r}$

Cette relation est appelée Formule de Wiener ou équation de Wiener-Hopf

$$\text{La solution est le filtre optimal } \mathbf{w}_{\text{opt}} : \mathbf{w}_{\text{opt}} = \mathbf{R}^{-1} \mathbf{r} \quad (1.8)$$

## 1.5 Définition d'un algorithme

Un algorithme est un processus à effectuer pour répondre à un problème. Ce processus est toujours le même pour un même problème et l'algorithme est la

description de la méthode à utiliser. Cette procédure de résolution de problème, décortique les étapes essentielles. Il suffit de donner l'algorithme d'un problème à un être humain ou à une machine pour que celui-ci puisse effectuer les bonnes actions dans le but de résoudre le problème.

## 1.6 Choix de l'algorithme

Le choix de l'algorithme se fera en fonction des critères suivants [2]:

- ✓ La rapidité de convergence qui sera le nombre d'itérations nécessaires pour converger « assez près » de la solution optimale de Wiener dans le cas stationnaire.
- ✓ La mesure de cette « proximité » entre cette solution optimale et la solution obtenue.
- ✓ La capacité de poursuite des variations (non stationnarités) du processus. On examinera quels sont les algorithmes vraiment adaptatifs.
- ✓ La robustesse au bruit.
- ✓ La complexité.
- ✓ Les propriétés numériques (stabilité –précision) dans le cas d'une précision limitée sur les données et les coefficients (problèmes d'implémentation en virgule fixe).

## 1.7 Algorithme du gradient stochastique (LMS)

### 1.7.1 Définition de base

L'algorithme du gradient stochastique est une approximation de l'algorithme du gradient déterministe. L'algorithme LMS est certainement l'algorithme adaptatif le plus populaire qui existe en raison de sa simplicité.

La matrice d'autocorrélation :  $\mathbf{R} = E\{\mathbf{x}(n) \mathbf{x}(n)\}$

Le vecteur d'intercorrélation :  $\mathbf{r} = E\{\mathbf{x}(n) d(n)\}$

Puisque  $\mathbf{R} = E\{\mathbf{x}(n) \mathbf{x}^T(n)\}$  et  $\mathbf{r} = E\{\mathbf{x}(n) d(n)\}$  sont inconnus, on approchera ces grandeurs déterministes par des estimées  $\hat{\mathbf{R}}(n)$  et  $\hat{\mathbf{r}}(n)$  à l'instant  $n$ . Dans le cas du LMS, on choisit les estimées les plus simples possibles, à savoir [7]:

$$\hat{\mathbf{R}}(n) = \mathbf{x}(n) \mathbf{x}^T(n) \quad (1.9)$$

$$\hat{r}(n) = \mathbf{x}(n) d(n) \quad (1.10)$$

Ce sont simplement les estimées instantanées des corrélations.

En remplaçant  $\hat{\mathbf{R}}(n)$  et  $\hat{r}(n)$  dans l'algorithme du gradient déterministe (première équation), on obtient:

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \mu [\hat{r}(n) - \hat{\mathbf{R}}(n) \mathbf{h}(n)] \quad (1.11)$$

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \mu \mathbf{x}(n)[d(n) - \mathbf{x}^T(n) \mathbf{w}(n)] \quad (1.12)$$

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

Qui est l'algorithme LMS. On remarquera que  $\mathbf{w}(n)$  est aléatoire, puisque à chaque nouvelle itération « n »,  $\mathbf{w}(n)$  dépend des processus aléatoires  $\mathbf{x}(n)$  et  $d(n)$ .

$$\text{Avec : } [e(n) = d(n) - y(n) \text{ et } y(n) = \mathbf{x}^T(n) \mathbf{w}(n)] \quad (1.14)$$

Avec  $\mu$  est le pas d'adaptation de l'algorithme qui démarre avec une initialisation quelconque  $\mathbf{w}$ .  $0 < \mu < \frac{2}{\sigma_x^2}$

L'algorithme LMS est très simple: il nécessite seulement  $2L + 1$  multiplications et  $2L$  additions par itération, et  $L$  est le nombre des coefficients du filtre [8].

Étape de l'algorithme	Nombre d'addition	Nombre de multiplication
$e(n) = d(n) - \mathbf{x}^T(n)\mathbf{w}(n)$	L	L
$\mathbf{w}(n) = \mathbf{w}(n-1) + \mu \mathbf{x}(n)e(n-1)$	L	L + 1
<b>Total par itération</b>	2L	2L + 1

**Tableau 1.1.** Opérations nécessaires dans l'algorithme LMS.

### 1.7.2 Convergence de l'algorithme LMS

L'analyse de la convergence du LMS se fait en utilisant les deux critères suivants [3]:

Convergence en moyenne du filtre  $\mathbf{w}(n)$ , c'est à dire:

$$\lim_{n \rightarrow \infty} E\{w(n)\} = w_{\text{opt}} \quad (1.15)$$

Convergence du critère  $J(n)$  (en moyenne quadratique), c'est à dire:

$$\lim_{n \rightarrow \infty} J(n) = J(\infty) = \text{constante} \quad (1.16)$$

### 1.7.3 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  $\lambda$  et  $\mathbf{I}$  de la matrice  $\mathbf{R}$ ,
  - iii.* la longueur  $L$  du filtre  $h$ .
- Avec un pas d'adaptation petit, le LMS converge lentement mais l'EQM excédentaire est petite.
- Avec un pas d'adaptation grand, le LMS converge rapidement mais l'EQM excédentaire est grande.
- Le temps de convergence de l'algorithme LMS dépend du conditionnement  $\kappa(\mathbf{R})$  de la matrice d'autocorrélation  $\mathbf{R}$ : plus  $\kappa(\mathbf{R})$  est grand et plus le LMS met du temps à converger [1].

### 1.8 Algorithme LMS Normalisé (NLMS)

L'algorithme NLMS (Normalized Least Mean Square) consiste à normaliser le pas dans l'algorithme LMS par rapport à l'énergie du signal d'entrée pour  $\mu$  d'adaptation réduire au minimum l'effet de variation de la puissance du signal d'entrée et de rendre minimum ainsi la convergence plus ou moins uniforme en passant d'une étape d'adaptation a une autre [1].

Dans l'équation qui va suivre, le pas d'adaptation  $\mu$  est alors remplacé par un pas  $\mu_n$  défini à chaque itération par :

$$\mathbb{E} = \frac{\mu_n}{\mathbf{x}(n)\mathbf{x}^T(n)} \quad (1.17)$$

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

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

Où  $\beta$  est un facteur permettant de suivre plus ou moins rapidement les variations d'énergie dans un signal d'entrée  $\mathbf{x}(n)$ . La convergence de cet algorithme est garantie pour un pas d'adaptation  $0 < \mu_n < 2$ .

## 1.9 Comparaison entre les algorithmes LMS et NLMS

L'intérêt de l'algorithme NLMS par rapport à l'algorithme LMS est de rendre l'algorithme indépendant de la variance du signal d'entrée. Cependant, la distribution des valeurs propres de la matrice d'autocorrélation du signal d'entrée n'est en rien modifiée. Ceci implique précisément la même dépendance, dans les deux cas, de la convergence vis à vis de la statistique du signal d'entrée. Pour les signaux stationnaires tels que le bruit blanc ou non-stationnaires tels que la parole, l'algorithme NLMS apporte une amélioration significative sur le taux de convergence par rapport au LMS grâce à la normalisation du pas d'adaptation. Cet algorithme peut être plus complexe que l'algorithme LMS mais il reste toujours l'un des algorithmes les plus simples à mettre en application. Il est souvent utilisé dans la technologie d'annulation d'écho avec ses différentes versions présentées dans les paragraphes qui suivent.

Un des inconvénients de l'algorithme NLMS par rapport au LMS est l'augmentation de la complexité de calcul au niveau du nombre de multiplications [1].

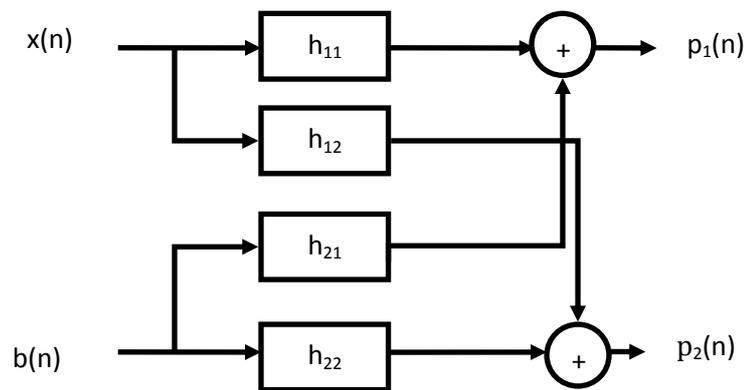
Étape de l'algorithme	Nombre d'addition	Nombre de multiplication
$e(n) = d(n) - \mathbf{x}^T(n)\mathbf{w}(n)$	L	L
$\mathbf{w}(n) = \mathbf{w}(n-1) + \frac{\mu}{\mathbf{x}^T(n)\mathbf{x}(n) + \beta} \mathbf{x}(n)e(n-1)$	L + 1	2L + 1
<b>Total par itération</b>	2L + 1	3L + 1

**Tableau 1.2.** Opérations nécessaires dans l'algorithme NLMS.

## 1.10 Réduction du bruit par les méthodes bi-capteurs adaptatives

Pour la réduction du bruit Il existe plusieurs méthodes en mono-capteur, en bi-capteur et en multi-capteurs. Nous allons travailler par les méthodes bi-capteurs qui sont dernièrement proposées dans le domaine de rehaussement de la parole et la réduction du bruit acoustique.

### 1.10.1 Mélange convolutif bi-capteur



**Figure 1.3.** Structure du mélange convolutif bi-capteur complet.

Dans cette structure, les deux microphones captent deux signaux bruités  $p_1(n)$  et  $p_2(n)$ . D'après la structure passée on a l'équation du mélange qui s'écrit comme suit :

$$p_1(n) = x(n) * h_{11}(n) + b(n) * h_{21}(n) \quad (1.19)$$

$$p_2(n) = b(n) * h_{22}(n) + x(n) * h_{12}(n) \quad (1.20)$$

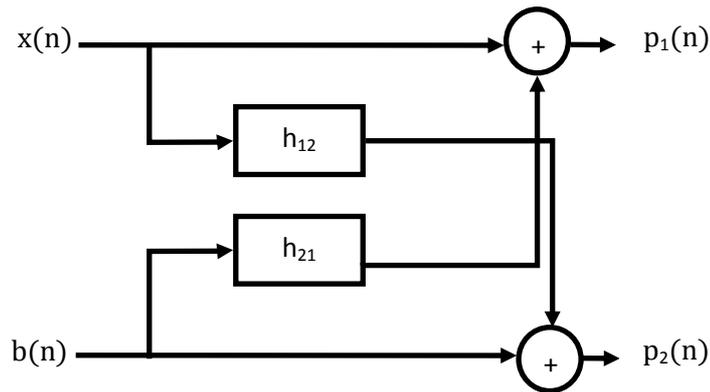
Où :

- $h_{11}(n)$  et  $h_{22}(n)$  représentent les réponses impulsionnelles des canaux directs de couplage.
- $h_{12}(n)$  et  $h_{21}(n)$  représentent les effets du couplage mutuel entre les signaux.
- $x(n)$  est le signal parole.
- $b(n)$  est le signal du bruit.

- $p_1(n)$  et  $p_2(n)$  représentent les deux signaux de la sortie du mélange.

On suppose que la source de la parole est très proche à la sortie  $p_2(n)$  comme quoi le  $h_{11}(n)$  n'existe pas (le  $h_{11}(n)$  devient comme impulsion du Dirac  $\delta(n)$ ) et la même chose pour la source du bruit et que :  $h_{11}(n) = h_{22}(n) = \delta(n)$ .

Donc la structure devient comme suit :



**Figure 1.4.** Structure du mélange convolutif bi-capteur simplifié.

Les deux sorties de la structure du mélange convolutif bi-capteur simplifié sont données par :

$$p_1(n) = b(n) * h_{21}(n) + x(n) \quad (1.21)$$

$$p_2(n) = x(n) * h_{12}(n) + b(n) \quad (1.22)$$

### 1.10.2 Séparation aveugle de sources

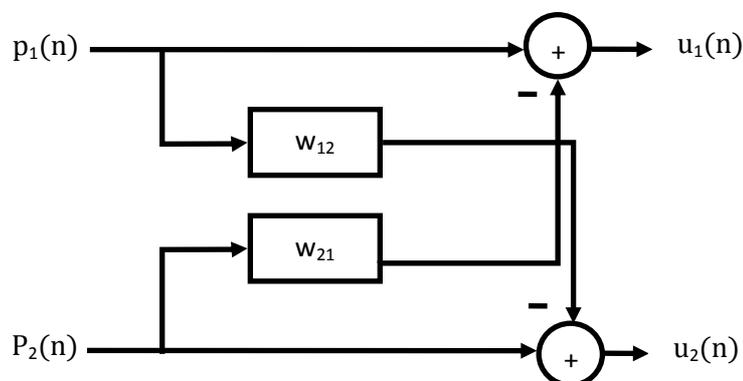
Depuis plusieurs années, la séparation de sources est un sujet de recherche particulièrement actif. Cet engouement peut s'expliquer par le large spectre d'applications; on peut notamment citer les télécommunications, l'acoustique, la sismique, le domaine biomédical. Le cadre aveugle est ici considéré, lequel offre l'avantage de ne requérir que des hypothèses faibles sur le mélange : en dehors de sa structure générale, souvent supposée linéaire, aucun paramètre de ce dernier n'est supposé connu.

Depuis plusieurs années, des méthodes efficaces existent pour la séparation des mélanges linéaires instantanés de sources. Le cas des mélanges convolutif demeure en revanche plus difficile [4]

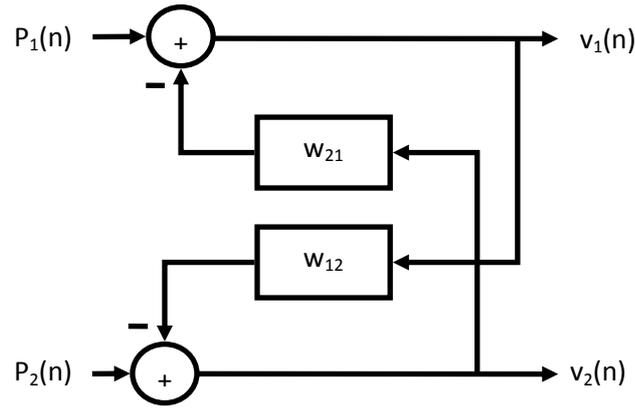
### *a Principe*

Le principe de la séparation aveugle de sources peut être énoncé comme suit : il s'agit de reconstituer les  $N$  signaux sources émis à partir de la seule connaissance de  $N$  signaux d'observations reçus correspondant à des mélanges de ces sources. Le traitement est alors aveugle, on ne possède aucune information sur le système de mélange et les sources sont inobservables. Il est nécessaire d'avoir une hypothèse supplémentaire, c'est pourquoi la plupart des techniques de la séparation aveugle de sources reposent sur l'hypothèse de l'indépendance des sources. Initialement, elle a été modélisée à base d'un simple produit, autrement dit les capteurs reçoivent à chaque instant une combinaison linéaire des signaux sources. Plus tard, des modélisations proches de la réalité ont été introduites. L'une de ces modélisations considère le canal de transmission comme un système reposant sur une opération de filtrage. En d'autres termes, les signaux captés dépendent linéairement à la fois des sources et de leurs versions retardées, Le système de mélange est dit linéaire convolutif. [5]

Le système de séparation devra réaliser une combinaison linéaire convolutive à partir de deux structures bien connues se représentent respectivement dans les figures suivantes :



**Figure 1.5.** Structure Symétrique Direct « Forward ».



**Figure1.6.** Structure Symétrique Réursive « Backward ».

Les deux signaux obtenus par la structure de séparation de sources directe (Forward) sont données par les deux équations:

$$u_1(n) = p_1(n) - p_2(n) * w_{21}(n) \quad (1.23)$$

$$u_2(n) = p_2(n) - p_1(n) * w_{12}(n) \quad (1.24)$$

En remplaçant les expressions  $p_1(n)$  et  $p_2(n)$  dans l'expression de  $u_1(n)$  et nous trouvons :

$$u_1(n) = b(n) * [h_{21}(n) - w_{21}(n)] + x(n) * [\delta(n) - h_{12}(n) * w_{21}(n)] \quad (1.25)$$

$$u_2(n) = b(n) * [h_{12}(n) - w_{12}(n)] + x(n) * [\delta(n) - h_{21}(n) * w_{12}(n)] \quad (1.26)$$

En utilisant l'hypothèse d'optimalité pour les deux filtres adaptatifs

$$(w_{21}^{opt} = h_{21} \text{ Et } w_{12}^{opt} = h_{12}),$$

On obtient les sorties  $u_1(n)$  et  $u_2(n)$  selon les deux expressions suivantes :

$$u_1(n) = x(n) * [\delta(n) - h_{12}(n) * w_{21}(n)] \quad (1.27)$$

$$u_2(n) = b(n) * [\delta(n) - h_{21}(n) * w_{12}(n)] \quad (1.28)$$

Et les sorties et de la structure de séparation de sources indirecte qui est représentée par la figure 1.6 « Backward » sont données par les équations suivantes:

$$v_1(n) = p_1(n) - v_2(n) * w_{21}(n) \quad (1.29)$$

$$v_2(n) = p_2(n) - v_1(n) * w_{12}(n) \quad (1.30)$$

En remplaçant les expressions  $p_1(n)$  et  $p_2(n)$  dans les expressions de  $v_1(n)$  et  $v_2(n)$ .

En utilisant l'hypothèse d'optimalité pour les deux filtres adaptatifs

$$(w_{21}^{\text{opt}} = h_{21} \text{ Et } w_{12}^{\text{opt}} = h_{12}),$$

On obtient les sorties  $v_1(n)$  et  $v_2(n)$  : selon les deux expressions suivantes :

$$v_1(n) = x(n)$$

$$v_2(n) = b(n)$$

Nous notons que les solutions optimales des filtres adaptatifs sont obtenues en utilisant les critères du minimum de l'erreur quadratique moyenne (MMSE ou bien EQMM) [5].

## 1.11 Conclusion

Dans le premier chapitre, nous avons présenté le principe du filtrage adaptatif avec ses paramètres et aussi les algorithmes les plus utilisés pour la réduction du bruit et qui sont le LMS et NLMS. De surcroît nous avons présenté le problème des mélanges convolutifs bi-capteurs entre un signal de la parole et le bruit (complet et simplifié). A la fin, nous avons introduit la notion de rehaussement de la parole et celle de la réduction du bruit, par les deux structures de séparation aveugle de sources (Forward et Backward).

# Chapitre 2 Algorithme Forward de Décorrélation: versions Normalisées avec pas d'adaptations variables

---

## 2.1 Introduction

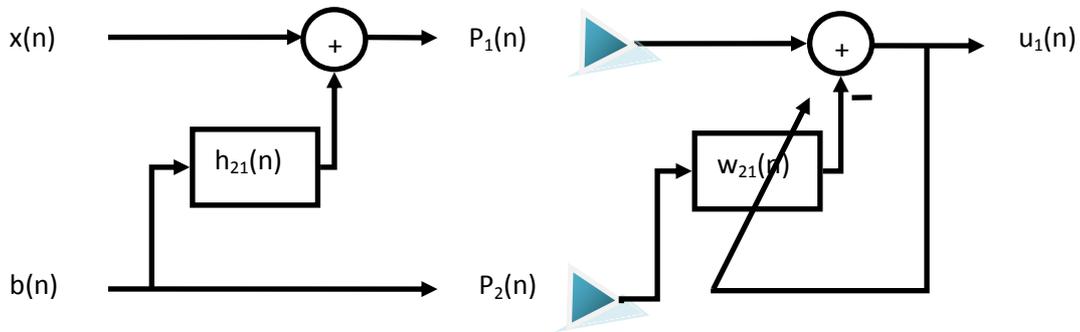
La performance des systèmes de réduction du bruit par le filtrage adaptatif est fortement liée aux signaux de parole et du bruit. Plusieurs techniques ont été proposées dans le domaine de traitement de la parole pour la réduction du bruit acoustique, comme la séparation aveugle de sources. Récemment, plusieurs algorithmes adaptatifs ont été appliqués aux structures de séparation aveugle de sources (Forward et Backward). Parmi ces algorithmes, nous citons l'algorithme de décorrélation symétrique adaptatif Forward (FSAD pour Forward symmetric adaptive decorrelation).

En ce document on discute une approche symétrique basée sur le principe de décorrélation et ceci pour un pas d'adaptation variable et normalisé dans la structure Forward.

On peut montrer que le critère des moindres carrés est équivalent à une décorrélation de l'estimation du signal avec référence bruit seul. Si la référence de bruit contient une partie du signal désiré, un tel critère semble être raisonnable et il serait meilleur si une estimation libre du bruit était disponible [15].

## 2.2 Principe d'annulation du bruit par le critère de Décorrélation

Le fonctionnement de l'algorithme de décorrélation est basé sur la minimisation de l'énergie d'erreur. Ce dernier est équivalent à l'intercorrrelation entre le signal estimé  $u_1(n)$  et le signal du mélange  $p_2(n)$ , voir la Figure 2.1 [1].



**Figure 2.1.** Principe de l'annulation du bruit acoustique.

$$\frac{\partial \varepsilon_1(n)}{\partial w_{21}(l)} = -2 C_{u_1 p_2}(l) \quad (2.1)$$

$$\frac{\partial \varepsilon_1(n)}{\partial w_{21}(l)} = 0 \quad (2.2)$$

Donc :

$$C_{u_1 p_2}(l) = 0 \quad l=0,1,\dots, L$$

L'équation de la mise à jour du filtre  $w_{21}(n)$  est donnée par [15] :

$$w_{21}^{(n)}(l) = w_{21}^{(n-1)}(l) - \gamma \frac{C_{u_1 p_2}(l)}{\nabla_1} \quad (2.3)$$

L'idée de cette algorithme est de remplacer le terme de l'intercorrrelation par ses valeurs instantanées et par un choix approprié de  $\gamma$  ( $0 < \gamma < 2 \rightarrow 0 < \mu < 2/\sigma_2^2$ ) avec  $\sigma_2^2$  est la variance du signal bruit  $p_2$ , on obtient la relation suivante :

$$w_{21}^{(n)}(l) = w_{21}^{(n-1)}(l) + \mu_{21} (u_1(n) p_2(n-l)) \quad (2.4)$$

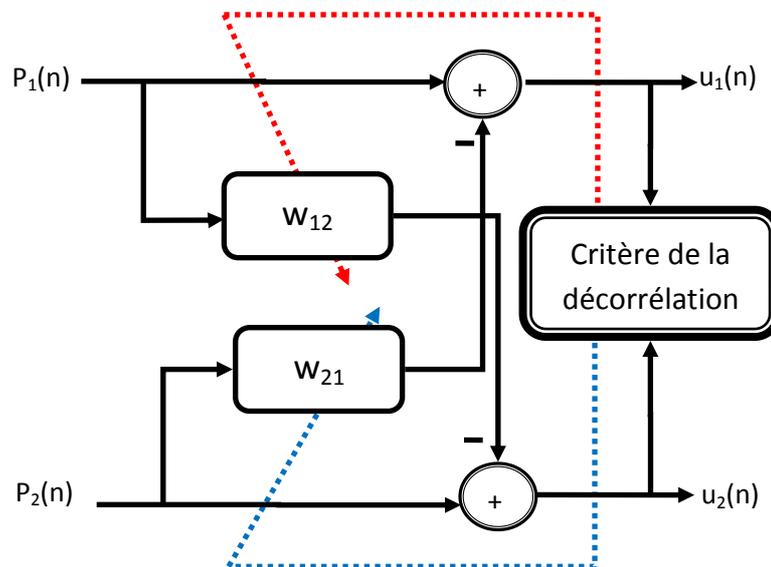
Donc, on peut dire que l'algorithme de décorrrelation symétrique adaptatif est identique à l'algorithme de LMS. [1]

## 2.3 Algorithme de décorrélation symétrique adaptatif Forward (FSAD)

La décorrélation se fait entre une estimation du signal de parole et une estimation du bruit. Le critère des moindres carrés est remplacé par le critère de décorrélation et en raison de sa symétrie complète, l'algorithme est un séparateur de signal plutôt qu'un annuleur de bruit. Il est désigné sous le nom de l'algorithme de décorrélation symétrique adaptatif. Nous supposons que les deux signaux  $x(n)$  et  $b(n)$  sont indépendants, donc, les valeurs du produit de l'intercorrélacion entre les deux sont nulles, ce dernière exigé pour le fonctionnement de l'algorithme SAD [1].

Dans une partie de notre travail, nous étudions l'algorithme SAD avec la structure Forward. Nous rappelons que la décorrélation se fait entre les deux sorties  $u_1(n)$  et  $u_2(n)$ , En utilisant le principe de décorrélation pour mettre à jour les deux filtres adaptatifs.

Dans le modèle de l'annulation du bruit acoustique donné par la Figure 2.1 L'idée de l'algorithme SAD est d'ajouter un filtre symétrique  $w_{12}(n)$ , donc l'algorithme de décorrélation symétrique adaptatif (FSAD) est schématisé par la Figure 2.2.



**Figure 2.2.** Structure d'algorithme de décorrélation symétrique adaptatif Forward.

En adaptant la formule (2.4) de mise à jour à la structure Forward bi-capteurs (SAD), nous obtenons respectivement les deux équations de mise à jour de deux filtres adaptatifs  $w_{12}(n)$  et  $w_{21}(n)$  :

$$w_{21}^{(n)}(l) = w_{21}^{(n-1)}(l) + \mu_{21} (u_1(l) u_2(n-l)) \quad (2.5)$$

$$w_{12}^{(n)}(l) = w_{12}^{(n-1)}(l) + \mu_{12} (u_2(l) u_1(n-l)) \quad (2.6)$$

Avec :  $l = 0, 1, \dots, L-1$ , ou  $L$  c'est la taille du filtre.

## 2.4 Algorithme FSAD normalisé (FNSAD)

Dans cette section, nous allons présenter l'analyse de l'algorithme FSAD normalisé (FNSAD) qu'est utilisé pour l'annulation du bruit et le rehaussement de la parole. En prenant le modèle de l'annulation du bruit présenté dans la Figure 2.1, les deux signaux observés  $p_1(n)$  et  $p_2(n)$  sont donnés par :

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

Avec  $p_1$  est le signal de la parole et  $p_2$  le bruit.

$$p_2(n) = b(n) \quad (2.8)$$

A la sortie du modèle de l'annuleur du bruit, nous définissons le signal de parole estimé  $u_1(n)$  et l'erreur a posteriori  $e_1(n)$  par :

$$u_1(n) = p_1(n) - \mathbf{p}_2^T(n) \mathbf{w}_{21}(n-1) \quad (2.9)$$

$$e_1(n) = p_1(n) - \mathbf{p}_2^T(n) \mathbf{w}_{21}(n) \quad (2.10)$$

L'équation (2.4) peut être s'écrire sous la forme suivante :

$$\mathbf{w}_{21}(n) = \mathbf{w}_{21}(n-1) + \mu_{21} u_1(n) \mathbf{p}_2(n) \quad (2.11)$$

Où  $\mathbf{p}_2(n) = [p_2(n), p_2(n-1), \dots, p_2(n-L+1)]^T$

Et  $\mathbf{w}_{21}(n) = [w_{21,0}(n), w_{21,1}(n), \dots, w_{21,L-1}(n)]^T$

Nous utilisons l'équation (2.9), et par l'insertion de l'équation (2.11) dans (2.10) avec la condition :  $e_1(n) = 0$

Seulement dans les périodes de silence, nous obtenons l'équation suivante :

$$e_1(n) = p_1(n) - \mathbf{p}_2^T(n)\mathbf{w}_{21}(n-1) - \mathbf{p}_2^T(n)[\mathbf{w}_{21}(n-1) + \mu_{21}u_1(n)\mathbf{p}_2(n)] = 0 \quad (2.12)$$

$$u_1(n)[1 - \mu_{21}\mathbf{p}_2^T(n)\mathbf{p}_2(n)] = 0 \quad (2.13)$$

Donc le pas d'adaptation normalisé est donné par:

$$\mu_{21} = \frac{1}{\mathbf{p}_2^T(n)\mathbf{p}_2(n)} \quad (2.14)$$

En appliquant le pas d'adaptation normalisé sur les deux équations (2.5) et (2.6) de l'algorithme de décorrélation symétrique adaptatif et nous insérons les nouveaux pas d'adaptation. Donc les deux équations de mise à jour de deux filtres adaptatifs  $\mathbf{w}_{21}(n)$  et  $\mathbf{w}_{12}(n)$  de l'algorithme proposé (FNSAD) sont données respectivement par :

$$\mathbf{w}_{21}(n) = \mathbf{w}_{21}(n-1) + \mu_{21,n}u_1(n) \frac{\mathbf{p}_2(n)}{\mathbf{p}_2^T(n)\mathbf{p}_2(n) + \beta} \quad (2.15)$$

$$\mathbf{w}_{12}(n) = \mathbf{w}_{12}(n-1) + \mu_{12,n}u_2(n) \frac{\mathbf{p}_1(n)}{\mathbf{p}_1^T(n)\mathbf{p}_1(n) + \beta} \quad (2.16)$$

où  $\beta$  est une constante positive utilisée pour éviter la division sur zéro,  $\mu_{12,n}$  et  $\mu_{21,n}$  sont les pas d'adaptations normalisés de deux filtres adaptatifs de l'algorithme (FNSAD).

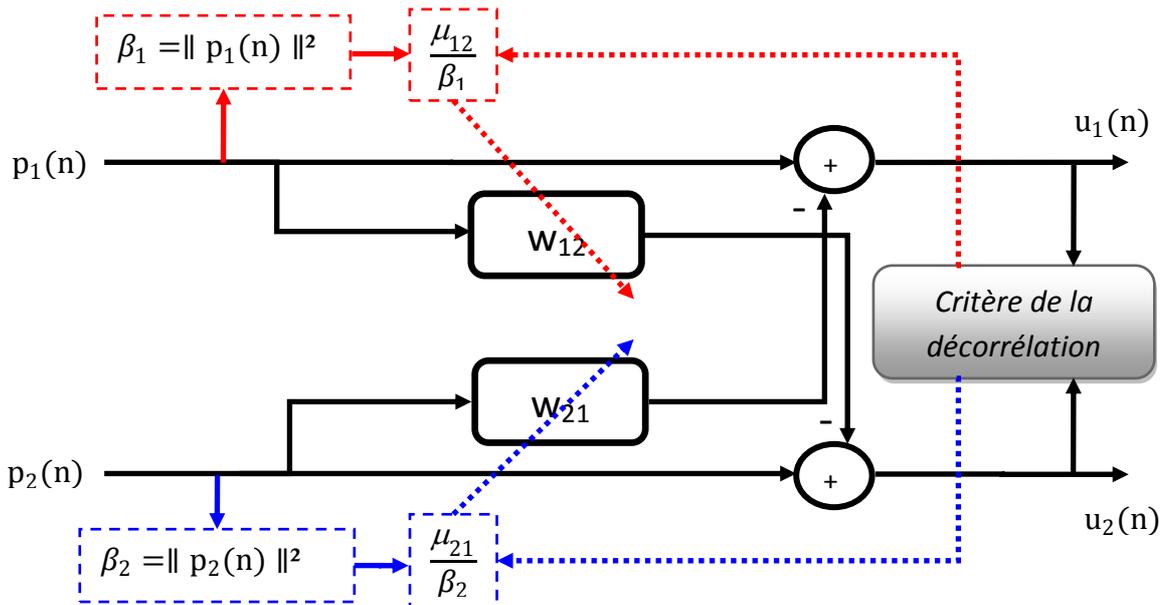


Figure 2.3. Algorithme FSAD normalisé (FNSAD).

## 2.5 Algorithmes FNSAD avec pas d'adaptation variable

La condition de stabilité et de convergence de l'algorithme FNSAD est donnée par :

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

Selon les études effectuées dans la littérature, le choix du pas d'adaptation est critique pour le bon fonctionnement des algorithmes adaptatifs :

- plus le pas d'adaptation  $\mu_{21,n}$  est grand, plus la vitesse de convergence est rapide. Cela se traduit par des fluctuations importantes autour de la trajectoire moyenne.
- plus  $\mu_{21,n}$  est faible, plus la convergence est lente, mais la variance est faible, la trajectoire sera déterministe.

Certains algorithmes avec pas d'adaptation variable sont proposés pour résoudre les deux problèmes, la vitesse de convergence et des fluctuations [14]. Leur principe est le suivant :

$\mu$  est choisi grand au départ afin de converger rapidement vers un optimum. Puis, plus l'erreur diminue, plus on diminue le pas d'adaptation afin d'obtenir une meilleure précision.

Dans les sous-sections suivantes, nous allons présenter quatre versions de l'algorithme FNSAD à pas d'adaptations variables.

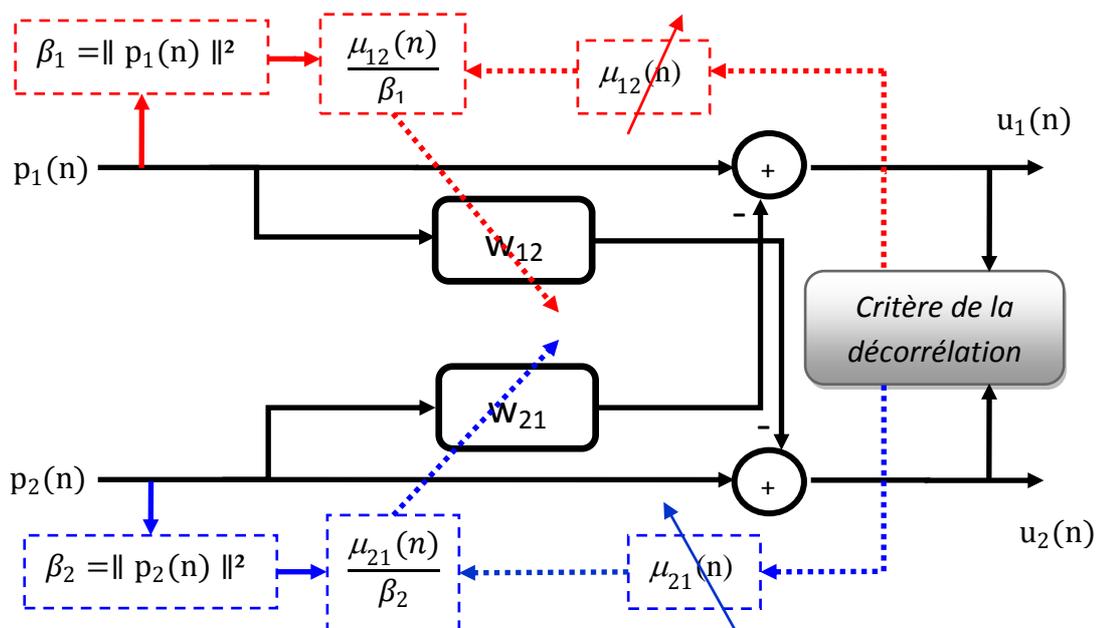


Figure 2.4. Algorithme FNSAD avec pas d'adaptations variables (VS-FNSAD).

Les deux équations de la mise à jour de deux filtres adaptatifs sont données par :

$$\mathbf{w}_{12}(n) = \mathbf{w}_{12}(n-1) + \mu_{12,n}(n) u_2(n) \frac{\mathbf{u}_1(n)}{\mathbf{p}_1(n) \mathbf{p}_1^T(n) + \beta} \quad (2.17)$$

$$\mathbf{w}_{21}(n) = \mathbf{w}_{21}(n-1) + \mu_{21,n}(n) u_1(n) \frac{\mathbf{u}_2(n)}{\mathbf{p}_2(n) \mathbf{p}_2^T(n) + \beta} \quad (2.18)$$

Dans les quatre versions de VS-FNSAD, les nouveaux pas d'adaptations  $\mu_{12,n}(n)$  et  $\mu_{21,n}(n)$  sont contrôlés par les nouveaux pas de contrôles  $\lambda_{12}(n)$  et  $\lambda_{21}(n)$  qui sont contrôlés par les relations suivantes:

$$\mu_{12,n}(n) = \begin{cases} \mu_{12,\max} & \text{si } \lambda_{12}(n) > \mu_{12,\max} \\ \mu_{12,\min} & \text{si } \lambda_{12}(n) < \mu_{12,\min} \\ \lambda_{12}(n) & \text{Ailleurs} \end{cases}$$

$$\mu_{21,n}(n) = \begin{cases} \mu_{21,\max} & \text{si } \lambda_{21}(n) > \mu_{21,\max} \\ \mu_{21,\min} & \text{si } \lambda_{21}(n) < \mu_{21,\min} \\ \lambda_{21}(n) & \text{Ailleurs} \end{cases}$$

### 2.5.1 Algorithme VS-FNSAD 1 :

Pour améliorer l'intelligibilité du signal de parole estimée en sortie de l'algorithme FNSAD, nous proposons une modification simple en adaptant la technique de pas variable [14] aux deux filtres de l'algorithme FNSAD.

La mise à jour des pas d'adaptations de contrôles  $\lambda_{12}(n)$  et  $\lambda_{21}(n)$  sont données par les deux équations suivantes :

$$\lambda_{12}(n) = \alpha \lambda_{12}(n-1) + \gamma e_1^2(n) \quad (2.19)$$

$$\lambda_{21}(n) = \alpha \lambda_{21}(n-1) + \gamma e_2^2(n) \quad (2.20)$$

Cet algorithme utilise les erreurs instantanées au carré pour mettre à jour les valeurs de  $\lambda_{12}(n)$  et  $\lambda_{21}(n)$ . Lorsque l'erreur est grande, le pas d'adaptation va augmenter, ce qui se conduit à une vitesse de convergence rapide, quand l'erreur diminue, le pas d'adaptation devient plus petit, la variance résiduelle devient faible.

Avec :

$$e_1^2(n) = [u_1(n)]^2 \quad (2.21)$$

$$e_2^2(n) = [u_2(n)]^2 \quad (2.22)$$

Donc La mise à jour des pas d'adaptations sont données par les équations suivantes :

$$\lambda_{12}(n) = \alpha \lambda_{12}(n-1) + \gamma [u_1(n)]^2 \quad (2.23)$$

$$\lambda_{21}(n) = \alpha \lambda_{21}(n-1) + \gamma [u_2(n)]^2 \quad (2.24)$$

Où  $\alpha$  et  $\gamma$  sont des paramètres positifs de contrôle qui sont données comme suite :

$0 < \alpha < 1$  Et  $\gamma > 0$ , et pour assurer le bon fonctionnement de l'algorithme VS-FNSAD, pour chaque itération  $\mu$  doit être borné entre  $[\underline{\mu}_{\min}, \underline{\mu}_{\max}]$ .

### 2.5.2 Algorithme VS-FNSAD 2:

Cet algorithme est une dérivée de l'algorithme précédant [17], le pas d'adaptation est ajusté en utilisant l'autocorrélation entre  $u_1(n)$  et sa valeur passée  $u_1(n-1)$ .

La mise à jour du pas d'adaptation de contrôle est donnée par l'équation suivante :

$$\lambda_{12}(n) = \alpha \lambda_{12}(n-1) + \gamma k_1^2(n) \quad (2.25)$$

$$\lambda_{21}(n) = \alpha \lambda_{21}(n-1) + \gamma k_2^2(n) \quad (2.26)$$

Avec :

$$k_1(n) = \beta k_1(n-1) + (1 - \beta) u_1(n) u_1(n-1) \quad (2.27)$$

$$k_2(n) = \beta k_2(n-1) + (1 - \beta) u_2(n) u_2(n-1) \quad (2.28)$$

Où  $\alpha$  et  $\gamma$  sont des paramètres positifs de contrôle qui sont données comme suite :  
 $0 < \alpha < 1$  et  $\gamma > 0$ , et  $0 < \beta < 1$ . Pour assurer le bon fonctionnement de  
l'algorithme VS-FNSAD, pour chaque itération le pas d'adaptation doit être borné entre  
 $[\mu_{\min}, \mu_{\max}]$ .

### 2.5.3 Algorithme VS-FNSAD 3 :

Dans cet algorithme, les pas d'adaptations variables sont évalués par les deux équations suivantes:

$$\lambda_{12}(n) = \alpha \lambda_{12}(n-1) + \gamma k_1^2(n) \quad (2.29)$$

$$\lambda_{21}(n) = \alpha \lambda_{21}(n-1) + \gamma k_2^2(n) \quad (2.30)$$

Comme dans l'algorithme précédent, le pas d'adaptation est ajusté en utilisant l'autocorrélation entre  $u_1(n)$  et sa valeur passée  $u_1(n-1)$ . Dans l'algorithme VS-FNSAD 3, en basant toujours sur la même idée mais en utilisant la fonction d'autocorrélation entre les L valeurs passées du signal d'erreur (avec L est la longueur de la réponse impulsionnelle). [18]

Considérant  $k_1(n)$  et  $k_2(n)$  comme l'estimation de la fonction de corrélation quadratique moyenne entre  $u(n)$  et ses valeurs passées  $u(n-1), u(n-2), \dots, u(n-L)$ . Donc les équations de mise à jour de  $k_1(n)$  et  $k_2(n)$  sont données par :

$$k_1(n) = \beta k_1(n-1) + (1-\beta) \sum_{l=1}^L u_1(n) u_1(n-l) \quad (2.31)$$

$$k_2(n) = \beta k_2(n-1) + (1-\beta) \sum_{l=1}^L u_2(n) u_2(n-l) \quad (2.32)$$

### 2.5.4 Algorithme VS-NFSAD 4 :

Dans cette partie, nous allons présenter les différentes formules du quatrième algorithme VS-FNSAD 4, en utilisant un nouvel critère pour estimer les deux pas d'adaptations variables [13], Ce critère est donné par les formules suivantes :

$$\mu_{12}(n) = \mu_{12_{\max}} \frac{\|\mathbf{k}_1(n)\|^2}{\|\mathbf{k}_1(n)\|^2 + \delta_1} \quad (2.33)$$

$$\mu_{21}(n) = \mu_{21_{\max}} \frac{\|\mathbf{k}_2(n)\|^2}{\|\mathbf{k}_2(n)\|^2 + \delta_2} \quad (2.34)$$

Où  $\delta_1$  et  $\delta_2$  sont des nombres positifs  $\mathbf{k}_1(n)$  et  $\mathbf{k}_2(n)$  sont des vecteurs de dimension  $(L \times 1)$  qui sont donnés par :

$$\mathbf{k}_1(n) = \mathbf{k}_1(n-1) + (1-\beta) u_2(n) \frac{\mathbf{u}_1(n)}{\mathbf{p}_1(n) \mathbf{p}_1^T(n) + \beta} \quad (2.35)$$

$$\mathbf{k}_2(n) = \mathbf{k}_2(n-1) + (1-\beta) u_1(n) \frac{\mathbf{u}_2(n)}{\mathbf{p}_2(n) \mathbf{p}_2^T(n) + \beta} \quad (2.36)$$

## 2.6 Conclusion :

Dans ce chapitre, nous avons présenté la théorie de l'algorithme de décorrélation symétrique adaptatif (SAD) pour la réduction du bruit et la séparation de sources, comme nous avons vu aussi la Structure Forward et ses différents algorithmes de décorrélation principaux pour un pas d'adaptation fixe et avec pas d'adaptations variables pour les quatre algorithmes proposés.

# Chapitre 3 Algorithme Backward de Décorrélation: Versions Normalisé avec pas d'adaptations variables

---

## 3.1 Introduction

Dans ce chapitre, on discute une approche symétrique basée sur le principe de décorrélation symétrique adaptative SAD avec pas d'adaptations variables et normalisés appliquée sur la structure inverse (Backward).

## 3.2 Algorithme de Décorrélation Symétrique Adaptative Backward (BSAD)

Dans l'algorithme de décorrélation adaptative symétrique Backward de base (BSAD), Nous supposons que les signaux de sources  $s(n)$  et  $b(n)$  sont statistiquement indépendants. Cela implique qu'ils sont non corrélés, c'est-à-dire:  $E[x(n)b(n-1)] = 0$ . La description de l'algorithme BSAD est présentée dans la Figure 3.1.

Le critère de performance de cet algorithme BSAD est de minimiser l'énergie des signaux de sortie estimés,  $v_1(n)$  et  $v_2(n)$ ,  $E[v_1^2(n)]$  et  $E[v_2^2(n)]$  respectivement.

$E[v_1^2(n)]$  et  $E[v_2^2(n)]$  Représentent la surface d'erreur quadratique avec une seule solution d'optimum. Dans [16], il a été prouvé que l'erreur quadratique est complètement équivalente à la minimisation de décorrélation entre les signaux de sortie estimés  $v_1(n)$  et  $v_2(n)$  avec la référence de bruit présente sur l'observation  $p_2(n)$  et  $p_1(n)$  respectivement sur l'étendue des filtres adaptatifs  $w_{12}(n)$  et  $w_{21}(n)$ :

$$C_{v_2v_1}(l) = E[v_1(n)v_2(n-l)] \text{ et } C_{v_1v_2}(l) = E[v_2(n)v_1(n-l)]$$

Les formules de base de l'algorithme BSAD sont obtenues lorsque nous mettons  $C_{v_1v_2}(l) = 0$  et  $C_{v_2v_1}(l) = 0$ , Les relations exactes de mise à jour des filtres adaptatifs  $w_{12}(n)$  et  $w_{21}(n)$  par l'algorithme BSAD sont données comme suit :

$$\mathbf{w}_{12}(n) = \mathbf{w}_{12}(n-1) + \mu_{12} \mathbf{v}_1(n)v_2(n) \quad (3.1)$$

$$\mathbf{w}_{21}(n) = \mathbf{w}_{21}(n-1) + \mu_{21} v_1(n)\mathbf{v}_2(n) \quad (3.2)$$

Avec :

$$\mathbf{v}_1(n) = [v_1(n) v_1(n-1), \dots, v_1(n-L+1)]^T \quad (3.3)$$

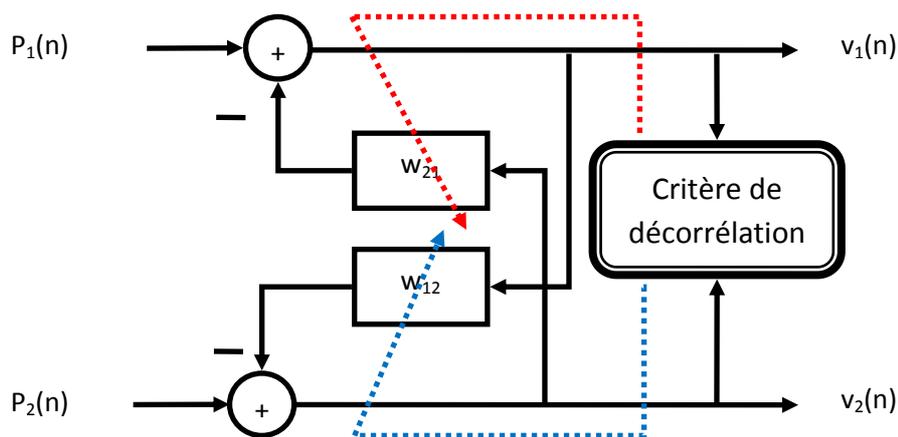
Et

$$\mathbf{v}_2(n) = [v_2(n)v_2(n-1), \dots, v_2(n-L+1)]^T \quad (3.4)$$

Où  $\mu_{12}$  et  $\mu_{21}$  représentent les pas d'adaptations fixes et de contrôle de l'algorithme BSAD. Ils sont choisis en fonction des relations données :

$$0 < \mu_{12} < 2/\sigma_1^2 \text{ Et } 0 < \mu_{21} < 2/\sigma_2^2 .$$

Où  $\sigma_1^2$  et  $\sigma_2^2$  représente la variance des deux entrées les signaux  $v_1(n)$  et  $v_2(n)$  [9, 10].



**Figure 3.1.** Structure d'algorithme de décorrélation symétrique adaptatif Backward (BSAD).

### 3.3 Algorithme BSAD normalisé (BNSAD)

Dans cette partie, nous allons concentrer notre intérêt sur l'algorithme SAD Backward avec pas d'adaptations normalisés.

Dans la sortie de l'algorithme BSAD présenté dans la figure 3.1, nous pouvons définir respectivement le signal de parole estimé  $v_1(n)$ , et le signal d'erreur a posteriori  $e_1(n)$  :

$$v_1(n) = p_1(n) - \mathbf{w}_{21}^T(n-1) \mathbf{v}_2(n) \quad (3.5)$$

$$e_1(n) = p_1(n) - \mathbf{w}_{21}^T(n) \mathbf{v}_2(n) \quad (3.6)$$

Avec la formule de mise à jour du filtre adaptatif  $\mathbf{w}_{21}(n)$  par le L'algorithme BSAD est donné comme suit :

$$\mathbf{w}_{21}(n) = \mathbf{w}_{21}(n-1) + \eta_{21} v_1(n) \mathbf{v}_2(n) \quad (3.7)$$

En utilisant (3.5), et en insérant (3.7) dans (3.6), on obtient,

$$e_1(n) = v_1(n) - \mathbf{w}_{21}^T(n-1) \mathbf{v}_2(n) - \mathbf{v}_{21}^T(n) [\mathbf{w}_{21}(n-1) + \eta_{21} v_1(n) \mathbf{v}_2(n)] \quad (3.8)$$

Seulement, dans les segments de bruit présents (silence périodes de parole) :  $e_1(n) = 0$ , nous obtenons :

$$v_1(n) [1 - \eta_{21} \mathbf{v}_2^T(n) \mathbf{v}_2(n)] = 0 \quad (3.9)$$

En se basant sur (3.9) et en supposant que  $v_1(n) \neq 0 \rightarrow [1 - \eta_{21} \mathbf{v}_2^T(n) \mathbf{v}_2(n)] = 0$

$\eta_{21}$  Est donnée par :

$$\eta_{21} = \frac{1}{\mathbf{v}_2^T(n) \mathbf{v}_2(n)} \quad (3.10)$$

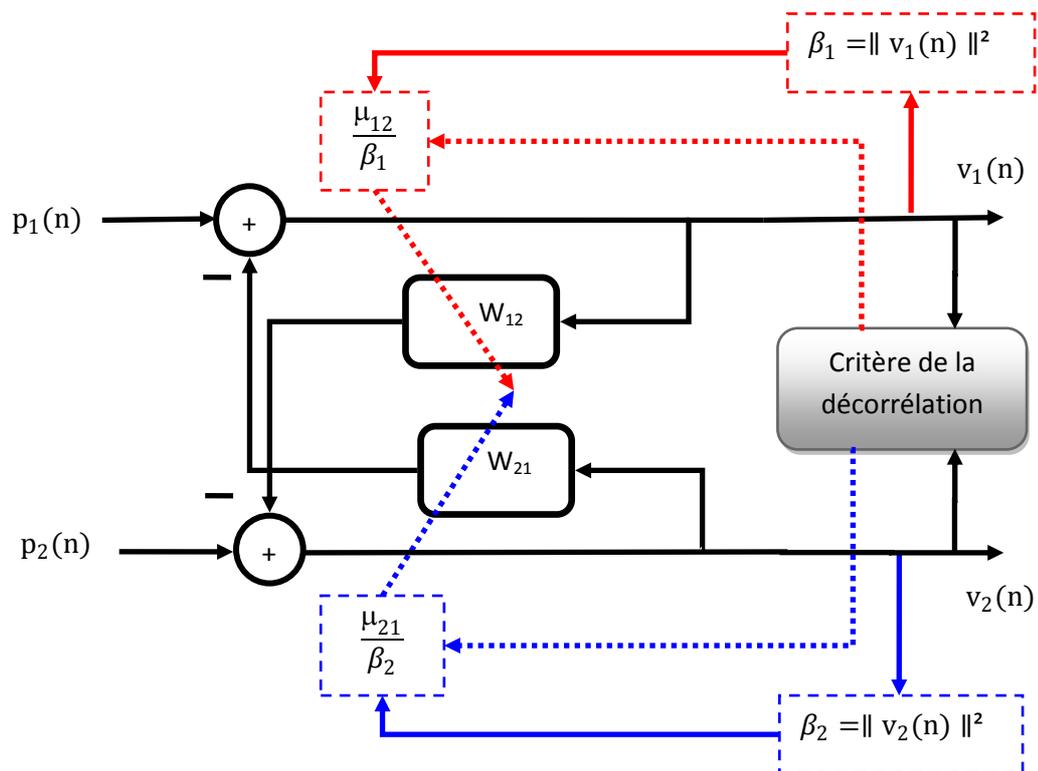
En insérant [(3.10) dans (3.1), (3.9) dans (3.2)] et en incorporant respectivement les nouveaux pas d'adaptations normalisés, notés: " $\mu_{12,n}$  et  $\mu_{21,n}$ ", les deux formules de mise à jour de  $\mathbf{w}_{21}(n)$  et  $\mathbf{w}_{12}(n)$  dans le algorithme de décorrélation normalisée proposé sont données par :

$$\mathbf{w}_{12}(n) = \mathbf{w}_{12}(n-1) + \mu_{12} \frac{v_2(n) \mathbf{v}_1(n)}{\mathbf{v}_1^T(n) \mathbf{v}_1(n) + \beta} \quad (3.11)$$

$$\mathbf{w}_{21}(n) = \mathbf{w}_{21}(n-1) + \mu_{21} \frac{\mathbf{v}_1(n) \mathbf{v}_2(n)}{\mathbf{v}_2^T(n) \mathbf{v}_2(n) + \beta} \quad (3.12)$$

Dans l'algorithme BNSAD modifié, nous avons proposé de normaliser les deux pas d'adaptations  $\mu_{12}$  et  $\mu_{21}$  respectivement par les signaux d'entrées d'énergies  $\mathbf{v}_1^T(n) \mathbf{v}_1(n)$  et  $\mathbf{v}_2^T(n) \mathbf{v}_2(n)$  des filtres  $\mathbf{w}_{12}(n)$  et  $\mathbf{w}_{21}(n)$ .

$\mu_{12}$  et  $\mu_{21}$  prennent leurs valeurs entre 0 et 2 pour garantir la convergence de  $\mathbf{w}_{12}(n)$  et  $\mathbf{w}_{21}(n)$ .



**Figure 3.2.** Structure d'algorithme BNSAD avec pas d'adaptation fixe.

### 3.4 Algorithmes BNSAD avec pas d'adaptations variables

Dans la figure 3.2, nous décrivons le diagramme global des algorithmes BNSAD à pas d'adaptations variables. Dans la nouvelle itération de ces algorithmes proposés, nous mettons à jour les deux pas d'adaptations variables qui sont appliquées à la structure Backward pour estimer les signaux de sources. L'objectif major de l'algorithme BNSAD avec pas d'adaptations variables est d'améliorer (augmenter) le taux de convergence

des filtres adaptatifs et en réduisant la distorsion de l'estimé signaux de sortie (minimisant l'erreur d'estimation en régime permanent).

Les équations de mise à jour des filtres adaptatifs  $w_{12}(n)$  et  $w_{21}(n)$  par les algorithmes NBSAD avec pas variables sont données par :

$$\mathbf{w}_{12}(n) = \mathbf{w}_{12}(n-1) + \mu_{12,n}(n) \frac{v_2(n) \mathbf{v}_1(n)}{\mathbf{v}_1^T(n) \mathbf{v}_1(n) + \beta} \quad (3.13)$$

$$\mathbf{w}_{21}(n) = \mathbf{w}_{21}(n-1) + \mu_{21,n}(n) \frac{v_1(n) \mathbf{v}_2(n)}{\mathbf{v}_2^T(n) \mathbf{v}_2(n) + \beta} \quad (3.14)$$

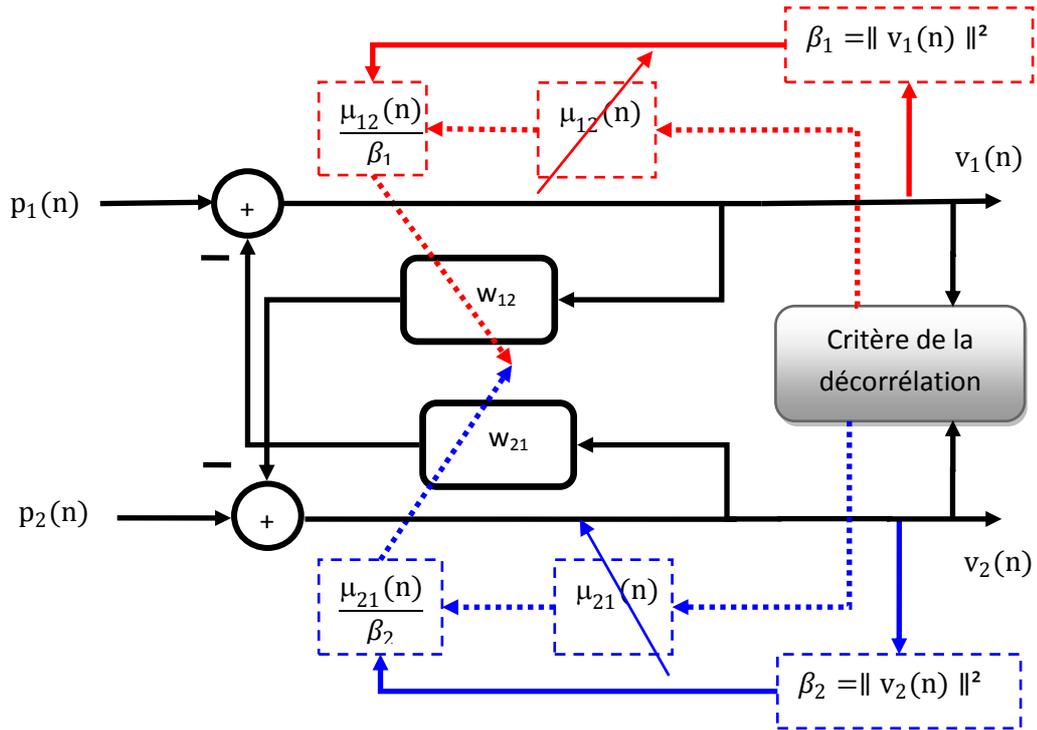
Où  $\mu_{12,n}(n)$  et  $\mu_{21,n}(n)$  présentent respectivement les pas variables des filtres adaptatifs  $w_{12}(n)$  et  $w_{21}(n)$ .

Dans la partie suivante, nous présentons les algorithmes BNSAD avec pas d'adaptation variables proposé qui reposent principalement sur la minimisation de  $\mu_{12,n}(n)$  et  $\mu_{21,n}(n)$  en utilisant des estimations récursives. Notons que les valeurs de ces pas variables sont limitées par ( $\mu_{12,\min}$  et  $\mu_{12,\max}$ ,  $\mu_{21,\min}$  et  $\mu_{21,\max}$ ) pour garantir la convergence adaptative des filtres de  $w_{12}(n)$  et  $w_{21}(n)$  respectivement, c'est-à-dire :

$$0 < \mu_{12,\min} < \mu_{12,n}(n) < \mu_{12,\max} \leq 2 \quad (3.15)$$

$$0 < \mu_{21,\min} < \mu_{21,n}(n) < \mu_{21,\max} \leq 2 \quad (3.16)$$

Où  $\mu_{12,\max}$  et  $\mu_{21,\max}$ , sont sélectionnés pour obtenir le taux de convergence maximum des filtres adaptatifs, et  $\mu_{12,\min}$  et  $\mu_{21,\min}$  sont choisis en fonction des niveaux désirés dans les signaux de sortie  $v_1(n)$  et  $v_2(n)$ .



**Figure 3.3.** Structure d'algorithme BNSAD avec pas d'adaptations variables (VS-BNSAD).

### 3.4.1 Algorithme VS-BNSAD 1 :

Pour améliorer l'intelligibilité du signal de parole estimée en sortie de l'algorithme BND, nous proposons une modification simple en adaptant et en appliquant la technique du pas variable aux deux canaux d'Algorithme BNSAD. Le VS-BNSAD proposé est maintenant contrôlé par les nouvelles tailles des pas variables  $\mu_{12,n}(n)$  et  $\mu_{21,n}(n)$  qui sont contrôlés par les relations suivantes:

$$\mu_{12,n}(n) = \begin{cases} \mu_{12,\max} & \text{si } \lambda_{12}(n) > \mu_{12,\max} \\ \mu_{12,\min} & \text{si } \lambda_{12}(n) < \mu_{12,\min} \\ \lambda_{12}(n) & \text{Ailleurs} \end{cases}$$

$$\mu_{21,n}(n) = \begin{cases} \mu_{21,\max} & \text{si } \lambda_{21}(n) > \mu_{21,\max} \\ \mu_{21,\min} & \text{si } \lambda_{21}(n) < \mu_{21,\min} \\ \lambda_{21}(n) & \text{Ailleurs} \end{cases}$$

La mise à jour des pas d'adaptations de contrôles  $\lambda_{12}(n)$  et  $\lambda_{21}(n)$  sont données par les deux équations suivantes :

$$\lambda_{12}(n) = \alpha \lambda_{12}(n-1) + \gamma e_1^2(n) \quad (3.17)$$

$$\lambda_{21}(n) = \alpha \lambda_{21}(n-1) + \gamma e_2^2(n) \quad (3.18)$$

Cet algorithme utilise les erreurs instantanées au carré pour mettre à jour la valeur de  $\lambda_{12}(n)$  et  $\lambda_{21}(n)$ . Lorsque l'erreur est grande, le pas d'adaptation va augmenter, ce qui conduit à une vitesse de convergence rapide ; quand l'erreur diminue, le pas d'adaptation devient plus petit, la variance résiduelle devient faible.

L'équation de l'erreur:

$$e_1^2(n) = [v_1(n)]^2 \quad (3.19)$$

$$e_2^2(n) = [v_2(n)]^2 \quad (3.20)$$

Donc, la mise à jour des pas d'adaptations est donnée par les équations suivantes :

$$\lambda_{12}(n) = \alpha \lambda_{12}(n-1) + \gamma [v_1(n)]^2 \quad (3.21)$$

$$\lambda_{21}(n) = \alpha \lambda_{21}(n-1) + \gamma [v_2(n)]^2 \quad (3.22)$$

Où  $\alpha$  et  $\gamma$  sont des paramètres positifs de contrôle qui sont données comme suit :

$0 < \alpha < 1$  et  $\gamma > 0$ , et pour assurer le bon fonctionnement de l'algorithme VS-BNSAD 1, pour chaque itération du pas d'adaptation doit être borné entre  $[\mu_{\min}, \mu_{\max}]$ .

Les deux équations de mise à jour des filtres adaptatifs sont données par :

$$\mathbf{w}_{12}(n) = \mathbf{w}_{12}(n-1) + \lambda_{12}(n) v_2(n) \frac{\mathbf{v}_1(n)}{\mathbf{v}_1(n) \mathbf{v}_1^T(n) + \beta} \quad (3.23)$$

$$\mathbf{w}_{21}(n) = \mathbf{w}_{21}(n-1) + \lambda_{21}(n) v_1(n) \frac{\mathbf{v}_2(n)}{\mathbf{v}_2(n) \mathbf{v}_2^T(n) + \beta} \quad (3.24)$$

### 3.4.2 Algorithme VS-BNSAD 2 :

Cet algorithme est une dérivée de l'algorithme précédant, le pas d'adaptation est ajusté en utilisant l'autocorrélation entre  $v_1(n)$  et sa valeur passée  $v_1(n-1)$ . Le pas d'adaptation variable est évalué par les équations suivantes :

$$\mu_{12,n}(n) = \begin{cases} \mu_{12,\max} & \text{si } \lambda_{12}(n) > \mu_{12,\max} \\ \mu_{12,\min} & \text{si } \lambda_{12}(n) < \mu_{12,\min} \\ \lambda_{12}(n) & \text{Ailleurs} \end{cases}$$

$$\mu_{21,n}(n) = \begin{cases} \mu_{21,\max} & \text{si } \lambda_{21}(n) > \mu_{21,\max} \\ \mu_{21,\min} & \text{si } \lambda_{21}(n) < \mu_{21,\min} \\ \lambda_{21}(n) & \text{Ailleurs} \end{cases}$$

La mise à jour des pas d'adaptations est donnée par les équations suivantes :

$$\lambda_{12}(n) = \alpha \lambda_{12}(n-1) + \gamma k_1^2(n) \quad (3.25)$$

$$\lambda_{21}(n) = \alpha \lambda_{21}(n-1) + \gamma k_2^2(n) \quad (3.26)$$

Avec :

$$k_1(n) = \beta k_1(n-1) + (1 - \beta) v_1(n) v_1(n-1) \quad (3.27)$$

$$k_2(n) = \beta k_2(n-1) + (1 - \beta) v_2(n) v_2(n-1) \quad (3.28)$$

Où  $\alpha$  et  $\gamma$  sont des paramètres positifs de contrôle qui sont donnés comme suit :  $0 < \alpha < 1$  et  $\gamma > 0$ , et pour assurer le bon fonctionnement de l'algorithme VS-BNSAD, pour chaque itération  $\mu_n$  doit être borné entre  $[\mu_{\min}, \mu_{\max}]$ .

Les équations de mise à jour du filtre adaptatif sont données par :

$$\mathbf{w}_{12}(n) = \mathbf{w}_{12}(n-1) + \mu_{12,n}(n) \mathbf{v}_2(n) \frac{\mathbf{v}_1(n)}{\mathbf{v}_1(n) \mathbf{v}_1^T(n) + \beta} \quad (3.29)$$

$$\mathbf{w}_{21}(n) = \mathbf{w}_{21}(n-1) + \mu_{21,n}(n) \mathbf{v}_1(n) \frac{\mathbf{v}_2(n)}{\mathbf{v}_2(n) \mathbf{v}_2^T(n) + \beta} \quad (3.30)$$

### 3.4.3 Algorithme VS-BNSAD 3 :

$$\mu_{12}(n) = \begin{cases} \mu_{12,\max} & \text{si } \lambda_{12}(n) > \mu_{12,\max} \\ \mu_{12,\min} & \text{si } \lambda_{12}(n) < \mu_{12,\min} \\ \lambda_{12}(n) & \text{Ailleurs} \end{cases}$$

$$\mu_{21}(n) = \begin{cases} \mu_{21,\max} & \text{si } \lambda_{21}(n) > \mu_{21,\max} \\ \mu_{21,\min} & \text{si } \lambda_{21}(n) < \mu_{21,\min} \\ \lambda_{21}(n) & \text{Ailleurs} \end{cases}$$

Dans cet algorithme, Les pas d'adaptations variables sont évalués par les deux équations suivantes :

$$\lambda_{12}(n) = \alpha \lambda_{12}(n-1) + \gamma k_1^2(n) \quad (3.31)$$

$$\lambda_{21}(n) = \alpha \lambda_{21}(n-1) + \gamma k_2^2(n) \quad (3.32)$$

Comme dans l'algorithme VS-BNSAD 2, le pas d'adaptation est ajusté en utilisant l'autocorrélation entre  $v_1(n)$  et sa valeur passée  $v_1(n-1)$ . Dans l'algorithme VS-BNSAD 3 proposé, en se basant toujours sur la même idée mais en utilisant la fonction d'autocorrélation entre les L valeurs passées du signal d'erreur (avec L est la longueur de la réponse impulsionnelle).

Considérant  $k_1(n)$  et  $k_2(n)$  comme l'estimation de la fonction de corrélation quadratique moyenne entre  $v_1(n)$  et ses valeurs passées  $v_1(n-1), v_1(n-2), \dots, v_1(n-L)$ . Donc les équations des mises à jour de  $k_1(n)$  et

$\mathbf{k}_2(n)$  sont données par :

$$\mathbf{k}_1(n) = \beta \mathbf{k}_1(n-1) + (1-\beta) \sum_{l=1}^L \mathbf{v}_1(n) v_1(n-l) \quad (3.33)$$

$$\mathbf{k}_2(n) = \beta \mathbf{k}_2(n-1) + (1-\beta) \sum_{l=1}^L \mathbf{v}_2(n) v_2(n-l) \quad (3.34)$$

#### 3.4.4 Algorithme VS-BNSAD 4 :

Dans l'algorithme VS-BNSAD 4, on utilise un vecteur d'erreur pour ajuster les pas d'adaptations variables  $\mu_{12,n}(n)$  et  $\mu_{21,n}(n)$ . Les équations de mise à jour des coefficients du filtre adaptatif avec l'algorithme VS-BNSAD 4 sont données par :

$$\mathbf{w}_{12}(n) = \mathbf{w}_{12}(n-1) + \mu_{12,n}(n) v_2(n) \frac{\mathbf{v}_1(n)}{\mathbf{v}_1(n) \mathbf{v}_1^T(n) + \beta} \quad (3.35)$$

$$\mathbf{w}_{21}(n) = \mathbf{w}_{21}(n-1) + \mu_{21,n}(n) v_1(n) \frac{\mathbf{v}_2(n)}{\mathbf{v}_2(n) \mathbf{v}_2^T(n) + \beta} \quad (3.36)$$

Dans cet algorithme, en utilisant un nouvel critère pour estimer les deux pas d'adaptations variables. Ce critère est donné par les formules suivantes :

$$\mathbb{Q}_{12,n}(n) = \mathbb{Q}_{12,\max} \frac{\|\mathbf{k}_1(n)\|^2}{\|\mathbf{k}_1(n)\|^2 + \delta_1} \quad (3.37)$$

$$\mathbb{Q}_{21,n}(n) = \mathbb{Q}_{21,\max} \frac{\|\mathbf{k}_2(n)\|^2}{\|\mathbf{k}_2(n)\|^2 + \delta_2} \quad (3.38)$$

Où  $\delta_1$  et  $\delta_2$  sont des nombres positifs,  $\mathbf{k}_1(n)$  et  $\mathbf{k}_2(n)$  sont des vecteurs de dimension  $(L \times 1)$  qui sont donnés par :

$$\mathbf{k}_1(n) = \mathbf{k}_1(n-1) + (1-\beta) v_2(n) \frac{\mathbf{v}_1(n)}{\mathbf{v}_1(n) \mathbf{v}_1^T(n) + \beta} \quad (3.39)$$

$$\mathbf{k}_2(n) = \mathbf{k}_2(n-1) + (1-\beta) v_1(n) \frac{\mathbf{v}_2(n)}{\mathbf{v}_2(n) \mathbf{v}_2^T(n) + \beta} \quad (2.40)$$

### **3.5 Conclusion**

Dans ce chapitre, nous avons présenté la structure de décorrélation symétrique adaptative SAD backward, le BSAD normalisé, et les quatre algorithmes du pas d'adaptation variable pour la méthode bi-capteur backward, qui sont basés sur l'estimation du pas d'adaptation variable en utilisant la décorrélation entre les deux sortis.

# Chapitre 4 Résultats des simulations

---

## 4.1 Introduction

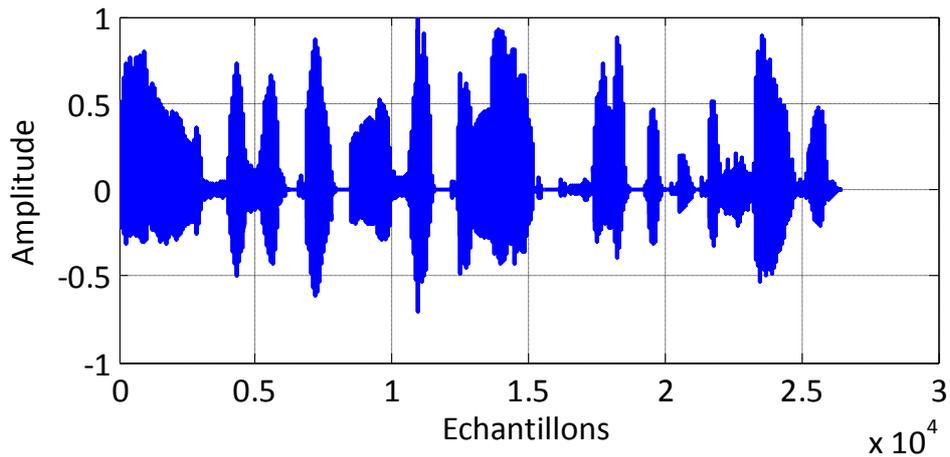
Dans ce chapitre nous présentons les résultats de séparation de sources utilisant l'algorithme de décorrélation symétrique adaptative avec les structures Forward et Backward.

Nous commençons par la présentation de la structure du mélange convolutif, les différentes sources des signaux (parole et bruit) qui sont utilisées dans les simulations, et les réponses impulsionnelles. Puis nous présentons les résultats de la simulation des deux structures Forward et Backward. Nous présentons également l'influence des paramètres (la taille du filtre  $L$ , le pas d'adaptation  $\mu$  et le rapport signal à bruit RSB).

## 4.2 Signaux de test

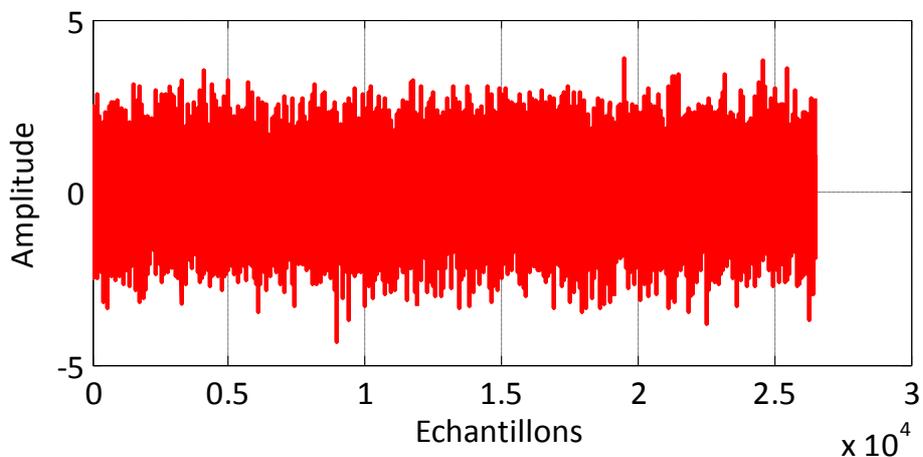
Nous allons présenter les différents signaux utilisés dans toutes les simulations, la figure suivante représente le signal de parole.

Notons tout d'abord qu'une fréquence d'échantillonnage de 8kHz a été utilisée pour obtenir l'ensemble des simulations présentées dans ce chapitre. De plus, les résultats présentés sont obtenus à partir d'un signal de parole de durée 4 secondes (locuteur masculin) représenté ci-dessous.



**Figure 4.1.** Signal de parole utilisé.

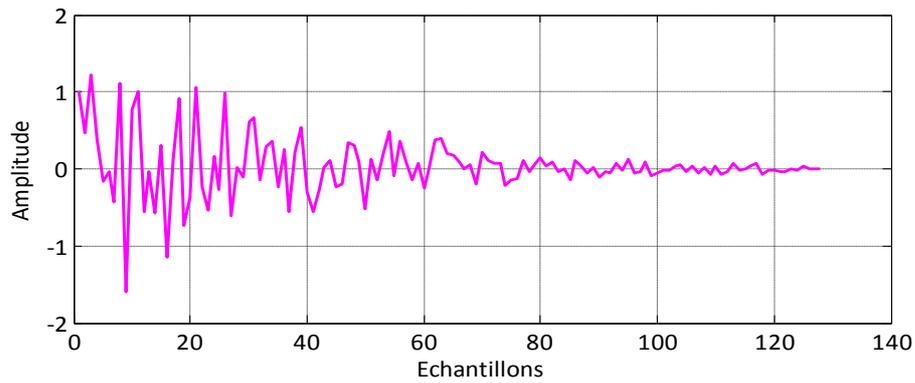
En ce qui concerne la seconde source, elle correspond à un bruit stationnaire. Un bruit blanc gaussien qui est une réalisation d'un processus aléatoire dans lequel la densité spectrale de puissance est la même pour toutes les fréquences et sert surtout à vérifier la stabilité numérique de l'algorithme utilisé.



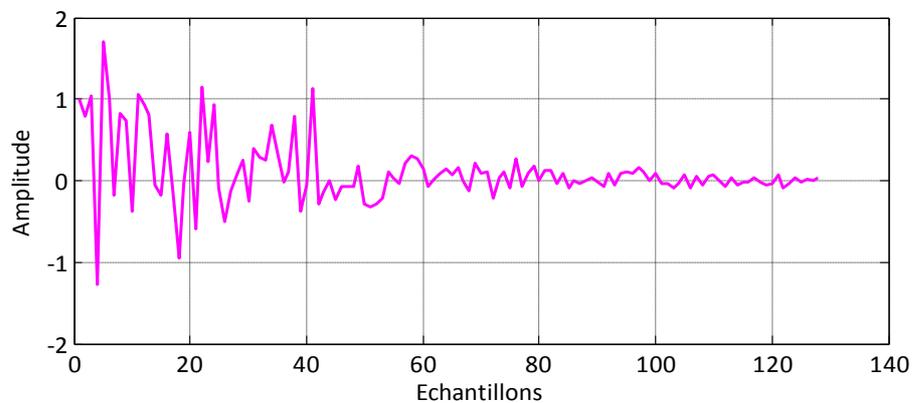
**Figure 4.2.** Bruit blanc.

### 4.3 Réponses impulsionnelles

Les deux figures (4.3 et 4.4) représentent les deux réponses impulsionnelles  $h_{12}$  et  $h_{21}$  avec une taille égale à 128.



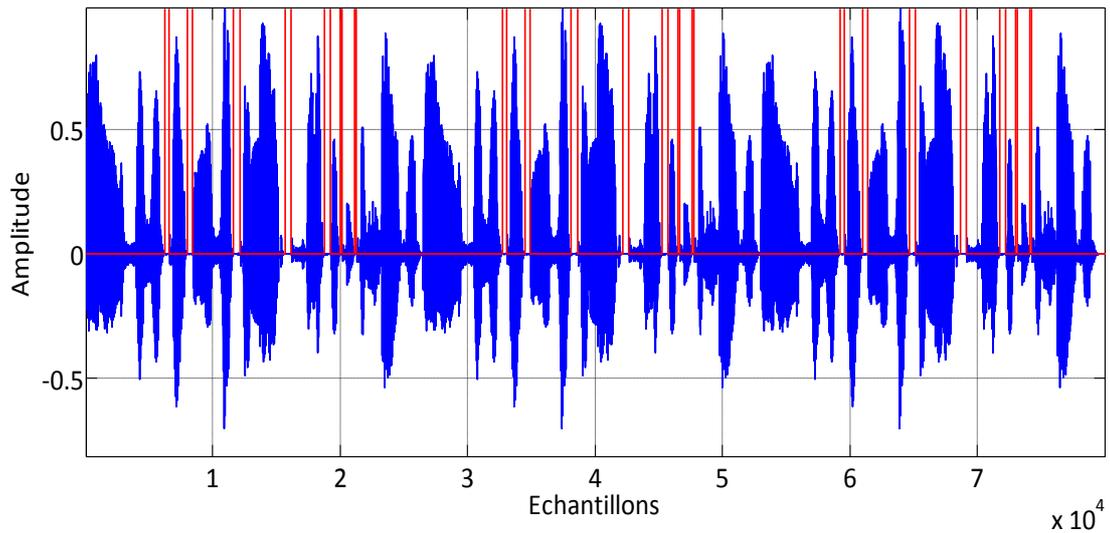
**Figure 4.3.** Exemple de la réponse impulsionnelle simulée  $h_{12}$ .



**Figure 4.4.** Exemple de la réponse impulsionnelle simulée  $h_{21}$ .

## 4.4 Détection d'activité vocale (DAV)

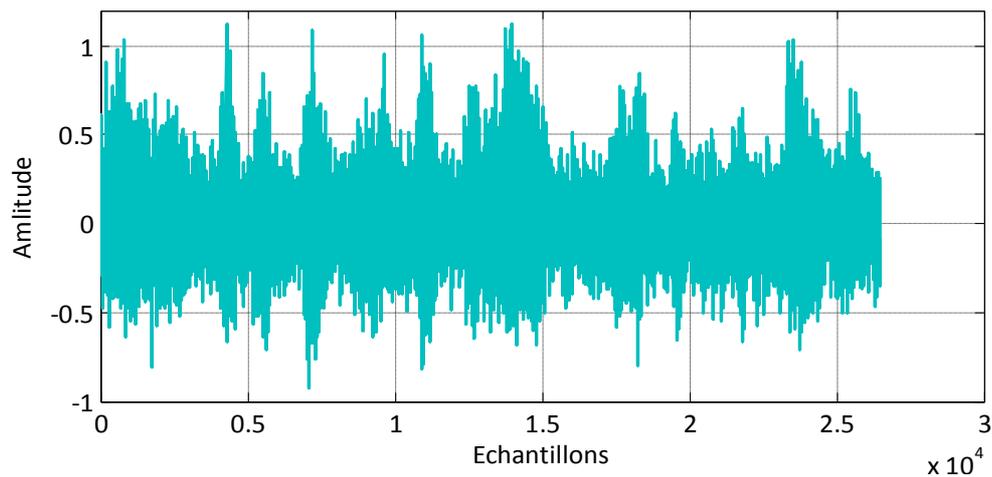
Il a déjà été noté que les signaux en sortie de la structure de séparation symétrique de sources de type Directe et Récursive sont obtenus selon la configuration souhaitée par l'utilisateur, il est possible de maîtriser le fait que l'estimée d'un signal source soit disponible sur la sortie primaire ou secondaire de cette structure. Nous utilisons dans la structure de séparation de sources avec l'algorithme (SAD) un mécanisme de détection d'Activité Vocale (DAV) manuel pour contrôler l'adaptation des filtres  $w_{21}$  et  $w_{12}$ . Cette adaptation est contrôlée de la manière suivante : le filtre  $w_{21}$  est adapté uniquement en période de bruit seul, alors que le filtre  $w_{12}$  est adapté uniquement pendant les périodes d'activité vocale (période de présence du signal de parole utile). Figure 4.5 présente un exemple de DAV manuelle sur le même signal de parole que celui décrit précédemment.



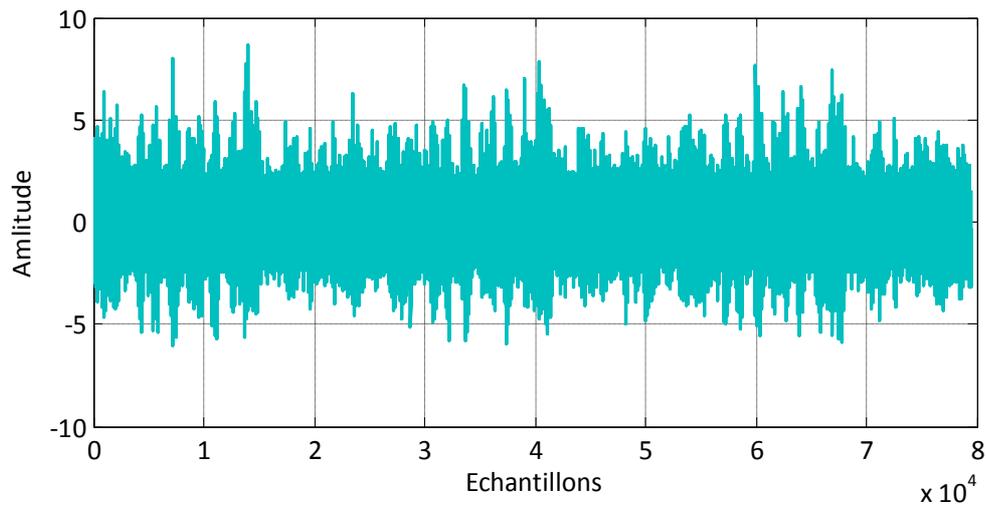
**Figure 4.5.** Signal parole avec DAV.

## 4.5 Simulation du mélange adaptatif (signaux bruités)

Dans cette partie, nous présentons les signaux dans les deux sorties de la structure de mélange convolutif (voir la figure 1.4), les signaux de source sont la parole et le bruit, dans les deux figures ci dessous, nous représentons un exemple de signaux de mélange  $p_1(n)$  et  $p_2(n)$ .



**Figure 4.6.** Le signal de mélange  $p_1$  (signal bruité).



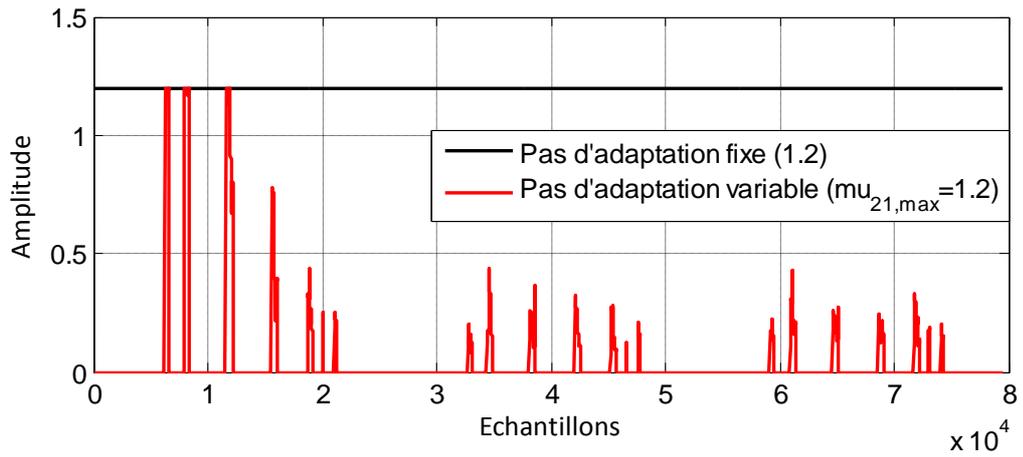
**Figure 4.7.** Le signal de mélange  $p_2$  (signal bruité).

## 4.6 Résultats de simulations de l’algorithme de décorrélation SAD

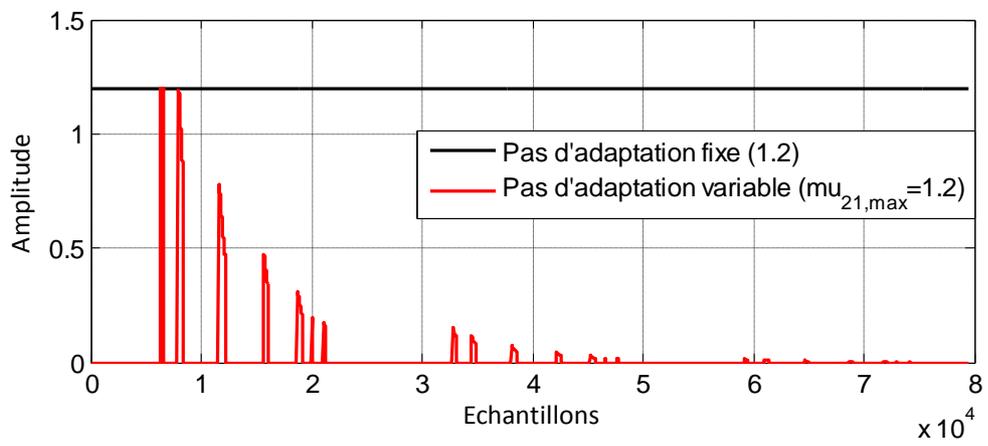
Dans cette partie, nous allons présenter les résultats de simulation de deux versions algorithmiques du SAD. La première version de cet algorithme est utilisée avec la structure de séparation de sources Forward. La deuxième version de ce dernier est utilisée avec la structure de séparation de sources Backward. Pour cela, nous allons commencer par présenter les résultats de simulation de l’algorithme SAD avec la structure Forward et Backward et ceci par différents types de paramètres.

### 4.6.1 Simulation d’algorithme SAD avec la structure Forward

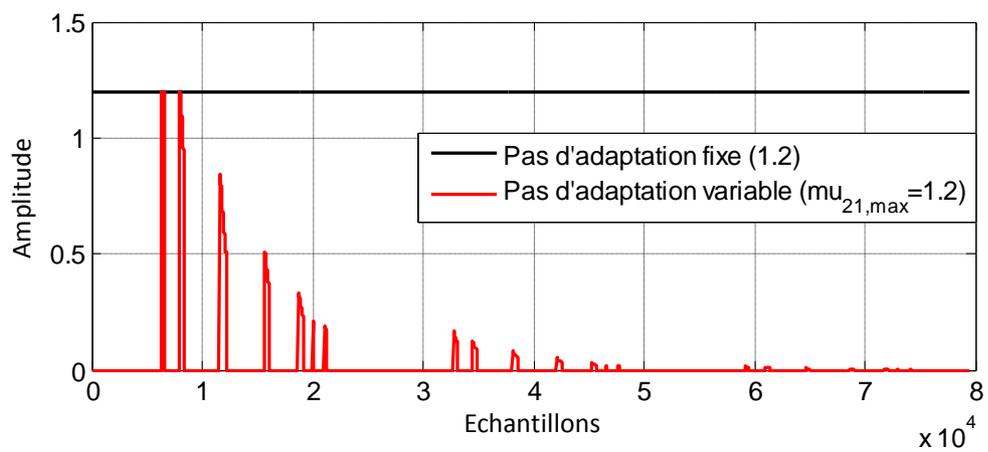
Nous allons voir dans cette partie la variation du pas d’adaptation selon les quatre algorithmes du pas d’adaptation variable proposés dans le chapitre 1, premièrement nous allons voir les variations des pas d’adaptations fixes à la valeur  $\mu = 1.2$ .



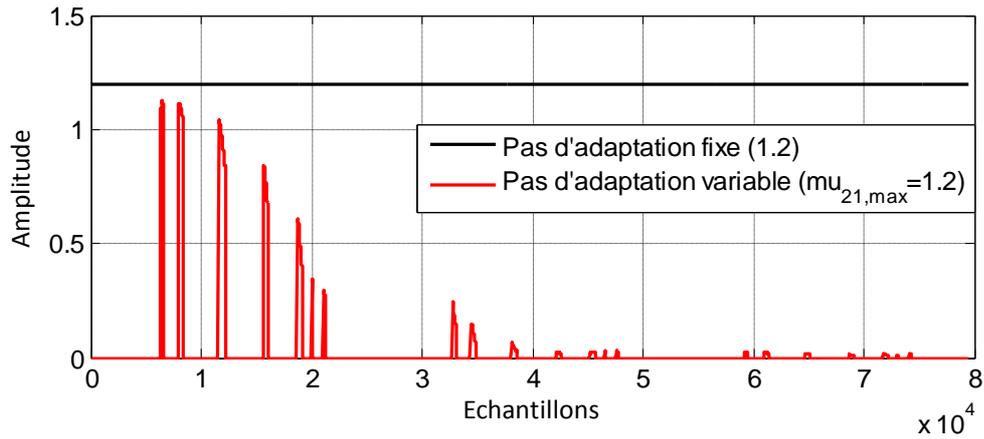
**Figure 4.8.** Variation du pas d'adaptation variable dans l'algorithme 1.



**Figure 4.9.** Variation du pas d'adaptation variable dans l'algorithme 2.



**Figure 4.10.** Variation du pas d'adaptation variable dans l'algorithme 3.

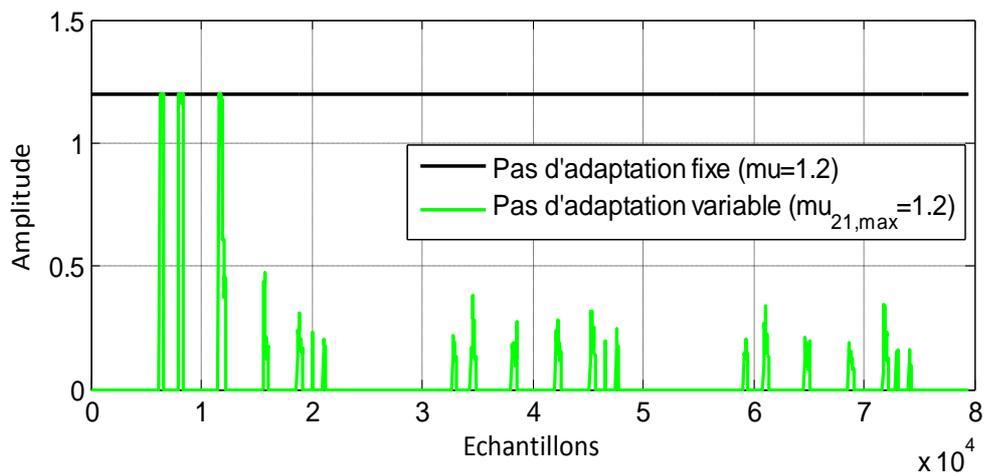


**Figure 4.11.** Variation du pas d'adaptation variable dans l'algorithme 4.

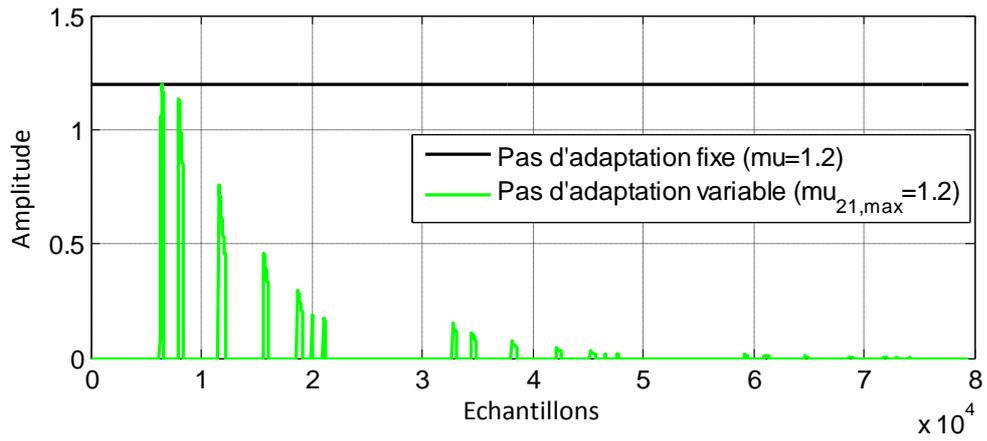
Nous remarquons que l'amplitude du pas d'adaptation au premier temps est grande presque (1.2) pour les premiers échantillons, ensuite elle commence à diminuer et converge vers la valeur minimale.

#### 4.6.2 Simulation d'algorithme SAD avec la structure Backward

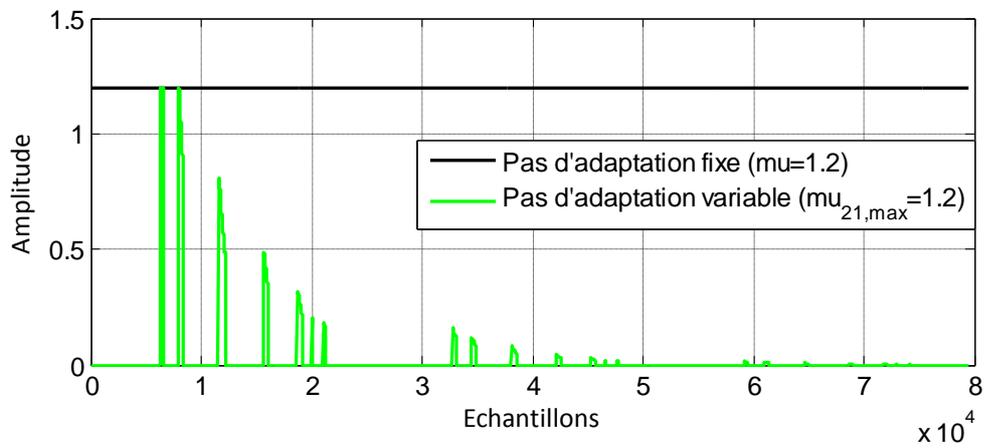
Nous allons faire exactement le même travail qu'on a fait pour la structure SAD Forward.



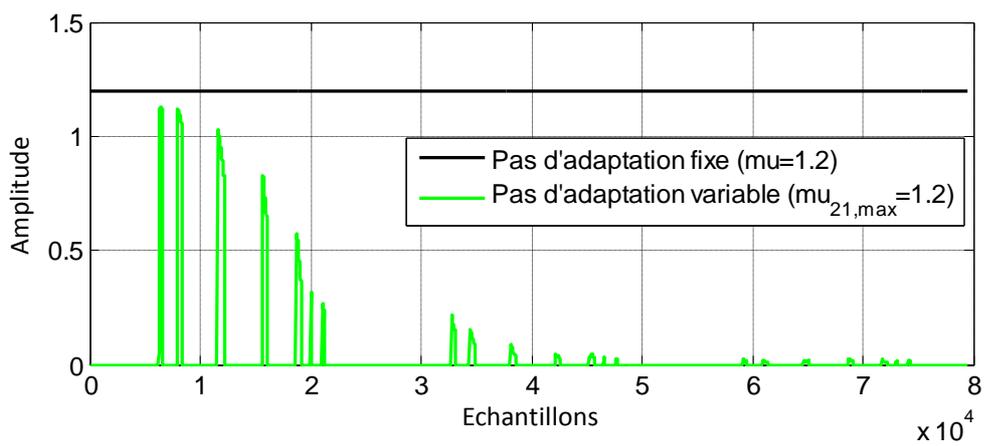
**Figure 4.12.** Variation du pas d'adaptation variable dans l'algorithme 1.



**Figure 4.13.** Variation du pas d'adaptation variable dans l'algorithme 2.



**Figure 4.14.** Variation du pas d'adaptation variable dans l'algorithme 3.



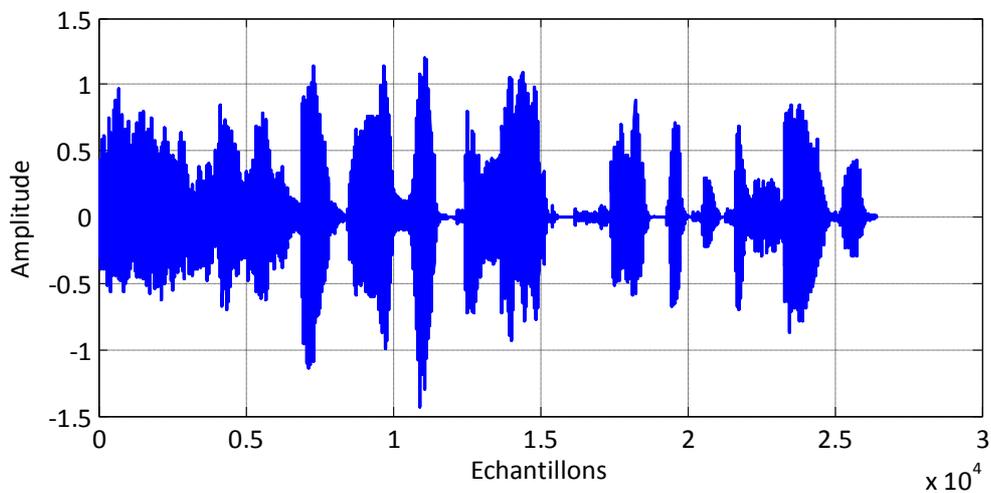
**Figure 4.15.** Variation du pas d'adaptation variable dans l'algorithme 4.

D'après ces figures de 4.12 à 4.15 qui montrent la variation du pas d'adaptation variable dans l'algorithme pour les quatre algorithmes proposés, nous remarquons que le pas d'adaptation commence par une grande valeur  $\mu=1.2$  au départ et diminuer jusqu'à la valeur minimale.

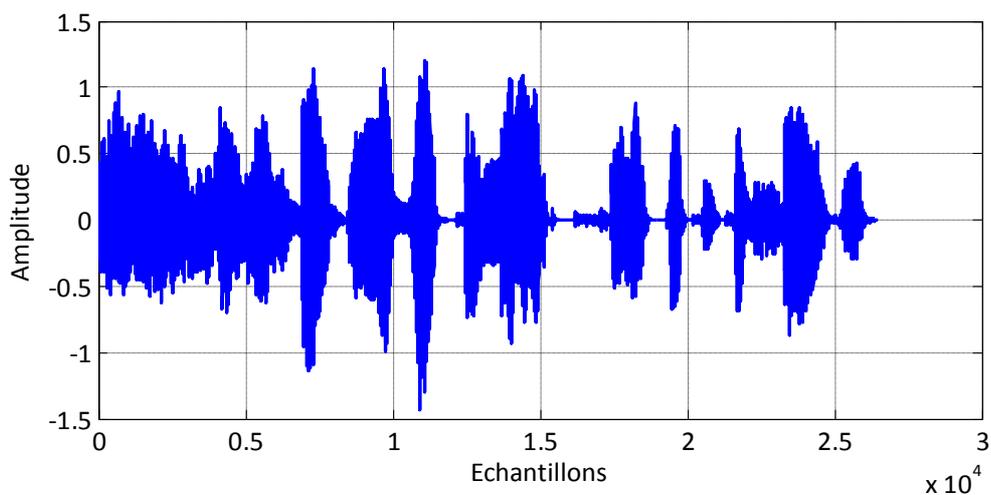
### 4.6.3 Signaux de parole estimés par l'algorithme de décorrélation

#### Forward avec pas variable

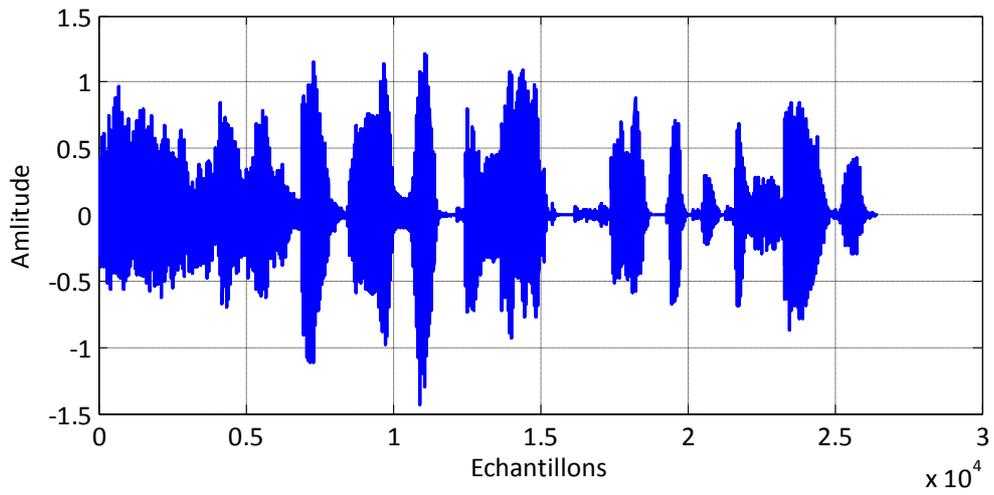
Nous allons voir les signaux de paroles estimés d'après l'utilisation du pas d'adaptation variable c'est à dire selon les quatre algorithmes proposés.



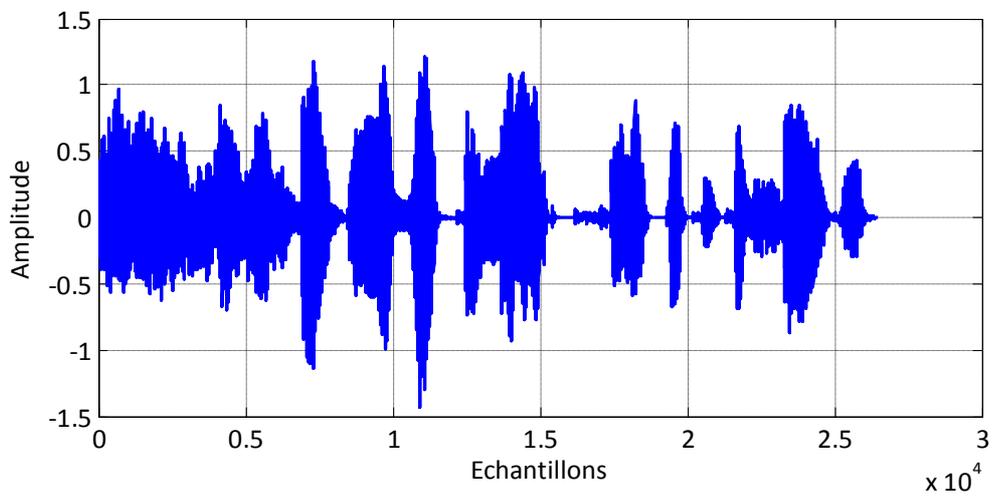
**Figure 4.16.** Signal de parole estimé par l'algorithme FNSAD.



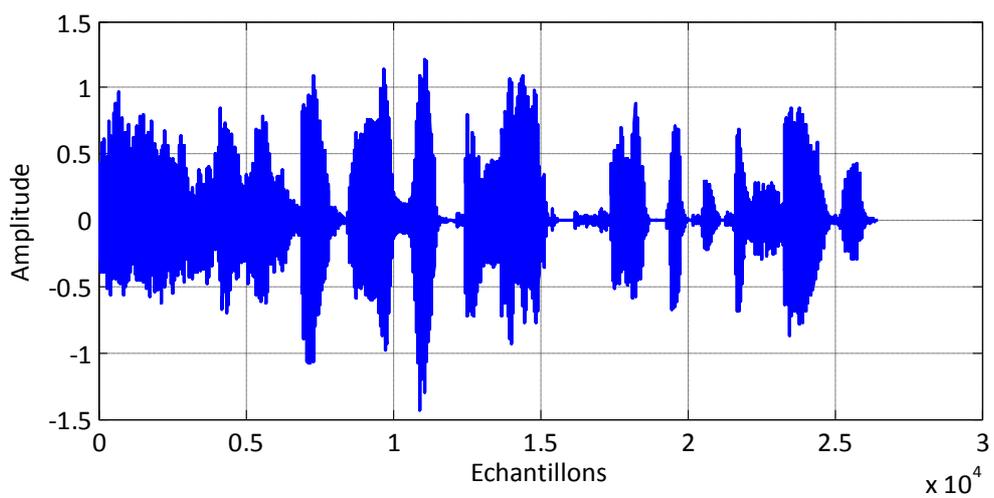
**Figure 4.17.** Signal de parole estimé par l'algorithme VS-FNSAD 1.



**Figure 4.18.** Signal de parole estimé par l'algorithme VS-FNSAD 2.



**Figure 4.19.** Signal de parole estimé par l'algorithme VS-FNSAD 3.



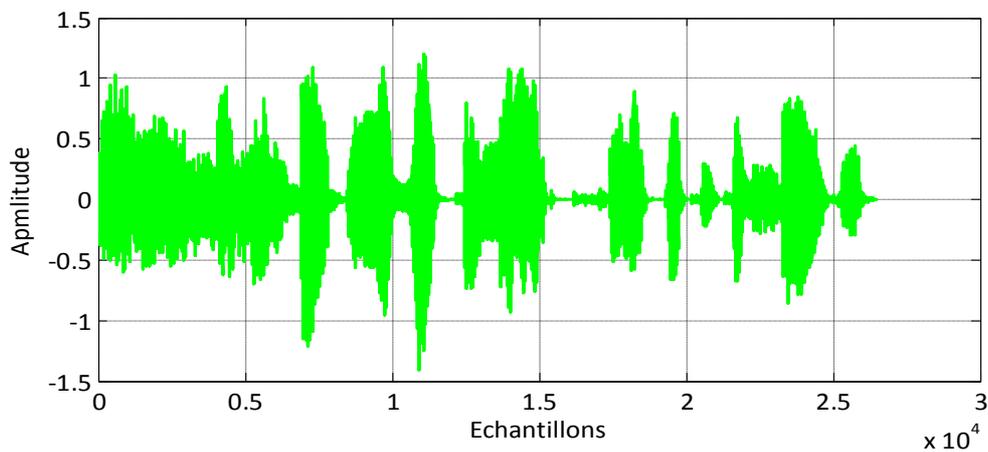
**Figure 4.20.** Signal de parole estimé par l'algorithme VS-FNSAD 4.

Dans toutes les simulations que nous avons faites, nous avons fixé la taille des filtres adaptatifs à 128 et les deux rapports signal sur bruit d'entrée à -3dB.

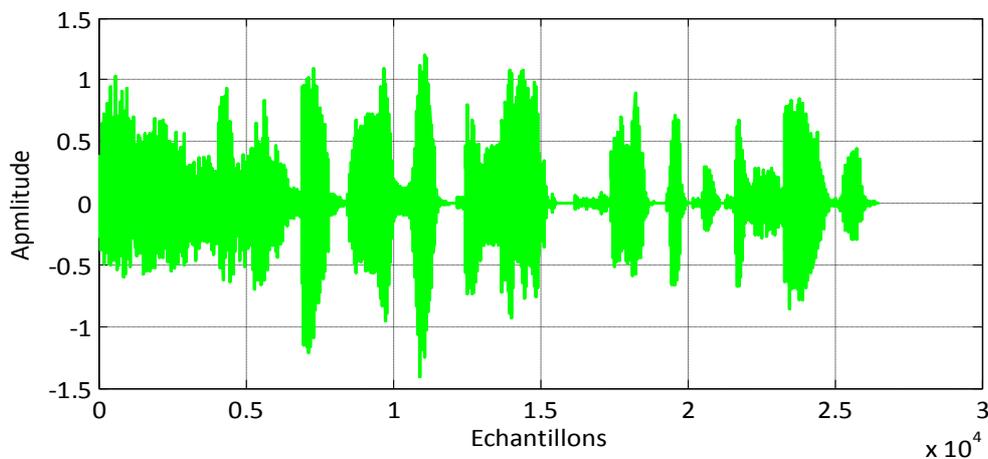
Nous remarquons dans les figures de (4.16 à 4.20), qui nous montrent les quatre algorithmes proposés dans la structure Forward FNSAD, une petite variation dans l'évaluation du signal parole, cette variation signifie l'amélioration du signal parole et la minimisation du bruit, et nous remarquons que la VS-FNSAD a une meilleure qualité par rapport à la SAD classique.

#### 4.6.4 Signaux de parole estimés par l'algorithme de decorrelation Bakward avec pas variable

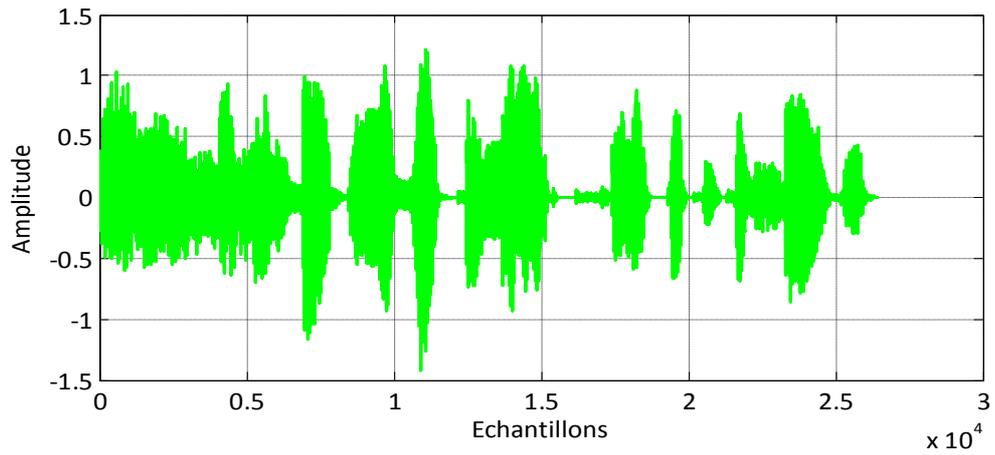
Nous allons montrer les différents signaux de parole estimés d'après la SAD Backward



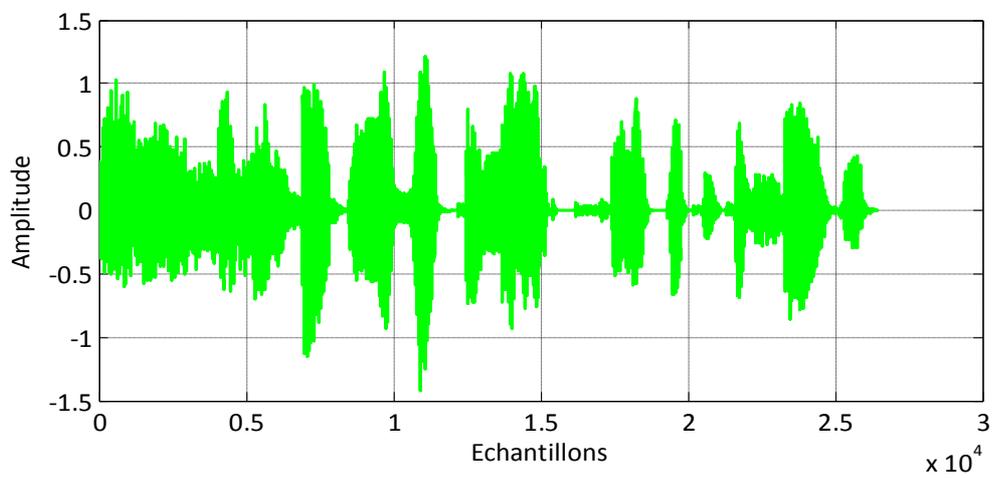
**Figure 4.21.** Signal de parole estimé par l'algorithme BNSAD.



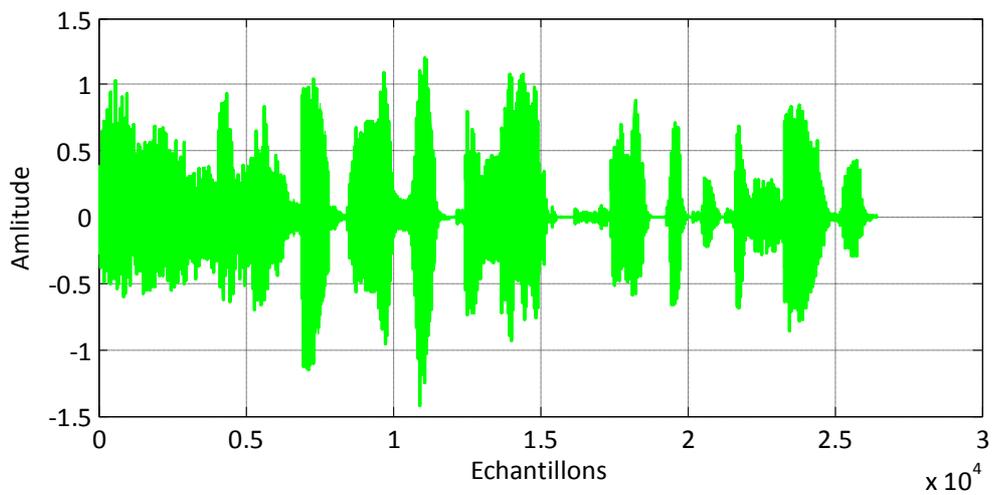
**Figure 4.22.** Signal de parole estimé par l'algorithme VS-BNSAD 1.



**Figure 4.23.** Signal de parole estimé par l'algorithme VS-BNSAD 2.



**Figure 4.24.** Signal de parole estimé par l'algorithme VS-BNSAD 3.



**Figure 4.25.** Signal de parole estimé par l'algorithme VS-BNSAD 4.

Nous remarquons dans les figures de (4.21 à 4.25), qui montrent le signal de parole estimé par les quatre algorithmes proposés dans la structure Backward BNSAD, une petite variation dans l'évaluation du signal parole, cette variation représente une amélioration du signal parole et la minimisation du bruit, et nous remarquons que la SAD classique a moins de qualité que la VS-BNSAD.

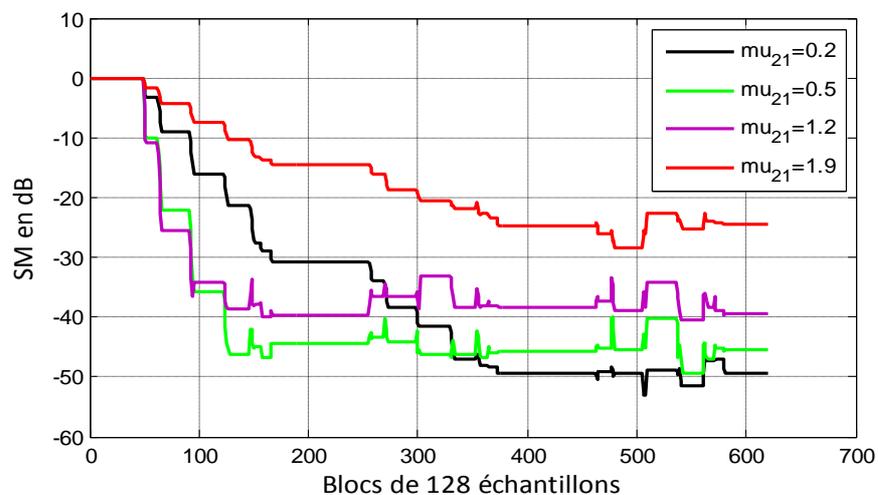
## 4.7 Problématique

Dans cette section, on va présenter les résultats des simulations de l'algorithme de décorrélation normalisé Forward et Backward avec plusieurs pas d'adaptations fixes. Dans toutes les simulations on a  $\alpha$  et  $\gamma$  qui sont des paramètres de contrôle et qui sont données comme suit :

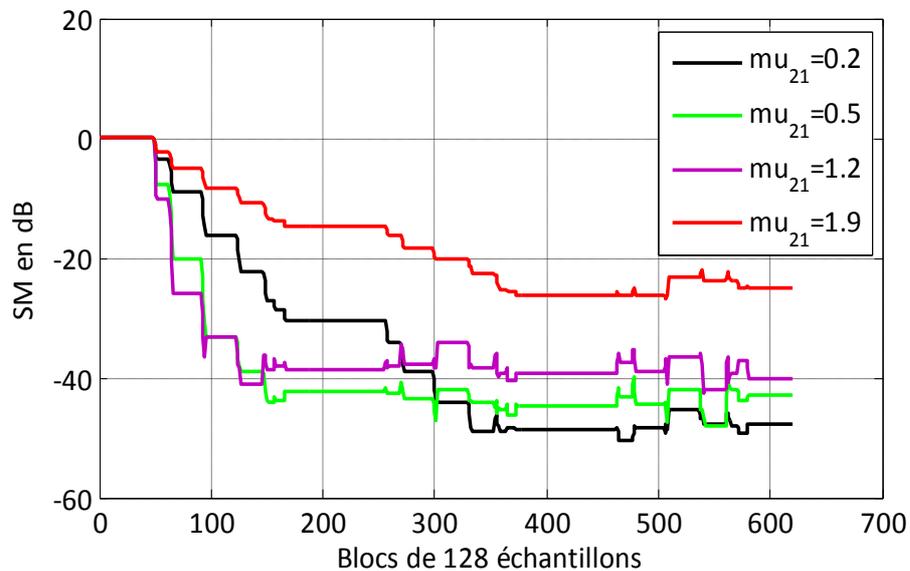
$0 < \alpha < 1$  et  $\gamma > 0$ , et  $0 < \beta < 1$ . Pour assurer le bon fonctionnement de l'algorithme VS-FNSAD et VS-BNSAD, pour chaque itération le pas d'adaptation doit être borné entre  $[\mu_{\min}, \mu_{\max}]$ .

Nous avons choisi les paramètres suivants à cause de leurs rendements :

- $\alpha = 0.999$ ;
- $\gamma = 0.999$ ;
- $\beta = 0.9$ ;
- $\mu_{\min} = 0$ ;
- $\mu_{\max} = 1.2$ ;



**Figure 4.26.** System Mismatch (SM) obtenu par l'algorithme FNSAD avec plusieurs pas fixes.

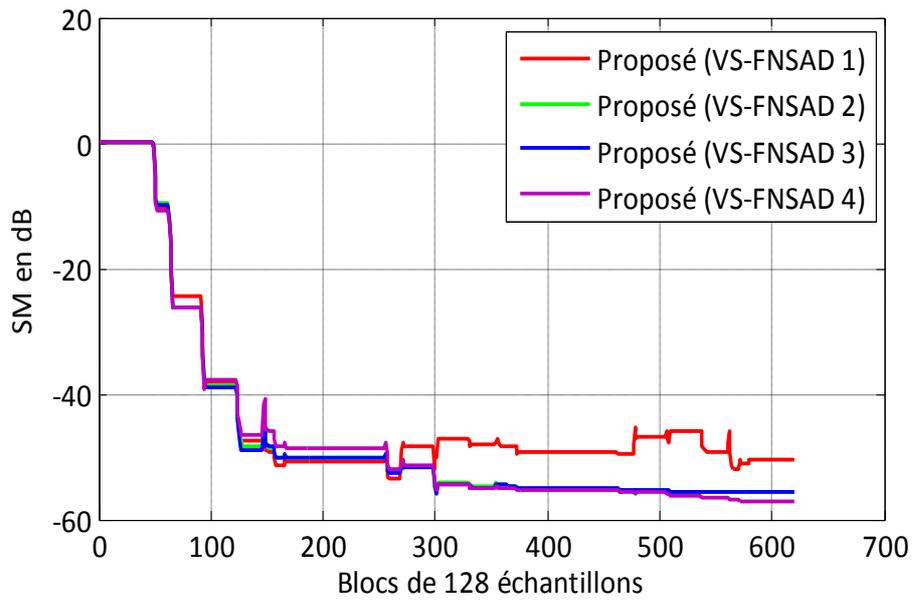


**Figure 4.27.** System Mismatch (SM) obtenu par l’algorithme BNSAD avec plusieurs pas fixes.

En nous basant sur les résultats du system mismatch présentés dans les deux figures (4.26 et 27), nous remarquons que pour une faible valeur du pas adaptatif par exemple ( $\mu=0.2$ ) on a un bon niveau du SM, mais la vitesse de convergence est un peu lente, par contre si le pas a une valeur plus élevée la vitesse est très rapide mais la qualité est mauvaise.

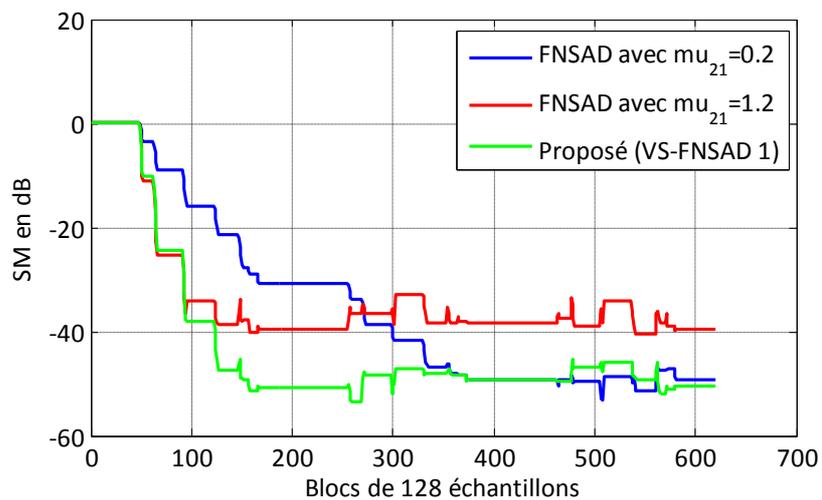
#### 4.7.1 Résultats comparatifs

Dans cette partie, nous allons présenter les résultats de simulations des algorithmes proposés (VS-FNSAD) et (VS-BNSAD), Nous nous basons sur le System Mismatch (SM) et sur le rapport signal à bruit pour faire une comparaison entre les quatre algorithmes (VSS) proposés en Forward et en Backward.

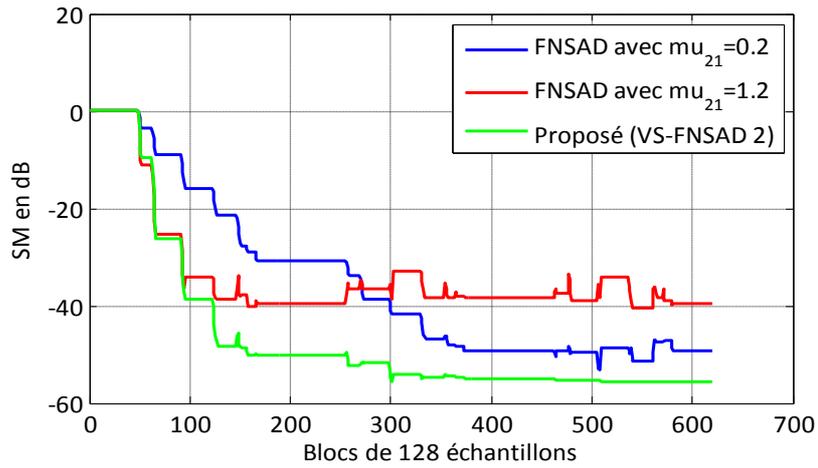


**Figure 4.28.** System Mismatch (SM) obtenu par les quatre algorithmes VS-FNSAD.

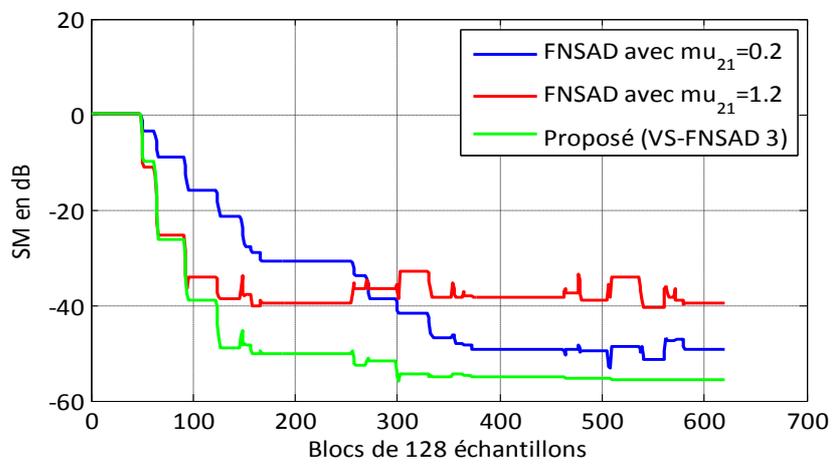
Nous remarquons dans la figure 4.28 que le system mismatch pour l’algorithme 4 est le meilleur en termes de la qualité et de la vitesse de convergence que les trois autres algorithmes proposés (VS-FNSAD 1,2 et 3) et que ces dernier sont meilleurs que la SAD classique.



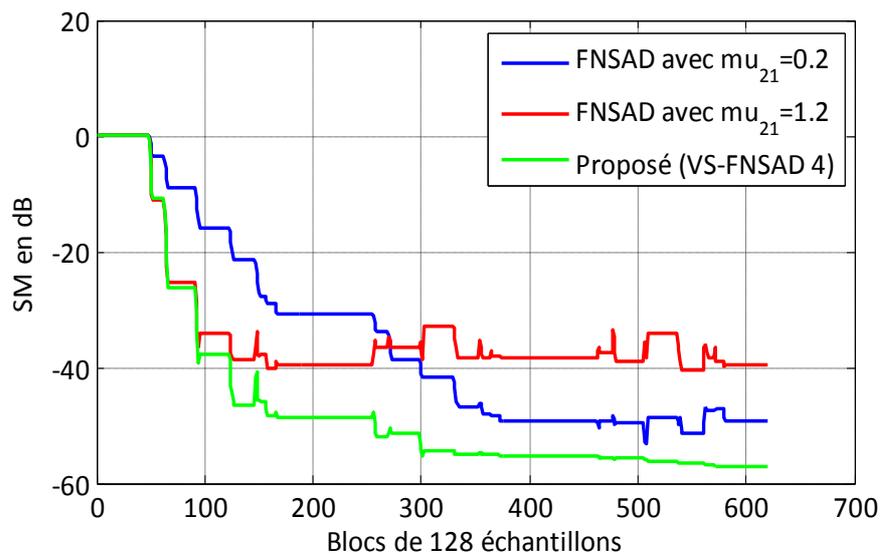
**Figure 4.29.** System Mismatch (SM) obtenu par l’algorithme VS-FNSAD 1.



**Figure 4.30.** Système Mismatch (SM) obtenu par l’algorithme VS-FNSAD 2.

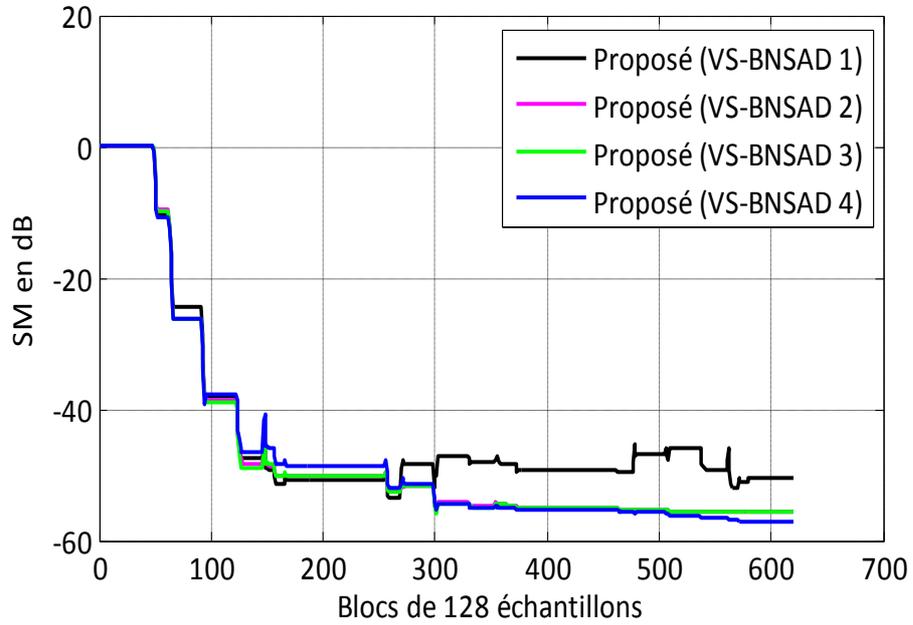


**Figure 4.31** System Mismatch (SM) obtenu par l’algorithme VS-FNSAD 3.

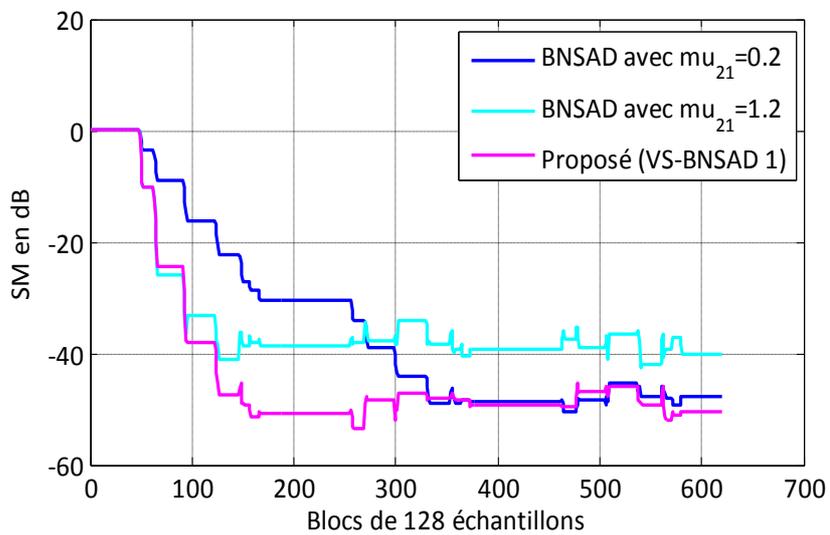


**Figure 4.32.** System Mismatch (SM) obtenu par l’algorithme VS-FNSAD 4.

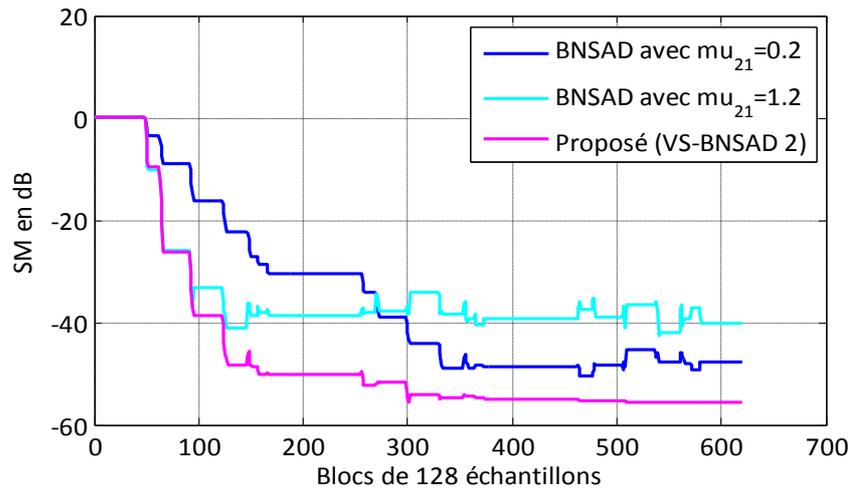
Le system mismatch nous montre que les quatre algorithmes proposés ont une meilleure qualité que la SAD classique, et que la VS-FNSAD 4 est la meilleure en termes de bonne qualité et de vitesse de convergence.



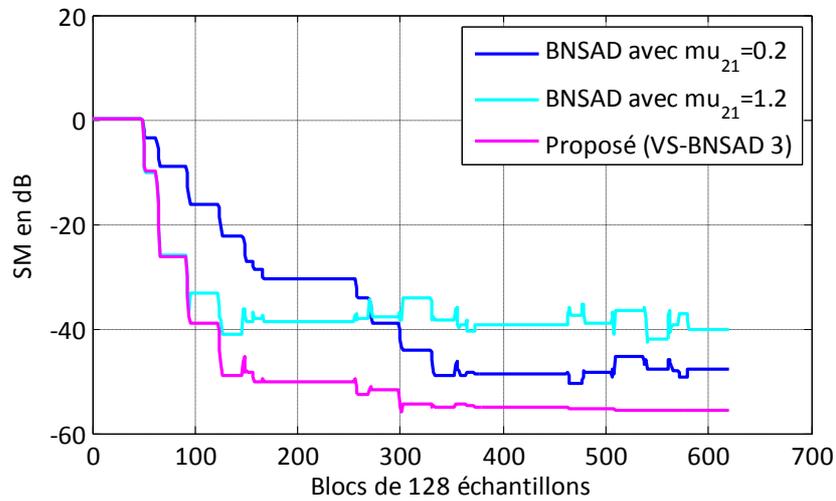
**Figure 4.33.** System Mismatch (SM) obtenu par les quatre algorithmes VS-BNSAD.



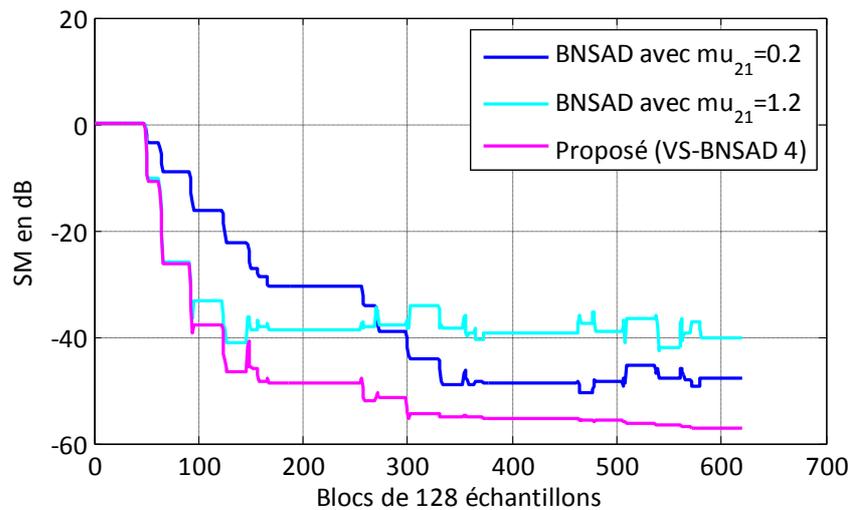
**Figure 4.34.** System Mismatch (SM) obtenu par l’algorithme VS-BNSAD 1.



**Figure 4.35.** System Mismatch (SM) obtenu par l'algorithme VS-BNSAD 2.



**Figure 4.36.** System Mismatch (SM) obtenu par l'algorithme VS-BNSAD 3.



**Figure 4.37.** System Mismatch (SM) obtenu par l'algorithme VS-BNSAD 4.

D'après les figures précédentes du (SM) nous remarquons que les algorithmes FNSAD et BNSAD lorsque le pas  $\mu=0.2$  :

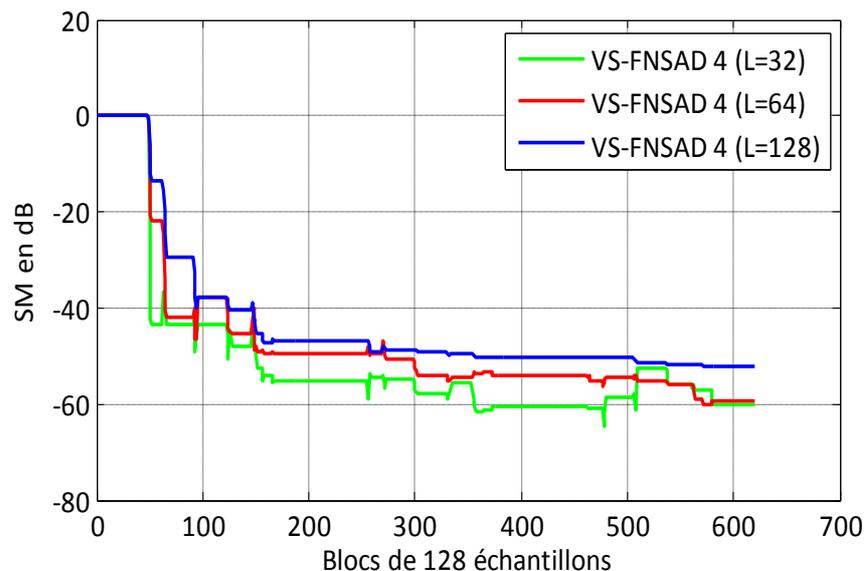
- La vitesse de convergence est un peu lente, elle n'est pas vraiment intéressante.
- la qualité est bonne.

Lorsque le pas  $\mu=1.2$  :

- la vitesse converge rapidement.
- la qualité n'est pas vraiment intéressante.

Les algorithmes proposés de pas variable gardent la vitesse de convergence quand le pas est de grande valeur, et gardent aussi la meilleure qualité quand le pas est de faible de valeur, donc nous avons la vitesse et qualité et c'est vraiment intéressant.

## 4.8 Effet de la taille L

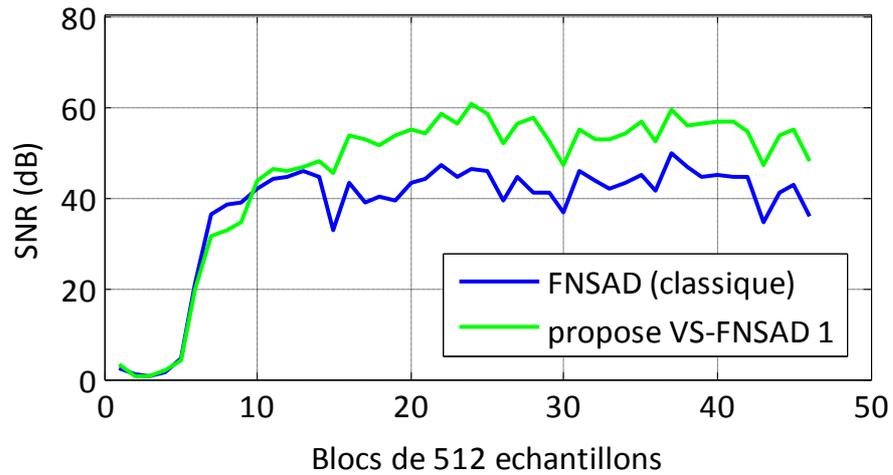


**Figure 4.38.** L'effet de la taille.

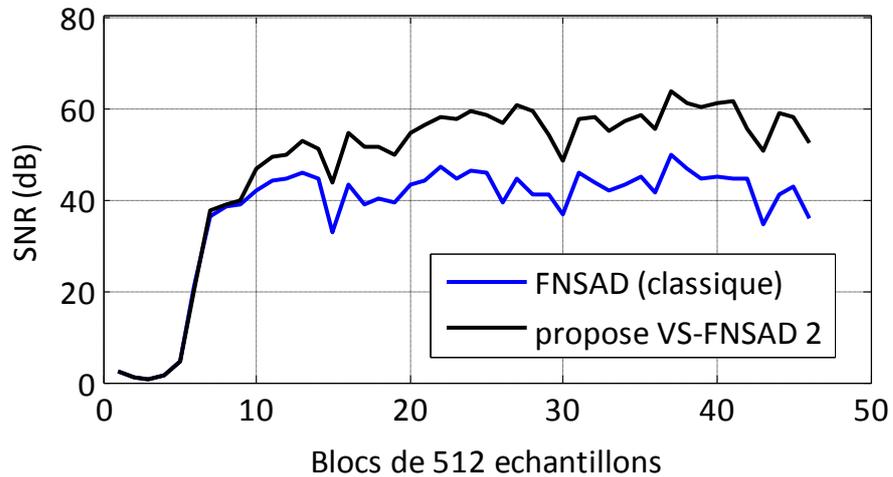
Nous avons mentionné dans la figure 4.38 l'effet de la taille de filtre sur la vitesse de convergence et la qualité avec l'algorithme VS-FNSAD que nous avons choisi comme un exemple, et nous remarquons qu'à chaque fois que la taille du filtre augmente on aura une moindre de qualité, et cette remarque est identique pour les huit algorithmes dans les deux structures Forward et Backward.

## 4.9 Rapport signal sur bruit SNR

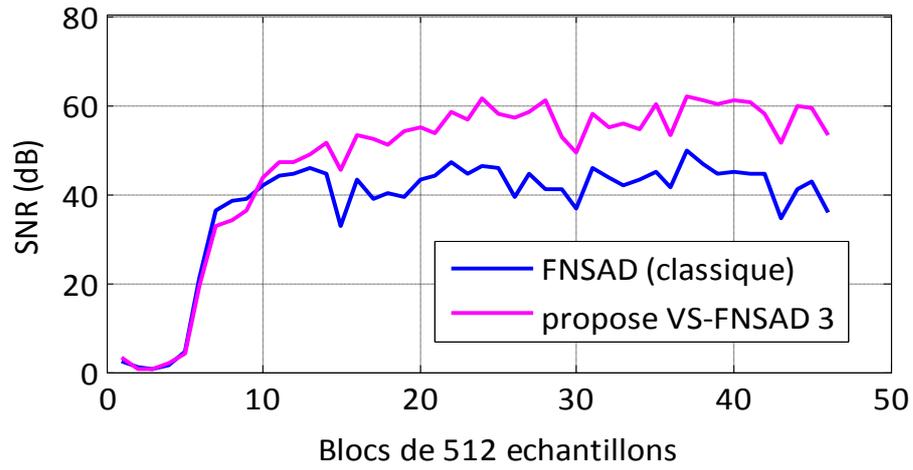
Dans cette partie nous allons voir les évaluations du SNR de sortie sur les huit algorithmes proposés en Forward et Backward. Nous avons utilisé les mêmes paramètres prononcés dans la section. Les résultats obtenus du SNR sont donnés par les figures suivantes :



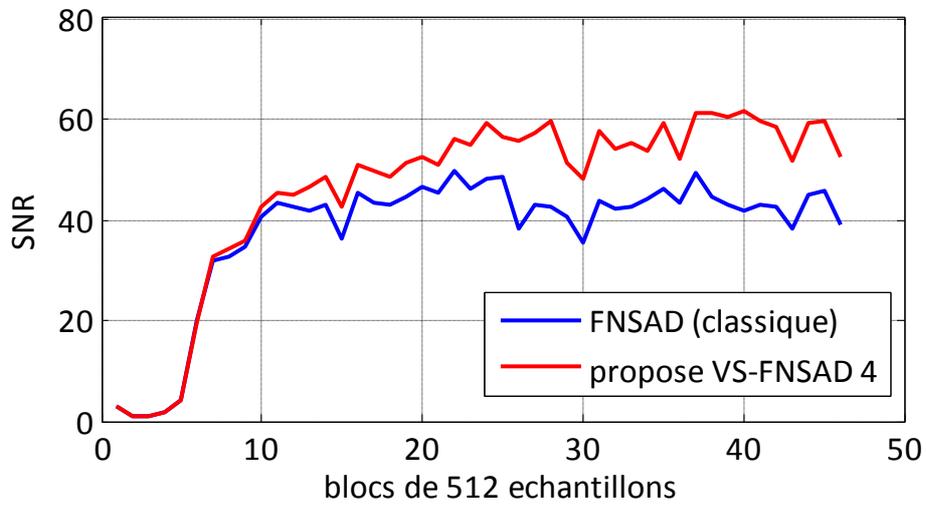
**Figure 4.39.** Comparaison du SNR obtenu par l'algorithme FNSAD et VS-FNSAD 1.



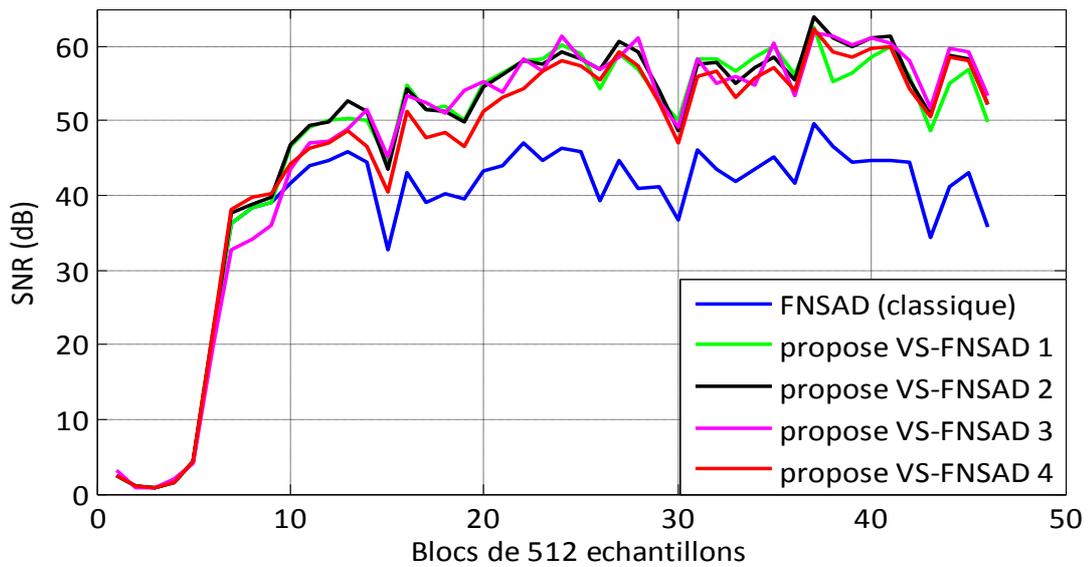
**Figure 4.40.** Comparaison du SNR obtenu par l'algorithme FNSAD et VS-FNSAD 2.



**Figure 4.41.** Comparaison du SNR obtenu par l'algorithme FNSAD et VS-FNSAD 3.

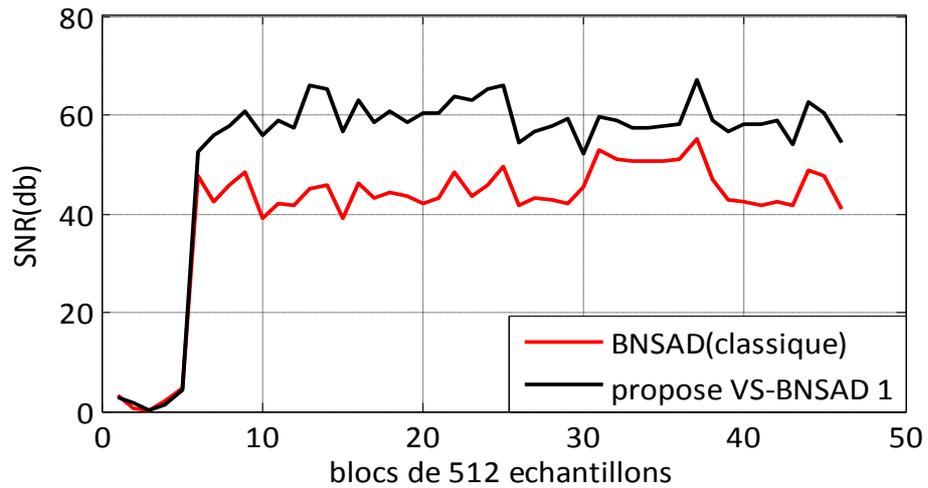


**Figure 4.42.** Comparaison du SNR obtenu par l'algorithme FNSAD et VS-FNSAD 4.

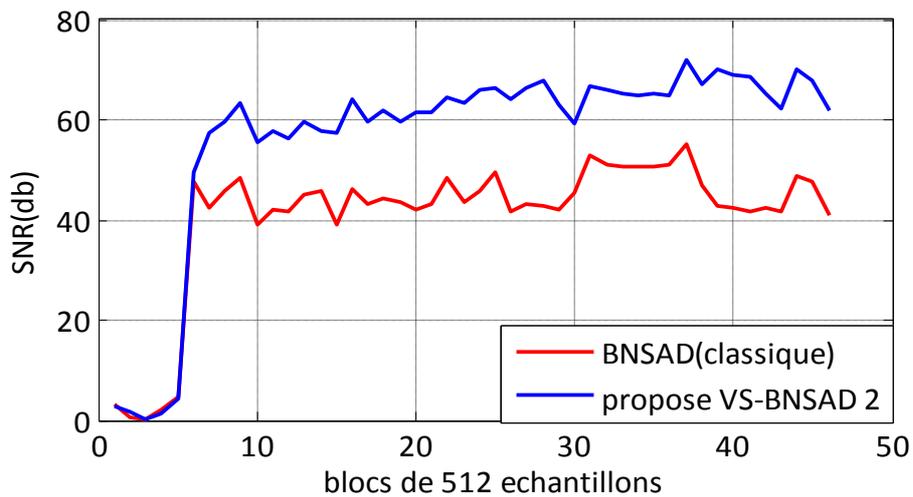


**Figure 4.43.** Comparaison pour le SNR du FNSAD classique et le SNR des VS-FNSAD.

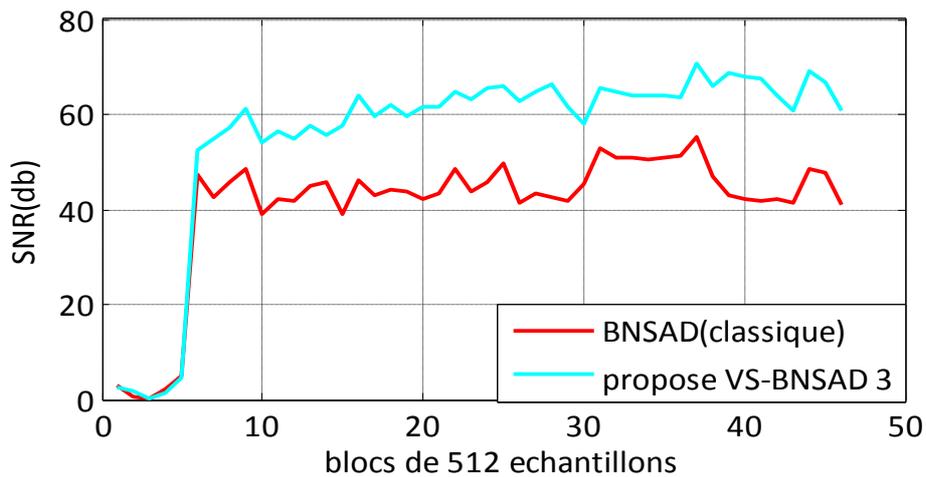
Selon les figures passées du SNR, nous remarquons que pour les quatre algorithmes proposés Forward donnent des bons résultats en terme de qualité du signal de parole estimé en comparaison avec leur version classique à pas d'adaptation fixe (FNSAD).



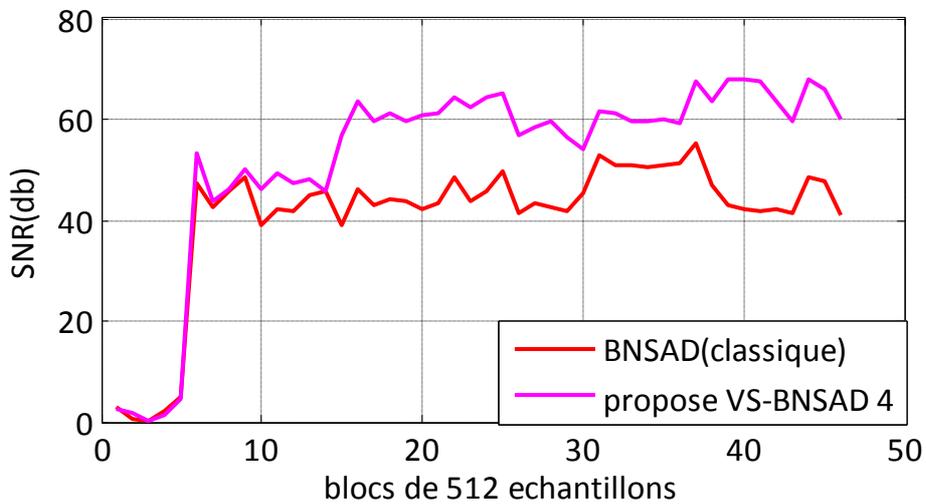
**Figure 4.44.** Comparaison du SNR obtenu par l'algorithme BNSAD et VS-BNSAD 1.



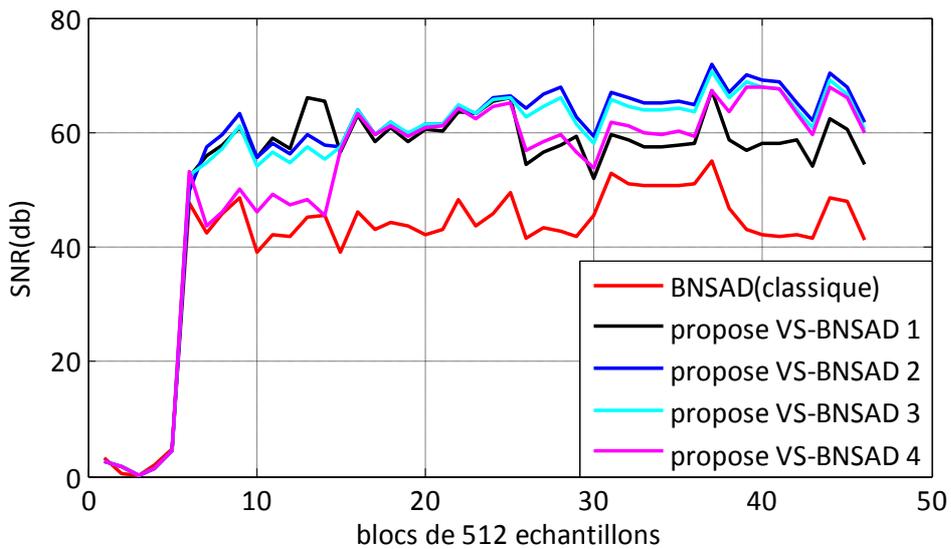
**Figure 4.45.** Comparaison du SNR obtenu par l'algorithme BNSAD et VS-BNSAD 2.



**Figure 4.46.** Comparaison du SNR obtenu par l'algorithme BNSAD et VS-BNSAD 3.



**Figure 4.47.** Comparaison du SNR obtenu par l'algorithme BNSAD et VS-BNSAD 4.



**Figure 4.48.** Comparaison entre le SNR du BNSAD classique et le SNR des VS-BNSAD.

Nous avons vu le SNR pour la BNSAD et VS-NSAD dans les figures 4.44 à 4.48, et nous constatons des très bons résultats du SNR pour les VS-BNSAD par rapport à la NSAD classique.

#### **4.10 Conclusion**

Nous avons vu dans ce chapitre 4 les résultats des simulations d'algorithme SAD pour les deux structures Forward et Backward pour les quatre algorithmes proposés, à partir des figures des simulations obtenus, nous constatons que les huit algorithmes proposés donnent des bons résultats, la vitesse de la convergence rapide et la bonne qualité des signaux estimés par rapport à la SAD normalisé.

## Conclusion générale

---

Dans un premier volet de ce mémoire, nous avons présenté les différentes méthodes de réduction du bruit et le rehaussement du signal de parole par les algorithmes de filtrage adaptatif. Nous nous sommes basés en particulier sur la structure de séparation aveugle de sources employant deux capteurs de prise de son.

D'après l'étude des performances des deux versions de l'algorithme symétrique de décorrélation (SAD) avec pas d'adaptation variable et normalisé VS-NSAD, nous avons remarqué que ces derniers donnent des bons résultats pour la séparation de sources et le rehaussement du signal de parole.

Si on utilise l'algorithme SAD normalisé avec pas d'adaptation fixes, Nous remarquons que la vitesse de convergence est très rapide lorsque les valeurs des pas sont grandes, par contre, la vitesse est très faible si on utilise des pas d'adaptation fixes faibles.

Nous remarquons aussi que la vitesse de convergence obtenue avec les huit versions est très rapide. On a aussi montré, à travers les résultats de simulations réalisés, que les signaux obtenus sont de bonnes qualités. Donc, nous notons la supériorité des algorithmes VS-NSAD en termes de vitesse de convergence et qualité des signaux estimés.

Nous constatons que les huit algorithmes proposés donnent des bons résultats pour la réduction du bruit acoustique et le rehaussement du signal de parole.

# Bibliographie

---

- [1] BENDOUMIA Rédha, DEBA ABDELLAH, M. Djendi, : 'rehaussement du signal de parole par l'algorithme de décorrélation symétrique', mémoire de projet de fin d'études pour l'obtention du diplôme master, L'Algérie, 2001.
- [2] J. El Mhamdi, F. Regragui & M. Harnafi : 'Traitement adaptatif appliqué au signal sismique' Bulletin de l'Institut Scientifique, Rabat, section Sciences de la Terre, 2008, n°30.
- [3] J.Benesty : 'traitement de signal numérique II, Filtrage de Wiener',
- [4] CASTELLA1, Marc : 'Méthodes séquentielles de séparation aveugle de sources en présence'.1 Institut National des Télécommunications (INT).France.
- [5] Y.Benabderrahman : ' Séparation aveugle des signaux de parole ', Thèse de Doctorat. Université du Québec. Québec, 2011.
- [6] M. Djendi, A. Gilloire et P. Scalart : 'Frequency post-filters for noise cancellation using two clasly speced microphones '. Eurasip. Eusipco 2009, Poznan, Polgne. 2007.
- [7] INRS-EMT J. Benesty : 'Algorithme du gradient stochastique' (least-mean-square LMS).
- [8] H.Alaeddine : ' Application de la transformée en nombres entiers à la conception d'algorithmes de faible complexité pour l'annulation d'échos acoustiques' thèse de Doctorat, l'université de Bretagne occidentale. 12 Juillet 2007
- [9] Van Gerven S, Van Compernelle D : 'Signal separation by symmetric adaptive decorrelation stability ', convergence, and uniqueness. IEEE Trans. Signal Process',1995, vol. 43, no. 7, pp.1602-1612.

- [10] S. Van Gerven, D. van Compernelle, : 'Feedforward and feedback in symmetric adaptive noise canceller: stability analysis in a simplified case', in Proceedings of the European Signal Processing Conference, Brussels, Belgium, August 1992, pp. 1081–1084.
- [11] Bendoumia R, Djendi M: 'Two-channel variable-step-size forward-and-backward adaptive algorithms for acoustic noise reduction and speech enhancement', Signal Process 2015; 108, 226–44.
- [12] J.-F. Bercher & P. : 'Jardin Introduction au filtrage adaptatif', ESIEE Paris I4-TTS 2003.
- [13] Hyun-Chool Shin, Ali H. Sayed and Woo-Jin Song ;' Variable Step-Size NLMS and Affine Projection Algorithms' IEEE SIGNAL PROCESSING LETTERS, VOL. 11, NO. 2, FEBRUARY 2004.
- [14] Kwong R. H, Johnston E. W : 'A variable step size LMS algorithm'. IEEE Trans. Signal Process (July) 1992; vol. 40, no. 7, pp. 1633-1642.
- [15] Rédha Bendoumia, Mohamed Djendi et Abderrezak Guessoum, Laboratoire de traitement du Signal et imagerie (LATSI), dept. Electronique Université Saad Dahlab de Blida, B.P. 270, Blida 09000, Algeria conf 3
- [16] Djendi M et Bendoumia R : 'A new coefficient two-channel backward algorithm for speech intelligibility enhancement: A subband approach' Appl.Acous.2014, vol.76, pp. 209 - 222.
- [17] Aboulnasr T, Mayyas K : 'A robust variable step-size LMS-type algorithm: analysis and simulations'. IEEE Trans. Signal Process., vol. 45, no. 3, pp. 631-639, Mar. 1997.
- [18] José Gil F. Zipf, Orlando J. Tobias, and Rui Seara: 'A VSSLMS algorithm based on error autocorrelation. ' 16 th European signal processing conference (EUSIPCO), Lausanne, Switzerland, August 25-29, 2008.

