UNIVERSITE SAAD DAHLEB DE BLIDA

Faculté des Sciences de l'ingénieur

Département d'électronique

MEMOIRE DE MAGISTERE

Spécialité : Electronique

Option : SIGNAUX ET SYSTEMES

ETUDE DES ALGORITHMES DE PROJECTIONS AFFINES APPLICATION A L'ANNULATION D'ECHO ACOUSTIQUE

Par

BELHOUT M'HAMED

Devant le jury composé de

Mr. H.SALHI	Maître de conférence A,	U. de Blida	Président
Mr. H.SAYOUD	Professeur,	USTHB	Examinateur
Mr. M.BENSELAMA	Maître de conférence A,	U. de Blida	Examinateur
Mr. M.DJENDI	Maître de conférence A,	U. de Blida	Rapporteur
Mr. A.GUESSOUM	Professeur,	U. de Blida	Co-rapporteur

Blida, Novembre 2011.

ملخص

في هذه المذكرة ، قمنا بمعالجة مشكلة إلغاء الصدى الصوتي بواسطة خوارزميات الإسقاط الدقيق.

حيث درسنا أداء هذه الخوارزميات، و أداء NLMS و FRLS

اقترحنا في هذه المذكرة ثلاثة إصدارات جديدة تستند على خوارزمية شبه الإسقاط الدقيق

كما أننا قدمنا مقارنة بين هذه الإصدارات الثلاثة الجديدة و بين NLMS و خوارزمية الإسقاط الدقيق الأصلي

RESUME

Dans ce mémoire, nous avons traité le problème d'annulation d'écho acoustique par les algorithmes de projections affines. Nous avons étudié les performances des algorithmes de projections affines, du NLMS, et FRLS.

Nous avons proposé dans ce mémoire trois nouvelles versions basées sur l'algorithme de pseudo projection affine.

Nous avons présenté aussi une étude comparative entre ces trois nouvelles versions algorithmiques, le NLMS, et l'algorithme de projection affine exacte.

ABSRACT

In this memory, we treated the problem of acoustic echo cancellation by the affine projection algorithms. We studied the performances of the affine projection algorithms, of the NLMS, and of the FRLS.

We proposed three new versions based on the pseudo affine projection algorithm.

We also presented a comparative study between these three new versions, the NLMS, and the original affine projection algorithm.

LISTE DES SYMBOLES ET DES ABREVIATIONS

$.^{T}$ Transposé dôune matrice ou dôun vecteur.

$.^{*}$ Conjugué dôun vecteur, matrice, ou nombre complexe.

$.^{H}$ Hermitien dôun vecteur, matrice.

$._{a}$ Indice désigne une variable aller.

$._{b}$ Indice désigne une variable retour.

$E[.]$ Espérance mathématique.

w_n Réponse impulsionnelle du filtre, dôordre M.

$d(n)$ Réponse impulsionnelle désirée, scalaire.

d_n Réponse désirée, vecteur dôordre K.

$y(n)$ Sortie du filtre adaptatif, scalaire.

u_n Signal dôentrée, vecteur dôordre M.

μ Pas dôadaptation.

U_n Matrice des vecteurs du signal dôentrée de taille $K \times M$

R Matrice de corrélation.

r Vecteur dôinter-corrélation.

$e(n)$ Erreur a priori de sortie du filtre, scalaire.

$\varepsilon(n)$ Erreur a posteriori de sortie du filtre, scalaire.

e_n Erreur à priori de sortie du filtre, vecteur dôordre K.

ε_n Erreur à posteriori de sortie du filtre, vecteur dôordre K.

a_M Vecteur prédicteur aller dôordre M.

b_M Vecteur prédicteur retour dôordre M.

$e_a(n)$ Erreur a priori de prédiction aller, scalaire.

$e_b(n)$ Erreur a priori de prédiction retour, scalaire.

E_a Puissance de prédiction arrière.

E_b Puissance de prédiction arrière.

$K_{M,n}$ Vecteur désigne gain de Kalman de taille M x 1

λ Facteur dôoubli, scalaire.

$\alpha(n)$ Variance des erreurs de prédiction aller.

$\beta(n)$ Variance des erreurs de prédiction retour, scalaire.

$\gamma_{M,n}$ Variable de vraisemblance.

LMS Algorithme du gradient stochastique.

NLMS Algorithme du gradient stochastique normalisé

APA Algorithme de projection affine.

FAP Algorithme de projection affine rapide.

GS-FAP Algorithme de projection affine rapide par la méthode de Gauss-Seidel.

PAPA Algorithme du pseudo projection affine.

EQM Erreur quadratique moyenne.

RLS Algorithme des moindres carrés récursif.

FRLS Algorithme des moindres carrés récursif rapide.

LISTE DES ILLUSTRATIONS GRAPHIQUES ET TABLEAUX

TABLE DES MATIERES

INTRODUCTION GENERALE

La communication dans ses diverses formes est devenue un moyen et un outil de développement indispensable, comme le téléphone et les réseaux informatiques. En effet, la conversation téléphonique permet déjà, un haut niveau de communication par le choix des mots et le ton de la voix, la visiophonie ajoute de nouvelles dimensions à cette communication (gestes, expressions de visages, langage de corps, environnement visible,é). Des documents avec des textes, des images ou autres données peuvent être utilisés pour venir appuyer sur le face à face, dans le but dôune meilleure compréhension.

Les systèmes de communications devront avoir une qualité suffisante pour que les personnes soient placées dans des conditions de confort telles quôelles nôaient plus besoin de se déplacer pour échanger des informations. Toutefois, la qualité de la communication est souvent fortement dégradée par le phénomène dôécho acoustique.

Il sôagit du signal qui apparait, en raison du couplage acoustique, entre le haut-parleur et le microphone du terminal et qui est renvoyé au locuteur distant. Ce dernier reçoit ainsi un écho de sa voix, qui devient gênant dès que les retards de transmission atteignent la trentaine de millisecondes.

L'annulation d'écho acoustique est un problème qui se pose dans un grand nombre d'application (téléphonie main libre, téléconférenceé ..). En télécommunication il est souvent nécessaire d'éliminer des échos gênants, c'est notamment le cas pour la transmission de données en mode bidirectionnel simultané sur deux fils ou pour la transmission téléphonique par satellite les échos proviennent de réflexions des signaux électriques. Les réflexions acoustiques peuvent aussi être gênantes dans les terminaux téléphoniques à mains libres pour les salles d'audio ou de visioconférence.

Les difficultés essentielles rencontrées sont d'une part liées à la durée des réponses impulsionnelles des canaux acoustiques à identifier et d'autre part à la nature des signaux à traiter [1].

Lôécho acoustique joue un rôle décisif quant à la poursuite ou non dôune conversation conviviale. Un écho non contrôlé peut sôavérer intolérable.

Un traitement spécifique doit être impérativement mis en î uvre pour préserver la qualité de la communication. Ce traitement, qui consiste en lôannulation de lõ echo acoustique, est une application typique du filtrage adaptatif.

Le principe de cette application réside à lôidentification dôun canal de couplage acoustique dôentrée (parole du locuteur distant) et de sortie (écho plus éventuellement parole du locuteur local) observables, et se résout par le moyen des algorithmes de filtrage adaptatif.

Plusieurs études algorithmiques publiées ont été menées à résoudre le problème dôannulation dôécho acoustique. Dans notre présent mémoire, nous nous intéressons à étudier les algorithmes de type gradient et de projection affine, dont nous proposons trois nouvelles versions algorithmiques basées sur lôalgorithme de pseudo projection. Pour cela, ce mémoire sera présenté comme suit :

Dans le premier chapitre, nous introduisons de la manière la plus simple des notions et termes essentiels de lôacoustique et du traitement de signal. Nous présentons aussi dans ce chapitre le problème de lôannulation dôécho acoustique.

Le deuxième chapitre décrit le problème dôidentification résolu par le filtrage optimal de Wiener, la prédiction linéaire, et le filtrage adaptatif. Les algorithmes de base des deux familles sont aussi étudiés dans ce chapitre en mettant en î uvre lôessentiel des équations qui les formulent.

Dans le troisième chapitre, nous proposons trois nouvelles versions basées sur lôalgorithme de pseudo projection affine, en utilisant trois structures de blanchiment. Nous avons publié ces versions dans un papier IEEE intitulé « Three pre-whitened versions of the pseudo affine projection algorithm for acoustic echo cancellation » [23].

Le quatrième chapitre concerne lôétude comparative entre les performances des différents algorithmes adaptatifs présentés dans le deuxième et le troisième chapitre. Ces performances sont exprimées en qualité de vitesse de convergence, dôerreur résiduelle et en terme de complexité de calcul.

Enfin, notre mémoire sera terminé par une conclusion générale.

CHAPITRE 1
GENERALITES SUR L'ACOUSTIQUE ET NOTION DE BASESUR LE TRAITEMENT DE SIGNAL

1.1 Introduction

L'acoustique peut être définie comme la science qui étudie le son, incluant sa production, transmission et ses effets. Elle n'est donc pas limitée au seul phénomène responsable de la sensation d'audition. Elle se distingue de l'optique par le caractère mécanique plutôt qu'électromagnétique des ondes sonores [1].

Dans ce chapitre, nous présentons quelques définitions de l'acoustique et de traitement du signal ; le but ici n'étant pas de dresser un compte rendu exhaustif des connaissances en acoustique mais d'introduire de la manière la plus accessible possible des notions et termes essentiels à la bonne compréhension du manuscrit, des problèmes et des choix ultérieurs. Nous introduirons tout d'abord des notions générales concernant la parole en qualité de niveau acoustique. Nous définirons également la notion de réponse impulsionnelle d'un canal acoustique, essentielle pour la suite du document. Puis nous présenterons le phénomène de la réverbération. Enfin nous aborderons le problème d'écho acoustique.

1.2 Théorie et Traitement de Signal

La théorie et le traitement des signaux est une discipline auxquelles elle apporte des bases théoriques fondamentales et des techniques particulières .En faite la théorie et le traitement des signaux intéressent tous les secteurs techniques et scientifiques dans lesquels l'information et perçue par l'intermédiaire d'observation expérimentales des grandeurs mesurables [2].

1.2.1 Notion de signal

On peut définir un signal comme le support physique d'une information, par exemple, les signaux sonores sont des fluctuations de la pression de l'air transportant un message à notre oreille [2].

Les signaux visuels sont des ondes de lumière apportant une information à notre œil. En mathématique, les signaux sont en fonction d'une variable. Généralement, cette variable est le temps [2].

Soit x (t) une grandeur physique mesurable dépendant d'un paramètre t appartenant à A où A est l'ensemble de variable t tel que x (t) $\in$ R.

- Alors la nature de x (t) peut représenter n'importe qu'elle phénomène physique soit :

- Electrique
- Signal optique
- Température
- Signal magnétique
- Pression
- Mode acoustique

- La variable t peut désigner n'importe qu'elle variable indépendante.

- Temps
- Position spatiale
- Position angulaire

1.2.2 Classe des signaux

Les signaux peuvent être répartis en deux grandes catégories qui sont les signaux analogiques et les signaux numériques.

1.2.2.1 Signaux analogiques

Ce sont des fonctions continues, définies dans un espace à deux dimensions dont leurs courbes représentatives indiquent les variations des signaux en fonction du temps.

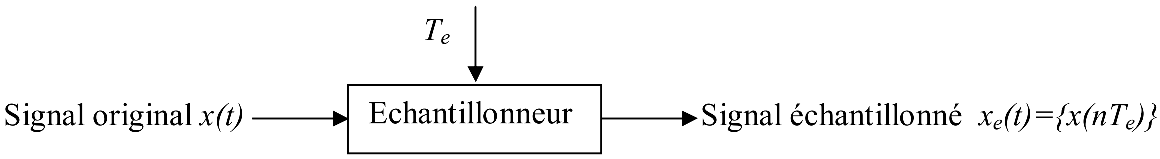
1.2.2.2 Signaux numériques

Afin de rendre possible le traitement d'un signal x(t), on doit procéder a la numérisation qui est illustrée par les opérations de la Figure 1.1, c'est-à-dire:

- <u>Echantillonnage</u>

L'échantillonnage consiste à transformer un signal analogique (continu) en signal numérique (discret), en capturant des valeurs à intervalle de temps régulier t_n [3]. Généralement les t_n sont régulièrement espacés d'une période $Te = t_{n+1} - t_n$, appelé période d'échantillonnage. On obtient la suite de valeurs $x_e(t) = \{x(t_n)\}$ avec

$t_n = nTe.$



Echantillonnage idéal : prélèvement pendant un temps infiniment court des valeurs de x(t) a $t = nTe$ (multiple entier de Te).

- <u>Quantification</u>

En traitement du signal, la quantification est le procédé qui permet d'approximer un signal continu (ou à valeurs dans un ensemble discret de grande taille) par des valeurs d'un ensemble discret d'assez petite taille [3].

- <u>Codage</u>

Le codage consiste à associer à un ensemble de valeurs discrètes un code composé d'éléments binaires. Les codes les plus connus: code binaire naturel, code binaire décalé, code complément à 2, code DCB, code Gray.

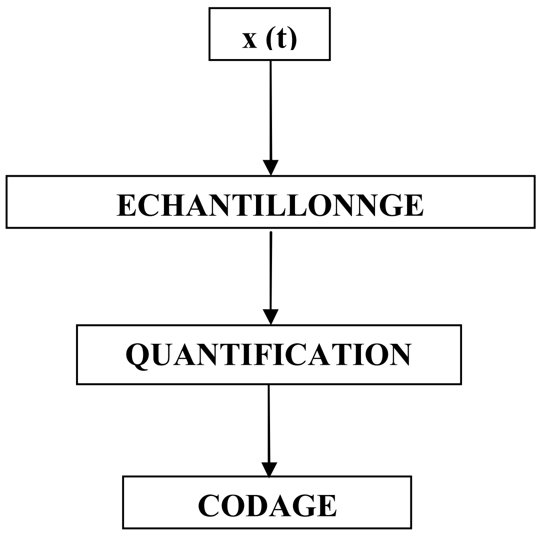


Figure 1.1 **:** opération d'un signal subit avant son traitement sur un calculateur numérique

1.2.3 Notion de système

Un système est un dispositif ou bien un phénomène physique qui transforme un signal d'entrée (cause, excitation) en un signal de sortie (l'effet, réponse). On a plusieurs types de systèmes (électrique, biologique (la parole), optique).

Un système est dit analogique si les signaux traités par ce système sont analogiques (exemple : circuit RLC)

Un système est dit numérique si les signaux traits par ce système sont numériques (exemples : porte logique, unité arithmétique et logique).

1.2.4 Principales fonctions (opérations)

Les principales fonctions du traitement du signal sont :

- Convolution
- Filtrage
- Corrélation
- Représentation spectrale (TF)
- Echantillonnage # reconstitution
- Quantification (code en binaire DCD)
- Détection, estimation décision

1.2.4.1 Convolution

Soient deux signaux discrets *x (n)* et *y (n),* on définie la convolution linéaire par :

$$z(n) = x(n) * y(n) = \sum_{m=-\infty}^{+\infty} x(m)y(n-m)$$

1.2.4.2 Filtrage numérique

Le filtrage numérique est une opération qui est utilisée pour modifier la distribution fréquentielle des composantes d'un signal selon des spécifications données. Le filtrage numérique permet la modification de cette distribution. Le problème général du filtrage numérique consiste à déterminer une fonction de transfert H (z) qui représente d'une part la réponse fréquentielle voulue et d'autre part, se prêt à une réalisation efficace [2]. Les filtres numériques se divisent en deux classes selon la durée de la réponse impulsionnelle.

- Filtre non récursif (RIF)

Un filtre a réponse impulsionnelle de durée finie est toujours stable pour autant que les valeurs *h(n)* de la réponse impulsionnelle soient toutes finies. Son fonctionnement est régi par une équation de convolution portant sur un nombre fini de termes :

$$y(n) = \sum_{i=0}^{N-1} h(i)x(n-i) \tag{1.1}$$

Et sa transformée en z est donnée par :

$$H(z) = \frac{Y(z)}{X(z)} = \sum_{n=0}^{N-1} h(n)z^{-n} \tag{1.2}$$

De cette relation on a *h (n) =0* pour : *n > N-1* avec *N* est le nombre de coefficients du filtre. Ce type de filtre ne tient pas compte des réponses précédentes, d'où l'absence de bouclage de la sortie avec l'entrée, pour cela on l'appelle non récursif.

- Filtre récursif (RII)

Ce sont des systèmes linéaires invariants dans le temps, leur fonctionnement est régi par une équation de convolution portant sur une infinité de termes.

$$y(n) = \sum_{i=0}^{M} b_i x(n-i) - \sum_{i=1}^{L} a_i y(n-i) \tag{1.3}$$

En appliquant la transformée en *z* à *y (n)* on obtient :

$$Y(z) = \sum_{i=0}^{M} b_i X(z) z^{-i} - \sum_{i=1}^{L} a_i Y(z) z^{-i} \tag{1.4}$$

$$\Rightarrow Y(z)[1 + \sum_{i=1}^{L} a_i z^{-i}] = \sum_{i=0}^{M} b_i z^{-i}.$$

D'où la réponse impulssionelle de *H(z)* devient :

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{i=0}^{M} b_i z^{-i}}{1 + \sum_{i=1}^{L} a_i z^{-i}} \tag{1.5}$$

Tel que : $M \leq L$.

Ce filtre garde une trace des réponses précédentes durant une durée infinie, ce sont des filtres à mémoire, pour cela on les appelle filtres récursifs [2].

1.2.4.3 Corrélation

On peut définir une certaine fonction de corrélation entre deux signaux *x (n)* et *y (n)*. Cette fonction exprime l'influence d'un signal sur un autre. Son rôle consiste à déterminer à partir d'une méthode de comparaison s'il existe une relation entre les deux signaux, le résultat de cette comparaison est un réel appartenant à l'intervalle [-1,-1] .on dit qu'il y a une forte corrélation si le résultat se rapproche de 1 .on distingue deux opérations :

- auto-corrélation

Elle consiste a comparer une fonction *x(t)* avec elle-même durant un intervalle de temps, dont l'une est décale d'une certaine valeur τ , elle est définie par [4] :

$$R_{xx}(\tau) = \sum_{i=-\infty}^{+\infty} x(i)x^*(i-\tau) \tag{1.6}$$

- Inter-corrélation

Elle consiste à comparer deux fonctions différentes x (t) et y (t) dont l'une est décale d'une certaine valeur τ , telle que [4].

$$R_{xy}(\tau) = \sum_{i=-\infty}^{+\infty} x(i)y^*(i-\tau) \tag{1.7}$$

1.2.4.4 Analyse de Fourrier

L'analyse de Fourier est l'instrument majeur de la théorie du signal. Elle permet la décomposition d'un signal en une somme de signaux élémentaires, qui ont la propriété d'être facile à mettre en œuvre et à observer. La représentation spectrale des signaux déterministes est obtenue grâce à la transformation de Fourier généralisée par l'emploi des distributions, cette représentation exprime la composition fréquentielle de l'amplitude, de la phase et de l'énergie ou de la puissance des signaux considérés [3].

Il y a deux représentations importantes de signal selon la nature de la variable indépendant :

Représentation temporelle qui a comme variable le paramètre temps *t*, et représentation fréquentielle qui a comme variable le paramètre fréquence *f*.

1.2.4.5 Représentation fréquentielle

Les méthodes utilisées pour calculer les représentations spectrales, ne sont pas les mêmes selon ces différents types de signaux. Pour le cas d'un signal numérique on utilise LA TFD.

On appelle Transformée de Fourier Discrète (TFD ou DFT : Discrete Fourier

Transform) d'un signal défini par N échantillons *x(n),* la suite de N termes *X(k)* définie par :

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{\frac{-j2\pi nK}{N}}$$

NB: Nous notons que ces deux représentations du signal sont reliées entre-elles par la transformation de Fourier, cette Transformation joue un rôle fondamental en traitement du signal.

1.2.4.6 Echantillonnage et la reconstitution du signal

Les signaux primaires porteurs d'information sont pratiquement toujours analogiques (amplitude et temps continus). Un ordinateur ou tout autre système numérique est un dispositif qui traite les données numériques, il y a apparemment incompatibilité.

Si l'on veut traiter un signal par la technique numérique il faut le représenter au préalable par une suite de valeurs ponctuelles régulière à un tel prélèvement. Cette opération est appelée échantillonnage [5].

L'opération inverse de l'échantillonnage est la reconstitution et pour qu'elle soit correcte il faut au moins la fréquence d'échantillonnage fe soit deux fois plus grande que la plus grande des fréquences du spectre du signal (Théorème de Shannon).

1.3 Généralité sur l'acoustique

Dans cette partie, nous nous intéressons au signal de parole qui est utilisé dans la communication en téléphonie, en audio et visioconférence.

1.3.1 Description du signal parole

La parole est la faculté de communiquer la pense par un système de sons articulés ; c'est le moyen de communication privilégié entre les humains qui sont les seuls êtres vivants a' utiliser un tel système .l'information d'un message parlé réside dans les fluctuations de la pression de l'air, engendrées, puis émises, par l'appareil phonatoire. Ces fluctuations constituent le signal vocal ; elles sont détectées par l'oreille, laquelle procède à une certaine analyse [1].

L'intensité du son émis est liée à la pression de l'air en amont du l'harynx ; sa hauteur est fixée par la fréquence de vibration des cordes vocales, appelée fréquence fondamental ou pitch. Cette dernière peut varier comme suit :

- ✓ de 70 à 150 Hz pour une voix masculine.
- ✓ de 100 à 400 Hz pour une voix féminine.
- ✓ de 200 à 600 Hz pour une voix d'entant.

Donc la parole est le résultat volontaire et coordonnée des appareils respiratoire et masticatoire. Cette action se déroule sous le contrôle du système nerveux central qui reçoit en permanence des informations par rétroaction auditive et par la sensation cénesthésique [1].

1.3.2 Audiogramme

L'échantillonnage transforme le signal à temps continu *x(t)* en signal à temps discret *x(nTe)* défini aux instants d'échantillonnage, multiples entiers de la période d'échantillonnage *Te*; celle-ci est elle-même l'inverse de la fréquence d'échantillonnage *fe*. Pour ce qui concerne le signal vocal, le choix de *fe* résulte d'un compromis. Son spectre peut s'étendre jusque 12 kHz. Il faut donc en principe choisir une fréquence *fe* égale à 24 kHz au moins pour satisfaire raisonnablement au théorème de Shannon [3]. Cependant, le coût d'un traitement numérique, filtrage, transmission, ou simplement enregistrement peut être réduit d'une façon notable si l'on accepte une limitation du spectre par un filtrage préalable.

C'est le rôle du filtre de garde, dont la fréquence de coupure *fc* est choisie en fonction de la fréquence d'échantillonnage retenue. Pour la téléphonie, on estime que le signal garde une qualité suffisante lorsque son spectre est limité à 3400 Hz et l'on choisit *fe* = 8000 Hz. Pour les techniques d'analyse, de synthèse ou de reconnaissance de la parole, la fréquence peut varier de 6000 à 16000 Hz. Par contre pour le signal audio (parole et musique), on exige une bonne représentation du signal jusque 20 kHz et l'on utilise des fréquences d'échantillonnage de 44.1 ou 48 kHz. Pour les applications multimédia, les fréquences sous-multiples de 44.1 kHz sont de plus en plus utilisées : 22.5 kHz, 11.25 kHz. Parmi le continuum des valeurs possibles pour les échantillons *x(nTe)*, la quantification ne retient qu'un nombre fini *2b* de valeurs (*b* étant le nombre de bits de la quantification), espacées du pas de quantification *q*. Le signal numérique résultant est noté *x(n)*. La quantification produit une erreur de quantification qui normalement se comporte comme un bruit blanc; le pas de quantification est donc imposé par le rapport signal à bruit à garantir. Si le pas de quantification est constant, ce rapport est fonction de l'amplitude du signal; les signaux de faible amplitude sont dès lors mal représentés. Aussi adopte-t-on pour la transmission téléphonique une loi de quantification logarithmique et chaque échantillon est représenté sur 8 bits (256 valeurs). Par contre, la quantification du signal musical exige en principe une quantification linéaire sur 16 bits (65536 valeurs) [2].

Une caractéristique essentielle qui résulte du mode de représentation est le débit binaire, exprimé en bits par seconde (b/s), nécessaire pour une transmission ou un enregistrement du signal vocal. La transmission téléphonique classique exige un débit de 8 kHz x 8 bits = 64 kb/s; la transmission ou l'enregistrement d'un signal audio exige en principe un débit de l'ordre de 48 kHz x 16 bits = 768 kb/s (à multiplier par deux pour un signal stéréophonique) [2].

La figure 1.2 représente l'évolution temporelle, ou *audiogramme*, du signal vocal pour les mots 'parenthèse', et 'effacer'. On y constate une alternance de zones assez périodiques et de zones bruitées, appelées zones *voisées et non voisées*. La figure 1.3 donne une représentation plus fine de tranches de signaux voisés et non voisés.

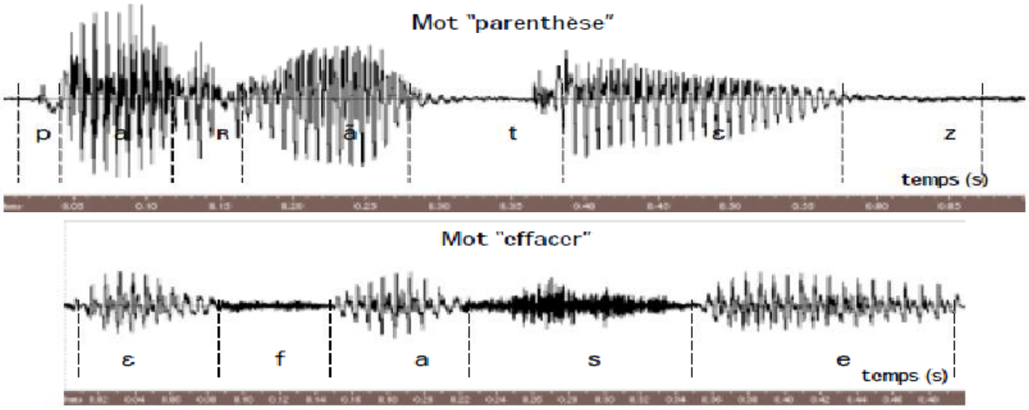


Figure 1.2 : Audiogramme de signaux de parole.

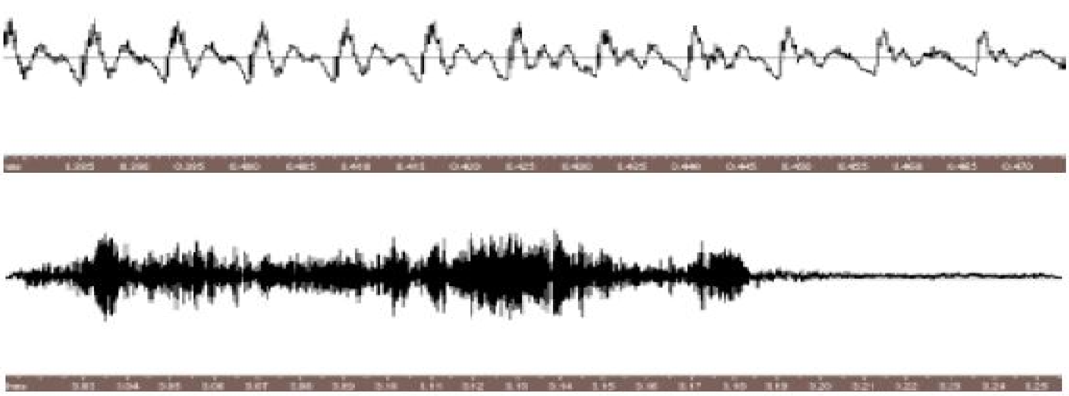


Figure 1.3 : Exemples de son voisé (haut) et non-voisé (bas).

1.3.3 Réponse impulsionnelle et canal acoustique

Supposons à présent qu'une source émette une impulsion de Dirac, et considérons le signal reçu par un récepteur. Au cours de leur propagation dans leur milieu (nous supposerons dans la suite de ce document qu'il s'agit de l'air), les ondes sonores sont modifiées de manière complexe (nous le verrons par la suite). Le signal reçu par le récepteur ne sera donc pas dans le cas général une impulsion de Dirac.

On supposera que les phénomènes de propagation dépendent de paramètres physiques (pression atmosphérique, température) qui varient lentement avec le temps et seront donc considérés indépendants du temps. On appellera alors *réponse impulsionnelle* le signal (pression acoustique) reçu par le récepteur lorsque la source émet une impulsion de pression (ponctuelle) dans le temps (mesure de Dirac).

La transformation subie par un signal sonore, que nous supposerons linéaire, peut en conséquence être exprimée comme un produit de convolution dans le domaine temporel, correspondant à un *filtrage* du signal d'origine :

$$y(t) = \int_{-\infty}^{+\infty} h(\tau)x(t-\tau)d\tau \qquad (1.7)$$

Où $y(t)$ est le signal reçu, $x(t)$ le signal d'origine et $h(t)$ est la réponse impulsionnelle de l'environnement.

On peut également exprimer cette relation par un produit simple dans le domaine fréquentiel :

$$Y(f) = X(f)H(f) \qquad (1.8)$$

Où $Y(f)$ et $X(f)$ sont les transformées de Fourier des signaux $y(t)$ et $x(t)$ et $H(f)$ est la *fonction de transfert* de l'environnement, transformée de Fourier de la réponse impulsionnelle $h(t)$.

La réponse impulsionnelle est porteuse de toute l'information concernant les modifications du son durant sa transmission depuis la source jusqu'au récepteur que l'on référence souvent comme le *canal acoustique*.
Sa connaissance permet donc de reconstruire, grâce à l'équation de filtrage (1.7), le signal reçu par un récepteur depuis une source dans un environnement donné.
On notera ici que l'équation (1.7) est généralement utilisée en traitement numérique du signal sous sa forme discrète :

$$y(n) = \sum_{m} h(m)x(n-m)\ ,\ (n,m) \in N^2 \qquad (1.9)$$

1.3.4 La réverbération

La réverbération tient une place très importante dans la musique : elle est utilisée pratiquement tout le temps. Elle permet d'apporter de la profondeur à la musique, mais aussi également de gommer les petits défauts.
La qualité de la réverbération est cruciale car ce n'est pas un effet comme les autres : C'est un phénomène qui existe à l'état naturel. Une bonne réverbération doit donc avant tout être réaliste et belle, puisque tout le monde à comme point de repère sa manifestation naturelle. Tout le monde connaît le son qu'il y a dans une mosquée, dans un hall, dans un appartement vide dans lequel on emménage.

1.3.4.1 Origine de la réverbération

Lorsqu'un objet produit un son, celui-ci se propage non pas directement de la source à l'auditeur, mais de manière diffuse. Certaines ondes qui n'étaient pas initialement dirigées vers l'auditeur lui parviendront quand même après s'être réfléchies sur les parois de l'environnement.

1.3.4.2 Structure temporelle de la réverbération

Il est intéressant d'examiner la structure temporelle et fréquentielle de la réponse impulsionnelle d'un environnement réverbérant, en particulier dans le cas d'un volume clos puisque cela va traduire l'effet de salle. Généralement il est plus simple de l'étudier en examinant un *échogramme*, représentant la puissance instantanée dans la réponse impulsionnelle en fonction du temps.

La structure temporelle de la réponse impulsionnelle d'un environnement réverbérant peut être généralement divisée en trois parties, clairement visibles sur l'´echogramme (*cf.* Figure 1.9 (a)) :

- ✓ le *son direct* arrivant directement depuis la source jusqu'au récepteur.
- ✓ les *réflexions précoces*, contributions de l'onde sonore ayant subi un faible nombre de réflexions (de l'ordre de 1 `a 5 en moyenne) avant d'arriver au récepteur et qui sont temporellement séparables.
- ✓ la *réverbération tardive*, dans laquelle de très nombreuses réflexions d'ordre élevé se superposent, formant un continuum et ne pouvant plus être individuellement séparées.

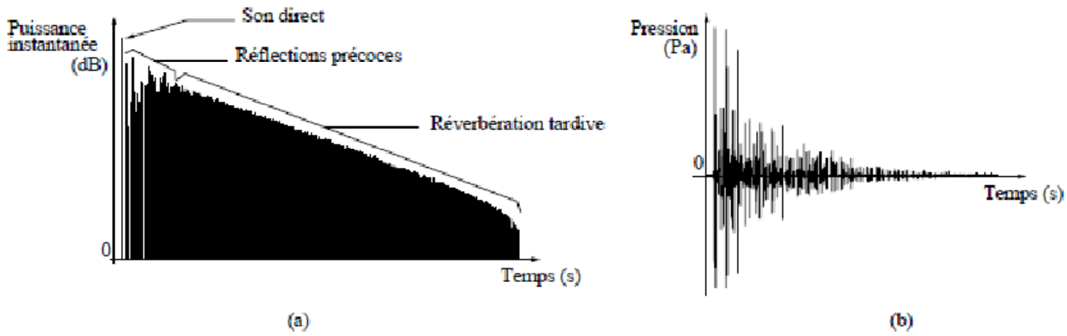


Figure 1.4 : (a) Echogramme et structure temporelle de la réponse impulsionnelle. (b) Réponse impulsionnelle en pression

1.3.5 Echo

L'écho est une onde électrique, acoustique ou électromagnétique qui parvient à un point donné après une réflexion ou une propagation indirecte, avec une intensité et un retard suffisants pour être perçue, en ce point, comme distincte de l'onde directe [1].

L'écho est employé utilement dans les sonars et radars pour la détection et l'exploration. Alors que dans les télécommunications, l'écho peut dégrader la qualité de service. Dans un système de communication, l'effet perçu de l'écho dépend de son amplitude et de sa temporisation. En général, les échos avec une amplitude sensible et un retard de plus de 1 ms sont perceptibles.

1.3.6 L'annulation d'écho acoustique

L'audioconférence est une technique permettant d'établir la communication entre deux salles de réunion. La prise du son dans chacun des salles considère l'orateur comme la source utile.

Pour un délai de transmission important, les personnes présentées dans la salle réentendent leurs propres voix, c'est le phénomène d'écho acoustique dû au canal acoustique du couplage.

Malheureusement, tous les équipements mains libres classiques ont un problème de réaction acoustique qui dépend de la disposition du haut- parleur et du microphone, de leur environnement immédiat, et de l'endroit où ils sont utilisés [1].

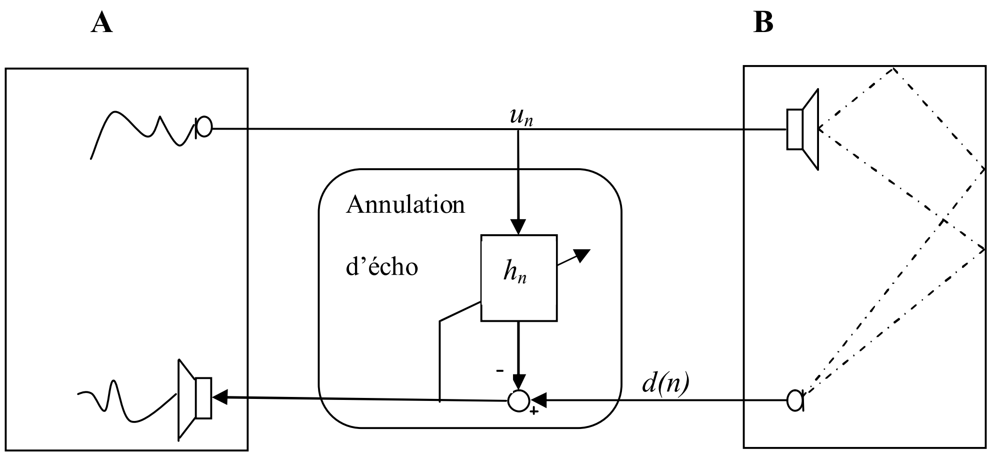


Figure 1.5 : Structure classique d'annulation d'écho acoustique.

Le schéma ci-avant représente un système classique d'annulation d'écho dans un système de communication sonore (téléphone mains libres, téléconférence,…), ou u_n est le signal reçu du locuteur lointain, $d(n)$ est le signal d'écho du locuteur lointain vers lui-même.

Lorsqu'un locuteur parle dans la pièce A, le haut-parleur de la pièce B émet le signal u_n. Le microphone de la pièce B reçoit une version filtrée de u_n. Etant directement relié au haut-parleur de la pièce A, le locuteur va s'entendre parler.

Pour éviter cela, on estime de manière adaptative le filtre h_n et on envoie sur le haut-parleur de la pièce A uniquement l'erreur commise.

La solution optimale du problème de l'estimation h_n est fournie par la solution de l'équation de Wiener-Hopf que nous allons voir dans le prochain chapitre.

1.4 Conclusion

Dans ce chapitre, nous avons présenté quelques bases de traitement de signal et d'acoustique physique, ainsi que le problème de l'annulation d'écho acoustique. Dans le prochain chapitre nous allons aborder l'identification du canal acoustique par le filtrage adaptatif en mentionnant les équations nécessaires, en parlant aussi sur les deux grandes familles des algorithmes de filtrage adaptatifs LMS et RLS.

CHAPITRE 2

NOTIONS DE FILTRAGE ADAPTATIF, DE PREDICTION ET REPRESENTATION DES ALGORITHMES ADAPTATIFS DES TYPE GRADIENT ET MOINDRES CARREES

2.1 Introduction

Le présent chapitre concerne des tâches d'optimisation du traitement du signal. Il en présente le filtrage optimal de *Wiener*, la prédiction, et le filtrage adaptatif qui est utilisé de façon répandue en annulation d'échos, en égalisation de canaux, en réduction du bruit, en annulation de la rétroaction dans les appareils auditifs, et en contrôle actif des bruits, etc.

Les différentes classes dôalgorithmes adaptatifs sont construites et discutées : LMS et RLS. Nous intéressons aussi aux différents algorithmes adaptatifs de projection affine (APA) qui présentent un bon candidat, et qui sont basés comme le NLMS et LMS sur lôapproximation instantanée [4] de la matrice de corrélation et de vecteur de covariance.

2.2 Filtrage de Wiener

Dans cette partie nous exposerons lôapproche statistique du problème (*filtrage de Wiener*) qui suppose la disponibilité de certaines grandeurs statistiques (*moyenne et auto-corrélation*) du signal utile et du bruit. Lôapproche consiste alors à minimiser la moyenne statistique du carré de lôerreur (EQM) entre lôinformation désirée et la sortie du filtre.

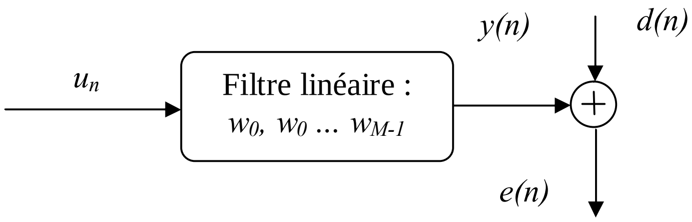


Figure 2.1 : représentation du problème statistique du filtrage.

Le problème du filtrage optimal est de trouver le meilleur filtre côest à dire celui permettant dôobtenir en sortie une réponse *y(n)* la plus proche possible dôune réponse désirée *d(n)* lorsque lôentrée est une certaine séquence u_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 la réponse impulsionnelle du filtre.

La sortie du filtre $y(n)$ s'écrit :

$$y(n) = \sum_{k=0}^{M-1} w^*(k)u(n-k) \quad (2.1)$$

Où :

$w_n = [w_0, w_1 \ldots\ldots\ldots\ldots w_{M-1}]$

$u_n = [u(n), u(n-1) \ldots\ldots\ldots\ldots u(n-M+1)]$

Le filtre de Wiener est celui qui minimise l'erreur quadratique moyenne (EQM).

$$J = E\left[e(n)e^*(n)\right] = E\left[e(n)^2\right] \quad (2.2)$$

En introduisant les vecteurs : w_n et u_n on aura :

$$e(n) = d(n) - w_n^H u_n \quad (2.3)$$

d'où : $J = E\left[(d(n) - w_n^H u_n)(d^*(n) - w_n^H u_n^*)\right]$ (2.4)

$$J = E\left[d^2(n)\right] - w_n^H E\left[u_n d^*(n)\right] - w_n^T E\left[u_n^* d(n)\right] + w_n^H E\left[u_n u_n^H\right] w_n$$

(2.5)

Par conséquent on aura:

$$J = \sigma_d^2 - w_n^H r - w_n^T r^* + w_n^H R\ w_n \quad (2.6)$$

Avec :

r : Le vecteur d'inter-corrélation entre la sortie désirée $d(n)$ et l'entrée u_n.

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

Cherchons le vecteur optimum celui qui annule le gradient du critère : $\nabla J_k = 0$.

En écrivant J sous la forme $J = E\left[e(n)e^*(n)\right]$

On prend : $w(k) = a(k) + jb(k)$

Donc on aura :

$$\nabla_k J = E\left(e_n \frac{\partial e_n^*}{\partial a(k)} + e_n^* \frac{\partial e_n}{\partial a(k)} + e_n \frac{\partial e_n^*}{\partial b(k)} + e_n^* \frac{\partial e_n}{\partial b(k)} \right) \tag{2.7}$$

Où les dérivées partielles intervenant dans lʼéquation (2.7) sont comme suit :

$$\frac{\partial e_n^*}{\partial a(k)} = \frac{\partial \left[d(n) - \sum_{k=0}^{M-1} w^*(k) u(n-k) \right]^*}{\partial a(k)} = \frac{\partial \left[d^*(n) \right]}{\partial a(k)} - \frac{\partial \left[\sum_{k=0}^{M-1} w\ (k) u^*(n-k) \right]}{\partial a(k)} = -u^*(n-k)$$

$$\frac{\partial e_n}{\partial a(k)} = \frac{\partial \left[d(n) - \sum_{k=0}^{M-1} w^*(k) u(n-k) \right]}{\partial a(k)} = \frac{\partial \left[d(n) \right]}{\partial a(k)} - \frac{\partial \left[\sum_{k=0}^{M-1} w^*(k) u(n-k) \right]}{\partial a(k)} = -u(n-k)$$

$$\frac{\partial e_n^*}{\partial b(k)} = \frac{\partial \left[d(n) - \sum_{k=0}^{M-1} w^*(k) u(n-k) \right]^*}{\partial b(k)} = \frac{\partial \left[d^*(n) \right]}{\partial b(k)} - \frac{\partial \left[\sum_{k=0}^{M-1} w\ (k) u^*(n-k) \right]}{\partial b(k)} = -ju^*(n-k)$$

$$\frac{\partial e_n}{\partial b(k)} = \frac{\partial \left[d(n) - \sum_{k=0}^{M-1} w^*(k) u(n-k) \right]}{\partial b(k)} = \frac{\partial \left[d(n) \right]}{\partial b(k)} - \frac{\partial \left[\sum_{k=0}^{M-1} w^*(k) u(n-k) \right]}{\partial b(k)} = ju(n-k)$$

Par conséquent on a :

$$\nabla J_k = -2E\left[u(n-k) e_n^* \right] \qquad \text{k=0,1,…} \tag{2.8}$$

On note e_0 la valeur à lʼoptimum :

$$E\left[e_o^* u_n \right] = 0 \tag{2.9}$$

Cʼest le *principe d’orthogonalité* [3], signifiant que toutes les entrées u_n sont décorrélées de e_n^*.

En développant lœquation (2.8), on obtient :

$$E(u_n d_n^* - u_n u_n^H w) = 0 \tag{2.10}$$

Soit : $R\ w_n = r$ (2.11)

Cette relation (2.11) est appelée Formule de *Wiener* ou équation de *Wiener-Hopf*
Cette solution donne le filtre optimal de *Wiener* :

$$w_n = R_u^{-1} R_{du} \tag{2.12}$$

Lœquation de Wiener-Hopf qui permet de calculer le filtre de Wiener optimal conduit à résoudre un système de M équations à M inconnues :
Il peut être préférable de résoudre ce système par une méthode itérative (algorithme), notamment en se souvenant que la fonction de coût est quadratique, ce qui entraîne que le minimum est unique.

Les algorithmes adaptatifs permettent lôestimation du filtre adaptatif par le vecteur w_n de taille M à lôaide dôun critère basé sur lôerreur dôestimation a priori [4].

Cette erreur dôestimation, appelée précédemment signal de différence, sœcrit pour chaque échantillon n :

$$e(n) = y(n) - w_n^T u_n \tag{2.13}$$

Où : u_n est le vecteur colonne des M derniers échantillons du signal haut-parleur.

w_n^T : désigne un vecteur ligne dôordre M contenant les coefficients de la réponse impulsionnelle finie. Lôexposant T : désigne lôopérateur de transposition.

La mise à jour du filtre à chaque instant est effectuée par une contre réaction de lôerreur dôestimation proportionnellement au gain dôadaptation (terme de correction).

2.3 Prédiction linéaire

La prédiction linéaire fait partie intégrante des filtres adaptatifs basés sur les algorithmes de moindres carrés rapides. De plus, elle joue un rôle important dans de nombreuses applications, notamment lôanalyse des signaux et la compression.

On a deux types de prédiction, la prédiction linéaire avant et la prédiction retour (arrière) [4] :

2.3.1 Prédiction linéaire avant

Le but de la prédiction linéaire avant est dőestimer la valeur dőun signal à lőinstant n à partir de ses valeurs aux instants antérieurs $n-1$, $n-2$, . . .

Lőerreur de prédiction linéaire avant sőécrit:

$$e_a(n) = u(n) - \sum_{m=1}^{M} a_{M,m}\, u(n-m) \tag{2.14}$$

Où : $a_M = \left[a_{M,1}, a_{M,1} \ldots a_{M,M}\right]^T$ est le vecteur prédicteur avant à M élément, et

$u_{n-1} = \left[u(n-1), u(n-2), \ldots u(n-M)\right]^T$.

On cherchera à minimiser le critère suivant :

$$J_a = E\left[e_a(n)^2\right] \tag{2.15}$$

Or on a : $\dfrac{\partial J_a}{\partial a_M} = 2\,E\left[e_a(n)\ \dfrac{\partial e_a(n)}{\partial a_M}\right] = -2\,E\left[e_a(n)\ u_{n-1}\right]$

A lőoptimum on a : $\dfrac{\partial J_a}{\partial a_M} = 0$ ce qui donne :

$$E\left[e_a(n)\ u_{n-1}\right] = 0_{M\times 1} \tag{2.16}$$

Remplaçant (2.14) dans (2.16) on trouve : $E\left[\left(u(n) - u_{n-1}^T\ a_M\right)u_{n-1}\right] = 0_{M\times 1}$

Cőest-à-dire :

$$E\left[u_{n-1}\ u_{n-1}^T\right]\ a_M = E\left[u_{n-1}\ u(n)\right] \tag{2.17}$$

Finalement on aura :

$$R_M \, a_M = r_a \tag{2.18}$$

Où : R_M est la matrice dôauto-corrélation de dimension MxM, r_a est le vecteur dôinter-corrélation, et lôindice a : désigne une variable aller.
Le système précèdent peut être formulé différemment en augmentant sa taille :

$$\begin{bmatrix} r(0) & r_a^T \\ r_a & R_M \end{bmatrix} \begin{bmatrix} 1 \\ -a_M \end{bmatrix} = \begin{bmatrix} E_a \\ 0_{M\times 1} \end{bmatrix} \Rightarrow R_{M+1} \begin{bmatrix} 1 \\ -a_M \end{bmatrix} = \begin{bmatrix} E_a \\ 0_{M\times 1} \end{bmatrix} \tag{2.19}$$

Où $E_a = r(0) - r_a^T a_M$ est la puissance de prédiction avant (aller). Cette dernière est donnée par : $E_a = J_{a,\min}$ $(EQM \text{ min})$.

2.3.2 Prédiction linéaire retour (arrière)

Le but de la prédiction linéaire arrière est dôestimer la valeur du signal à lôinstant n-M à partir de ces valeurs aux instants futur n, n-1, é
Lôerreur de prédiction linéaire arrière sôécrit :

$$e_b(n) = u(n-M) - \sum_{m=1}^{M} b_{M,m} u(n-m+1) \tag{2.20}$$

Où $b_M = \left[b_{M,1} \;\; b_{M,2} \;\; \dots \;\; b_{M,m} \right]$ est le vecteur de prédiction arrière à M élément.
et $u_n = \left[u(n), u(n-1), ..u(n-M+1) \right]^T$

On cherche à minimiser le critère :

$$J_{b,M} = E\left[e_b^2(n) \right] \tag{2.21}$$

Ce qui donne : $\frac{\partial J_b}{\partial b_M} = 2\,E\left[e_b(n)\ \frac{\partial e_b(n)}{\partial b_M}\right] = -2\,E\left[e_b(n)\ u_n\right]$ (2.22)

A lôoptimum on a : $\frac{\partial J_b}{\partial b_M} = 0$ ce qui donne : $E\left[e_b(n)\ u_n\right] = 0_{M\times 1}$ (2.23)

Remplaçant (2.20) dans (2.23) on trouve :

$$E\left[\left(u(n-M) - u_{n-1}^T\, b_M\right) u_{n-M}\right] = 0_{M\times 1} \quad (2.24)$$

C'est-à-dire que : $E\left[u_n\, u_n^T\right]\ b_M = E\left[u_n\, u(n-M)\right]$ (2.25)

Finalement on aura : $R_M\ b_M = r_b$ (2.26)

Le système précèdent peut être formulé différemment en augmentant sa taille :

$$\begin{bmatrix} R_M & r_b \\ r_b^T & r(0) \end{bmatrix} \begin{bmatrix} -b_M \\ 1 \end{bmatrix} = \begin{bmatrix} 0_{M\times 1} \\ E_b \end{bmatrix} \Rightarrow R_{L+1} \begin{bmatrix} -b_M \\ 1 \end{bmatrix} = \begin{bmatrix} 0_{M\times 1} \\ E_b \end{bmatrix} \quad (2.27)$$

Où $E_b = r(0) - r_b^T b_M$ est la puissance de prédiction arrière, qui est donnée aussi par : $E_b = J_{b,\min}$ $(EQM$ min) .

2.3.3 Relation entre la prédiction avant et arrière

Soit la matrice carré dôordre M, dite co-identité suivante :

$$J_M = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 1 & 0 \\ . & . & . & . & . \\ 0 & 1 & \cdots & 0 & 0 \\ 1 & 0 & . & 0 & 0 \end{bmatrix}$$

On peut facilement vérifier que : $R_M\ J_M = J_M\, R_M$

La matrice R_M est aussi symétrique par rapport à la seconde diagonale, elle est dite doublement symétrique ou persymetrique [3].

Rappelons que : $R_M \, b_M = r_b$

En multipliant lôexpression (2.26) par la matrice co-identité J_M :

$$J_M \, R_M \, b_M = J_M \, r_b \Rightarrow R_M \, J_M \, b_M = r = R_M \, a_M$$

Comme R_M est supposé inversible il vient :

$$a_M = J_M b_M$$

Dôautre part :

$$E_b = r(0) - r_b^T \, b_M$$

En introduisant la matrice co-identité J_M , on aura :

$$E_b = r(0) - r_b^T \, J_M \, J_M \, b_M$$

Remplaçons $J_M b_M$ par sa valeur on trouve :

$$E_b = r(0) - (J_M \, r_b)^T a_M$$

Donc :

$$E_b = r(0) - r_a^T a_M = E_a = E$$

Pour un signal dôentrée stationnaire, les puissances dôerreur de prédiction avant et arrière sont égales et les coefficients sont les mêmes, mais dans lôordre inverse.

2.3.4 Calcul des coefficients de prédiction linéaire

Pour le calcul des coefficients de prédiction linéaire, nous allons décrire une méthode directe qui est récursive dans sa nature, et qui utilise une structure particulière dite Toeplitz de la matrice de corrélation, appelée algorithme de Levinson-Durbin.

Cette récursivité peut être formulée avec lôune des deux façons suivantes :

1. Calcul à partir du vecteur predicteur avant :

$$a_{M,m} = \begin{bmatrix} a_{M-1} \\ 0 \end{bmatrix} + k_M \begin{bmatrix} 0 \\ b_{M-1} \end{bmatrix} \tag{2.28}$$

$$a_{M,m} = a_{M-1,m} + k_l\, b_{M-1,M-l} \quad m = 0,1...M \tag{2.29}$$

Où $a_{M,m}$ est le $m^{ème}$ élément du vecteur predicteur avant dôordre *M*.

$b_{M-1,M-l}$ *est* le $m^{ème}$ élément du vecteur predicteur arrière dôordre *M-1*.

2. calcul à partir du vecteur predicteur arrière :

$$b_{M,m} = \begin{bmatrix} 0 \\ b_{M-1} \end{bmatrix} + k_M \begin{bmatrix} a_{M-1} \\ 0 \end{bmatrix} \tag{2.30}$$

$$b_{M,M-l} = b_{M-1,M-l} + k_l\, a_{M-1,M-l} \tag{2.31}$$

Où $b_{M,M-l}$ *est* le $m^{ème}$ élément du vecteur prédicteur arrière dôordre *M*.

Et les autres éléments sont calculés comme ci-dessus.

Pour établir la condition que la constante k_l doit satisfaire à fin de valider les équations de Levinson Durbin, procédons à ce qui suit :

1. Multiplions la formule (2.28) par R_{M+1}, le membre coté gauche de cette formule devient :

$$R_{M+1}\, a_M = \begin{bmatrix} E \\ 0_{M\times 1} \end{bmatrix} \tag{2.32}$$

2. La partie droite du deuxième membre est donnée par :

$$R_{M+1} \begin{bmatrix} a_{M-1} \\ 0 \end{bmatrix} = \begin{bmatrix} R_M & r_b \\ r_b^T & r(0) \end{bmatrix} \begin{bmatrix} a_{M-1} \\ 0 \end{bmatrix}$$

$$R_{M+1} \begin{bmatrix} a_{M-1} \\ 0 \end{bmatrix} = \begin{bmatrix} R_M\, a_{M-1} \\ r_b^T\, a_{M-1} \end{bmatrix} \tag{2.33}$$

D'autre part on a : $R_M \; a_{M-1} = \begin{bmatrix} E_{M-1} \\ 0_{(M-1)\times 1} \end{bmatrix}$ (2.34)

Notons : $\Delta_{M-1} = r_b^T \, a_{M-1} = \sum_{m=0}^{M-1} r(m-M) a_{M-1,m}$

L'équation (2.32) devient : $R_{M+1} \begin{bmatrix} a_{M-1} \\ 0 \end{bmatrix} = \begin{bmatrix} E_{M-1} \\ 0_{(M-1)\times 1} \\ \Delta_{M-1} \end{bmatrix}$ (2.35)

3. Le second terme du deuxième membre est donnée par :

$$R_{M+1} \begin{bmatrix} 0 \\ b_{M-1} \end{bmatrix} = \begin{bmatrix} r(0) & r^T \\ r & R_M \end{bmatrix} \begin{bmatrix} 0 \\ b_{M-1} \end{bmatrix}$$

C'est-à-dire : $R_{M+1} \begin{bmatrix} 0 \\ b_{M-1} \end{bmatrix} = \begin{bmatrix} r^T b_{M-1} \\ R_M \; b_{M-1} \end{bmatrix}$ (2.36)

Où on aura : $r^T b_{M-1} = \Delta^*_{M-1}$ et $R_M \; b_{M-1} = \begin{bmatrix} 0_{(M-1)\times 1} \\ E_{M-1} \end{bmatrix}$

Donc : $R_{M+1} \begin{bmatrix} 0 \\ b_{M-1} \end{bmatrix} = \begin{bmatrix} \Delta^*_{M-1} \\ 0_{(M-1)\times 1} \\ E_{M-1} \end{bmatrix}$ (2.37)

4. De ces dernières étapes 1, 2, 3 on obtient :

$$\begin{bmatrix} E_M \\ 0 \end{bmatrix} = \begin{bmatrix} \Delta_{M-1} \\ 0_{(M-1)\times 1} \\ E_{M-1} \end{bmatrix} + k_l \begin{bmatrix} \Delta^*_{M-1} \\ 0_{M-1} \\ E_{M-1} \end{bmatrix} \quad (2.38)$$

On peut déduire à partir de cette équation que :

1. $E_M = E_{M-1} + k_l \; \Delta^*_{M-1}$
2. $0 = \Delta_{M-1} + k_l \; \Delta_{M-1} \Rightarrow k_l = -\dfrac{\Delta_{M-1}}{E_{M-1}}$
3. $E_M = E_{M-1} \left(1 - k_l^2\right)$

2.3.5 Interprétation des paramètres k_l et Δ_{M-1}

Le paramètre k_l qui résulte de lôapplication de lôalgorithme de Levinson-Durbin est appelé coefficient de réflexion dont : $|k_l|^2 \leq 1$ puisque $0 \leq E_M \leq E_{M-1}$.

Le paramètre Δ_{M-1} , on peut lôinterpréter comme lôinter-corrélation entre lôerreur de prédiction arrière retardé $\Delta_{M-1} = E\left[e_{b,M-1}(n-1)\, e^{*}_{a,M-1}(n)\right]$

2.4. Filtrage adaptatif

Le principal but des filtres adaptatifs est de déterminer un ensemble de coefficients d'un système qui évolue dans le temps.

Un filtrage est rendu adaptatif si ses paramètres, et ses coefficients sont modifiés selon un critère donné, dès quôune nouvelle valeur du signal devient disponible. Ces modifications doivent suivre lôévolution du système.

Dans leur environnement aussi rapidement que possible, l'adaptation dans le filtre numérique est généralement associée à un fonctionnement en temps réel (dans le cas ou les coefficients du filtre seraient variables dans le temps, pour simuler ou modéliser un système dont les caractéristiques évoluent dans le temps).

Le filtrage adaptatif conduit a la mise en ouvre de filtre à coefficients variables ou ces derniers sont modifiés selon un critère d'optimisation donné des qu'une nouvelle valeur du signal devient disponible.il est réalisé, ainsi, suivant un algorithme d'adaptation qui est déterminé en fonction de lôapplication [4].

2.4.1 Structure du filtre programmable

La structure du filtre quôon doit la programmer, peut être de type RII (réponse impulsionnelle infinie) ou RIF(réponse impulsionnelle finie). Cependant, les effets de la limitation de la précision des calculs et de complexité arithmétique variant avec la structure comme pour les filtres à coefficients fixes. En outre, la structure du filtre influe sur la complexité des algorithmes. La structure RIF directe ou transversale est pratiquement simple à étudier et à réaliser [5].

2.4.2 Rôle de filtre adaptatif

Le rôle primordial d'un filtre adaptatif est d'ajuster le paramètre w pour un objectif bien défini (minimisation de l'EQM : erreur quadratique moyenne)[4]. Le principe d'un filtre adaptatif bouclé par un algorithme d'adaptation est représenté sur la figure 2.3.

2.4.3 Principe de base d'un filtre adaptatif

Le principe de base du filtre adaptatif est représente par la figure suivante :

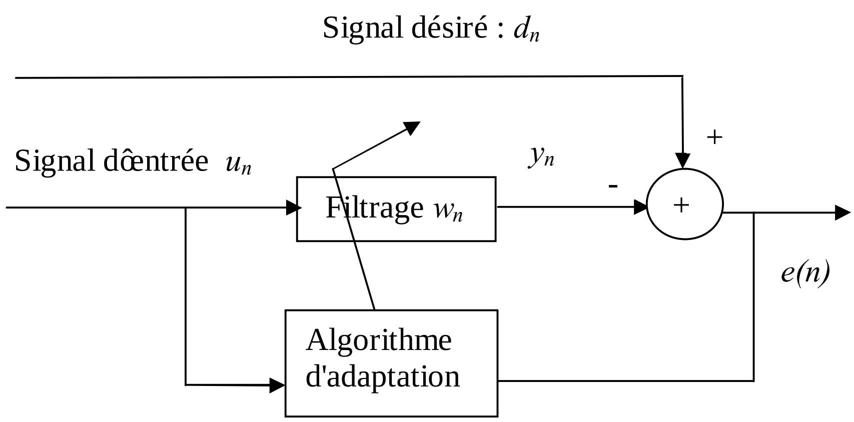


Figure 2.2 : schéma de principe de base d'un filtre adaptatif appliqué à l'annulation d'écho acoustique

Le fonctionnement du filtre adaptatif se décrit de la façon suivante. Premièrement, le signal dœntrée u_n est convolué avec le filtre w_n. Le résultat de cette convolution donne y_n.Le signal à la sortie du filtre y_k est comparé au signal désiré d_n. Le signal dœrreur $e(n)$, qui est la différence entre le signal à la sortie du filtre y_n et le signal désiré d_n , permet de faire la mise à jour des coefficients du filtre adaptatif w_n . A chaque itération, les coefficients du filtre varient en fonction du signal dœrreur e_n et ce pour faire diminuer la différence entre la sortie du filtre y_n et le signal désiré d_n. Le signal dœrreur diminue jusquà atteindre dans certains cas une valeur nulle. Lorsque la valeur est atteinte, les coefficients du filtre adaptatif cessent de sâdapter.

2.4.4 Présentation des algorithmes de filtrage adaptatif

Nous allons présenter dans cette partie trois grandes familles des algorithmes de filtrage adaptatif, soit la famille des moindres carrées récursifs RLS (Recursive Least Square), la famille du gradient stochastique LMS (Least Mean Square), et lôautre des projections affine. Pour la famille du RLS, la formulation de base du RLS et sa version rapide FRLS seront décrites. Pour la famille LMS, notre intérêt porte sur le LMS, et le LMS normalisé. Pour la famille des projections affines, on sôintéresse à étudier plusieurs algorithmes tels que lôAPA original, et ses versions rapide FAPA, GS-FAPA.

2.4.4.1 La famille du RLS

Lôalgorithme LMS a pour but de minimiser la moyenne stochastique des carrées des erreurs. Cette approche diffère de lôalgorithme des moindres carrées récursif RLS (Recursive Least Square) qui cherche à minimiser la somme des carrées des erreurs, par la méthode des moindres carrées.

2.4.4.1.1 Algorithme RLS

Dans la méthode des moindres carrés exacte, on vise à minimiser par rapport au vecteur des paramètres w_n à chaque instant n, un critère défini sur les erreurs commises depuis lôinstant initial. Ce critère est donné par [12] :

$$J = \sum_{i=1}^{\infty} \lambda^{n-i} \left(d(i) - w^T u_i\right)^2 \tag{2.39}$$

Où λ $(0 < \lambda \leq 1)$ est un facteur dôoubli exponentiel qui permet à lôalgorithme dôoublier le passé trop lointain et de poursuivre les non stationnarités intervenantes dans les signaux.

On suppose que les signaux u_n et $d(n)$ sont nuls avant lôinstant initial $n = 0$; côest à dire on se place dans le cas de la fenêtre antérieure.

La solution qui exprime la nullité du gradient de la fonctionnelle J :

$$\nabla J = 0 \Leftrightarrow \quad \nabla J = -2\left[\sum_{i=1}^{n} \lambda^{n-i}\left(u_i d(i)\right) - \sum_{i=1}^{n} \lambda^{n-i} u_i u_i^T w_n\right] \tag{2.40}$$

est donnée par :

$$R_{M,n}\, w_n = r_{M,n} \tag{2.41.a}$$

$$w_n = R^{-1}_{M,n}\, r_{M,n} \tag{2.41.b}$$

Où $R_{M,n}$ représente la matrice dóauto-corrélation à court terme à lóitération n qui est donnée par la relation suivante *:*

$$\begin{aligned} R_{M,n} &= \sum_{i=1}^{n} \lambda^{n-i} u_i u_i^T \\ &= \lambda R_{M,n-1} + u_n u_n^T \end{aligned} \tag{2.42}$$

Le vecteur $r_{M,n}$ désigne le vecteur dóinter-corrélation à court terme à lóinstant n donné par la relation :

$$\begin{aligned} r_{M,n} &= \sum \lambda^{n-i} u_i d(i) \\ &= \lambda\, r_{n-1} + u_n d(n) \end{aligned} \tag{2.43}$$

Par substitution des équations (2.42) et (2.43) dans (2.41.a) on obtient :

$$\lambda R_{L,n-1} w_{n-1} = \lambda\, r_{M,n-1}$$

On obtient la solution équivalente à (2.41.b) mais sous une forme récursive :

$$w_n = w_{n-1} - K_{M,n} e(n) \tag{2.44}$$

avec

$$K_{M,n} = \begin{bmatrix} k_{M,n}^{1} \\ k_{M,n}^{2} \\ . \\ . \\ . \\ k_{M,n}^{M} \end{bmatrix} = -R_n^{-1} u_n \tag{2.45}$$

$$e(n) = d(n) - w_n^T u_n$$

Où le vecteur $K_{M,n}$ est appelé gain de Kalman.

La formule (2.44) nécessite lôinversion dôune matrice carré dôordre M dont le coût de calcul est de lôordre M^3 opérations arithmétiques par itération.
Lôalgorithme des moindres carrés récursifs (RLS : Récursive Least Square) résout ce problème avec un nombre dôopérations arithmétiques proportionnel à M^2.

En appliquant à la matrice $R_{M,n}$ le lemme dôinversion matricielle suivant [16] :

$$\left(A + BV^T\right)^{-1} = A^{-1} - \frac{A^{-1}BV^T A^{-1}}{1 + V^T A^{-1} B}$$

Avec

$$A = \lambda R_{M,n-1} \qquad ; \qquad B = V = u_n$$

il vient

$$R_{M,n}^{-1} = \lambda^{-1}\left[R_{M,n-1}^{-1} - \frac{R_{M,n-1}^{-1} u_n u_n^T R_{M,n-1}^{-1}}{\lambda + u_n^T R_{M,n-1}^{-1} u_n} \right] \tag{2.46}$$

Cet algorithme, appelé « algorithme des moindres carrés récursifs (RLS) », permet, en propageant une matrice carré *M x M*, de trouver itérativement la solution qui minimise le critère (2.41.a). Cette solution nécessite un coût en opération arithmétique proportionnel à M^2. Cette dernière décennie, on sôest aperçu que la solution des moindres carrés pouvait être obtenue avec un coût opératoire proportionnel à M ; ceux sont les algorithmes des moindres carrés rapides.

On se limite aux algorithmes des moindres carrés transversaux rapides non normalisés à oubli exponentiel obtenus dans le cas dôun pré-fenêtrage des signaux dôentrée (les signaux sont supposés nuls avant lôinstant initial). On utilisera souvent la dénomination algorithme des moindres carrés rapides (MCR) pour désigner ce type dôalgorithme. Dans cette catégorie, on génère la solution des moindres carrés, à chaque instant n, pour un ordre donné M en propageant trois vecteurs dôordre M (deux prédicteurs aller/retour et le gain de Kalman) au lieu dôune matrice. Les algorithmes des moindres carrés rapides les plus rapides connus appartiennent à cette catégorie.

2.4.4.1.2 Algorithmes des moindres carrés transversaux rapides

Dans ce paragraphe nous décrivons brièvement lôessentiel des équations qui permettent dôobtenir les algorithmes des moindres carrés transversaux rapides à oubli exponentiel pour le cas de la fenêtre antérieure, faisant intervenir des erreurs a posteriori [17].

Le but de cette méthode des moindres carrés est de trouver le vecteur w_n qui minimise le critère (2.40). Une solution récursive à ce problème est donnée par lôalgorithme (2.44). Cet algorithme utilise une erreur de filtrage a priori et un vecteur gain dôadaptation dôordre M, quôon a appelé gain de Kalman. Si on substitue (2.42) et (2.43) dans (2.41.a), on obtient une autre forme récursive du vecteur w_n, strictement équivalente à (2.13), faisant intervenir une erreur de filtrage a posteriori et un gain dôadaptation appelé gain de Kalman dual :

$$w_{n+1} = w_n - \tilde{K}_{L,n}\varepsilon(n) \tag{2.47}$$

avec

$$\varepsilon(n) = d(n) - w_{n+1}^T u_n \tag{2.48}$$

$$\tilde{K}_{M,n} = -\lambda^{-1} R_{M,n-1}^{-1} u_n \tag{2.49}$$

Où $\varepsilon(n)$ donnée par lôexpression (2.48) désigne lôerreur de filtrage à posteriori (calculée après la mise à jour du filtre) et $\tilde{K}_{M,n}$ donnée par lôexpression (2.49) désigne le gain de Kalman dual.

Lôalgorithme (2.47) ne peut être appliqué tel quel car lôadaptation et le filtrage ne peuvent être réalisés en même temps. Comme nous allons le voir par la suite, lôun des avantages des algorithmes MCR est que les erreurs a posteriori se calculent à partir des erreurs a priori avant lôopération dôadaptation.

Pour cela, on exploite certaines propriétés dôinvariance par décalage du vecteur signal dôentrée étendu à lôordre *M+1* :

$$u_{M+1,n} = \begin{bmatrix} u(n) \\ u_{n-1} \end{bmatrix} \qquad (2.50.a)$$

$$u_{M+1,n} = \begin{bmatrix} u_n \\ u(n) \end{bmatrix} \qquad (2.50.b)$$

Ces deux formes du vecteur signal permettent de définir deux matrices dôautocorrélation dôordre (*M+1*) partitionnées

La première qui correspond au vecteur (2.50.a) sôécrit :

$$R_{M+1,t} = \sum_{i=1}^{t} \lambda^{t-i} \begin{bmatrix} u(i) \\ u_{n-1} \end{bmatrix} \begin{bmatrix} u(i) & u_{i-1}^T \end{bmatrix} = \begin{bmatrix} r_a(0) & r_{a,M}^T \\ r_{a,M} & R_{M,i-1} \end{bmatrix} \qquad (2.51.a)$$

Avec :

$$r_a(0) = \sum_{i=1}^{t} \lambda^{t-i} u^2(i)$$

$$r_{a,M} = \sum_{i=1}^{n} \lambda^{t-i} u(i) u_i = \lambda\, r_{a,t-1}^a + u(i) u_{t-1}$$

La deuxième, qui correspond au vecteur (2.19.b) s’écrit :

$$R_{M,n} = \sum_{i=1}^{n} \lambda^{t-i} \begin{bmatrix} u_i \\ u(i-M) \end{bmatrix} \begin{bmatrix} u_i^T & u(i-M) \end{bmatrix} = \begin{bmatrix} R_{M,i-1} & r_b \\ r_b^T & r_b(0) \end{bmatrix} \qquad (2.51.b)$$

Avec

$$r_b(0) = \sum_{i=1}^{n} \lambda^{t-i} u^2(i-M)$$

$$r_{b,n} = \sum_{i=1}^{n} \lambda^{n-i} u^2(i-M) u_i = \lambda\, r_{b,n-1} + u(n-M) u_n$$

Où l’indice b désigne une variable retour.

Le but des algorithmes MCR est de propager un vecteur gain. Le calcul du gain de Kalman (2.45) ou du gain de Kalman dual (2.49) fait intervenir l’inverse de la matrice d’autocorrélation à court terme.

Pour réaliser cette inversion, on utilise le lemme d’inversion d’une matrice partitionnée [17]. Si M désigne une matrice partitionnée :

$$M = \begin{bmatrix} M_1 & M_2 \\ M_3 & M_4 \end{bmatrix}$$

L’inverse de cette matrice M s’écrit de manière générale (on suppose la compatibilité des dimensions et l’existence des inverses de certaines matrices de M).

$$M^{-1} = \begin{bmatrix} 0 & 0 \\ 0 & M_4^{-1} \end{bmatrix} + \begin{bmatrix} I \\ -M_4^{-1} M_3 \end{bmatrix} \left(M_1 - M_2 M_4^{-1} M_3 \right)^{-1} \begin{bmatrix} I & -M_2 M_4^{-1} \end{bmatrix} \qquad (2.52.a)$$

Où

$$M^{-1} = \begin{bmatrix} M_1^{-1} & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} -M_1^{-1} M_2 \\ I \end{bmatrix} \left(M_4 - M_3 M_1^{-1} M_2 \right)^{-1} \begin{bmatrix} -M_3 . M_1^{-1} & I \end{bmatrix} \qquad (2.52.b)$$

En utilisant la forme (2.52.a) pour inverser la matrice (2.51.a) et la forme (2.52.b) pour inverser la matrice (2.51.b), on aura les expressions suivantes :

$$R_{M,n}^{-1} = \begin{bmatrix} 0 & 0 \\ 0 & R_{M,n-1}^{-1} \end{bmatrix} + \begin{bmatrix} 1 \\ -R_{M,n-1}^{-1} P_{L,t}^{a} \end{bmatrix} \left(r_a(0) - r_{a,n}^{T} R_{M,n-1}^{-1} r_{a,n} \right)^{-1} \left[1 - r_{a,n}^{T} R_{M,n-1}^{-1} \right] \qquad (2.53.a)$$

$$R_{M+1,n}^{-1} = \begin{bmatrix} R_{M,n}^{-1} & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} -R_{M,n}^{-1} r_{b,n} \\ 1 \end{bmatrix} \left(r_b(0) - r_{b,n}^{T} R_{M,n}^{-1} r_{b,n} \right)^{-1} \left[-r_{b,n}^{T} R_{M,n}^{-1} \quad 1 \right] \qquad (2.53.b)$$

La forme (2.53.a) fait apparaître un prédicteur aller optimal au sens des moindres carrés (le terme aller désigne, la modélisation de lôéchantillon *u(n)* par une combinaison linéaire de son passé) :

$$a_{M,n} = R_{M,n-1}^{-1} r_{a,n} \qquad (2.54)$$

Ce vecteur peut être obtenu en minimisant le critère des moindres carrés suivants :

$$J_a = \sum_{i=1}^{n} \lambda^{n-i} \left(u(i) - a_{M,n}^{T} u_{M,i-1} \right)^2 \qquad (2.55)$$

Les versions récursives du prédicteur aller (2.54) faisant intervenir soit le gain de Kalman (2.45), soit le gain de Kalman dual (2.49), sôobtiennent de la même façon que les versions récursives (2.44) et (2.47) du filtre transverse w_n :

$$a_{M,n} = a_{M,n-1} - K_{M,n-1} \overline{e}_a(n) \qquad (2.56.a)$$

$$a_{M,n} = a_{M,n-1} - \tilde{K}_{M,n-1} e_a(n) \qquad (2.56.b)$$

Où $e_a(n)$ et $\varepsilon_a(n)$ désignent respectivement lôerreur de prédiction aller à priori et lôerreur de prédiction aller à posteriori, qui sont données par :

$$e_a(n) = u(n) - a_{M,n-1}^{T} u_{n-1} \qquad (2.57.a)$$

$$\varepsilon_a(n) = u(n) - a_{M,n}^{T} u_{n-1} \qquad (2.57.b)$$

De la même façon, la forme (2.53.b) fait apparaître un prédicteur retour optimal au sens des moindres carrés (le terme retour désigne la modélisation de lôéchantillon $u(n-L)$ par une combinaison linéaire des échantillons futurs) :

$$b_{M,n} = R_{M,n}^{-1} r_{b,n} \tag{2.58}$$

Ce vecteur peut être obtenu en minimisant le critère des moindres carrés suivant :

$$\mathrm{J}_b = \sum_{\mathrm{i}=1}^{n} ə^{n-\mathrm{i}} \left(u(i-L) - \mathrm{b}_{M,n}\, u_i\right)^2 \tag{2.59}$$

Les versions récursives de $b_{M,n}$, sôbtiennent en substituant dans la solution (2.58) des versions récursives de $R_{M,n}$ et de $r_{b,n}$. Elles sœcrivent :

$$b_{M,n} = b_{M,n-1} - K_{M,n} e_b(n) \tag{2.60.a}$$

$$b_{M,n} = b_{M,n-1} - \widetilde{K}_{M,n} \varepsilon_b(n) \tag{2.60.b}$$

Où $e_b(n)$ et $\varepsilon_b(n)$ désignent respectivement lœrreur de prédiction retour a priori et lœrreur de prédiction retour a posteriori, qui sont données par :

$$e_b(n) = u(n-M) - b_{M,n-1}^{T} u_n \tag{2.61.a}$$

$$\varepsilon_b(n) = u(i-M) - b_{M,n}^{T} u_n \tag{2.61.b}$$

Les termes entre parenthèses dans les expressions (2.53.a) et (2.53.b) représentent respectivement les variances des erreurs de prédiction aller et retour (minima des critères (2.55) et (2.59)).

Leurs versions récursives sœcrivent [17] :

$$\alpha_{M,n} = \lambda.\alpha_{M,n-1} + e_a(n)\varepsilon_a(n) \tag{2.62}$$

$$\beta_{M,n} = \lambda.\beta_{M,n-1} + e_b(n)\varepsilon_b(n) \tag{2.63}$$

Où $\alpha_{M,n}$ désigne la variance des erreurs de prédiction aller et $\beta_{M,n}$ désigne la variance des erreurs de prédiction retour.

Les équations qui vont permettre de propager les gains de Kalman dœrdre ($M+1$) sôbtiennent en multipliant à droite les expressions (2.53.a) et (2.53.b) par le vecteur $\left[- X_{M+1,n}\right]$ convenablement partitionné :

$$K_{M+1,n} = \begin{bmatrix} 0 \\ K_{M,n-1} \end{bmatrix} - \frac{\varepsilon_a(n)}{\alpha_{M,n}} \begin{bmatrix} 1 \\ -a_{M,n} \end{bmatrix} \tag{2.64.a}$$

$$K_{M+1,n} = \begin{bmatrix} K_{M,n} \\ 0 \end{bmatrix} - \frac{\varepsilon_b(n)}{\beta_{M,n}} \begin{bmatrix} -b_{M,n} \\ 1 \end{bmatrix} \tag{2.64.b}$$

De la même façon en multipliant à droite les expressions (2.53.a) et (2.53.b) prises aux instants $n\text{ -}1$ par $\left[-\lambda^{-1} u_{M+1,n}\right]$ convenablement partitionnés, on obtient les deux expressions du gain de Kalman dual dôordre $M+1$:

$$\tilde{K}_{M+1,n} = \begin{bmatrix} 0 \\ \tilde{K}_{M,n-1} \end{bmatrix} - \frac{e_a(n)}{\lambda\alpha_{M,n-1}} \begin{bmatrix} 1 \\ -a_{M,n-1} \end{bmatrix} \tag{2.65.a}$$

$$\tilde{K}_{M+1,n} = \begin{bmatrix} \tilde{K}_{M,n} \\ 0 \end{bmatrix} - \frac{e_b(n)}{\lambda\beta_{M,n-1}} \begin{bmatrix} -b_{M,n-1} \\ 1 \end{bmatrix} \tag{2.65.b}$$

Il faut noter que les expressions (2.64) et (2.65) font apparaître des erreurs de prédiction qui peuvent être calculées, sans les relations de convolutions (2.57) et (2.61), si on dispose de la première ou de la ($M+1$)ième composante du gain de Kalman ou du gain de Kalman dual :

$$\varepsilon_a(n) = -\alpha_{M,n} K^{1}_{M+1,n} \tag{2.66.a}$$

$$e_a(n) = -\lambda.\alpha_{M,n-1} \tilde{K}^{1}_{M+1,n} \tag{2.66.b}$$

$$\varepsilon_b(n) = -\beta_{M,n} K^{M+1}_{M+1,n} \tag{2.67.a}$$

$$e_b(n) = -\lambda.\beta_{M,n-1} \tilde{K}^{M+1}_{M+1,n} \tag{2.67.b}$$

Par ailleurs, parmi les relations qui permettent de réduire la complexité dans les algorithmes, on trouve les relations qui lient les erreurs a priori aux erreurs a posteriori. Pour voir cela, il suffit de prendre par exemple lôerreur de filtrage a posteriori (2.48) et remplacer le filtre w_n par son équation dôadaptation (2.44) il vient :

$$\varepsilon(n) = e(n)\left(1 + K_{M,n}^{T} u_n\right) \qquad (2.68)$$

Le terme entre parenthèse dans cette expression définit ce que lôon appelle la variable de vraisemblance

$$\gamma_{M,n} = 1 + K_{M,n}^{T} u_n = 1 - u_n^{T} R_{M,n}^{-1} u_{nt} \qquad (2.69)$$

En théorie, la matrice $R_{M,n}$ et son inverse sont définis positifs. La variable $\gamma_{M,n}$ sera donc comprise entre 0 et 1 pour un fonctionnement normal de lôalgorithme.

De la même façon, on obtient les relations entre les erreurs de prédiction a priori et a posteriori

$$\varepsilon_a(n) = \gamma_{M,n-1} e_a(n) \qquad (2.70)$$

$$\varepsilon_b(n) = \gamma_{M,n}\, e_b(n) \qquad (2.71)$$

Dôautre part, les équations de mise à jour des vecteurs w_n , $a_{M,n}$ et $b_{M,n}$ utilisant soit les erreurs a priori, soit les erreurs a posteriori, sont strictement équivalentes. Ceci conduit, en égalant deux dôentre elles, à la relation qui lie le gain de Kalman et le gain de Kalman dual :

$$K_{M,n} = \gamma_{M,N} \tilde{K}_{M,n} \qquad (2.72)$$

En remplaçant cette relation (2.72) dans (2.68) on obtient la relation suivante :

$$\gamma_{M,n} = \frac{1}{1 - \tilde{K}_{M,n}^{T} u_n} \qquad (2.73)$$

Enfin, au lieu dôutiliser les définitions (2.69) ou (2.73) de la variable de vraisemblance, on utilise des versions plus rapides pour réduire la complexité. Ceci sôobtient en appliquant à la définition (2.69) les formes (2.64).

En multipliant (2.64.a) à gauche par le vecteur $\left[u(n)\ u_{n-1}^T\right]$, on trouve lœexpression de la variable de vraisemblance dœordre $M+1$:

$$\gamma_{M+1,n} = \frac{\lambda\alpha_{M,n-1}}{\alpha_{M,n}}\gamma_{M,n-1} \tag{2.74}$$

De la même manière, en multipliant (2.64.b) à gauche par le vecteur $\left[u_n^T\ \ u(n-M)\right]$ il vient:

$$\gamma_{M+1,n} = \frac{\lambda\beta_{M,n-1}}{\beta_{M,n}}\gamma_{M,n} \tag{2.75}$$

et si on pose

$$\theta_{M,n} = \frac{\lambda\beta_{M,n-1}}{\beta_{M,n}} \tag{2.76.a}$$

En manipulant les expressions (2.63) et (2.67), on aura dœautres formes dœécritures :

$$\begin{aligned}\theta_{M,n} &= 1+e_b(n)K_{M+1,t}^{M+1} \\ &= 1+e_b(n)\gamma_{M+1,n}\widetilde{K}_{M+1,t}^{M+1}\end{aligned} \tag{2.76.b}$$

On calcule alors la variable de vraisemblance dœordre M par la relation suivante :

$$\gamma_{M,n} = \frac{\gamma_{M+1,n}}{\theta_{M,n}} \tag{2.77}$$

On montre aussi, en utilisant lœexpression (2.46), la définition de la variable de vraisemblance (2.69), la propriété det[I+XY] = det [I+YX] (det: déterminant dœune matrice) et les liens entre les déterminants des matrices $R_{M+1,n}$ et $R_{M,n}$ que la variable de vraisemblance peut se mettre sous la forme suivante [17] :

$$\begin{aligned}\gamma_{M,n} &= \lambda^M \frac{\det\left[R_{M,n-1}\right]}{\det\left[R_{M,n}\right]} \\ &= \lambda^M \frac{\beta_{M,n}}{\alpha_{M,n}}\end{aligned} \tag{2.78}$$

2.4.4.1.3 Algorithme FTF 7M

A partir du paragraphe précédent (2.3.2.2), un grand nombre d'algorithmes peuvent être obtenus. Tous ces algorithmes sont théoriquement équivalents et calculent la solution des moindres carrés avec un nombre de multiplications proportionnelles à l'ordre M. Les algorithmes des MCR les plus rapides se caractérisent par le gain de Kalman dual et font intervenir les erreurs à posteriori calculées de manière simple à partir des erreurs a priori. Leur complexité arithmétique est de l'ordre de 7M multiplications par échantillons.

Une version obtenue par [18], appelé FTF (Fast Transversal Filter), propage la variable de vraisemblance $\gamma_{M,n}$, et le gain de Kalman dual. Cet algorithme FTF 7M est résumé dans la table 2.1 (7M pour la complexité).

- Résumé de lôalgorithme FTF 7M

variables disponibles à lôinstant t $$a_{M,n-1};b_{M,n-1};\tilde{K}_{M,t-1};\gamma_{M,n-1};\alpha_{M,n-1};\beta_{M,n-1};w_n$$ nouvelles informations $u(n)$ et $u(n-M)$
modélisation de $u(n)$ et $u(n-M)$ $$e_a(n)=u(n)-a^T_{M,n-1}u_{M,n-1}$$ $$\alpha_{M,n}=\lambda\alpha_{M,n-1}+\gamma_{M,n-1}e_a^2(n)$$ $$\gamma_{M+1,n}=\frac{\lambda\alpha_{M,n-1}}{\alpha_{M,n}}\gamma_{M,n-1}$$ $$\tilde{K}_{M+1,n}=\begin{bmatrix}0\\ \tilde{K}_{M,n-1}\end{bmatrix}-\frac{e_a(n)}{\lambda\alpha_{M,n-1}}\begin{bmatrix}1\\ -a_{M,n-1}\end{bmatrix}$$ $$a_{M,n}=a_{M,n-1}-e_a(n)\gamma_{M,n-1}\tilde{K}_{M,t-1}$$ $$e_b=-\lambda\beta_{M,n-1}\tilde{K}^{M+1}_{M+1,n}$$ $$\gamma_{M,n}=\frac{\gamma_{M+1,N}}{1+\gamma_{M+1,n}\bar{r}_{M,n}\tilde{K}^{M+1}_{M+1,n}}$$ $$\begin{bmatrix}\tilde{K}_{M,n}\\ 0\end{bmatrix}=\tilde{K}_{M+1,n}-\tilde{K}^{M+1}_{M+1,n}\begin{bmatrix}-b_{M,n-1}\\ 1\end{bmatrix}$$ $$b_{M,n}=b_{M,n-1}-e_b(n)\gamma_{M,n}\tilde{K}_{M,n}$$ $$\beta_{M,n}=\lambda\beta_{M,n-1}+\gamma_{M,n}e_b^2(n)$$
filtrage de $d(n)$ $$e(n)=d(n)-w_n^Tu_n$$ $$w_{n+1}=w_n-e(n)\gamma_{M,n}\tilde{K}_{M,n}$$

Table 2.1 : Algorithme FTF 7M

- Stabilisation numérique de lôalgorithme FTF7M

Il est bien connu que lôefficacité en complexité de calcul des algorithmes des moindres carrés transversaux rapides (FTF) est payée par une dégradation importante de leurs propriétés numériques.

Les erreurs numériques se propagent dôune manière non bornée au cours du temps, pour un facteur dôoubli inférieur à 1, ce qui conduit à une solution instable à plus au moins long terme. Des efforts importants ont été fait pour expliquer lôorigine de cette instabilité numérique.

Plusieurs méthodes ont été proposées pour combattre le problème de lôinstabilité numérique [19],[20].

En ce qui concerne lôalgorithme FTF7M, nous avons utilisé la version de stabilisation de base qui a été proposé en [7].

Cette dernière qui est basées sur la rétroaction des erreurs numériques dans le calcul des variables de prédiction rétrogrades est donnée par les équations suivantes :

La variable « indicateur de divergence $\xi_{M,n}$ est choisi et donnée par la relation suivante :

$$\xi_{M,n} = e_b^{conv}(n) - e_b^{f_0}(n) \tag{2.79}$$

avec

$$e_b^{conv}(n) = u(n-M) - b_{M,n-1}^T u_n \tag{2.80}$$

$$e_b^{f_0} = -\lambda \beta_{M,n-1} \widetilde{K}_{M+1,t}^{M+1} \tag{2.81}$$

Pour assurer la stabilité de lôalgorithme, il faut satisfaire la condition suivante [7].

$$\lambda > 1 - \frac{1}{2M + 3.5} \tag{2.82}$$

- <u>Initialisation de lôalgorithme FTF 7M</u>

Dans le cas de la fenêtre antérieure, les algorithmes MCR supposent que les signaux $u(n)$ et d(n) sont nuls avant lôinstant initial $t = 0$. Par conséquent, les erreurs de prédiction retour et le prédicteur retour doivent être nuls avant lôinstant $n = M$.

pour $n = 0$

$$a_{M,n} = 0_M^{\downarrow} \ ; \ \widetilde{K}_{M,n} = 0_M^{\downarrow} \ ;$$
$$e(n) = 0 \ ; \ \alpha_{M,n} = E_0 \lambda^M \ ; \ \beta_{M,n} = E_0 \ ; \ \gamma_{M,n} = 1$$
$$w_n = w_0$$

w_0 : vecteur arbitraire

pour $n < M$

$$b_{M,n} = 0_M^{\downarrow}$$
$$e_b(n) = 0$$

La constante E_0 (strictement positive) est la seule qui doit être convenablement choisie. Pendant les premières itérations, les valeurs prises par les variables internes de lôalgorithme sont étroitement liées au choix de la constante E_0.
En pratique, il faut assurer le démarrage de lôalgorithme on pourra choisir par exemple la constante E_0 qui vérifie lôinégalité suivante [18] :

$$E_0 \geq \frac{M}{100}\sigma_u^2$$

Où σ_u^2 est lôénergie du signal u_n.

Des valeurs de E_0 assurant le bon fonctionnement initial sont données par [21].

2.4.4.2 La Famille du LMS

La première grande famille de filtres est celle du LMS. Le LMS est introduit pour la Première fois par Widrow-Hoff au début des années 60. Lôalgorithme se base sur une estimation simple et peu complexe du gradient.
Ce type de filtre est très simple, mais inefficace pour la problématique de lôannulation dôécho en raison des grandes variations dôénergie contenues dans la voix. Ces variations dôénergie, souvent brusques, tendent à faire diverger le filtre [5].

Pour palier aux problèmes reliés à la variation dôénergie, Haykin a introduit le gradient normalisé NLMS [4]. La modification apportée consiste à normaliser la correction des coefficients en fonction de lôénergie du signal. Ce calcul fait en sorte que le pas dôadaptation varie de façon inversement proportionnelle à lôénergie contenue dans le signal. Ainsi en présence dôune grande énergie, lôadaptation du filtre est ralentie. Ce ralentissement permet dôéviter les cas de divergence qui pourraient subvenir avec le filtre LMS.

Avant de mettre en î uvre les équations dôadaptation des algorithmes de gradient, on procède tout dôabord à les démontrer en passant par lôalgorithme de Newton.

2.4.4.2.1 Algorithme du gradient

Les différents algorithmes utilisés pour lôidentification du canal acoustique se distinguent par le calcul du gain (terme de correction).
Pour avoir une convergence vers la solution optimale, on utilise la formule générale :
Nouvelle Estimation =Ancienne Estimation + (terme de correction).

$$w_{n+1} = w_n + \mu P \qquad (2.83)$$

Où :
P : Vecteur direction de taille $M \times 1$, correspond au terme de correction.
μ : Pas dôadaptation.

Les différents algorithmes utilisés pour lôidentification du filtre se distinguent par le calcul du gain P [5].
La condition de choisir μ et P est :

$$J(w_{n+1}) < J(w_n) \qquad (2.84)$$

Tel que $J(w_n)$ représente l'erreur quadratique moyenne décrite par lôéquation (2.6) :

$$J = \sigma_d^2 - w_n^H r - w_n^T r^T + w_n^H R \ w_n$$

On fait le développement limité de $J(w_{n+1})$:

$$J(w_{n+1}) = J(w_n) + \mu(\nabla_w J(w_n))P + \mu^2 P^T (\nabla_w^2 (J(w_n)))P$$

$$\Rightarrow J(w_{n+1}) = J(w_n) + \mu(-r^T + w_n^T R \ )P + \mu^2 P^T R \ P \qquad (2.85)$$

Par substitution de (2.84) en (2.85) on trouve :

$$[\nabla_w J(w_n)]p < 0 \quad \Rightarrow \quad [-r^T + w_n^T R\ ]\ P < 0 \tag{2.86}$$

Il y a une infinité de vecteur P qui vérifie lôinéquation (2.86).

On prend $P = -B[\nabla_w J(w_n)]^T$ (2.87)

Tel que B est une matrice positive non nulle.

Et prenons $B = I \quad \Rightarrow \quad P = [r - R\ w_n]$ (2.88)

On remplace (2.88) dans (2.83)

On trouve : $w_{n+1} = w_n + \mu[r - R\ w_n]$ (2.89)

La dernière équation (2.89) représente lôalgorithme itératif du gradient déterministe.

Et la condition de convergence de lôalgorithme du gradient (*en notant λ_{max} la valeur propre maximale de la matrice de corrélation* R) est donnée par:

$$0 < \mu < \frac{2}{\lambda_{\max}}$$

2.4.4.2.2 Algorithme de Newton

Pour lôalgorithme de Newton [10], on fait un autre choix de la matrice B citée dans (2.87).

Prenons : $B = [\nabla_w^2 J(w_n)]^{-1}$

On aura : $w_{n+1} = w_n + \mu[\nabla_w^2 J(w_n)]^{-1}[\nabla_w J(w_n)]^*$

Or on a :

$$J = \sigma_d^2 - w_n^H r - w_n^T r^T + w_n^H R\ w_n$$

Donc : $\nabla_w^2 J(w_n) = R$ et $\nabla_w J(w_n) = r - R\,w_n$

Remplaçons les dernières équations dans la dernière formule itérative, on trouve

$$w_{n+1} = w_n + \mu \ R^{-1}\left[r - R \ w_n\right] \tag{2.90}$$

Après régularisation :

$$w_{n+1} = w_n + \mu \ \left[R \ + \delta I\right]^{-1}\left[r - R \ w_n\right] \tag{2.91}$$

La condition de convergence de cet algorithme est donnée par : $0 < \mu < 2$

2.4.4.2.3 Algorithme du gradient stochastique LMS

L'idée des algorithmes de type gradient stochastique et de remplacer la moyenne statistiques dans l'algorithme du gradient déterministe d'équation (2.89) par sa valeur instantanée.[9]

$$r \cong d(n)u_n \tag{2.92}$$

et

$$R \ = u_n u_n^T \tag{2.93}$$

Remplaçant (2.92) et (2.93) dans (2.25) On obtient la relation suivante :

$$w_{n+1} = w_n + 2\mu e_n \, u_n \tag{2.94}$$

Tel que : $e_n = d(n) - w_n^T u_n$

La condition nécessaire et suffisante de convergence de l'algorithme (LMS) est :

$$0 < \mu < \frac{2}{\lambda_{max}}$$

Une étude plus précise (en moyenne quadratique), mais qui repose également sur des hypothèses contestables conduit à la condition plus contraignante :

$$0 < \mu < \frac{1}{Trace(R \ )} = \frac{1}{M\sigma_u^2}$$

Trace (R) : désigne la somme des éléments de la diagonale de la matrice R.

σ_u^2 : désigne l'énergie du signal d'entrée u_n.

2.4.4.2.4 Algorithme du gradient stochastique normalisé NLMS

L'algorithme du gradient stochastique normalisé NLMS est une variante du LMS dont le gain d'adaptation est normalisé par l'énergie du signal d'entrée u_n. Avant d'exposer leurs équations on doit les démonter d'abord en partant de la formule récursive de Newton (2.91) qui est donnée par :

$$w_{n+1} = w_n + \mu \left[R + \delta I\right]^{-1}\left[r - R\, w_n\right] \tag{2.95}$$

Les quantités $\left[R + \delta I\right]^{-1}$ et $\left[r - R\, w_n\right]$ peuvent s'exprimer les approximations instantanées [5] suivantes : $\left(\delta I + u_n\, u_n^T\right)$ et $u_n\left[d(n) - w_n^T u_n\right]$ respectivement.

Remplaçant ces dernières quantités dans (2.95) on trouve :

$$w_{n+1} = w_n + \mu\left[u_n u_n^T + \delta I\right]^{-1} u_n \left[d(n) - w_n^T u_n\right] \tag{2.96}$$

La formule récursive (2.96), requiert une inversion matricielle à chaque itération. Une forme plus simple peut être donnée en simplifiant le terme $\left[u_n u_n^T + \delta I\right]^{-1}$ [8]:

$$\left[u_n u_n^T + \delta I\right]^{-1} = \delta^{-1} I - \frac{\delta^{-2}}{1 + \delta^{-1} u_n^T u_n} u_n u_n^T \tag{2.97}$$

En multipliant (2 .97) par u_n on obtient :

$$\begin{aligned}
\left[u_n u_n^T + \delta I\right]^{-1} u_n &= \delta^{-1} u_n - \frac{\delta^{-2}}{1 + \delta^{-1} u_n^T u_n} u_n u_n^T u_n \\
&= \delta^{-1} u_n \left[1 - \frac{u_n u_n^T}{\delta + u_n^T u_n}\right] \\
&= \frac{u_n}{\delta + u_n^T u_n}
\end{aligned} \tag{2.98}$$

Par substitution (2.98) dans (2.96) on trouve :

$$w_{n+1} = w_n + \frac{\mu}{u_n^T u_n} e_n u_n \tag{2.99}$$

Où lôerreur de filtrage est représentée par : $e_n = d(n) - w_n^T u_n$

Et ε représente le pas dôadaptation de lôalgorithme NLMS. La condition suffisante de convergence est : $0 < \mu < 2$.

2.4.4.3 Famille des algorithmes de projection affine

Il est possible dôaméliorer la convergence de lôalgorithme LMS en modifiant la direction dôadaptation des coefficients du filtre dôidentification. Cette analyse est lôorigine des algorithmes de projection affine qui sont obtenus par une projection dôordre multiple *K* [28]. Les algorithmes APA dôordre *K* sont caractérisés par le fait quôils annulent les *K* erreurs a posteriori produites par le filtre dôidentification. Ces algorithmes possèdent de ce fait de bien meilleures propriétés de convergence sur des signaux fortement corrélés que lôalgorithme LMS. Toutefois, leur complexité arithmétique initiale est trop importante. Plusieurs versions rapides de ces algorithmes ont été proposées afin de réduire la complexité initiale.

2.4.4.3.1 Algorithme de projection affine exacte

Prenons lôéquation (2.91) celle de la méthode de Newton régularisé :

$$w_{n+1} = w_n + \mu\left[R_u + \delta I\right]^{-1}\left[r - R_u w_n\right]$$

Remplaçons la moyenne statistique par sa valeur instantanée [5]. Et choisissons un entier *K* ($K \leq M$) tel que M est la taille du vecteur poids *w*.

$$R_u \cong \frac{1}{K}\left(\sum_{j=n-K+1}^{n} u_j^* u_j\right) \tag{2.100}$$

$$r \cong \frac{1}{K}\left(\sum_{j=n-K+1}^{n} d(j) u_j^* \right) \tag{2.101}$$

C'est-à-dire pour chaque itération *n*, on prend les *K* vecteurs les plus récents du signal dœntrée et les *K* vecteurs les plus récents du signal sortie

$U_n = [u_n, u_{n-1} \ldots\ldots u_{n-K+1}]^T$ de taille $K \times M$

$u_n = [\, u(n)\, u(n-1) \, . \, . \, . \, u(M) \,]$

$d_n = [d(n), d(n-1) \ldots\ldots\ldots d(n-K+1)]$ $K \times 1$

On réécrire (2.100) et (2.101) de la façon suivante :

$$R \cong \frac{1}{K} U_n^* U_n \tag{2.100.a}$$

$$r \cong \frac{1}{K} U_n^* d_n \tag{2.101.a}$$

Remplaçant ces dernières équations dans la formule récurrente de Newton, on trouve :

$$w_{n+1} = w_n + \frac{\mu}{K}\left[U_n^* U_n + \delta I\right]^{-1}\left[U_n^* d_n - U_n^* U_n w_n\right]$$

$$w_{n+1} = w_n + \mu\left[U_n^* U_n + \delta I\right]^{-1} U_n^*\left[d_n - U_n w_n\right] \tag{2.102}$$

Or on a : $\left(\delta I + U_n^* U_n\right)^{-1} U_n^* = U_n^*\left(\delta I + U_n U_n^*\right)^{-1}$ (2.103)

Par substitution de (2.103) dans (2.102) on aura :

$$w_{n+1} = w_n + \mu U_n^*\left[U_n U_n^* + \delta I\right]^{-1}\left[d_n - U_n w_n\right] \tag{2.104}$$

Cette dernière formule représente lœalgorithme de lœAPA ou APA régularisé.

- <u>Condition de stabilité de lœalgorithme APA</u>

Pour simplifie, on suppose que δ =0.

On définit le vecteur erreur a posteriori par lœexpression suivante :

$$\varepsilon_n = d_n - U_n\, w_{n+1} \tag{2.105}$$

Cette erreur se calcule une fois que la mise a jour de filtre a été effectuée.

Lœalgorithme peut être considéré comme stable si :

$\varepsilon_n^T \varepsilon_n < e_n^T e_n$

A partir lœéquation (2.60), on peut vérifier que : $\varepsilon_n = (1-\mu)e_n$

Dans ce cas lœéquation $\varepsilon_n^T \varepsilon_n < e_n^T e_n$ devient :

$(1-\mu)^2 < 1$

Et la condition de la stabilité sera donnée par :

$$0 < \mu < 2 \qquad (2.106)$$

Cette condition est identique à celle obtenue pour lœalgorithme NLMS

- <u>Résumé de lôalgorithme APA</u>

Initialisation : $w_0 = \underline{0}, U_n = [\underline{0}\ \ \underline{0}\ .\ .\ .\ \underline{0}]$
Mise a jour de la matrice U_n a partir du vecteur u_n $e_n = d_n - U_n^T w_n$. $\xi_n = \left[U_n U_n^* + \delta I\right]^{-1} e_n$ $w_{n+1} = w_n + \mu U_n^* \xi_n$

Table 2.2 : Algorithme de projection affine exacte

- <u>Interprétation géométrique</u>

Dans ce paragraphe nous allons donner une explication géometrique de convergence de lôalgorithme NLMS et APA.

La figure1 ci-dessous donne une représentation géométrique de la convergence de lôalgorithme NLMS. On désigne w le vecteur optimal du filtre adaptatif ; w_o est le vecteur initial du filtre adaptatif. Quand on fait passer le 1er vecteur dôentrée u_1, le vecteur w_0 suit la même direction que celle du vecteur u_1, c'est-à-dire w_0 est projetée dans la direction orthogonale au vecteur dôentrée u [9].

La figure 2, représente géométriquement la convergence de lôalgorithme APA dont lôordre de projection égale à 2. Ici la projection de w_0 se faite dans lôintersection des deux plans orthogonaux aux vecteurs dôentrée u_1 et $u_{2\,;}$ ça signifie que lôAPA représente une extension de lôalgorithme NLMS, avec une convergence plus rapide que celle du NLMS.

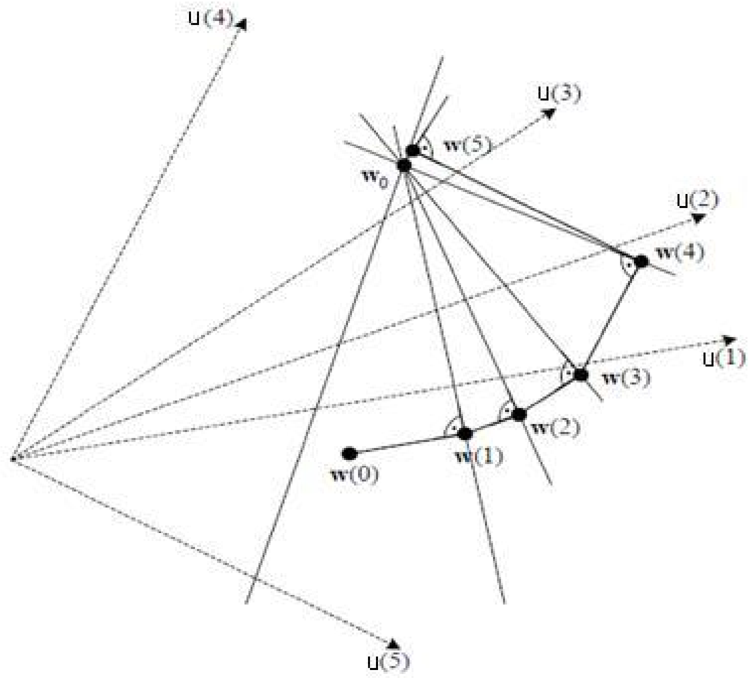


Figure 2.3: Interprétation géométrique du NLMS

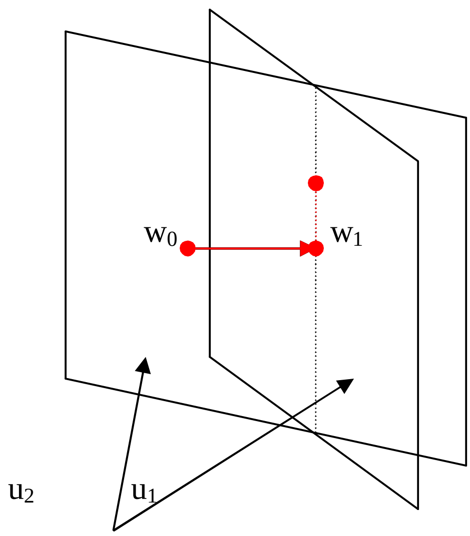


Figure 2.4 : Interprétation géométrique de lôAPA.

2.4.4.3.2 Algorithme de projection affine rapide (FAP)

La complexité de lôAPA augmente considérablement avec lôaugmentation dôordre de projection. A fin de remédier, un nouveau algorithme FAP été présenté dans [11] 1990 par Steven L.Grant. La complexité de calcul et de mémoire requise de FAP sont proches à celles de NLMS [11][12].

La réduction de complexité de lôAPA est faite en trois étapes :

- *Première étape :*

La complexité est réduite de *KM* à *M*.

Considérons les équations (2.104), et comme référence on a le développement des équations présentées dans [11] :

$$e_n = d_n - U_n^T w_n = \begin{bmatrix} d(n) - u_n^T w_{n-1} \\ \overline{d}_n - \overline{U}_n w_{n-1} \end{bmatrix} = \begin{bmatrix} e(n) \\ (1-\mu)\overline{e}_{n-1} \end{bmatrix} \tag{2.107}$$

Tel que : $\overline{d}_n = [d(n-1)\ d(n-2)\ ...\ d(n-M+1)]$

et $\overline{U}_n = [u_{n-1}\ \ u_{n-2}\ .\ .\ .\ u_{n-K+1}]^T$.

- *Deuxième étape :*

la complexité de calcul de ε_n qui définie ci-dessous est réduite de lôordre K^2 à K.

On a : $R \cong U_n^T U_n + \delta I$ (2.108)

$$\xi_n = R_n^{-1}.e_n \tag{2.109}$$

Où R_n^{-1} représente la matrice de corrélation à la *nième* itération.

R_n^{-1} peut être définie à partir R_{n-1}^{-1} :

Or on a dans [11] :

$$R_n = R_{n-1} + \alpha_n \alpha_n^T - \alpha_{n-L} \alpha_{n-L}^T \tag{2.110}$$

Tel que : $\alpha_n = [u(n), u(n-1)\ ...\ u(n-K+1)]^T$

On peut utiliser pour le calcul de ε_n la méthode des moindres carrées rapide à fenêtre glissante (*Sliding Windowed Fast Recursive least squares SW-FRLS*) afin de réduire le nombre de calcul.

Finalement, la complexité de w_n est réduite de *KM* à *M*, en remplaçant w_n par $\hat{w}_n$.

$\hat{w}_n$ utilise le dernier vecteur de la matrice U_n qui est définie par :

$$\hat{w}_n = \hat{w}_{n-1} + \mu u_{n-K+1} E_{K-1,n} \quad (2.111)$$

Tel que: $E_{K-1,n} = [\xi_{K-1}(n) + \xi_{K-2}(n-1) + ... + \xi_0(n-K+1)]$, le vecteur erreur e_n est calculé à partir $\hat{w}_n$ au lieu de w_n.

Le FAP utilise la technique de la fenêtre coulissante à fin de mettre à jour les données de la matrice de corrélation et le vecteur de corrélation dans sa forme implicite régulière. Les erreurs dues aux calculs arithmétiques de l'algorithme influent directement sur le calcul de la matrice de corrélation. Pour cela un algorithme dit SW-FRLS est introduit avec une complexité *14M* multiplications par itération. Autrement, l'algorithme est stable juste pour les signaux stationnaires, une classe de signaux qui n'incluse pas la parole. Une autre approche qui est directe et élégante pour le FAP, dont on doit démarrer avec une nouvelle fenêtre coulissante en parallèle avec l'ancienne, et quand les données seront les mêmes dans les deux processus, on remplace les paramètres de l'ancienne fenêtre par la nouvelle [17].

- <u>Résumé des équations de lalgorithme FAP</u>

Initialisation : $E_{a,n} = \delta$ et $a_0 = [1\ 0\ 0\ ...\ 0]$
La mise a jour de $E_{a,n}$ et a_n avec lalgorithme FRLS $\tilde{r}_{u,n} = \tilde{r}_{u,n-1} + u(n)\alpha_n + u(n-M)\alpha_{n-M}$. $\hat{e}_n = d_n - U^T w_n$ $e_n = \hat{e} - \mu \tilde{r}_{u,n}\ \overline{E}_{n-1}$ $E_n = \begin{bmatrix} 0 \\ \overline{E}_n \end{bmatrix} + \frac{1}{E_{a,n}} \mu \xi_n$ $\hat{w}_n = \hat{w}_{n-1} + \mu u(n-K+1) E_{N-1,n}$

Table 2.3 : algorithme de projection affine rapide (FAP)

<u>2.4.4.3.3 Algorithme de Gauss Seidel projection affine rapide (GS-FAP)</u>

La version rapide de lAPA qui utilise algorithme FRLS (Fast Recursive Least Squares) souffre d'instabilité numérique [12]. La complexité de cet algorithme est 2K+20M, où M est la longueur du filtre et K est l'ordre de projection. D'autres difficultés se présentent en besoins dune grande capacité de mémoire. Autre formes de FAP standard qui utilisent la fenêtre coulissante-RLS ont été proposées dans [13] et [14]. Ces autres algorithmes FAP conduisent à bien estimer linverse de la matrice dauto-corrélation. Si les estimations s'écartent, les erreurs se propagent à lautre itération, posant le filtre adaptatif à l'échec [15]. Une solution fréquemment proposée consiste à démarrer périodiquement un nouveau processus d'inversion.

Lalgorithme de GS-FAP présenté dans [15] à fin de remédier à ce problème et améliorer lalgorithme FAP en utilisant la méthode de Gauss-Seidel pour faire linversion matricielle. Il a été prouvé qu'il est stable et facile à mettre en î uvre dans une comparaison avec d'autres algorithmes de projection affine.

- <u>Résumé de lôalgorithme GS-FAP</u>

Lôalgorithme de projection affine rapide en utilisant la méthode Gauss-Seidel GS-FAP [15], est résumé dans le tableau suivant:

Initialisation : $E_{a,n} = \delta$ et $a_0 = [1\ 0\ 0\ ...\ 0]$.

La mise a jour de $E_{a,n}$ et a_n en utilisant FRLS

$$\tilde{r}_{u,n} = \tilde{r}_{u,n-1} + u(n)\alpha_n + u(n-M)\alpha_{n-M} .$$

$$\hat{e}_n = d_n - U^T w_n$$

$$e_n = \hat{e} - \mu \tilde{r}_{u,n} \overline{E}_{n-1}$$

Mise à jour du R_u à partir de $\tilde{r}_{u,n}$

Résoudre $R\ p_n = b$ avec la méthode de Gauss-Seidel.

$$\xi_n = e_n\ p_n$$

$$E_n = \begin{bmatrix} 0 \\ \overline{E}_n \end{bmatrix} + \frac{1}{E_{a,n}} \mu \xi_n$$

$$\hat{w}_n = \hat{w}_{n-1} + \mu u(n-K+1) E_{N-1,n}$$

Table 2.4 : Algorithme de projection affine rapide en utilisant la méthode de Gauss-Seidel (GS-FAP).

Nous résolvons $R\ p_n = b$ par la méthode de Gauss-Seidel qui sera décrite dans ce qui suit.

- Méthode de Gauss Seidel

La méthode de Gauss-Seidel [22] est une méthode itérative de résolution de système linéaire de la forme $\boldsymbol{A\,x = B}$.
Pour cela, on va construire une suite de vecteurs :
$\boldsymbol{x^{(0)}, x^{(1)}, \ldots\ x^{(k)}, x^{(k+1)}, \ldots}$ qui converge vers $\boldsymbol{x}$ **,** solution du système d'équations linéaires.

Nous notons que chacun des vecteurs successifs est identifié par un numéro placé en exposant et entre parenthèses.

✓ Description de la méthode

Un vecteur initial $\boldsymbol{x}^{(0)}$ étant donné, l'algorithme suivant permet de déterminer les éléments successifs de la suite.
On décompose la matrice A en trois matrices L, D et U. La matrice L est constituée par des termes qui se trouvent au-dessous de la diagonale principale de $A(j<i)$; la matrice D contient les termes diagonaux de $A(j = i)$; la matrice U est constituée des termes qui se trouvent au-dessus de la diagonale principale de $A(j > i)$.
Le système à résoudre peut alors s'écrire :

$$(D+L)\,x = B - U\,x$$

d'où l'on tire la formule de récurrence :

$$\boldsymbol{x^{(k+1)}} = (D + L\,)^{-1}\,(B - U\,x^{(k)})$$

Qui permet de calculer les composantes de $\boldsymbol{x}^{(k+1)}$ lorsque celles de $\boldsymbol{x}$ sont connues.

$$x_i^{(k+1)} = \frac{B_i - \sum_{j=1}^{i=1} A_{ij}\, x_j^{(k)} - \sum_{j=i+1}^{n} A_{ij}\, x_j^{(k)}}{A_{ii}}$$

On remarque que chaque composante de $\boldsymbol{x}^{(k)}$ n'est utilisée que jusqu'au calcul de la composante correspondante de $\boldsymbol{x}^{(k+1)}$. Ces deux vecteurs peuvent donc être stockés dans le même tableau.

✓ Variante

Il est aussi possible de calculer les $\boldsymbol{x}_i^{(k+1)}$ à partir du dernier. Cela revient à permuter le rôle des matrices L et U. La formule de récurrence devient :

$$x^{(k+1)} = (D + L)^{-1} (B - U x^{(k)})$$

qui permet de calculer les composantes de $\boldsymbol{x}^{(k+1)}$ lorsque celles de $\boldsymbol{x}$ sont connues.

$$x_i^{(k+1)} = \frac{B_i - \sum_{j=1}^{i=1} A_{ij} x_j^{(k)} - \sum_{j=i+1}^{n} A_{ij} x_j^{(k)}}{A_{ii}}$$

✓ Erreur

A chaque itération, le vecteur trouvé $\boldsymbol{x}^{(k+1)}$ comporte une certaine erreur

$$E^{(k+1)} = x^{k+1} - x^{(k)}$$

On pose :$P = D + L^{-1} U$. Il vient alors

$$E^{(k+1)} = P^{k+1} . x^{(0)}$$

✓ Convergence

L'algorithme converge si $\lim_{k\to\infty} \left\| E^{(k)} \right\| = 0$ ou, ce qui revient au même,. $\lim_{k\to\infty} \left\| P^{(k)} \right\| = 0$

✓ Théorème 1

Une condition nécessaire et suffisante pour que $\lim_{k\to\infty} \left\| P^{(k)} \right\| = 0$ est que les modules de toutes les valeurs propres de P soient strictement inférieures à 1[20].

✓ Théorème 2

La formule de récurrence converge, quel que soit $x^{(0)}$, si la matrice A est à diagonale dominante, c'est-à-dire si la valeur absolue de chaque terme diagonal est supérieure à la somme des valeurs absolues des termes rectangles placés sur la même ligne [20].

2.5 Conclusion

Dans ce chapitre nous avons présenté le problème dõidentification résolu par le filtrage optimal de Wiener, la prédiction linéaire, et le filtrage adaptatif en décrivant lõessentiel des équations qui formulent les algorithmes adaptatifs de type gradient stochastique, NLMS, les algorithmes de projection affine, moindres carrés transversaux rapides FTF.

Dans le prochain chapitre on sõintéresse à étudier une nouvelle version des algorithmes de projection affine : algorithmes de pseudo projection affine.

CHAPITRE 3

PROPOSITION ET ETUDE DE TROIS NOUVELLES VERSIONS ALGORITHMIQUES DU PSEUDO PROJECTION AFFINE

3.1 Introduction

Dans ce chapitre nous avons proposé trois nouvelles versions de lôalgorithme de Pseudo projection affine (PAP)[23].

Le principe de ces trois variantes est basé sur lôutilisation de trois structures de blanchiment [25], des signaux intervenant dans le fonctionnement de la version de base du Pseudo-APA [24][28].

Nous avons utilisé également deux méthodes, pour estimer les prédicteurs aller et retour qui interviennent directement à la phase de blanchiment. Ces méthodes sont celle de Levinson-Durbin et la méthode de covariance [6][26].

Dans ce qui suit, nous allons détailler les trois nouvelles versions avec leurs formules correspondantes.

3.2 Algorithme de Pseudo-APA

Dans lôalgorithme de projection affine [24], le filtre adaptatif ⍰⍰ est décrit comme suit :

$$w_{n+1} = w_n + \mu U_n^* \left[U_n U_n^* + \varepsilon I \right]^{-1} \left[d_n - U_n w_n \right] \qquad (3.1)$$

Tel que :

µ est le pas dôadaptation,

$d_n = [d(n), d(n-1), \ldots d(n-K+1)]$ est le vecteur des derniers K échantillons du signal référence.

La matrice $U_n = \left[u_n, u_{n-1} \ldots\ldots\ldots u_{n-K+1} \right]^T$ est de dimension ⍰ × ⍰

Et $u_n = [u(n)\, u(n-1) \ldots u(M)]$

Quand K est égal à un, lôalgorithme APA devient NLMS, et côest pour cette raison, on dit que lôAPA est une généralisation de NLMS.

Un autre paramètre appelé erreur a posteriori dôordre K est défini par la formule suivante:

$$\varepsilon_n = d_n - U_n^T w_{n+1} \qquad (3.2)$$

Où : $\varepsilon_n = [\varepsilon(n)\ \varepsilon(n-1)\ \ldots\ \varepsilon(n-K+1)]$

On définit aussi lôerreur a priori e_n par la formule suivante:

$$e_n = d_n - U_n^T w_n \tag{3.3}$$

Tel que : $e_n = [e(n)\ e(n-1)\ ...\ e(n-K+1)]$.

On suppose aussi que $\mu = 1$, ε_n est alors nul, on peut réécrire lôéquation (3.1) différemment côest à dire :

$$w_{n+1} = w_n + [U_n^T.U_n]^{-1}.U_n\,(d(n) - u_n^T.w_n) \tag{3.4}$$

Adoptant la stationnarité du signal dôentrée [1], on aura la relation entre le vecteur des coefficients de prédiction linéaire avant $a_n = [1\,a(1)\ a(2)\ ...\ a(K-1)]^T$ et la puissance dôerreur de prédiction E_n comme suit :

$$a_n = \frac{1}{K}.[U_n^T.U_n]^{-1}.\begin{bmatrix} E_n \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{3.5}$$

Soit donc, le vecteur dôerreur de prédiction :

$$u_n^f = U_n^T.a_n \tag{3.6}$$

A partir de lôéquation (3.6) et (3.5) on obtient :

$$E_{K-1} = u_n^f.u_n \tag{3.7}$$

Par substitution des équations (3.5), (3.6) et (3.7) dans lôéquation (3.4) on obtient :

$$w_{n+1} = w_n + \frac{u_n^f}{((u_n^f)^T.u_n)}.(d(n) - u_n^T.w_n) \tag{3.8}$$

A fin dôaméliorer les performances de cette adaptation, on réintroduit le paramètre ε [23], et finalement on peut définir lôalgorithme du pseudo projection affine comme suit :

$$w_{n+1} = w_n + \mu.\frac{u_n^f}{((u_n^f)^T.u_n)}.(d(n) - u_n^T.w_n) \tag{3.9}$$

3.3 Structures blanchissantes

Lôidée de base, est quand on utilise à lôentrée du système, en question, un signal non stationnaire, on introduit un filtre decorrélateur (blanchisseur) dans le but dôaméliorer les performances du filtre adaptatif.

On présente notamment, dans cette partie trois structures blanchissantes dont le filtre blanchisseur est calculé par deux méthodes, la première est la récursion de Levinson-Durbin décrite dans [6], la seconde est la méthode de covariance [7] qui nous donne le filtre blanchissant (vecteur de prédiction aller pour le vecteur dôentrée □□) sous la forme suivante :

$$A_K = \left[\frac{1}{\left[U^T_{n-1,K-1} . U_{n-1,K-1}\right]^{-1} . U_{n-1,K-1}\, u_n} \right] \tag{3.10}$$

Dans ce qui suit nous allons présenter les trois structures de blanchiment.

3.3.1 Structure blanchissante N° 1

La figure 3.1 représente une structure dôannulation dôécho, où on applique un filtre blanchissant □□ au signal dôentrée qui est de type parole. La sortie de ce filtre représente le vecteur dôerreur de prédiction ainsi écrite :

$$u_n^f = U_n^T . a_n \tag{3.11}$$

Le filtre de prédiction □□ est appliqué aussi à lôerreur dôadaptation *e(k).* On peut réécrire lôéquation dôadaptation de lôalgorithme Pseudo projection affine comme suit :

$$w_{n+1} = w_n + \mu \frac{u_n^f}{((u_n^f)^T . u_n)} . e^f(n) \tag{3.12}$$

Où :

$$e^f(n) = A_K^T . \begin{bmatrix} d(n) - u_n^f . w_n \\ \vdots \\ d(n-K+1) - u_{n-K+1}^f . w_{n-K+1} \end{bmatrix}$$

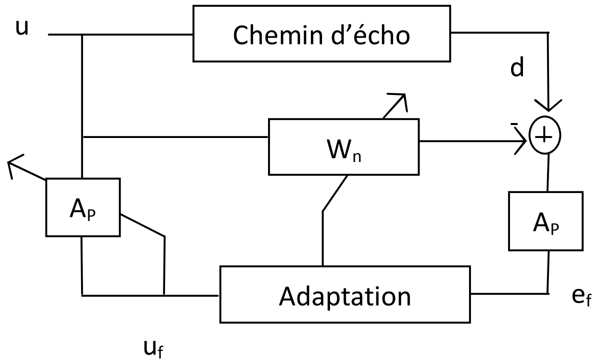


Figure 3.1 : 1ère Structure Blanchissante

Préblachiment de lôentrée ainsi lôerreur dôadaptation.

3.3.2 Structure blanchissante N° 2

Dans cette structure représentée par la figure 3.2, le filtre blanchissant est appliqué à lôentrée du système et à la réponse désirée du chemin dôécho.

Lôéquation dôadaptation devient :

$$w_{n+1} = w_n + \mu . \frac{u_{t,L}}{(u_{t,L}^T . u_{t,L})} . (d^f(n) - u_n^T . w_n) \qquad (3.13)$$

Où :

$$d_n^f = A_K^T . \begin{bmatrix} d(n) \\ : \\ d(n-K+1) \end{bmatrix} .$$

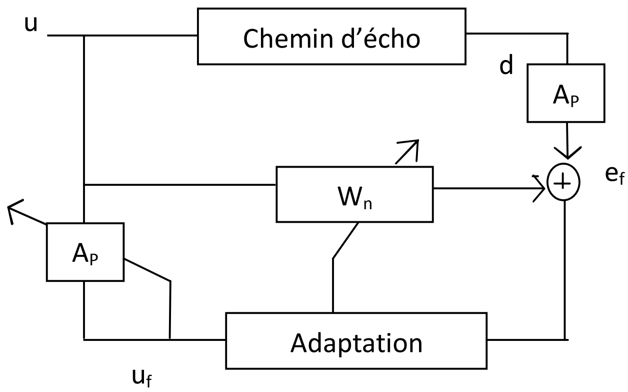


Figure 3.2 : 2ème structure blanchissante.

Pré-blanchiment de lôentrée seulement.

3.3.3 Structure blanchissante N° 3

Le filtre blanchisseur est appliqué seulement au signal dœntrée [4][2], comme le montre la figure 3.3.

Lœquation dœadaptation appropriée à cette structure est la suivante :

$$w_{n+1} = w_n + \mu \frac{u_n^f}{u_n^f\, u_n}\left(d(n) - w_n^T\, u_n^f\right) \qquad (3.14)$$

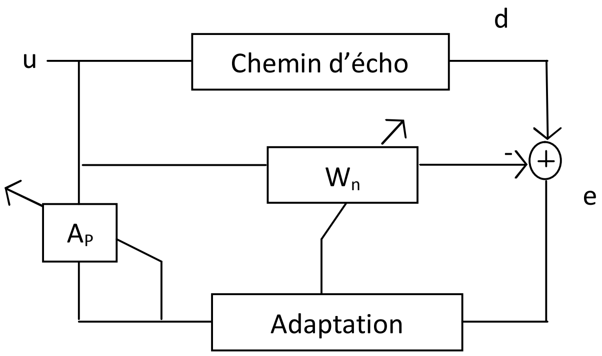


Figure 3.3 : 3ème Structure blanchissante

Préblanchiment de lœntrée et la sortie du système.

3.4 Equivalence entre les structures de blanchiment 1 et 2

Dans ce chapitre, nous allons montrer lœquivalence analytique des deux structures blanchissantes n° 1 et 2.

Nous avons réalisé cette démonstration pour corriger lœrreur du papier [20], qui stipule une distinction entre ces deux structures qui nœst pas correcte.

On peut écrire lœrreur présenté dans la figure 3.1 comme suit:

$$e^f(n) = A_K^T \begin{bmatrix} d(n) - u_n^f w_n \\ \vdots \\ d(n-K+1) - u_{n-K+1}^f w_{n-K+1} \end{bmatrix} \qquad (3.1)$$

On peut réécrire cette dernière équation comme suit:

$$e^f(n) = [1\ \ a_1\ \ a_2\ \ \ldots a_{k-1}] \begin{bmatrix} d(n) - u_n\, w_n \\ \vdots \\ d(n-K+1) - u_{n-K+1} w_{n-K+1} \end{bmatrix} \qquad (3.2)$$

On peut simplifier lœéquation (3.16) comme suit:

$$e^f(n) = d(n) - u_n\, w_n + C1_n \tag{3.3}$$

Où $C1_n$ est donné par

$$C1_n = \sum_{i=1}^{K-1} a_i d(n-i) - \sum_{i=1}^{K-1} a_i\, u_{n-i} w_{n-i} \tag{3.4}$$

On peut réécrire lœéquation (3.17) de la façon suivante

$$e^f(n) = d(n) + \sum_{i=1}^{K-1} a_i d(n-i) - C2_n \tag{3.5}$$

Où $C2_n$ est exprimé par lœéquation suivante:

$$C2_n = u_n w_n + \sum_{i=1}^{K-1} a_i u_{n-i} w_{n-i} \tag{3.6}$$

Et la nouvelle expression de lœécho acoustique $d^f(n)$ est donnée par :

$$d^f(n) = A_K^T \begin{bmatrix} d(n) \\ \vdots \\ d(_{n-K+1}) \end{bmatrix} A_K^T \begin{bmatrix} u_n \\ \vdots \\ u_{n-K+1} \end{bmatrix} u_{n-1} \tag{3.7}$$

Dœoù, on trouve : $e^f(n) = d^f(n) - u_n w_n$ (3.8)

La dernière équation (3.8) représente lœerreur finale de la deuxième structure (voir figure 2).

De lœéquation 3.15 à lœéquation 3.22, on conclut que les technique de pré-blanchiment illustrées dans les figures 1 et 2 (structures 1et 2) sont équivalentes analytiquement.

Ceci sera vérifié via les simulations qui sont présentées dans le chapitre IV (Partie B).

3.5 Résumé des nouvelles versions de lôalgorithme Pseudo-APA

Dans ce qui suit nous allons présenter le nécessaire des équations affectant sur le déroulement des trois nouvelles versions algorithmiques du Pseudo-APA.

3.5.1 1ère version de lôalgorithme Pseudo-APA correspondante à la 1ère structure

Pour la première version, le filtre de blanchiment est appliqué au signal dôentrée u_n ainsi que lôerreur dôadaptation e_n.

Initialisation : $a_0 = \begin{bmatrix} 1 \\ \underline{0} \end{bmatrix}, w_{-1} = \underline{0}, U_n = \begin{bmatrix} \underline{0} & \underline{0} & . & . & . & \underline{0} \end{bmatrix}$
Mise a jour de la matrice U_n a partir du vecteur u_n $e_n = d_n - U_n^T w_{n-1}$.   Calcul du vecteur de prédiction a_n
Structure 1 $u_n^f = U_n a_n$ $e^f(n) = a_n^T . e_n$ $w_n = w_{n-1} + \mu \dfrac{u_n^f}{u_n^f u_n} e^f(n)$

Table 3.1: La 1ère version de lôalgorithme de Pseudo-APA

3.5.2 2^ère version de lâalgorithme Pseudo-APA correspondante à la 2^ère structure

Pour la deuxième version, le filtre de blanchiment est appliqué au signal dâentrée u_n et à la réponse désirée d_n .

Initialisation : $a_0 = \begin{bmatrix} 1 \\ \underline{0} \end{bmatrix}, w_{-1} = \underline{0}, U_n = [\underline{0} \ \ \underline{0} \ . \ . \ . \ \underline{0}]$
Mise a jour de la matrice U_n a partir du vecteur u_n $e_n = d_n - U_n^T w_{n-1}$.   Calcul du vecteur de prédiction a_n
Structure 2 $u_n^f = U_n a_n$ $d_n^f = d_n . a_n$ $w_n = w_{n-1} + \mu \dfrac{u_n^f}{u_n^f u_n} \left(d^f(n) - w_{n-1}^T u_n^f \right)$

Table 3.2: 2^ème version de lâalgorithme de Pseudo-APA

3.5.3 3ère version de lôalgorithme Pseudo-APA correspondante à la 3ère structure

Pour la troisième version, le filtre de blanchissant est appliqué seulement à lôentrée du système.

Initialisation : $a_0 = \begin{bmatrix} 1 \\ \underline{0} \end{bmatrix}, w_{-1} = \underline{0}, U_n = [\underline{0}\ \ \underline{0}\ .\ .\ .\ \underline{0}]$
Mise a jour de la matrice U_n a partir du vecteur u_n $e_n = d_n - U_n^T w_{n-1}$.   Calcul du vecteur de prédiction a_n
Structure 3 $u_n^f = U_n a_n$ $w_n = w_{n-1} + \mu \frac{u_n^f}{u_n^f u_n} \left(d(n) - w_{n-1}^T u_n^f \right)$

Table 3.3: 3ème version de lôalgorithme de Pseudo-APA

3.6 Conclusion

Dans ce chapitre, nous avons proposé trois nouvelles versions de lôalgorithme pseudo projection affine. Ces trois versions sont basées sur lôutilisation de trois structures blanchissantes en combinaison avec la version exacte de lôalgorithme de pseudo-APA proposé dans [19].

Nous notons aussi que ce travail a été publié dans la conférence IEEE, ICECS 2010 en Gréce sous le titre « Three pre-whitened versions of the pseudo affine projection algorithm for acoustic echo cancellation » [23].

CHAPITRE 4

ETUDE COMPARATIVE DES PERFORMANCES DES ALGORITHMES DE FILTRAGE ADAPTATIF / APPLICATION A L'ANNULATION D'ECHO ACOUSTIQUE

4.1 Introduction

Dans ce chapitre, nous allons, tester et comparer les performances des algorithmes d'identification adaptatifs présentés dans les chapitres 2 et 3. Nous présentons aussi dans ce chapitre les résultats de simulation des nouvelles versions de l'algorithme Pseudo-APA en utilisant trois structures de blanchiment. Les performances auxquelles nous nous intéressons dans cette étude sont exprimées en termes de vitesse de convergence, de niveau de l'erreur de filtrage en sortie, et de niveau de complexité de l'algorithme.

4.2 Principe général d'annulation d'écho acoustique

Dans le modèle d'annulation d'écho acoustique présenté dans la figure (4.1), si le filtre adaptatif modélise parfaitement le canal acoustique, l'erreur du filtrage est nulle et il n'y a plus d'écho acoustique. Plus les conditions s'éloignent du modèle exposé précédemment les performances se dégradent. En réalité, il y aura toujours une erreur de filtrage appelé erreur résiduelle, à un certain niveau d'écoute, n'est pas gênante et elle permet au locuteur qui se situe dans la salle distante où le son est capté, d'avoir une perception de l'ambiance sonore de téléconférence avec une prise et restitution du son stéréophonique.

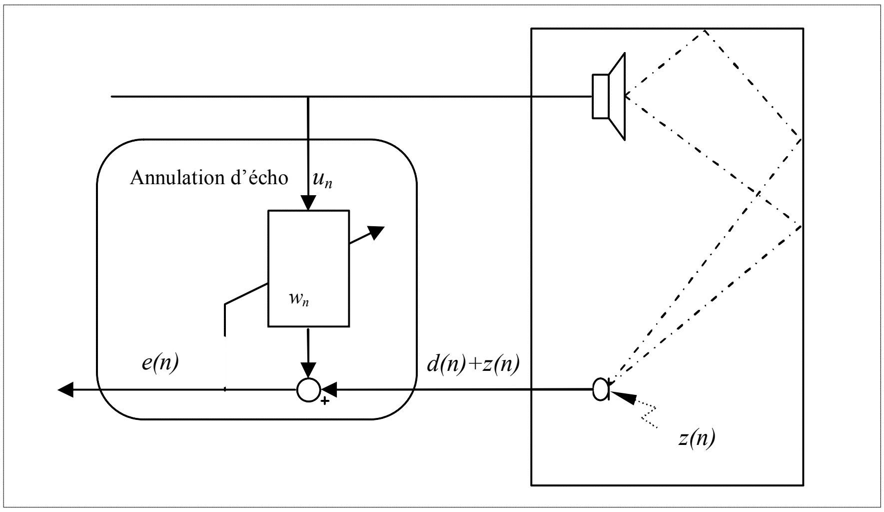


Figure 4.1: Structure générale d'annulation d'écho acoustique

Dans la structure présentée dans la figure 4.1, l'erreur de filtrage s'écrit :

$$e(n) = d(n) + z(n) - y(n) \tag{4.1}$$

Tel que : $y(n) = w_n^T u_n$ (4.2)

Un bruit $z(n)$ est modélisé avec la fonction Matlab *randn*.

Dans ce cas l'erreur résiduelle est équivalente au bruit $z(n)$.

Le rapport entre la puissance statistique moyenne du signal d'entrée u_n et celle du bruit z_n est appelé rapport signal bruit, et sera noté *RSB*.

Tel que : $RSB = 10 log_{10}\left(\dfrac{\sigma_u^2(n)}{\sigma_z^2(n)}\right)$ (4.3)

Où :

$\sigma_u^2(n)$: est la puissance statistique moyenne du signal d'entrée u_n.

et

$\sigma_z^2(n)$: est la puissance statistique moyenne du signal bruit $z(n)$.

Pour réaliser efficacement cette identification, on doit prendre en compte les propriétés particulières des canaux acoustiques et des signaux traités.
Les canaux acoustiques (trajets des ondes sonores) ont les propriétés caractéristiques suivantes :

- Réponse impulsionnelle à durée infinie ; la partie qu'il est utile d'identifier en pratique varie typiquement entre 30 ms et 250 ms suivant les applications, soit plusieurs centaines à plusieurs milliers de points aux fréquences d'échantillonnage audio (8 et 16 kHz) ; cette réponse a une structure temporelle complexe (ensemble de réflexions dépendant de la géométrie de la salle, des obstacles présents, etc...) et n'admet pas de modèle simple ayant peu de paramètres.

- Non-stationnarité due aux mouvements des personnes, aux déplacements d'objets, etc... , l'évolution temporelle peut être rapide, mais il n'y a pas de ruptures.

  Le signal à l'entrée est la parole ou un mélange bruit + parole; sa bande passante s'étend sur plusieurs octaves (300 Hz à 3 400 Hz pour la parole téléphonique, 150 Hz à 7 000 Hz pour la parole dite à « bande élargie »), son spectre n'est pas plat (formants, pente) et très variable dans le temps.

  Le bruit en sortie (bruit d'ambiance acoustique) est généralement non-stationnaire, non blanc et peut avoir un niveau élevé (parole locale appelée « double parole », bruit de roulement dans un véhicule). On doit donc utiliser des algorithmes d'identification adaptative (puisque les canaux acoustiques sont inconnus et évoluent au cours du temps) qui soient robustes aux perturbations en sortie.

4.3 Description des signaux de tests

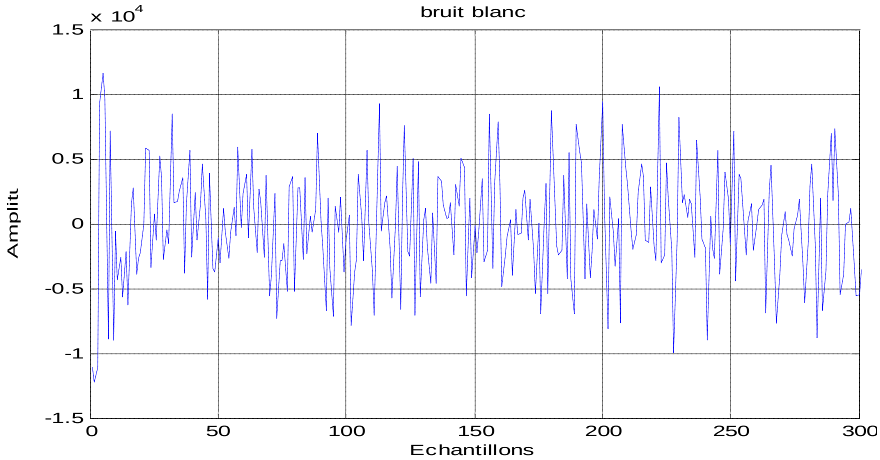
Les signaux utilisés dans les simulations que nous allons détaillées un peu loin dans ce document sont:

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.

Un bruit stationnaire qui a un spectre similaire au spectre moyen de la parole (bruit USASI), il est souvent utilisé comme signal de test dans les applications d'annulation d'écho acoustique pour évaluer la vitesse de convergence des algorithmes adaptatifs et leur capacité de poursuite des non stationnarités intervenant dans le chemin d'écho à identifier.

Un signal de parole constitué de deux phrases phonétiquement équilibrées, échantillonnées à 16 KHz codées sur 16 bits. La première phrase est prononcée par un locuteur masculin (signal parole 1) et la deuxième par un locuteur féminin (signal parole 2) [20].

(A)



(B)

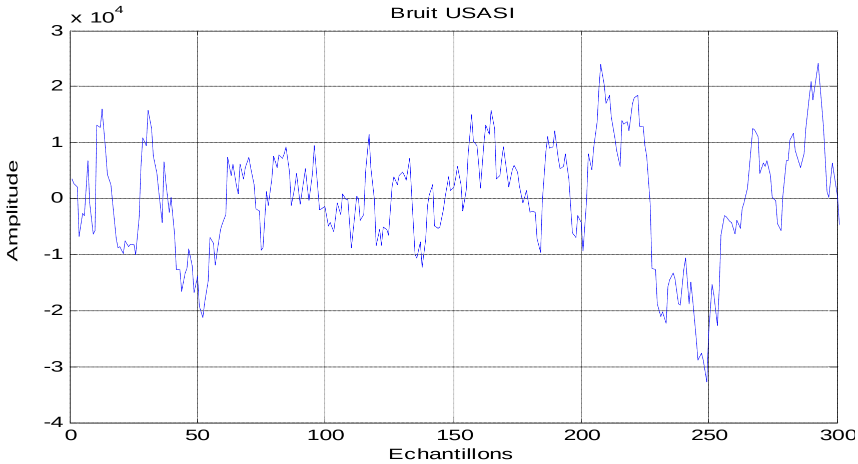
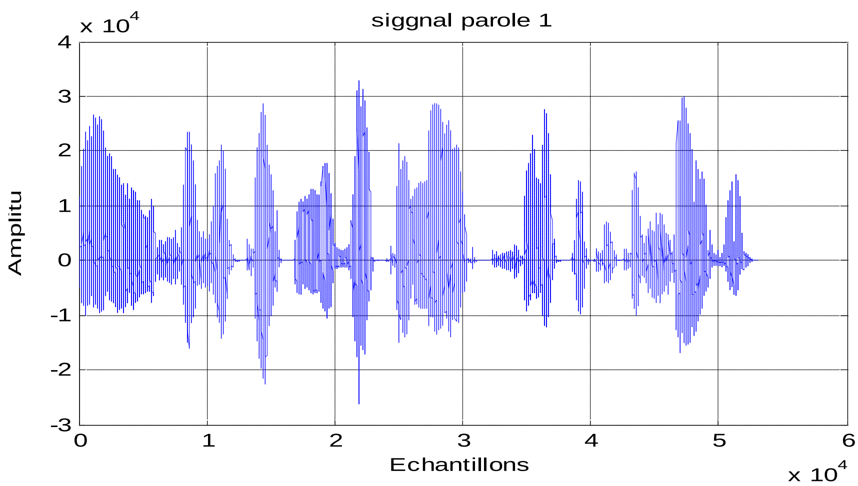


Figure 4.2 : Signaux simples. (A) : Bruit blanc, (B) : Bruit USASI.

(A)



(B)

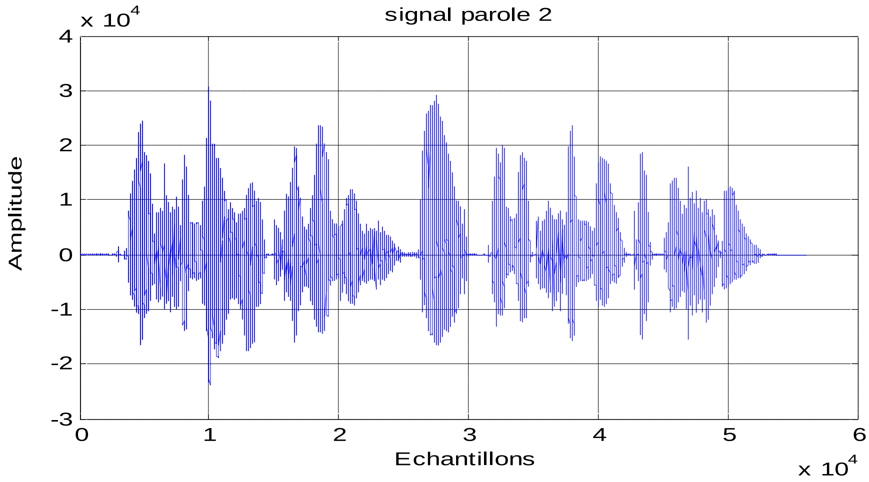


Figure 4.3 : description des signaux de parole, Fréquence d'échantillonnage $Fe = 16KHZ$. (A) : Signal de parole 1, (B) : Signal de parole 2.

Signal parole 1 **(A)** : « un loup se jeta immédiatement sur la petite chèvre ».

Signal parole 2 **(B)** : « il se garantira du froid avec ce bon capuchon »

4.4 Description des canaux de couplage acoustique

On dispose d'une réponse impulsionnelle de couplage acoustique, d'un habitacle de voiture contenant 1024 points, mesurée en régime stationnaire.

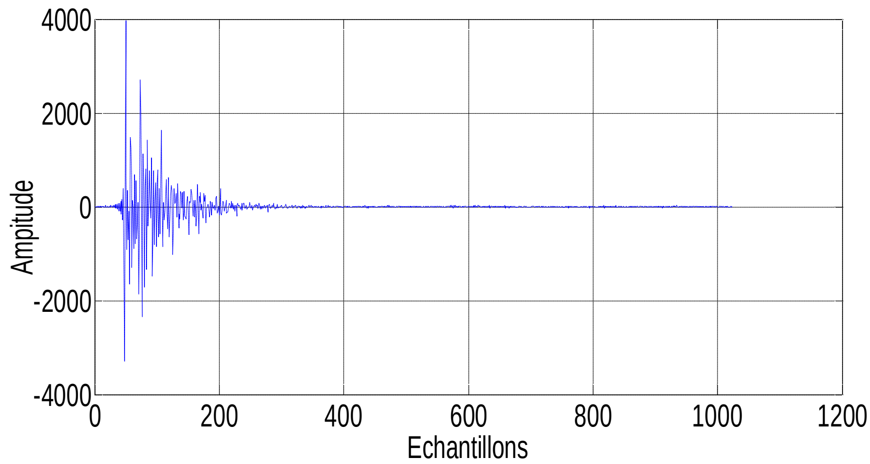


Figure 4.4 : Réponse impulsionnelle utilisée.

4.5 Description des critères de performance

Le critère de performance couramment utilisé en annulation d'écho est celui de l'évolution temporelle de l'EQM. Ce critère est donné par :

$$EQM(n) = 10\log\left(\sigma_e^2(n)\right) \tag{4.4}$$

Où $\sigma_e(n)$ symbolise une moyenne temporelle de M échantillons consécutifs, et $log(.)$ représente le logarithme à base 10, avec :

$$\sigma_e^2(n) = \frac{1}{M}\sum_{i=1}^{M} e^2(n) \tag{4.5}$$

Où $e(n)$ représente l'erreur de filtrage à priori calculé avant la mise à jour du filtre.

4.6 Test des algorithmes de filtrage adaptatif

Dans ce qui suit nous allons tester en deux parties les algorithmes de filtrage adaptatif qui ont été étudiés et discutés dans le deuxième chapitre.

La première partie concerne les algorithmes de gradient et de projection affine, et la deuxième concerne une étude comparative des trois nouvelles versions de pseudo projection affine décrite dans le troisième chapitre avec l'algorithme NLMS et le l'algorithme de projection affine exacte.

4.6.1 Test des algorithmes de filtrage adaptatif du gradient et de projection affine et de l'algorithme FRLS

Dans cette partie de ce chapitre, nous allons tester certains algorithmes qui existent dans la littérature. Nous avons sélectionné les plus utilisés dans les applications d'annulation d'écho acoustique. Parmi ces algorithmes nous avons choisi l'algorithme NLMS, APA, FAPA, GS-FAP.

Nous avons réalisé plusieurs simulations qui tiennent compte des points suivants :

M : Taille du filtre transverse.

K : Ordre de projection et la taille de prédiction aller/retour.

μ : Pas d'adaptation.

4.6.1.1 Test avec des signaux simples

Pour le cas stationnaire on prend comme signaux de test : un bruit blanc et bruit USASI.

4.6.1.1.1 Algorithme NLMS

A fin de percevoir le comportement de l'algorithme NLMS nous avons étudié l'effet de chaque paramètre influant sur cet algorithme en procédant à plusieurs expériences.

- Effet de la taille du filtre transverse

Les résultats de cette simulation obtenus pour plusieurs tailles du filtre transverse sont présentés dans le tableau et les figures 4.5 et 4.6, ci-dessous :

La taille du filtre M	EQM (dB)	
	Bruit blanc	Bruit USASI
32	87,85	85,8
64	87,25	87
128	86,5	87,5
256	86,4	87,3
512	86,5	87
1024	86.4	87

Tableau 4.1 : Effet de la taille de filtrage sur l'algorithme NLMS, μ=0.9, RSB=90.

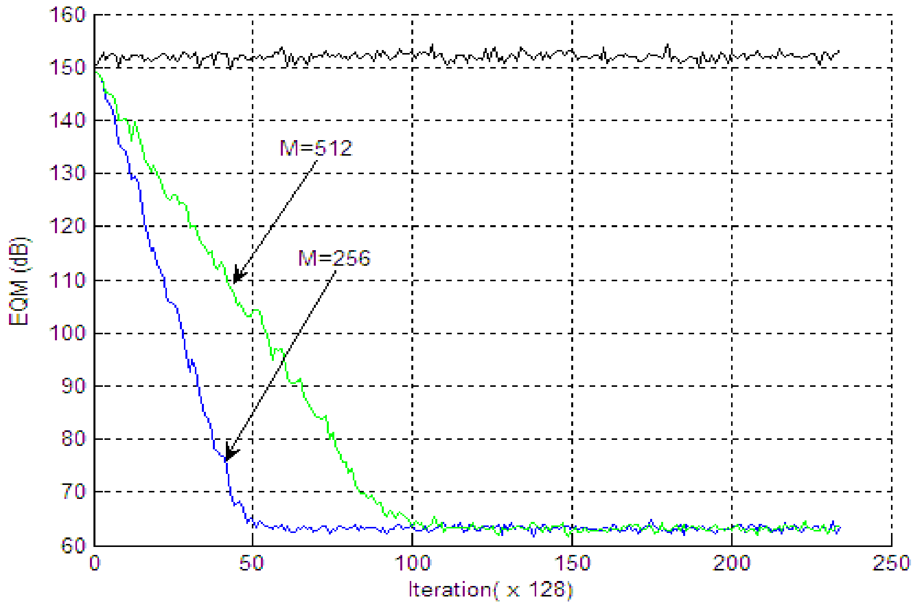


Figure 4.5 : Influence de la taille M sur le NLMS pour M=256, M=128, RSB=90, μ=0.9, et un signal d'entrée de type bruit blanc

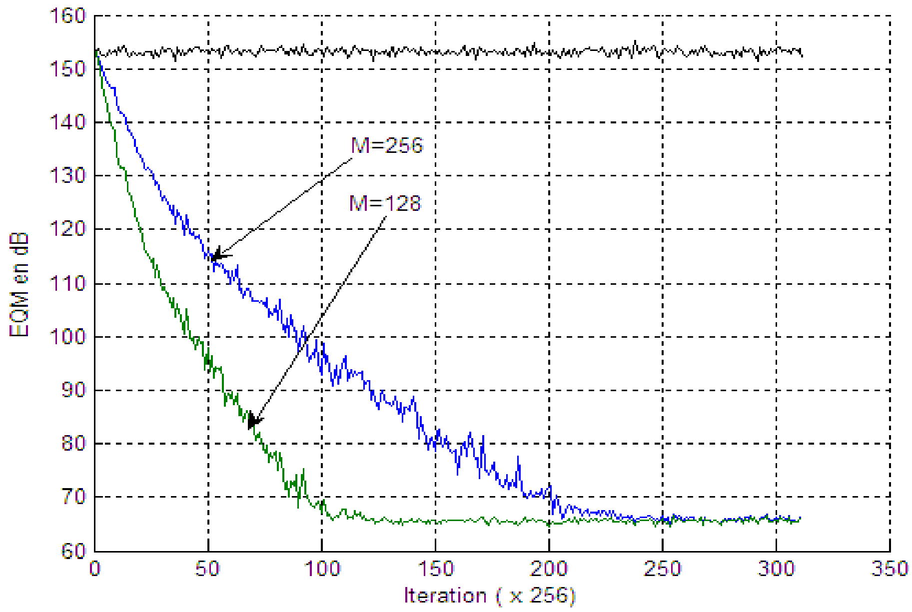


Figure 4.6 : Influence de la taille M sur le NLMS pour M=128, et M=256, μ=0.9, RSB=90 et un signal d'entrée de type bruit USASI.

Les résultats expérimentaux présentés dans le tableau 4.1 illustrent le bon comportement de l'algorithme NLMS avec les signaux de tests (bruit blanc, bruit USASI), on remarque une dégradation de l'énergie de l'erreur de filtrage (EQM) lorsqu'on a l'entrée un bruit USASI et pour une réponse impulsionnelle longue ce qui montre que la vitesse de convergence dépend de la nature du signal d'entrée, et les résultats montrent aussi que l'identification de la réponse impulsionnelle des systèmes avec l'algorithme NLMS nécessitent que la taille du filtre transverse soit égale ou supérieure à la taille réelle du système à identifier.

Les deux figures montrent que la vitesse de convergence de l'algorithme NLMS est inversement proportionnelle à la taille du filtre transverse.

- <u>Effet du pas d'adaptation μ</u>

Dans cette simulation, nous avons fixé la taille du filtre transverse M à 256 et nous avons fait varier le pas μ. Nous avons obtenus les résultats présentés sur la figure suivante :

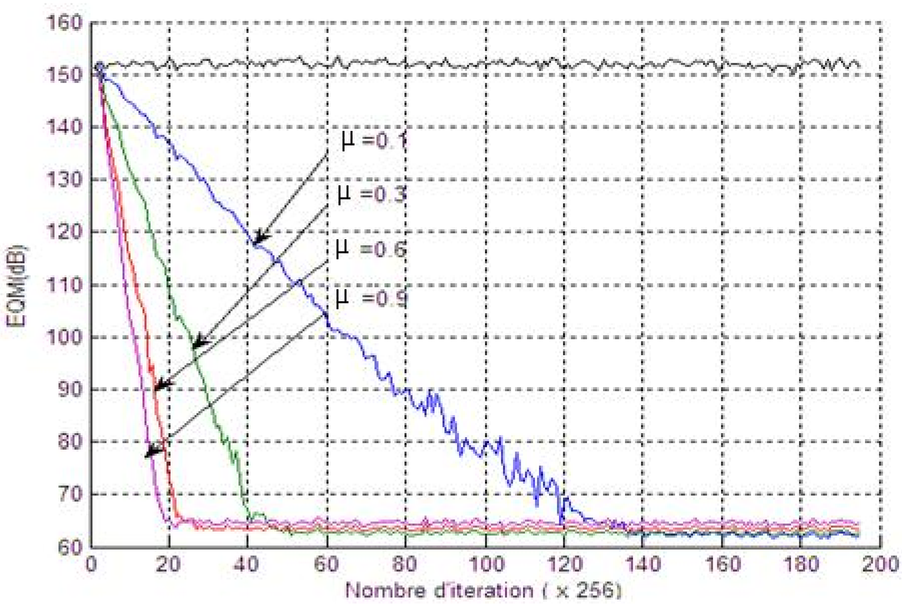


Figure 4.7 : Influence du pas d'adaptation dans l'algorithme NLMS, M=256, RSB=90 et un signal d'entrée de type bruit blanc

A partir des résultats de simulations illustrés dans la figure ci-dessus, on remarque bien que l'algorithme NLMS converge mieux lorsque le pas d'adaptation μ est choisi grand. Cette convergence devient plus importante dans le cas d'un bruit blanc Gaussien. On peut dire que l'algorithme NLMS a un bon comportement avec les signaux Gaussiens.

4.6.1 1.2Algorithme de projection affine exacte

Les paramètres intervenants dans le comportement de l'APA comprennent ceux du NLMS, la taille du filtre transversal et le pas d'adaptation, et un autre paramètre qui est l'ordre de projection.

- Influence de la taille du filtre transversal

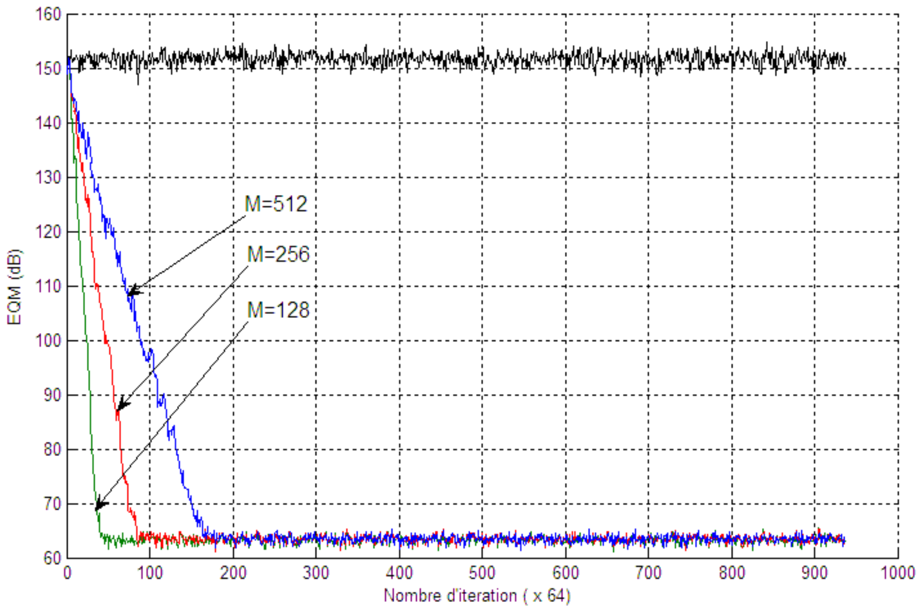


Figure 4.8 : Influence de la taille M sur l'algorithme APA, K=10, μ=0,2. RSB=90.
Signal d'entrée : bruit blanc.

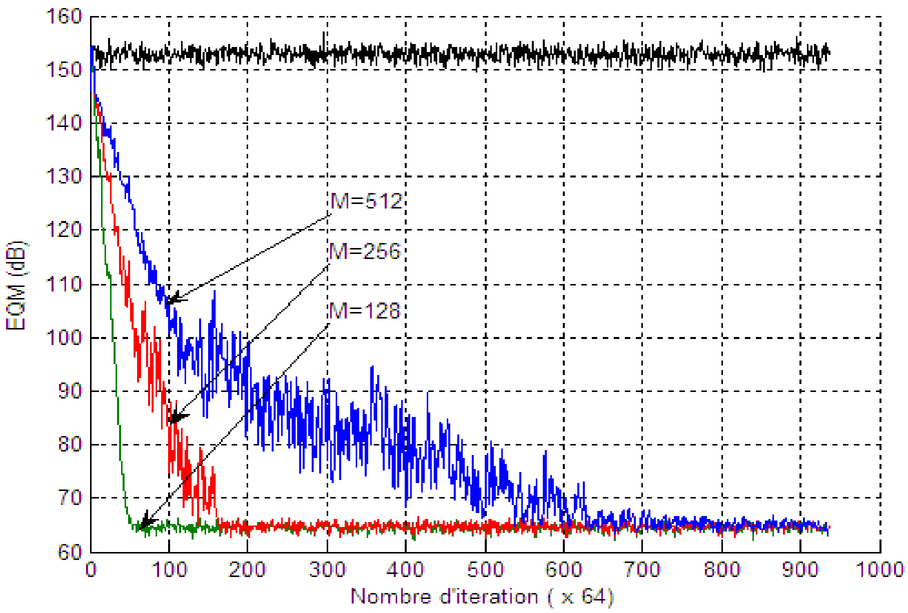


Figure 4.9 : Influence de la taille du filtre sur l'algorithme APA, K=10, μ=0,2, RSB=90 et un signal d'entrée de type bruit USASI.

Nous remarquons bien que l'algorithme de projection affine converge mieux lorsque la taille de filtre est faible cette convergence est rapide dans le cas d'un bruit blanc (voir les figures 4.8 et 4.9). Ceci qui montre que la vitesse de convergence de l'algorithme APA est inversement proportionnelle à la taille du filtre transverse.

- <u>Effet de l'ordre de projection et du pas d'adaptation μ</u>

Dans cette partie de simulation, on va étudier l'influence des paramètres μ (pas d'adaptation), et K (nombre de projection), dans l'algorithme de projection affine c'est-à-dire l'étude de performance de ces algorithmes en termes de vitesse de convergence et de l'EQM. Ces résultats sont obtenus à partir de 40000 itérations et en utilisant un filtre de taille (M= 256).

Les résultats de simulation obtenus avec un bruit blanc Gaussien à l'entrée et dans le cas d'une réponse impulsionnelle d'un habitacle de voiture sont illustrés dans le tableau suivant :

μ	Nombre de projection K	Nombre d'itération (x 64) pour atteindre le régime permanent
0.1	2	246
	4	139
	9	84
	20	63
0.2	2	148
	4	89
	9	74
	20	65
0.5	2	73
	4	67
	9	61
	20	57
0.9	2	66
	4	65
	9	64
	20	59

Tableau 4.2 : effet des paramètres le pas d'adaptation μ et l'ordre de projection M. Signal d'entrée bruit blanc.

Les résultats de simulation obtenus avec un bruit USASI à l'entrée, dans le cas d'une réponse impulsionnelle d'un habitacle de voiture, sont illustrés dans le tableau suivant :

μ	Nombre de projection K	nombre d'itération (x 64) pour atteindre le régime permanent
0.1	2	466
	4	205
	9	179
	20	124
0.2	2	307
	4	158
	9	130
	20	104
0.5	2	130
	4	119
	9	108
	20	99
0.9	2	118
	4	108
	9	98
	20	74

Tableau 4.3 : effet des paramètres : pas d'adaptation μ et l'ordre de projection M.

Signal d'entrée bruit USASI.

A partir des résultats de simulations répertoriés dans les deux tableaux 4.2 et 4.3 on remarque bien que l'algorithme de projection affine converge mieux lorsque le pas d'adaptation μ et le nombre de projection *K* sont choisis grands.

- <u>Influence de l'ordre de projection</u>

L'ordre de projection K dans cet algorithme correspond au nombre de ligne de la matrice de corrélation, dont on doit varier pour voir mieux leur influence sur le comportement de l'APA.

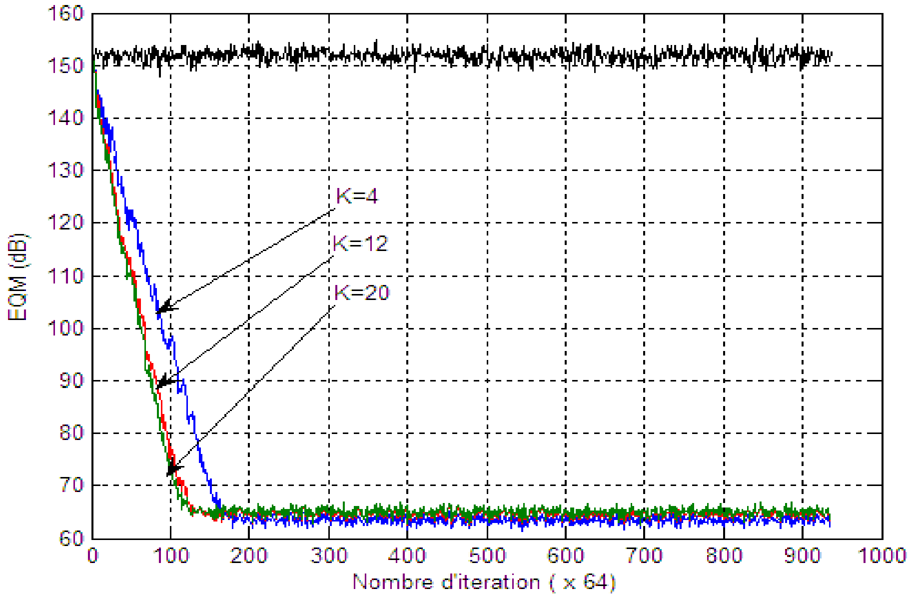


Figure 4.10 : Influence de l'ordre de projection dans l'algorithme APA, M=512, RSB=90, μ=0,2, et un signal d'entrée de type bruit blanc.

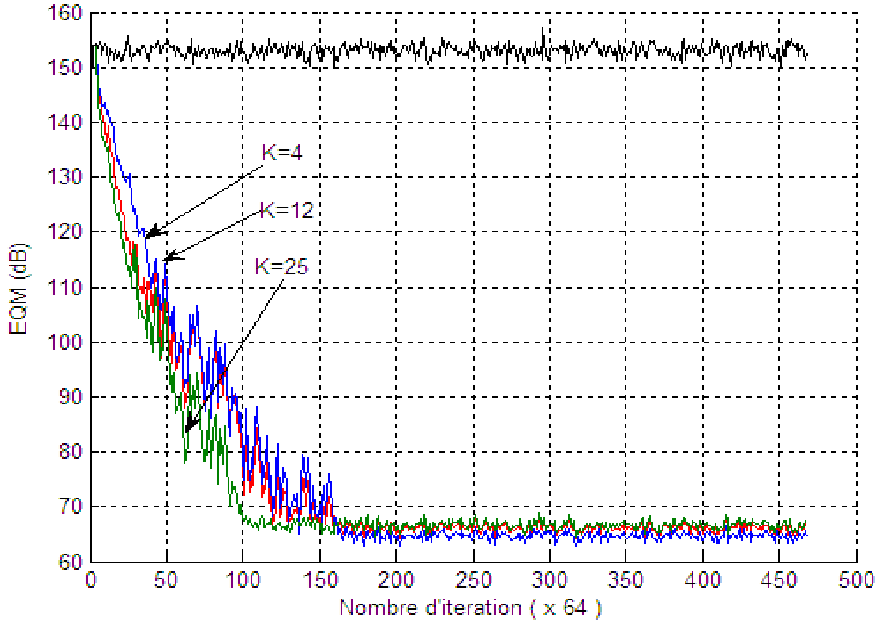


Figure 4.11 : Influence de l'ordre de projection dans l'algorithme APA, M=512, μ=0,2, RSB=90 et un signal d'entrée : bruit USASI.

Les figures 4.10 et 4.11 montrent le bon comportement de l'algorithme de projection affine APA qui converge mieux si le nombre de projection augmente surtout dans le cas d'un bruit blanc avec des réponses impulsionnelles acoustiques de taille faible (exemple : habitacle de voiture).

On peut conclure à partir de ces figures que l'algorithme de projection affine APA a un bon comportement que l'algorithme NLMS si $K>1$. Nous concluons aussi que l'inconvénient majeur de l'algorithme APA est la dégradation de l'EQM finale.

4.6.1.1.3 Algorithme FRLS

Pour étudier le comportement et la stabilité numérique de l'algorithme FRLS on doit procéder aux simulations suivantes :

- Influence de la taille du filtre transversal sur le FRLS

Dans cette simulation on a pris λ fixe à λ=0.999 et nous avons fait varier la taille du filtre. Les résultats sont présentés dans la figure suivante :

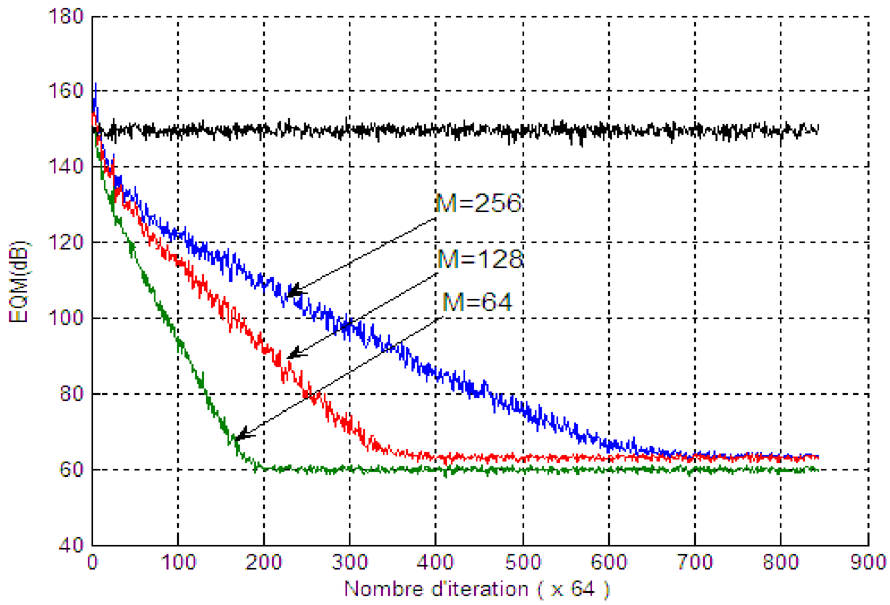


Figure 4.12 : Influence de la taille du filtre dans l'algorithme FRLS, $\lambda=0,999$,

RSB=90, et un signal d'entrée de type bruit USASI.

Nous avons simulé l'algorithme FRLS avec un signal de type bruit USASI pour différentes tailles du filtre adaptatif. On remarque que le FRLS converge lentement quand on augmente la taille du filtre.

- <u>Etude de la stabilité de l'algorithme</u>

On a pris le facteur d'oubli : $\lambda = 1 - \dfrac{1}{20 \times M}$, tel que M représente la taille du filtre.

Ci-dessous le tracé de la variable indicateur de divergence ξ pour tester la stabilité de l'algorithme FRLS.

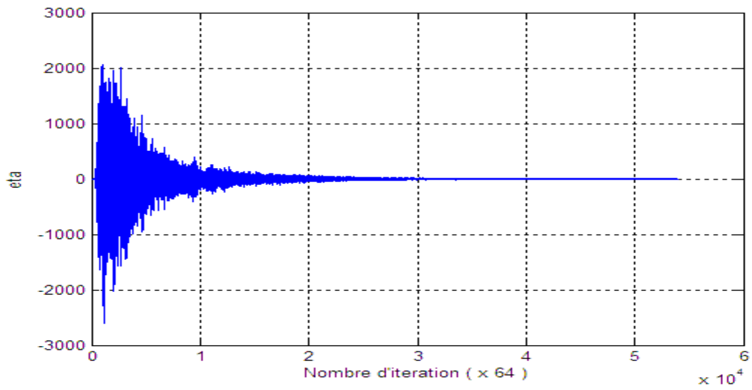


Figure 4.13 : Indicateur de divergence dans l'algorithme FRLS λ=0,999, taille du filtre M=256, signal d'entrée : bruit USASI.

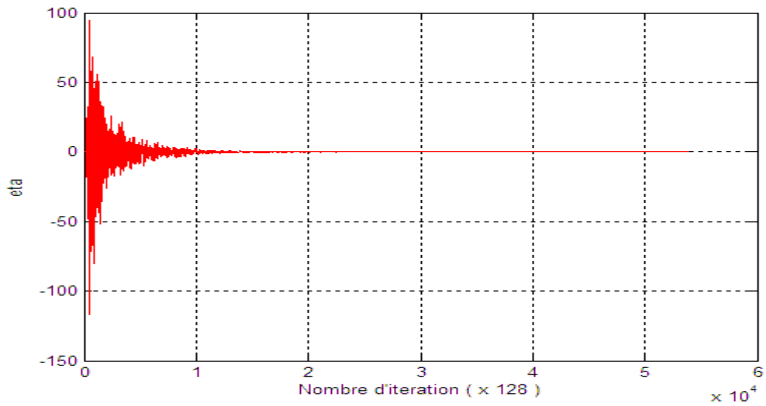


Figure 4.14 : Indicateur de divergence dans l'algorithme FRLS λ=0,999, taille du filtre M=128, signal d'entrée : bruit USASI.

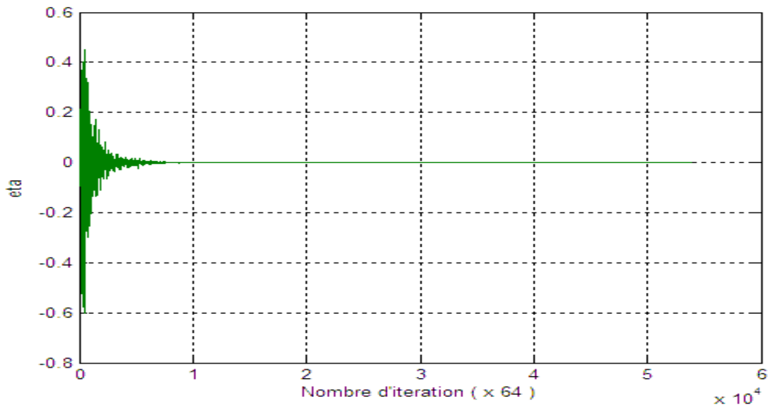


Figure 4.15 : Indicateur de divergence dans l'algorithme FRLS λ=0,999, taille du filtre M=64, signal d'entrée : bruit USASI.

Les figures précédentes montrent que le FRLS est stable numériquement et a un bon comportement quand le signal d'entrée est stationnaire.

4.6.1.1.4 Algorithme de projection affine rapide (FAP)

Les simulations du FAP nous a conduit à programmer le FRLS, à fin d'améliorer les performances de l'algorithme FAP en qualité de complexité et aussi pour le calcul de l'inverse la matrice de corrélation.

Les paramètres qui influent sur le FAP sont étudiés comme suit :

- Effet de la taille du filtre

Dans les simulations suivantes on a fais plusieurs expériences en variant dans la taille du filtre transversal avec $\mu=0.9$ et l'ordre de prédiction $K=10$.

Les résultats de simulation sont représentés dans les figures suivantes:

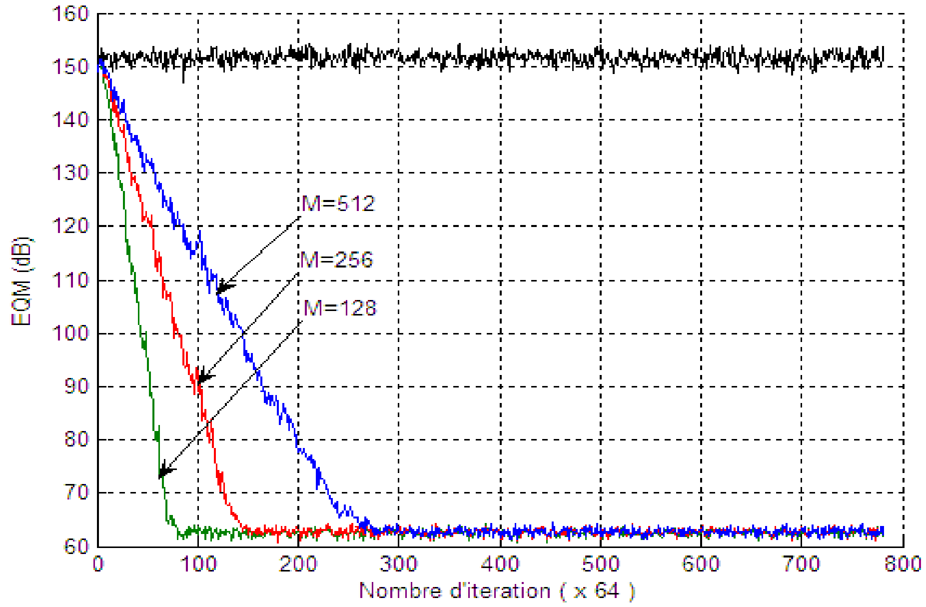


Figure 4.16 : Influence de la taille du filtre dans l'algorithme FAPA, K=10, μ=0,9, RSB=90, signal d'entrée : bruit blanc.

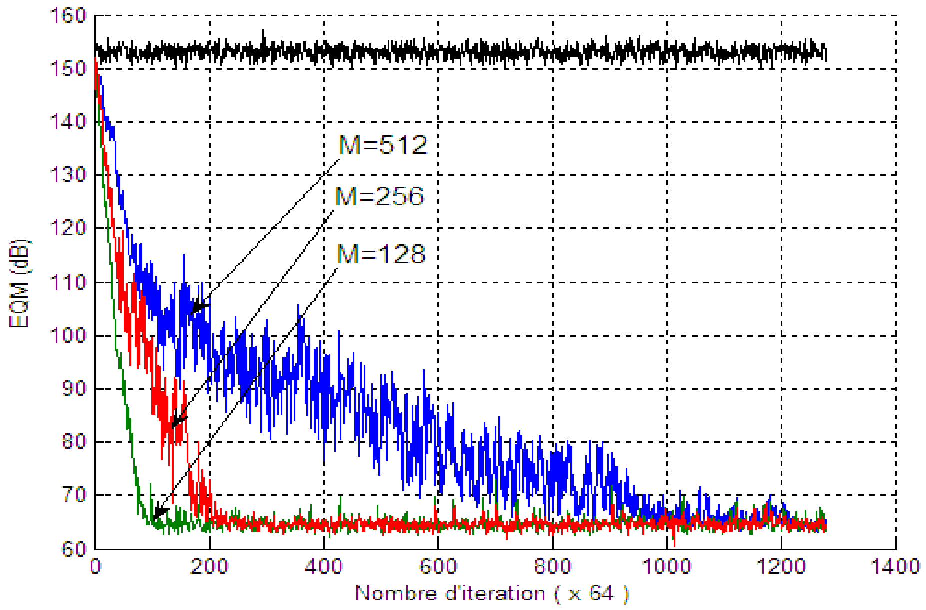


Figure 4.17 : Influence de la taille du filtre dans l'algorithme FAPA, K=10, μ=0,9, RSB=90, signal d'entrée : bruit USASI.

Le FRLS est une partie intégrante dans l'algorithme FAP, donc l'instabilité du FRLS influence directement sur le FAP et il le rend instable. Pour cela, on a du assurer que le FRLS programmé donne un résultat satisfaisant pour l'intégrer à notre algorithme FAP.

On remarque bien que l'algorithme de projection affine rapide (FAP), a un bon comportement avec le signal d'entrée bruit blanc alors que pour le bruit USASI on remarque qu'il y a une légère dégradation au niveau du régime transitoire.

Les résultats présentés dans les figures 4.16 et 4.17 montrent aussi que la vitesse de convergence du FAP est inversement proportionnelle à la taille du filtre.

- Effet de la taille de la prédiction (ordre de projection)

Pour voir mieux l'effet de l'ordre de prédiction sur le comportement du FAP on a fait les simulations illustrées dans la figure 4.18 pour un bruit blanc et figure 4.19 pour un bruit USASI en prenant μ=0.9 et la taille du filtre M=256.

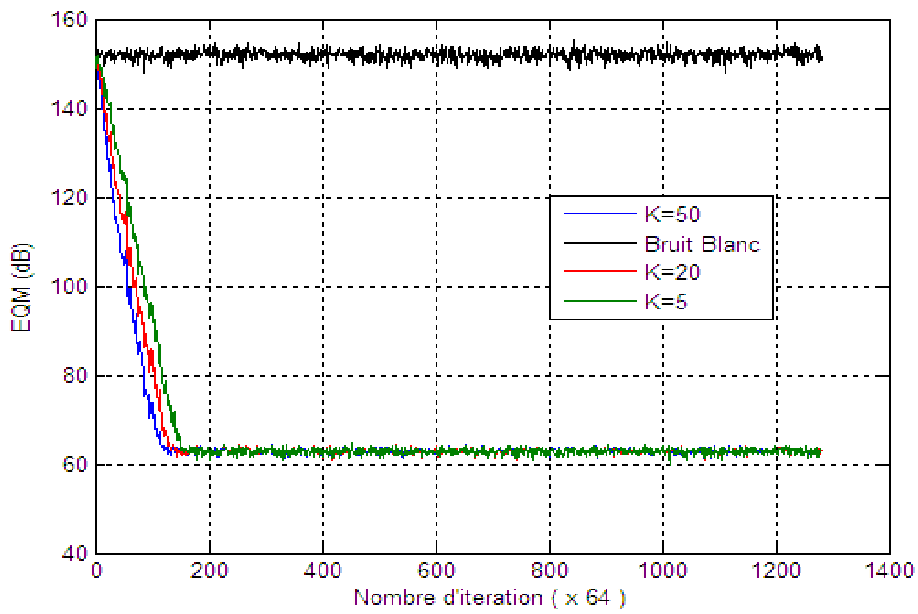


Figure 4.18 : Influence de l'ordre de prédiction dans l'algorithme FAPA, la taille du filtre M=256, μ=0,9, RSB=90, signal d'entrée : bruit blanc.

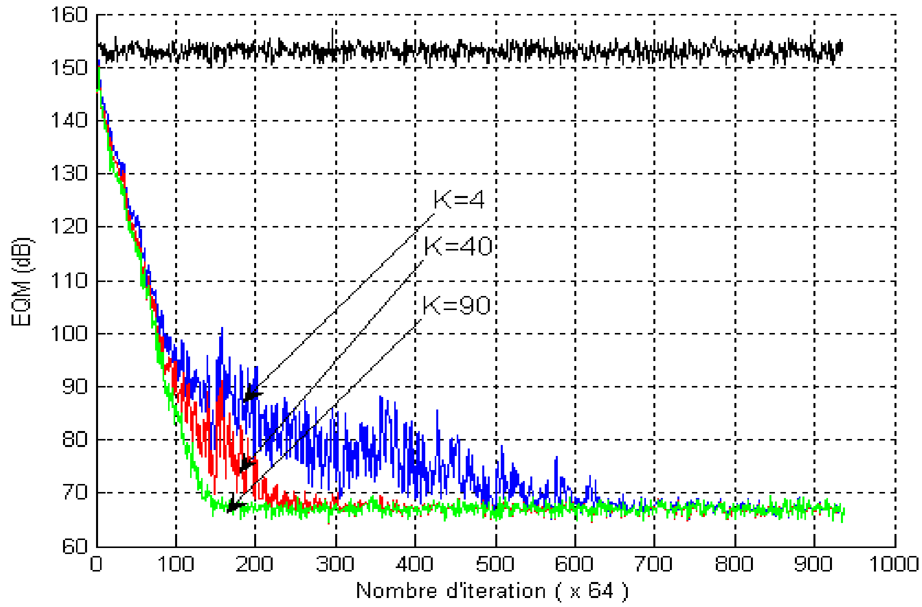


Figure 4.19 : Influence de l'ordre de prédiction dans l'algorithme FAPA, la taille du filtre M=256, µ=0,9, RSB=90, signal d'entrée : bruit USASI.

Les figures ci-dessus 4.17 et 4.18 (dont les signaux d'entrée sont un bruit blanc et un bruit USASI respectivement), représentent les variations de la vitesse de convergence de l'algorithme FAP en fonction du temps en variant dans le paramètre de l'ordre de prédiction. Le rapport signal sur bruit égale à 90dB.

Nous Remarquons bien que la vitesse de convergence est proportionnelle a la variation de l'ordre de prédiction.

4.6.1.1.5 Algorithme FAP en utilisant la méthode de Gauss-Seidel

Nous reprenons la même démarche pour étudier le comportement de l'algorithme de projection affine rapide qui est basé sur la méthode de Gauss-Seidel (GS-FAP).

- Effet de la taille du filtre

  Les résultats de simulation avec cet algorithme sont présentés dans les figures suivantes :

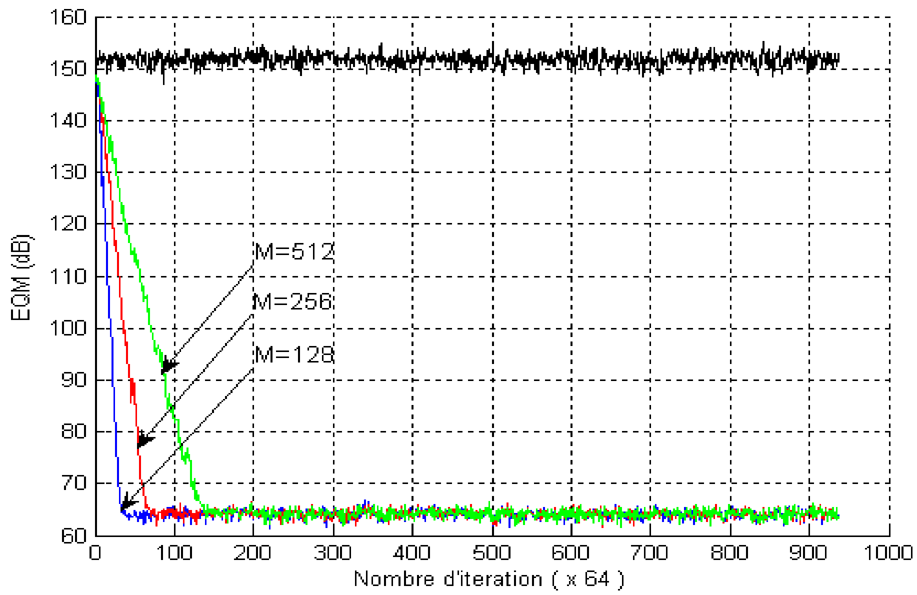


Figure 4.20 : Influence de la taille du filtre dans l'algorithme GS-FAP, l'ordre de projection K=10, μ=0,8, RSB=90, signal d'entrée : bruit blanc.

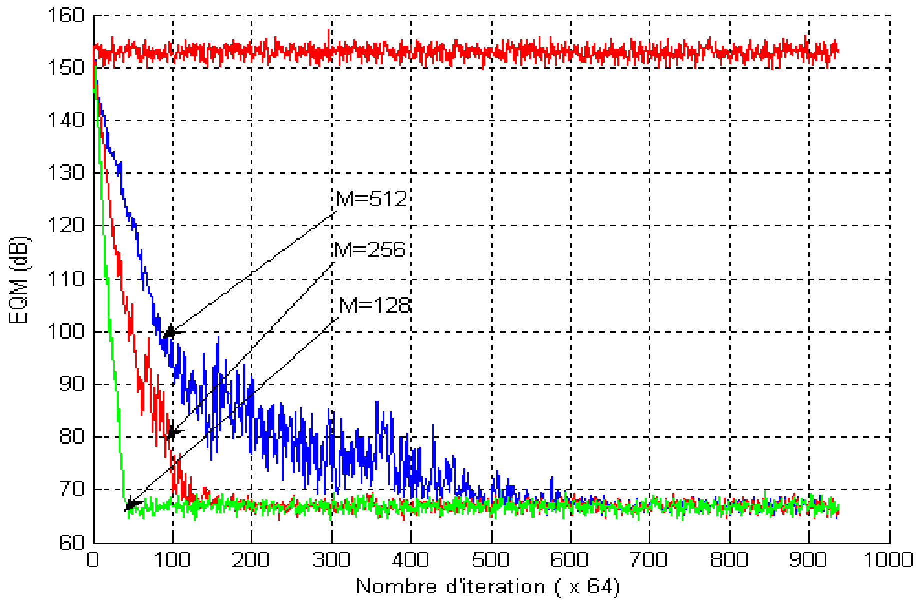


Figure 4.21 : Influence de la taille du filtre dans l'algorithme GS-FAP, l'ordre de projection K=10, μ=0,8, RSB=90, signal d'entrée : bruit USASI.

Les figures ci-dessous 4.20 et 4.21 présentent l'évolution temporelle des EQM de l'algorithme de GS-FAP avec un signal d'entrée de type bruit blanc et un autre de type USASI. On remarque que la vitesse de convergence est inversement proportionnelle à la variation de la taille du filtre. On remarque que le régime permanent est atteint après 500 blocs d'itérations avec une taille de 512, après 100 blocs avec une taille de 256 et enfin après 40 blocs d'itérations avec une taille de 128.

On remarque aussi qu'il y a une légère dégradation au régime transitoire pour le bruit USASI.

4.6.1.2 Test avec le signal parole

Dans ces simulations, le canal acoustique est formé des M premiers points d'une réponse impulsionnelle de couplage acoustique mesurée dans une voiture. La taille M du filtre transverse est variable à fin d'identifier le comportement des algorithmes discutés.

Le signal à l'entrée du canal est de la parole naturelle (phrases prononcées par un locuteur masculin). Les signaux sont échantillonnés à une cadence de 8 kHz.

4.6.1.2.1 Algorithme NLMS

Les résultats de simulations du NLMS pour un signal parole avec plusieurs tailles du filtre transverse sont présentés dans la figure suivante :

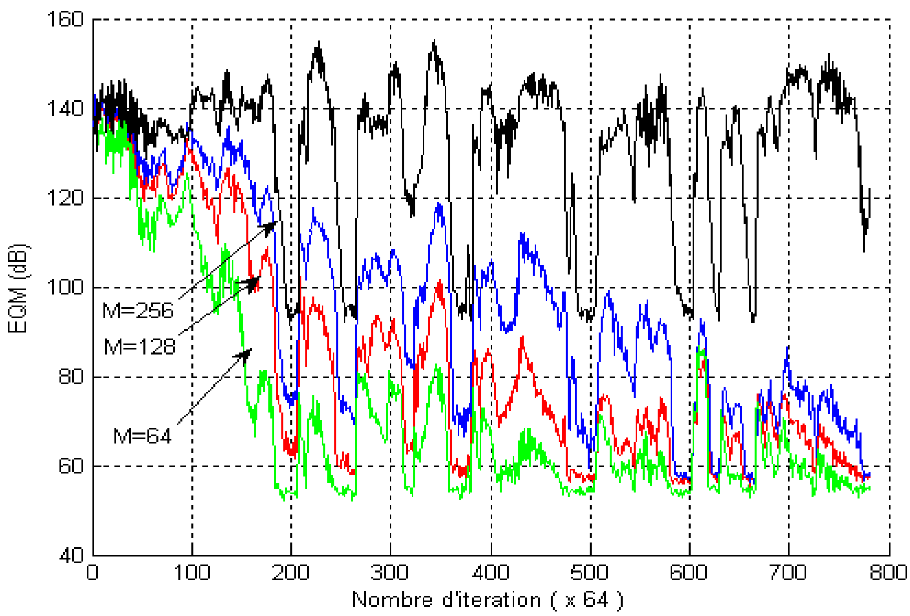


Figure 4.23 : Influence de la taille de dans l'algorithme NLMS, µ=0,6, RSB=90, signal d'entrée : Signal parole.

La figure ci-dessus montre le mauvais comportement de l'algorithme NLMS avec une excitation de type parole.

On remarque qu'il y a une dégradation qui devient plus importante quand on augmente la taille du filtre.

Pour le NLMS, la dégradation est considérable, on remarque qu'il n'atteint le régime permanent qu'après 38400 itérations pour une taille de filtre de 256 points.

4.6.1.2.2 Algorithme de Projection Affine exacte

A partir des résultats de simulations présenté sur les figures 4.20 et 4.19, on remarque que l'erreur quadratique moyenne (EQM) finale est presque la même pour les différentes tailles mais la vitesse de convergence est différente. On remarque que l'algorithme de projection affine converge après 12800 itérations.

On remarque aussi que l'augmentation de la taille du filtre est inversement proportionnelle à la vitesse de convergence, alors que cette dernière et beaucoup plus meilleur quand on augmente l'ordre de projection *K*.

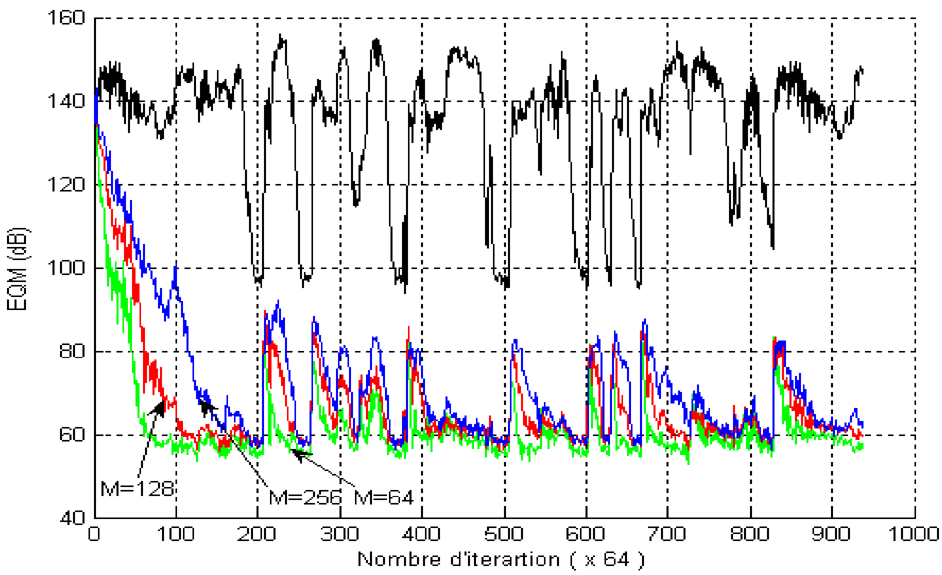


Figure 4.24 : Influence de la taille du filtre dans l'algorithme APA, l'ordre de projection M=10, µ=0,9, RSB=90, signal d'entrée : Signal parole.

4.6.1.2.3 Algorithme de projection affine rapide FAP

Pour traiter le comportement de l'algorithme FAP avec la parole on injecte à l'entrée du filtre transversal un signal parole, en étudiant les paramètres influents sur leur fonctionnement.

Pour étudier le comportement de l'algorithme FAP avec la parole on a fait plusieurs expériences pour des tailles différentes.

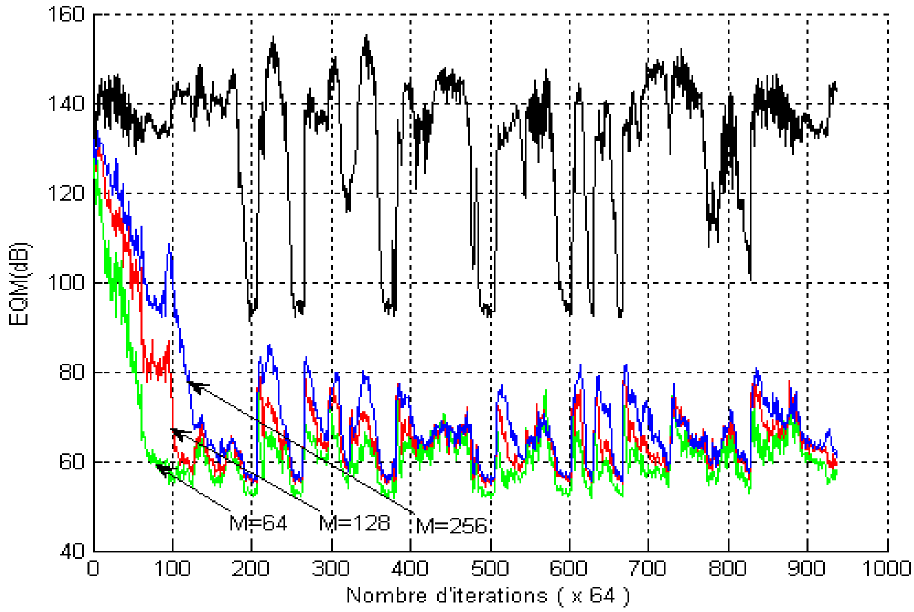


Figure 4.25 : Influence de la taille du filtre dans l'algorithme FAP, l'ordre de projection $K=5$, $\mu=0,9$, RSB=90, signal d'entrée : Signal parole.

D'après les simulations on remarque que la taille du filtre influe inversement sur le taux de convergence de l'algorithme FAP. Pour une taille de 64 on a une convergence après 6400 itérations alors que pour une taille de 256 points, l'algorithme atteint le régime permanent après 9600 itérations.

4.6.1.2.4 Algorithme de FAP avec la méthode de Gauss-Seidel (GS-FAP)

On fait les mêmes démarches précédentes, en étudiant l'effet de taille du filtre à fin d'améliorer les performances de l'algorithme GS-FAP avec une entrée de type parole.

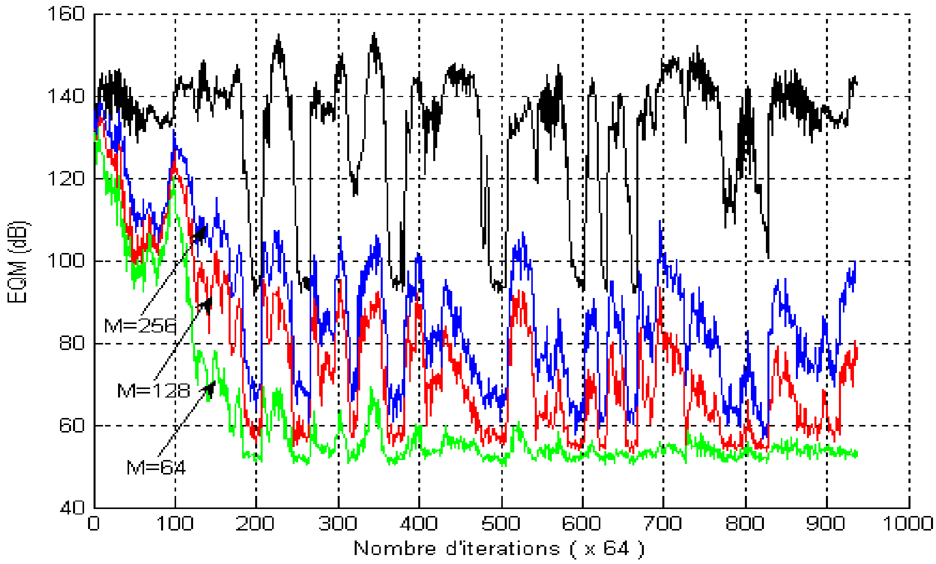


Figure 4.26 : Influence de la taille du filtre dans l'algorithme GS-FAP, l'ordre de projection $K=10$, $\mu=0,8$, RSB=90, signal d'entrée : Signal parole.

Les simulations montrent le bon comportement de l'algorithme de GS-FAP pour des tailles de filtre faibles, mais avec une légère dégradation au régime permanent qui est due à une excitation parole.

D'après les simulations de l'algorithme GS-PAP, on aura une convergence après 12800 itérations dont le niveau de l'EQM au régime permanent varie inversement avec l'augmentation de la taille du filtre adaptatif. L'avantage majeur par rapport à l'algorithme de projection affine est sa réduction au niveau de complexité avec $2M+K^2+3K+5$ entre multiplications et divisions.

4.6.2 Test des Nouvelles versions de l'algorithme de Pseudo-APA, application à l'annulation d'écho acoustique

Dans le cadre de ce sujet de magistère, nous avons proposé trois nouvelles versions de l'algorithme de Pseudo Projection Affine qui ont été détaillées dans le chapitre III. Dans cette partie de ce chapitre nous allons évaluer le comportement de chaque version en présence du signal de parole. Nous avons opté à comparer dans chaque expérience, le comportement de chaque version avec l'algorithme NLMS et la version exacte de l'algorithme de projection affine.

Pour tester les nouvelles versions proposées de l'algorithme PAP en contexte d'annulation d'écho acoustique. On injecte à l'entrée du filtre adaptatif un signal parole échantillonné à une fréquence de 16KHz, l'ordre de projection K est égale à 9, le rapport signal sur bruit RSB=90dB, la taille du filtre M=256 et le pas d'adaptation $\mu=0.9$.

Les figures 4.28, 4.29, présentent le comportement des nouvelles versions proposées dans [23] de l'algorithme PAP en comparaison avec l'APA et l'algorithme NLMS.

Nous Remarquons bien que le comportement de l'algorithme de PAP avec les trois structures proposées sont presque similaire à celui de l'APA.

Nous remarquons aussi qu'avec les trois structures 1, 2 et 3 dont $\mu=0.9$, l'algorithme PAP converge après 6400 itérations alors que le NLMS atteint son régime permanent après 20480 itérations.

Nous avons aussi noté que l'évaluation de l'EQM des deux variantes de PAPA (Structures 1 et 2) donne strictement le même comportement, ceci prouve et montre l'exactitude de notre développement d'équivalence entre ces deux structures qui était détaillé dans le chapitre 3.

Nous avons noté que la troisième structure a le même comportement des deux autres structures. L'avantage de cette structure est sa faible complexité par rapport aux deux autres structures.

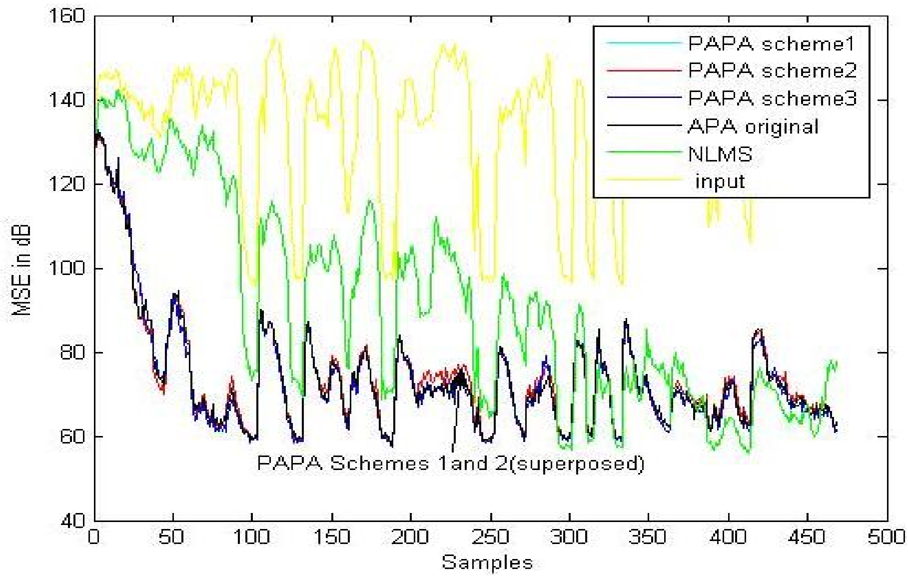


Figure 4.28 : EQM en dB pour le PAP avec une excitation parole.

(Les prédicteurs sont calculés avec la méthode de covariance, RSB=90, $\mu = 0.9$, $M = 256$, $K = 9$).

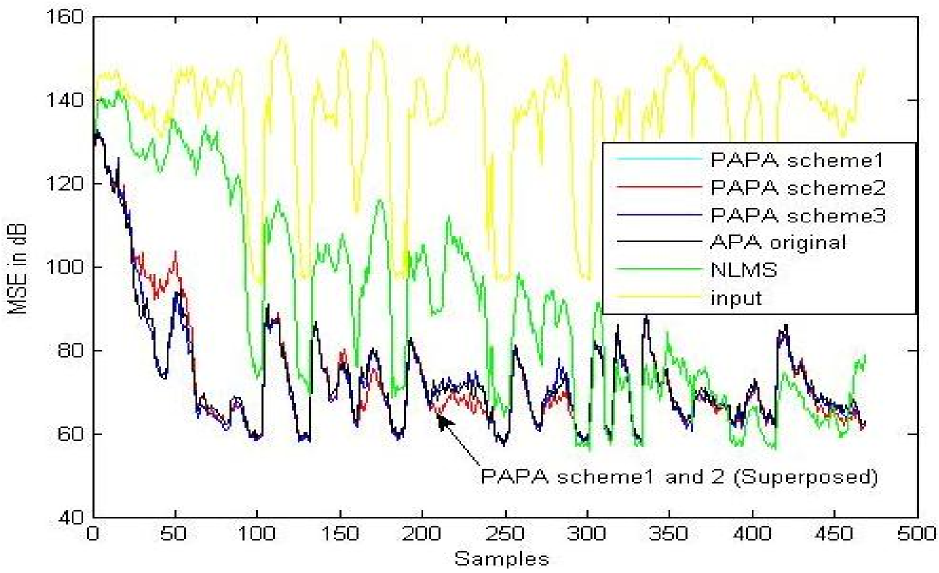


Figure 4.29 : EQM en dB pour le PAP avec une excitation parole.

(Les prédicteurs sont calculés avec la méthode de Levinson-Durbin, RSB=90, μ=0.9, $M = 256$, $K = 9$).

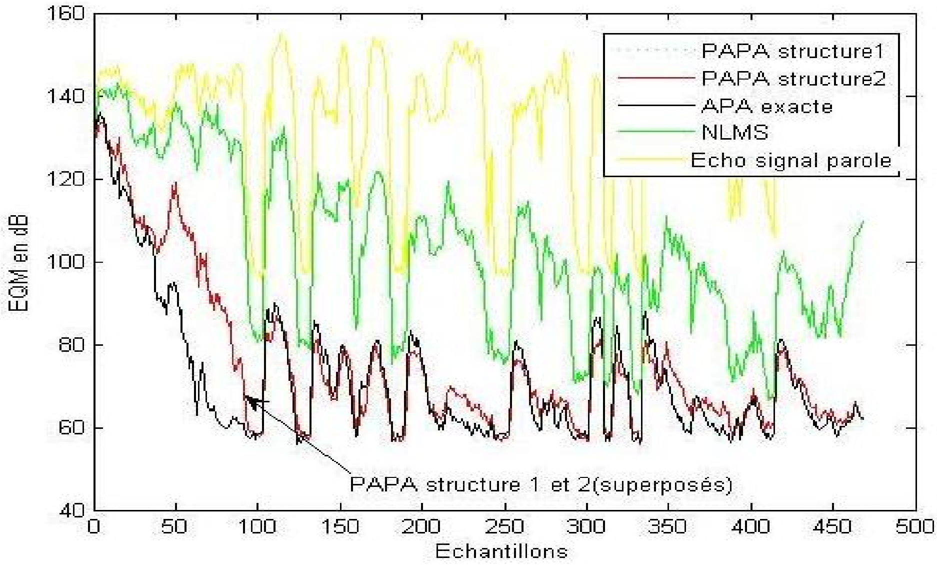


Figure 4.30 : EQM en dB pour le PAP avec une excitation parole.
(Les prédicteurs sont calculés avec la méthode de Levinson-Durbin, RSB=90, $\mu = 0.3$, M = 256, K = 9).

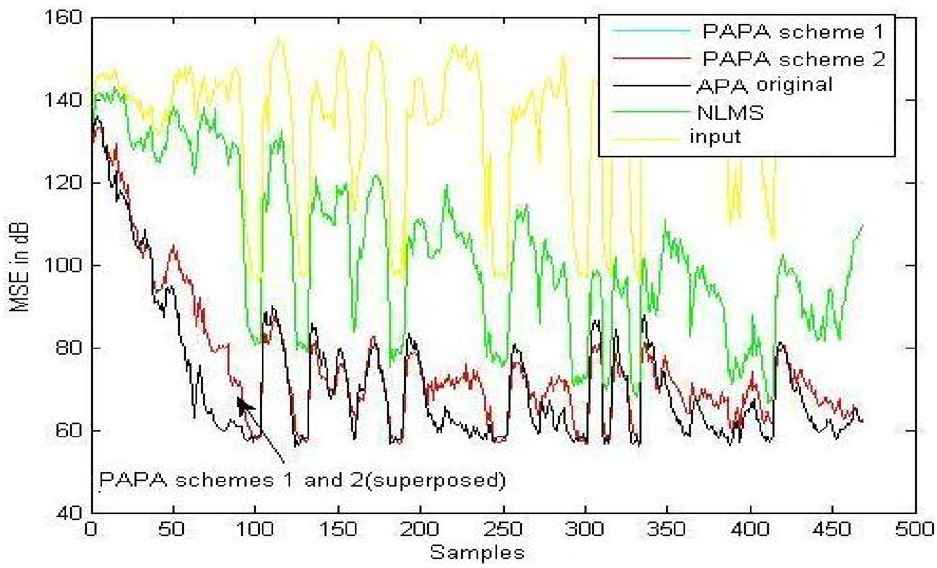


Figure 4.31 : EQM en dB pour le PAPA avec une excitation parole.
(Les prédicteurs sont calculés avec la méthode de covariance, RSB=90, $\mu = 0.3$, M = 256, K = 9).

Les résultats de simulations présentés dans les figures 4.30 et 4.31 avec les mêmes paramètres qu'auparavant, mais avec un pas d'adaptation μ=0.3.
Nous remarquons bien que le comportement des trois nouvelles versions de l'algorithme de Pseudo projection affine est presque similaire à celui de l'APA exacte même lorsqu'on change le pas d'adaptation μ mais avec une légère dégradation au régime permanent par rapport à l'algorithme NLMS.

4.7 Conclusion

Les simulations montrent bien que les algorithmes de filtrage adaptatif que nous avons présenté dans ce mémoire se comportent bien avec les deux signaux d'excitation (bruit USASI et bruit blanc) tel que le NLMS converge en prenant un pas d'adaptation égal à 0.9 et une taille M égale 256 après 51200 itérations avec une excitation de type bruit USASI ; pour l'APA en prenant les mêmes paramètres avec un l'ordre de projection K égale à 9, la convergence est atteinte après 6272 itérations.

Nous avons remarqué que la vitesse de convergence des algorithmes de filtrage adaptatif est inversement proportionnelle à la taille du filtre transverse, alors que ces algorithmes convergent mieux lorsque nous prenons le pas d'adaptation μ grand.

Pour un signal de type parole, nous avons remarqué que les algorithmes de projection affine (APA, FAP, GS-FAP, et les nouvelles versions de PAP) convergent mieux que l'algorithme du NLMS.
La dégradation des performances de l'algorithme NLMS est due aux non stationnarité que contient le signal de parole.

Les trois nouvelles versions du Pseudo Projection Affine que nous avons proposé dans [23] ont fait preuve de performance au niveau de convergence par rapport aux algorithmes adaptatifs, avec une convergence assez similaire à celle de l'APA exacte.

CONCLUSION GENERALE

Lôétude que nous avons présentée dans ce mémoire concerne lôannulation dôécho acoustique en utilisant les algorithmes de projection affine dont nous exposons des nouvelles versions basées sur lôalgorithme pseudo projection affine.

Afin dôaméliorer les performances de ces algorithmes nous avons fait une étude comparative avec les algorithmes de filtrage adaptatif classiques tels que le NLMS, les algorithmes de projection affine et lôalgorithme des moindres carrées rapide.

Dans ce contexte, nous avons étudié le comportement de lôalgorithme NLMS qui est très simple à mettre en î uvre mais il a une vitesse de convergence lente dans le cas dôun signal source de type parole alors que lôalgorithme de projection affine qui possède de meilleures performances en qualité de vitesse de convergence, il a une complexité nettement trop grande pour être utilisé dans une application réelle.

Un autre algorithme appelé algorithme de projection affine rapide « FAP » est étudié dans le but de réduire la complexité de calcul de lôAPA exacte. Dôaprès les simulations présentées dans le chapitre 4, on a remarqué que le FAP a un bon comportement avec une excitation parole mais il souffre dôune instabilité numérique à cause de la partie prédiction qui est calculé par lôalgorithme FRLS.

Afin de remédier à cette instabilité, lôalgorithme de GS-FAP est mis en î uvre dont il est basé sur la méthode de Gauss-Seidel pour inverser la matrice de corrélation, ceci met lôalgorithme FAP stable dans une comparaison avec d'autres algorithmes. Les simulations montrent bien le bon comportement de ce dernier avec les signaux fortement corrélé tel que la parole. La complexité de cet algorithme GS-FAP est de lôordre de $2.M+K^2+4.M-1$.

Les trois nouvelles versions du pseudo projection affine quôon a proposé dans ce mémoire, et qui sont établies à partir des structures de blanchiment décrites dans le chapitre III, ont permis dôobtenir des performances plus proches que celles de lôalgorithme de projection affine exacte avec une complexité assez faible que ce dernier qui est de lôordre de $2M+K^2+5K+4$. Les résultats de simulations montrent bien la supériorité de ces nouvelles versions par rapport à lôalgorithme NLMS.

Les futurs travaux qui peuvent sõinscrire dans la suite de ce travail sont :

- Proposition dõautres algorithmes qui se basent sur les trois nouvelles versions de pseudo projection affine [23], en utilisant la technique de la variation du pas dõadaptation
- Lõimplémentation de ces algorithmes sur des DSP ou bien sur des stations VHDL et FPGA.

REFERENCES

1. R.Bourlard, H.Boite, T.Dutoit, J.Hancq, H.Liech « Traitement de la parole »Presses polytechniques et universitaires Romandes 2000, Lausanne.
2. Francis Cottet « Aide mémoire traitement de signal » Dunod, Paris 2005.
3. M. Bellenger « traitement numérique du signal », MASSON ,1987.
4. Dimitris G. Manolakis, Vinay K. Ingle , Stephen M. Kogon «Statistical and Adaptive Signal Processing» ARTECH HOUSE 2005.
5. VijayK.Madisetti, Douglas B. Williams ñDigital Signal Processingò CRC Press 1999.

6. S. Haykin, Adaptive Filter Theory, 3rd Ed., Prentice Hall Inc., New York, 1996.

7. A.Benallal, « Etude des algorithmes des moindres carrés transversaux rapides et application à lôidentification de réponse impulsionnelles acoustiques», thèse de doctorat, université de rennes I, France, janvier 1989.

8. A. H. Sayed « Fundamentals of Adaptive Filtering » IEEE PRESS, A JOHN WILEY & SONS PUBLICATION, 2003.

9. Geert Rombouts « Adaptive filtering algorithms for acoustic echo and noise cancellation» dissertation, KATHOLIEKE UNIVERSITEIT LEUVEN, 2003.

10. A.sidi Moussa et D. Youssef Khoudja « segmentation automatique de la parole et application a lôannulation dôécho acoustique » thèse dôingénieur dôétat université Saad Dahleb Blida octobre 2000.

11. S. L. Gay and S. Tavathia, ñThe fast affine projection algorithm,ò in *Proc. Int. Conf. Acoust.. Speech, Signal Process.*, Detroit, MI, May 1995, pp. 3023ï 3026.

12. H. Ding, ñA stable fast affine projection adaptation algorithm suitable for low-cost processors,ò in Proc. Int. Conf. Acoust., Speech, Signal Process., Istanbul, Turkey, Jun. 2000, pp. I-360ï I-363.

13. H. Ding, ñAdaptive filtering using fast affine projection adaptation,ò U.S. Patent Applicat. No. 20060039458, Aug. 17, 2004, filed.

14. F. Albu, J. Kadlec, N. Coleman, and A. Fagan, ñThe Gauss-Seidel fast affine projection algorithmò in Proc. IEEE Workshop Signal Processing Systems (SIPS 2002), pp. 109 - 114, San Diego, U.S.A, October 2002.

15. F. Albu, and H.K. Kwan, ñCombined echo and noise cancellation based on Gauss- Seidel pseudo affine projection algorithmò, Proc. IEEE ISCAS 2004, Vancouver, Canada, pp. 505-508.

16. D.Lay ñ Linear Algebra And Its Apllivcations, Addison Wesley Publishing Company, 1994.

17. J. Benesty and T. Gaensler, ñA robust fast recursive least squares adaptive algorithm,ò in Proc. Int. Conf. Acoustics, Speech, Signal Processing (ICASSP), Salt Lake City, UT, 2001, pp. 3785ï3788.

18. J.M.Cioffi and T.Kailath, « Fast Recursive Least Square Transversal Filters for adaptive Filtering », IEEE Trans. On ASSP 32, N°2, pp.304 ï 337, Apr.1984.

19. H.Schutze, Z.Ren, « Numerical characteristics of Fast Least Squares Transversal Adaptation Algorithms - A comparative study », Signal processing N° 27, 1992.

20. Panagiotis P. Mavridis and George V. Moustakidiss « Simplified Newton-Type Adaptive Estimation Algorithms », IEEE Transaction on signal processing, Vol.44, NO .8, August 1996.

21. M.Djendi « Réduction de la complexité des algorithmes des moindres carrées transversaux rapides, application à lôannulation dôécho acoustique et implantation sur la carte DSP TMS320 C31 », Thèse de Magister université Saad Dahleb Blida, juillet 2000.

22. Philippe.GCiarlet « Introduction à lôanalyse numérique matricielle et à lôoptimisation », Dunod, 2007.

23. M.Djendi, M.Belhout, A.Guessoum « Three Prewhitened Versions of Pseudo Affine Projection for Acoustic Echo Cancellation» ICECS 2010, GRECE.

24. F. Bouteille, P. Scalart, M. Corzza, ñPseudo-APA: new soltion for adaptive identification,ò in Proc.Eurospeech, Budapest, Hungary, 1999, pp. 427-430.

25. S.Ben Jebara, M. J. Saidane, ñComparison of based adaptive predictive schemes for improvement of tracking randomly time-varying systemsò, in 5th Int. Conf. on Electronics, Circuits and Systems, Lisboa, Portugal, September 1998.

26. P. Scalart, F. Bouteille "on integrating speech coding functions into echo cancelling filters with decorrelating properties". In ICASSP, Orlando, USA, 2002.

27. K. Ozeki and T. Umeda, ñAn adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties,òElectronics and Communications in Japan, vol. 67-A, no. 5, 1984.

28. Lee,L, Park, Y.-C., Youn, D.-H ñ Robust pseudo affine projection algorithm with variable step-size ñ. Electronics Letters 44 (3), pp. 250-252, 2008.