



République Algérienne Démocratique et Populaire

MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE
SCIENTIFIQUE

UNIVERSITE DE BLIDA
INSTITUT D'ELECTRONIQUE

MEMOIRE

présenté par :
MOHAMED DJENDI

EN VUE DE L'OBTENTION DU DIPLOME
DE MAGISTER EN ELECTRONIQUE
OPTION : COMMUNICATION

THEME

**REDUCTION DE LA COMPLEXITE DE CALCUL DES
ALGORITHMES ADAPTATIFS ET IMPLANTATION
SUR LA CARTE OROS-AU32 (TMS320C31)**

devant le jury :

Mr A. GUESSOUM	<i>Professeur (Univ. Blida)</i>	Président
Mr D. BERKANI	<i>Professeur (ENP Alger)</i>	Examineur
Mr M. BENSEBTI	<i>maître de conférence (Univ. Blida)</i>	Examineur
Mr M. AREZKI	<i>maître assistant (Univ. Blida)</i>	Examineur
Mr A. BENALLAL	<i>maître de conférence (Univ. Blida)</i>	Rapporteur

BLIDA, ALGERIE JUIN 2000

Dédicaces

Je dédie ce modeste travail à :

- Ma mère et mon père pour leurs sacrifices et leurs encouragements ;
- Mes sœurs ;
- Mes frères ;
- A la petite siham que dieu la guérisse ;
- Toute ma famille ;
- Tous mes amis.

Mohamed DJENDI

Remerciements

- Ce mémoire a été élaboré au sein du laboratoire de l'acoustique des salles, institut d'électronique, université de Blida de septembre 1998 à février 2000.
- Au terme de cette étude, je tiens à présenter mes vifs remerciements à mon promoteur Monsieur Ahmed BENALLAL. Je lui exprime ici ma reconnaissance pour sa disponibilité en dépit de ses nombreuses responsabilités et d'avoir mis à ma disposition son logiciel de traitement du signal « ICHARA ».
- Je remercie monsieur A. GUESSOUM qui a donné une importance à mon travail et pour avoir accepté d'être président du jury.
- J'exprime toute ma reconnaissance à monsieur M. BENSEBTI pour l'aide qu'il m'a apportée en corrigeant certaines parties de ce travail et d'accepter d'être membre du jury.
- Je remercie monsieur D. BERKANI de l'école nationale polytechnique d'Alger (ENPA) qui m'a fait l'honneur d'être membre du jury.
- Un grand merci à monsieur M. AREZKI de l'USTB pour les nombreuses discussions que nous eûmes sur les aspects théoriques de l'adaptatif et de la stabilité numérique et pour avoir accepté d'être membre de jury.
- Je remercie également monsieur H. MELIANI pour les intéressantes discussions que nous avons eues sur les processeurs numériques.
- Mes remerciements s'adressent également aux collègues du laboratoire d'acoustique des salles de l'institut d'électronique pour leurs aides morale et matériels.
- Que toute personne ayant œuvré de près ou de loin à la réalisation de ce travail, par un simple conseil, un mot d'encouragement ou pour toute forme de contribution, trouve ici le témoignage de ma profonde reconnaissance.

Liste d'abréviation

- **EQM** : Erreur Quadratique Moyenne.
- **EQMN** : Erreur Quadratique Moyenne Normalisée.
- **LMS** : Least Mean Square Algorithm (algorithme du gradient stochastique).
- **NLMS** : Normalised Least Mean Square Algorithm (algorithme du gradient stochastique normalisé).
- **RLS** : Recursive Least square Algorithm (algorithme des Moindres carrés récursifs).
- **MCR** : Moindres carrés rapides.
- **FTF** : Fast Transversal Filter (Filtre transversal rapide).
- **FTF 7L** : Fast Transversal Filter Algorithm (algorithme du filtre transversal rapide de complexité 7L).
- **FTF8L** : Fast Transversal Filter Algorithm (algorithme du filtre transversal rapide numériquement stabilisé de complexité 8L).
- **FNTF12N2L** : Fast Newton Transversal Filter Algorithm (algorithme du filtre transversal de Newton rapide de complexité $12N+2L$).
- **FNS** : Simplified Fast Newton Algorithm (algorithme de Newton rapide simplifié).
- **BNLMS** : Bloc Normalised Least Mean square Algorithm (algorithme du gradient stochastique normalisé par blocs).
- **BFTF8L** : Bloc Fast Transversal Filter Algorithm (algorithme du filtre transversal rapide numériquement stabilisé par blocs).
- **BFNTF12N2L** : Bloc Fast Newton Transversal Filter (algorithme du filtre transversal de Newton rapide par blocs).
- **BFNS** : Simplified Bloc Fast Newton Algorithm (algorithme de Newton rapide simplifié par blocs).
- **DSP** : Digital Signal Processing (processeur de traitement numérique du signal).
- **PC** : Personnel Computer (ordinateur personnel)

SOMMAIRE

Introduction générale	1
-----------------------------	---

Chapitre I : Identification par les algorithmes de filtrage adaptatif

I.1 Introduction	3
I.2 Position du problème	4
I.3 Critères de comparaison	8
I.4 Complexité de calcul	8
I.5 Présentation des algorithmes	9
I.5.1 Algorithmes du gradient stochastique LMS et NLMS	9
I.5.2 Algorithmes des moindres carrés exacts	10
I.5.2.1 Algorithme des moindres carrés récurrents (RLS)	10
I.5.2.2 Algorithmes des moindres carrés transversaux rapides (FTF)	12
I.6 Algorithme FTF 7N	18
I.7 Stabilisation numérique de l'algorithme FTF7N	19
I.8 Initialisation de l'algorithme FTF 7N	19
I.9 Conclusion	20

Chapitre II : Réduction de la complexité des calculs des algorithmes d'identification adaptative

II.1 Introduction	21
II.2 Algorithme FTF12N2L	21
II.2.1 Initialisation de l'algorithme FTF12N2L	24
II.3 L'algorithme Fast Newton Simplifié (FNS)	25
II.4 Complexité des calculs	28
II.5 Algorithmes de filtrage adaptatif par blocs	29
II.5.1 Algorithme NLMS par blocs (Bloc-LMS)	29
II.5.2 Algorithmes (Bloc-FTF, Bloc-FTF12N2L et Bloc-FNS)	30
II.5.3 Stabilisation numérique des algorithmes FTF et FTF12N2L par blocs	31
II.6 Conclusion	31

Chapitre III : Etude comparative des algorithmes d'identification adaptative à complexité réduite

III.1 Introduction	32
III.2 Résultats de simulation	35

A- Cas stationnaire

III.2.1 Simulation des algorithmes de filtrage adaptatif à complexité réduite	35
III.2.1.1 Algorithme NLMS	35
III.2.1.2 Algorithme FTF8L	37

III.2.1.3	Algorithme FNTF12N2L.....	39
III.2.1.4	Algorithme FNS.....	42
III.2.2	Simulation des algorithmes de filtrage adaptatif par blocs à complexité réduite.....	46
III.2.2.1	Algorithme NLMS par bloc (BNLMS).....	46
III.2.2.2	Algorithme FTF8L par bloc (BFTF8L).....	47
III.2.2.3	Algorithme FTF12N2L par bloc (BFTF12N2L).....	49
III.2.2.4	Algorithme FNS par bloc (BFNS).....	50

B- Cas non stationnaire

III.2.3	Description des signaux de parole	52
III.2.4	Description des canaux de couplage acoustique	52
III.2.5	Simulation des algorithmes de filtrage adaptatif à complexité réduite.....	54
III.2.5.1	Algorithme FTF8L.....	54
III.2.5.2	Algorithme NLMS.....	56
III.2.5.3	Algorithme FTF12N2L.....	58
III.2.5.4	Algorithme FNS.....	60
III.2.6	Simulation des algorithmes de filtrage adaptatif par blocs à complexité réduite.....	62
III.2.6.1	Algorithme BNLMS.....	62
III.2.6.2	Algorithme BFTF8L.....	64
III.2.6.3	Algorithme BFTF12N2L.....	67
III.2.6.4	Algorithme BFNS.....	69
III.3	Comparaison des performances des algorithmes de filtrage adaptatif à complexité réduite.....	71
III.4	Conclusion.....	73

Chapitre IV : Implantation des algorithmes d'identification adaptative a complexité réduite sur la carte OROS-AU32

IV.1	Introduction.....	74
IV.2	Méthodologie de l'implantation sur la carte OROS-AU32.....	75
IV.3	Acquisition des données analogiques par la carte OROS-AU32.....	76
IV.4	Implantation temps différé.....	78
IV.5	Acquisition et calculs sur DSP.....	81
IV.6	Filtrage en temps réel sur PC.....	84
IV.7	Conclusion.....	87
	Conclusion générale et perspectives	88
	Bibliographie.....	90

Annexe A
 Annexe B
 Annexe C

INTRODUCTION GENERALE

Introduction générale

La parole moyen de communication privilégié entre les humains, constitue une grande part des messages transmis en télécommunication. Dans le système téléphonique classique, le message parlé est capté par le combiné puis transmis au locuteur lointain, le combiné téléphonique et le canal de transmission sont les seuls responsables de la qualité de la communication. Le phénomène d'écho qui peut apparaître dans ce système est la conséquence de la réflexion d'une partie du signal émis en un point où l'adaptation de ligne n'est pas parfaite. Le phénomène d'écho électrique devient très gênant pour des liaisons téléphoniques à grandes distances (communication par satellite par exemple).

Certains nouveaux services de télécommunication amènent à considérer le milieu acoustique (salle, cabine téléphonique, habitacle de voiture...) comme faisant part entière de la chaîne de communication. Ces nouveaux services correspondent à l'apparition des postes à haut-parleur, des postes mains libre et des systèmes de téléconférence (audio et visioconférence).

Des nouveaux problèmes sont apparus jusqu'alors négligeables ou inexistant grâce à la prise de son rapprochées dans le système téléphonique classique [1]. La réverbération, le bruit acoustique, et le phénomène d'écho acoustique.

Un traitement spécifique à la prise de son est nécessaire pour faire face à ces problèmes : déréverbération, débruitage de la parole et annulation d'écho acoustique. Ils peuvent être ramenés à l'identification du canal. L'objectif est de minimiser pour l'utilisateur la puissance des signaux perturbateurs (réverbération, bruit) sans dégrader le signal utile (parole). Dans notre étude on ne s'intéresse qu'à l'annulation d'écho acoustique. Pour réaliser efficacement cette identification, on doit prendre en compte les propriétés particulières des canaux acoustiques [1], [2] et des signaux traités.

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 bruits perturbateurs en sortie.

Dans ce contexte, plusieurs algorithmes de filtrage adaptatif à complexité réduite ont été développés dernièrement dans le but de réaliser efficacement l'identification des systèmes à réponses impulsionnelles longues. On cite à titre d'exemple la famille des algorithmes des moindres carrés transversaux rapides numériquement stabilisés, la famille des algorithmes de Newton rapides et la famille des algorithmes dite "par bloc". Ces différentes familles d'algorithmes permettent une importante réduction de la complexité des calculs. Cette réduction est importante lorsqu'on veut implanter ces algorithmes, dans le contexte d'annulation d'écho, sur des processeurs spécialisés tel que le TMS320C31 de Texas Instruments. Le travail présenté dans cette thèse s'inscrit dans cette ligne de recherche.

Le chapitre I définit le problème et présente les familles d'algorithmes d'adaptation de base : Les gradients stochastiques et les moindres carrés exacts. Une bonne partie est consacrée pour l'essentiel des équations qui permettent d'obtenir les algorithmes des moindres carrés transversaux rapides à facteur d'oubli exponentiel ainsi que sa version numériquement stable utilisée dans notre étude.

Au chapitre II, nous présentons les différentes versions d'algorithmes d'identification adaptative que nous avons proposées dans [3]. Ces nouvelles versions appartiennent à la famille dite "par bloc" qui permettent une importante réduction de la charge des calculs par rapport aux versions originales. D'autres modifications à d'autres algorithmes sont présentées dans ce chapitre.

Le chapitre III concerne la simulation et les tests des performances de tous les algorithmes d'identification adaptative que nous avons présentés dans le chapitre II. Cette étude comparative est divisée en deux parties. La première partie est consacrée au cas stationnaire c'est-à-dire on teste les algorithmes avec des signaux simples : bruit blanc et bruit USASI (qui est le spectre moyen de la parole). La deuxième partie est consacrée au cas non stationnaire où nous avons évalué les performances relatives des algorithmes avec la parole comme signal d'excitation. En fin on sélectionne les meilleurs algorithmes d'identification adaptative qui présentent les meilleurs rapports qualité/coût. Ces derniers vont être implantés sur le processeur spécialisé TMS320C31 de Texas Instruments de la carte OROS-AU32.

Au quatrième et dernier chapitre, on présente trois types d'implantations des algorithmes sélectionnés dans le chapitre III. Ces implantations ont permis l'exécution de l'identification adaptative avec les algorithmes sélectionnés en temps réel sur le TMS320C31 de la carte OROS-AU32 et sur le PC.

Chapitre I

IDENTIFICATION PAR LES ALGORITHMES DE FILTRAGE ADAPTATIF

I.1 Introduction

Les moyens de communications comme le téléphone et les réseaux informatique sont un élément essentiel des télécommunications à longue distance. 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 : expression du visage, gestes, langage du corps, environnement visible. Des documents avec des textes, des images ou autres données peuvent être utilisés pour venir appuyer le face à face, dans le but d'une meilleure compréhension [4].

L'audioconférence est une technique permettant d'établir la communication entre deux salles de réunion. La prise de son dans chacune des salles considère l'orateur comme la source utile. Malheureusement, tous les équipements mains libres classique [5] 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.

Le microphone mains libres n'est pas seulement exposé aux sons diffusés directement par l'interlocuteur proche. Une partie du son issue du haut-parleur, revient directement au microphone à travers l'air, en raison de la réflexion des ondes sonores par les objets environnants et puis est renvoyée à l'interlocuteur distant c'est ce qu'on a appelé écho acoustique.



Figure I.1 : Communication bidirectionnelle entre deux salles

Par définition [2], un canal acoustique de couplage représente la transformation du signal diffusée par le haut-parleur de sonorisation et capté par les microphones de prise de son. Par ailleurs, l'approximation de l'acoustique linéaire permet d'affirmer qu'un canal acoustique réalise un filtrage linéaire qui peut être étudié indépendamment des signaux. Il dépend essentiellement de la salle, des positions et des caractéristiques des sources et des récepteurs.

L'annulation d'écho acoustique est un problème qui se pose dans un grand nombre d'application (téléphonie main libre, téléconférence,...). 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. Dans ce qui suit, nous allons définir le problème et présenter les algorithmes d'identification adaptative de base ainsi que la notion de "complexité des calculs".

I.2 Position du problème

Le problème qui nous intéresse dans cette étude est l'estimation continue d'un signal s_t , éventuellement perturbé par un autre signal p_t , à partir d'un autre signal x_t , appelé signal de référence. La composante p_t est non mesurable, elle est incluse dans l'observation captée sur une voie dite principale : $y_t = s_t + p_t$. La composante p_t peut représenter le signal utile pour certaines applications. Par exemple dans le cas du débruitage de la parole avec voie de référence : p_t est la parole, s_t est le bruit perturbateur et x_t le bruit seul. Dans l'annulation d'écho acoustique, en absence ou en présence de la double parole, x_t est le signal reçu du locuteur lointain, y_t est le signal d'écho du locuteur lointain vers lui-même et p_t le bruit ambiant du milieu acoustique où le signal de la double parole du locuteur proche. Dans les deux cas, la restauration du signal utile est réalisée en soustrayant à l'observation y_t une estimée \hat{s}_t du signal s_t obtenus par un filtrage approprié du signal de référence x_t [5]. Le schéma de principe de ce filtrage est représenté sur la **figure I.2**, où les signaux sont connus aux instants d'échantillonnage t . La période d'échantillonnage est T_e , et l'on a : $x_t = x(t) = x(tT_e)$.

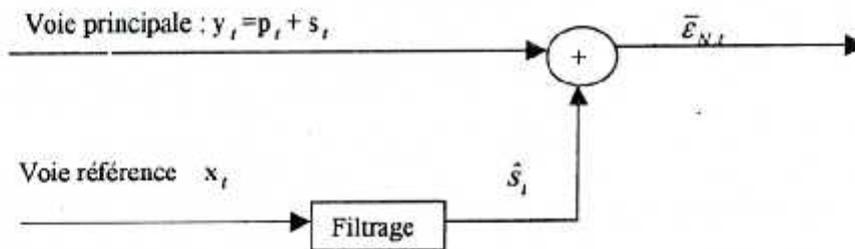


Figure I.2 : Schéma de principe d'un système d'annulation d'écho acoustique ou de débruitage de la parole.

Il faut noter que l'optimum en annulation d'écho acoustique est d'avoir la sortie $\bar{\varepsilon}_{N,t} = 0$ en absence de double parole ou du bruit perturbateur ($p_t = 0$), alors que cet optimum se transforme en $\bar{\varepsilon}_{N,t} = s_t$ dans le cas du débruitage de la parole.

Les hypothèses de base concernant le système de la **figure I.1** sont d'une part l'indépendance entre la composante p_t et les signaux s_t et x_t , et d'autre part la corrélation des signaux s_t et x_t .

D'après le schéma de la figure I.2, l'estimation du signal s_t à partir du signal x_t revient à trouver ou à estimer le filtre. Par conséquent, des applications telles que l'annulation d'écho acoustique ou le débruitage de la parole avec voie de référence bruit seul consistent essentiellement à identifier le filtre. Ce dernier doit en principe modéliser le trajet physique (canal acoustique de couplage en annulation d'écho) suivi par l'onde acoustique en se propageant de la voie dite de référence vers la voie dite principale. Le filtre modélisateur H_N est soit un filtre à réponse impulsionnelle finie (FIR), soit un filtre à réponse impulsionnelle infinie (IIR).

On se place d'emblée dans le cas de l'identification d'une réponse impulsionnelle acoustique où l'identification du filtre H_N par une réponse impulsionnelle finie a déjà fait ses preuves [4]. Par conséquent, on suppose que le signal s_t se déduit de x_t par un filtrage linéaire causal. Dans ces conditions, la sortie du système de la figure I.2 s'écrit :

$$\bar{e}_{N,t} = y_t - H_N^T X_{N,t} \quad (\text{I.1})$$

avec

$$H_N^T = [H_N^1, H_N^2, \dots, H_N^N] \quad (\text{I.1i})$$

$$X_N^T = [x_t, x_{t-1}, \dots, x_{t-N+1}] \quad (\text{I.1ii})$$

Où H_N désigne un vecteur d'ordre N contenant les coefficients de la réponse impulsionnelle finie et $X_{N,t}$ désigne un vecteur qui résume le passé du signal x_t sur une durée de N échantillons. L'exposant T désigne l'opérateur de transposition.

La solution optimale du problème de l'estimation de s_t à partir d'une combinaison linéaire x_t est fournie par la solution de l'équation de WIENER-HOPF, qui exprime la nullité du gradient de l'erreur quadratique moyenne en sortie

$$\nabla_H E\{\bar{e}_{N,t}^2\}_{H=H_{opt}} = 0 \quad (\text{I.2})$$

Où $E\{\}$ et ∇_H désignent respectivement la moyenne statistique et l'opérateur gradient. La solution est donnée par la relation suivante :

$$H_{N,opt} = R_{N,xx}^{-1} P_{N,xy} \quad (\text{I.3})$$

avec

$$P_{N,xy} = E\{X_{N,t} y_t\} \quad (\text{I.4})$$

$$R_{N,xx} = E\{X_{N,t}X_{N,t}^T\} \tag{I.5}$$

Où le vecteur $P_{N,xy}$ désigne l'intercorrélation entre l'observation y_t et le vecteur signal $X_{N,t}$ dans la mémoire du filtre ; et la matrice carré $R_{N,xx}$, d'ordre N , désigne la matrice d'autocorrélation du signal de référence dans le filtre, elle est symétrique et définie positive. Cette dernière propriété assure l'existence et l'unicité de la solution optimale. Cependant, dans la plupart des applications pratiques, les statistiques des signaux sont inconnues a-priori où les signaux sont souvent non-stationnaires. Par conséquent, les statistiques doivent être évaluées à partir des données recueillies aux entrées.

Ce type de problème amène à développer des méthodes de filtrage adaptatif où le filtre H_N change en fonction des informations reçues où l'erreur en sortie est utilisée pour la mise à jour des coefficients du filtre. Les coefficients du filtre sont ajustés à chaque instant t par un algorithme d'adaptation itératif. Une procédure itérative bien connue qui annule le gradient de l'erreur quadratique moyenne est donnée dans [6].

D'après [6], l'algorithme du gradient déterministe s'écrit :

$$H_{N,t} = H_{N,t-1} - \mu \left[\nabla_H E\{\bar{\epsilon}_{N,t}^2\} \right]_{H=H_{N,t-1}}$$

avec

$$\nabla_H E\{\bar{\epsilon}_{N,t}^2\} \Big|_{H=H_{N,t-1}} = 2R_{N,xx}H_{N,t-1} - 2P_{N,xy}$$

Où $P_{N,xy}$ et $R_{N,xx}$ sont respectivement données par les relations (I.4) et (I.5).
En utilisant l'égalité suivante :

$$R_{N,xx}H_{N,t-1} - P_{N,xy} = E\{X_{N,t}X_{N,t}^T H_{N,t-1} - X_{N,t}y_t\} = -E\{X_{N,t}\bar{\epsilon}_{N,t}\}$$

L'algorithme du gradient déterministe s'écrit :

$$H_{N,t} = H_{N,t-1} + 2\mu E\{\bar{\epsilon}_{N,t}X_{N,t}\} \tag{I.6}$$

Où μ est un pas d'adaptation, éventuellement variable, qui contrôle la stabilité et la vitesse de convergence de l'algorithme et $H_{N,t}$ est un vecteur dépendant du temps t donné par :

$$H_{N,t}^T = [H_{N,t}^1, H_{N,t}^2, \dots, H_{N,t}^N] \tag{I.6i}$$

et
$$\bar{\epsilon}_{N,t} = y_t - H_{N,t-1}^T X_{N,t} \tag{I.6ii}$$

Où $\bar{e}_{N,t}$ désigne l'erreur de filtrage à priori calculée avant la mise à jour du filtre.

Le principale rôle d'un algorithme adaptatif est d'ajuster un vecteur de paramètre $H_{N,t}$ pour un objectif bien défini (minimisation de l'EQM). Le schéma de principe d'un filtre adaptatif bouclé par un algorithme d'adaptation est représenté sur la **figure I.3**.

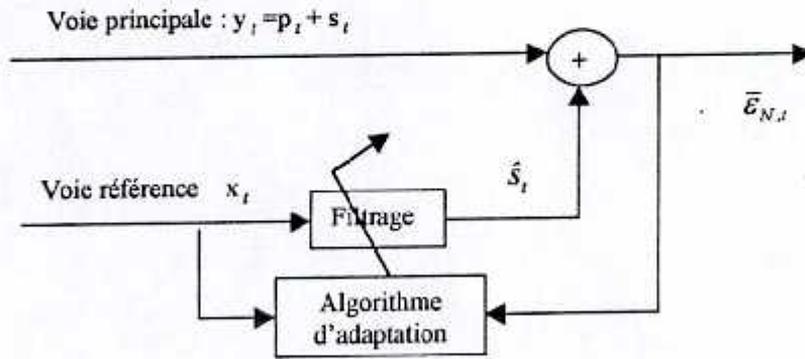


Figure I.3 : Schéma de principe d'un système d'annulation d'écho acoustique

Pour étudier le schéma de la **figure I.3**, diverses approches peuvent être utilisées. Elles correspondent d'une part à la structure du filtre adaptatif et d'autre part à la technique utilisée pour adapter les coefficients du filtre adaptatif. Les algorithmes adaptatifs ont été largement étudiés dans la littérature, il existe plusieurs ouvrages traitant des toutes sortes d'algorithmes adaptatifs [7],[8],[5],[9].

Par conséquent, la structure sera bien évidemment transversale. Le modèle de filtre transverse est donné par le schéma de la **figure I.4**.

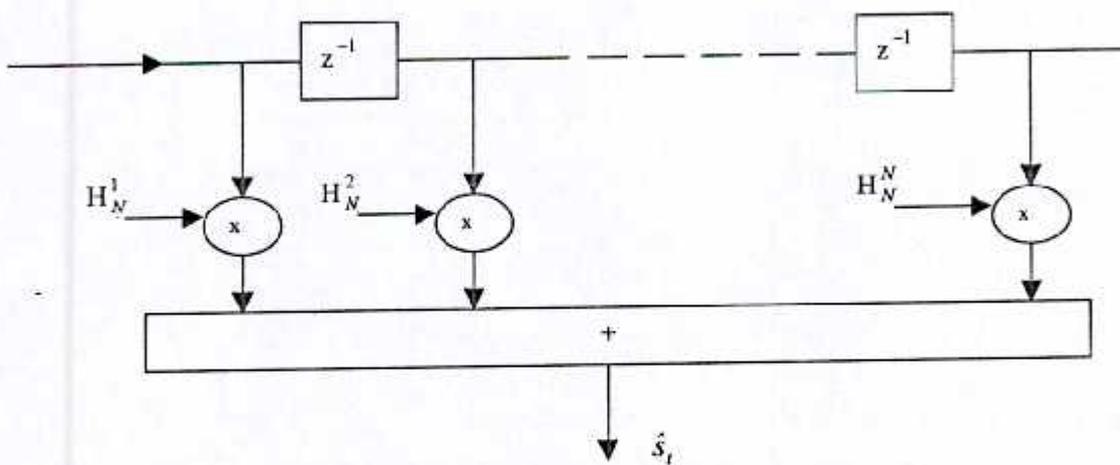


Figure I.4 : Schéma du modèle de filtre transverse

avec

$$\hat{s}_t = \sum_{i=1}^N H_N^i x_{t-i+1}$$

Le but de cette thèse est la maîtrise et le test des performances de ces nouvelles versions d'algorithmes dans des conditions d'utilisation réels (parole comme signal d'excitation).

Un autre objectif de cette thèse est la recherche de techniques qui permettent de réduire encore la complexité des calculs tout en s'approchant des performances des algorithmes exacts.

I.3 Critères de comparaison

Dans la comparaison des performances entre les algorithmes d'identification adaptative nous basons sur les critères objectifs suivants .

- L'erreur en régime permanent
- Vitesse de convergence en phase initiale
- Vitesse de convergence en régime permanent
- Influence de la taille du système sur les performances des algorithmes
- Influence de la parole comme signal d'excitation
- Influence du bruit additif sur le signal écho
- Stabilité numérique

Enfin, nous sélectionnons les algorithmes d'identification adaptative qui présentent le meilleur rapport qualité/coût.

I.4 Complexité de calcul

La complexité de calcul d'un algorithme d'identification adaptative est définis par le nombre de multiplications de divisions et d'additions que fait l'algorithme à chaque itération, récursive. La réduction de la complexité des calculs consiste à minimiser au maximum le nombre d'opération arithmétique (multiplication, division, addition) que fait l'algorithme à chaque itération recursive en gardant les mêmes performances originales de ces algorithmes, cette réduction est importante lorsqu'on veut faire fonctionner ces algorithmes en temps réel sur des processeurs spécilisés telque le TMS320C31 de TEXAS INSTRUMENTS.

Nous présentons dans le chapitre II une etude détaillée de tous les algorithmes d'identification adaptative, ainsi que des nouvelles versions de ces derniers qui réduisent la complexité des calculs[3].

Nous présentons aussi dans le chapitre IV l'implantation de quelques algorithmes d'identification adaptative sur le processeur du traitement numérique du signal TMS320C31 de TEXAS INSTRUMENTS, qui se sont avèrait meilleurs types d'algorithmes après une étude comparative détaillée des perforamances que ce soit avec des signaux stationnaires bruit blanc, bruit-Usasi,ou non stationnaires la parole, cette étude comparative est présentée dans le chapitre III.

1.5 Présentation des algorithmes de base

Nous allons présenter dans cette partie deux familles d'algorithmes : algorithmes du type gradient stochastique (LMS), algorithmes du type moindres carrés exacts (RLS) et un algorithme des moindres carrés transversaux rapides (FTF).

1.5.1 Algorithmes du gradient stochastique LMS et NLMS

L'idée des algorithmes de type gradient stochastique est de remplacer la moyenne statistique dans l'algorithme du gradient déterministe d'équation (1.6) par sa valeur instantanée. On obtient la relation suivante :

$$H_{N,t} = H_{N,t-1} + 2\mu \bar{e}_{N,t} X_{N,t} \quad (1.7)$$

où $\bar{e}_{N,t}$ est donnée par la relation (1.6ii).

D'après [10], la condition nécessaire et suffisante de convergence de l'algorithme (LMS) est :

$$0 < \mu < 1/\lambda_{\max} \quad (1.7i)$$

où λ_{\max} est la plus grande valeur propre de la matrice d'autocorrélation $R_{N,xx}$ donnée par (1.5). En pratique, on utilise souvent, pour le choix de la constante μ , une condition plus restrictive :

$$0 < \mu \leq \frac{1}{\text{Trace}(R_{N,xx})} = \frac{1}{N\sigma_x^2} \quad (1.7ii)$$

où $\text{Trace}(\cdot)$ représente la somme des éléments de la diagonale d'une matrice et σ_x^2 désigne l'énergie du signal d'entrée.

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 x_t ; il est donné par les équations suivantes:

- Filtrage de y_t

$$\bar{e}_{N,t} = y_t - H_{N,t-1}^T X_{N,t}$$

- Adaptation du filtre

$$H_{N,t} = H_{N,t-1} + \frac{\delta}{X_{N,t}^T X_{N,t}} \bar{e}_{N,t} X_{N,t}$$

où δ représente le pas d'adaptation de l'algorithme NLMS. La condition suffisante de convergence est :

$$0 < \delta < 2$$

Les avantages de l'algorithme NLMS sont : la faible complexité des calculs et la stabilité numérique. Son principal inconvénient est que le transitoire dépend de la dispersion spectrale du signal d'entrée ; la convergence peut devenir très lente dans le cas de la parole comme on le verra au chapitre III.

1.5.2 Algorithmes des moindres carrés exacts

Il existe Deux types d'algorithmes de la famille des moindres carrés exacts : Les algorithme récursif et les algorithmes transversaux rapides. Ces deux types d'algorithmes sont présentés dans les paragraphes suivants.

1.5.2.1 Algorithme des moindres carrés récursifs (RLS)

Dans la méthode des moindres carrés exacts, on vise à minimiser par rapport au vecteur des paramètres $H_{N,t}$, à chaque instant t , un critère défini sur les erreurs commises depuis l'instant initial. Ce critère est donné par [6] :

$$J_{N,t} = \sum_{i=1}^t \lambda^{t-i} (y_i - H_{N,t}^T X_{N,i})^2 \tag{I.8}$$

Où λ ($0 < \lambda \leq 1$) est un facteur d'oubli exponentiel qui permet à l'algorithme d'oublier le passé trop lointain et de poursuivre les non-stationnarités intervenant dans les signaux.

On suppose que les signaux x_i et y_i sont nuls avant l'instant initial $t = 0$; c'est à dire on se place dans le cas de la fenêtre antérieure.

La solution qui exprime la nullité du gradient de la fonctionnelle $J_{N,t}$

$$\nabla_H J_{N,t} = 0 \Leftrightarrow -2 \left[\sum_{i=1}^t \lambda^{t-i} X_{N,i} y_i - \sum_{i=1}^t \lambda^{t-i} X_{N,i} X_{N,i}^T H_{N,t} \right] = 0$$

est donnée par

$$R_{N,t} H_{N,t} = P_{N,t} \tag{I.9}$$

$$H_{N,t} = R_{N,t}^{-1} P_{N,t} \tag{I.10}$$

Où la matrice $R_{N,t}$ représente une matrice d'autocorrélation à court terme qui est donnée par la relation suivante

$$R_{N,t} = \sum_{i=1}^t \lambda^{t-i} X_{N,i} X_{N,i}^T \tag{I.11}$$

$$= \lambda R_{N,t-1} + X_{N,t} X_{N,t}^T \tag{I.11i}$$

Le vecteur $P_{N,t}$ désigne un vecteur d'intercorrélation à court terme donnée par la relation :

$$P_{N,t} = \sum_{i=1}^t \lambda^{t-i} X_{N,i} y_i \tag{I.12}$$

$$= \lambda P_{N,t-1} + X_{N,t} y_t \tag{I.12i}$$

En substituant les quantités (déduites de (I.11i) et (I.12i))

$$\lambda R_{N,t-1} = R_{N,t} - X_{N,t} X_{N,t}^T$$

$$\lambda P_{N,t-1} = P_{N,t} - X_{N,t} y_t$$

dans la version de (I.9) il vient :

$$\lambda R_{N,t-1} H_{N,t-1} = \lambda P_{N,t-1}$$

On obtient la solution équivalente à (I.10) mais sous une forme récursive :

$$H_{N,t} = H_{N,t-1} - C_{N,t} \bar{e}_{N,t} \tag{I.13}$$

avec

$$C_{N,t} = \begin{bmatrix} C_{N,t}^1 \\ C_{N,t}^2 \\ \cdot \\ \cdot \\ \cdot \\ C_{N,t}^N \end{bmatrix} = -R_{N,t}^{-1} X_{N,t} \tag{I.14}$$

$$\bar{e}_{N,t} = y_t - H_{N,t-1}^T X_{N,t}$$

Où le vecteur $C_{N,t}$ est appelé gain de Kalman.

L'algorithme (I.13) nécessite l'inversion d'une matrice carrée d'ordre N dont le coût de calcul est de l'ordre N^3 opérations arithmétiques par itération. L'algorithme des moindres carrés récursifs (RLS) résout ce problème, avec un nombre d'opérations arithmétiques proportionnel à N. En appliquant à la matrice $R_{N,t}$ le lemme d'inversion matricielle suivant [6] :

$$(A + UV^T)^{-1} = A^{-1} - \frac{A^{-1}UV^T A^{-1}}{1 + V^T A^{-1}U}$$

avec

$$A = \lambda R_{N,t-1} \quad ; \quad U = V = X_{N,t}$$

Il vient

$$R_{N,t}^{-1} = \lambda^{-1} \left[R_{N,t-1}^{-1} - \frac{R_{N,t-1}^{-1} X_{N,t} X_{N,t}^T R_{N,t-1}^{-1}}{\lambda + X_{N,t}^T R_{N,t-1}^{-1} X_{N,t}} \right] \tag{I.15}$$

Cet algorithme, appelé «algorithme des moindres carrés récursifs (RLS)», permet, en propageant une matrice carrée $N \times N$, de trouver itérativement la solution qui minimise le critère (I.9). Cette solution nécessite un coût en opération arithmétique proportionnel à N.

Cette dernière décennie, on s'est aperçu que la solution des moindres carrés pouvait être obtenue avec un coût opératoire proportionnel à N ; 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 d'un pré-fenêtrage des signaux d'entrée (les signaux sont supposés nuls avant l'instant initial).

On utilisera souvent la dénomination algorithme des moindres carrés rapides (MCR) ou Fast transversal filters (FTF) pour désigner ce type d'algorithme. Dans cette catégorie, on génère la solution des moindres carrés, à chaque instant t , pour un ordre donné N en propageant trois vecteurs d'ordre N (deux prédicteurs aller/retour et le gain de Kalman) au lieu d'une matrice. Les algorithmes des moindres carrés rapides les plus rapides connus appartiennent à cette catégorie.

1.5.2.2 Algorithmes des moindres carrés transversaux rapides (FTF)

Dans ce paragraphe nous décrivons brièvement l'essentiel des équations qui permettent d'obtenir les algorithmes des moindres carrés transversaux rapides à oubli exponentiel pour le cas de la fenêtre antérieure, faisant intervenir des erreurs a posteriori [11], [12].

Le but de cette méthode des moindres carrés est de trouver le vecteur $H_{N,t}$ qui minimise le critère (L.8). Une solution récursive à ce problème est donnée par l'algorithme (L.13). Cet algorithme utilise une erreur de filtrage a priori et un vecteur gain d'adaptation d'ordre N , qu'on a appelé gain de Kalman (L.14).

Si on substitue (L.11i) et (L.12i) dans (L.9), on obtient une autre forme récursive du vecteur $H_{N,t}$, strictement équivalente à (L.13), faisant intervenir une erreur de filtrage à posteriori et un gain d'adaptation appelé gain de Kalman dual :

$$H_{N,t} = H_{N,t-1} - \tilde{C}_{N,t} \varepsilon_{N,t} \quad (\text{L.16})$$

avec

$$\varepsilon_{N,t} = y_t - H_{N,t}^T X_{N,t} \quad (\text{L.17})$$

$$\tilde{C}_{N,t} = -\lambda^{-1} R_{N,t-1}^{-1} X_{N,t} \quad (\text{L.18})$$

Où $\varepsilon_{N,t}$ donnée par l'expression (L.17) désigne l'erreur de filtrage à posteriori (calculée après la mise à jour du filtre) et $\tilde{C}_{N,t}$ donnée par l'expression (L.18) désigne le gain de Kalman dual.

L'algorithme (L.16) ne peut être appliqué tel quel car l'adaptation et le filtrage ne peuvent être réalisés en même temps. Comme nous allons le voir par la suite, l'un des avantages des algorithmes MCR est que les erreurs à posteriori se calculent à partir des erreurs à priori avant l'opération d'adaptation.

Pour cela, on exploite certaines propriétés d'invariance par décalage du vecteur signal d'entrée étendu à l'ordre $N+1$:

$$X_{N+1,t} = \begin{bmatrix} x_t \\ X_{N,t-1} \end{bmatrix} \quad (\text{L.19i})$$

$$X_{N+1,t} = \begin{bmatrix} X_{N,t} \\ x_{t-N} \end{bmatrix} \quad (\text{L.19ii})$$

Ces deux formes du vecteur signal permettent de définir deux matrices d'autocorrélation d'ordre $(N+1)$ partitionnées.

La première qui correspond au vecteur (L.19i) s'écrit :

$$\begin{aligned} R_{N+1,t} &= \sum_{i=1}^t \lambda^{t-i} X_{N+1,i} X_{N+1,i}^T = \sum_{i=1}^t \lambda^{t-i} \begin{bmatrix} x_i \\ X_{N,i-1} \end{bmatrix} \begin{bmatrix} x_i & X_{N,i-1}^T \end{bmatrix} \\ &= \begin{bmatrix} P_{0,t}^a & P_{N,t}^{aT} \\ P_{N,t}^a & R_{N,t-1} \end{bmatrix} \end{aligned} \quad (\text{L.20i})$$

avec

$$\begin{aligned} P_{0,t}^a &= \sum_{i=1}^t \lambda^{t-i} x_i^2 \\ P_{N,t}^a &= \sum_{i=1}^t \lambda^{t-i} x_i X_{N,i} = \lambda P_{N,t-1}^a + x_t X_{N,t-1} \end{aligned}$$

Où l'exposant a , désigne une variable aller.

La deuxième, qui correspond au vecteur (L.19ii) s'écrit :

$$\begin{aligned} R_{N+1,t} &= \sum_{i=1}^t \lambda^{t-i} \begin{bmatrix} X_{N,i} \\ x_{i-N} \end{bmatrix} \begin{bmatrix} X_{N,i}^T & x_{i-N} \end{bmatrix} \\ &= \begin{bmatrix} R_{N,t} & P_{N,t}^{rT} \\ P_{N,t}^{rT} & P_{0,t}^r \end{bmatrix} \end{aligned} \quad (\text{L.20ii})$$

$$P_{0,t}^r = \sum_{i=1}^t \lambda^{t-i} x_{i-N}^2$$

avec

$$P_{N,t}^r = \sum_{i=1}^t \lambda^{t-i} x_{i-N} X_{N,i} = \lambda P_{N,t-1}^r + x_{t-N} X_{N,t}$$

Où l'exposant r , désigne une variable retour.

Le but des algorithmes MCR est de propager un vecteur gain. Le calcul du gain de Kalman (L.14) ou du gain de Kalman dual (L.18) 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 [11].
 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).

$$M^{-1} = \begin{bmatrix} 0 & 0 \\ 0 & M_4^{-1} \end{bmatrix} + \begin{bmatrix} I \\ -M_4^{-1}M_3 \end{bmatrix} (M_1 - M_2M_4^{-1}M_3)^{-1} \begin{bmatrix} I & -M_2M_4^{-1} \end{bmatrix} \quad (\text{I.21i})$$

ou

$$M^{-1} = \begin{bmatrix} M_1^{-1} & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} -M_1^{-1}M_2 \\ I \end{bmatrix} (M_4 - M_3M_1^{-1}M_2)^{-1} \begin{bmatrix} -M_3M_1^{-1} & I \end{bmatrix} \quad (\text{I.21ii})$$

En utilisant la forme (I.21i) pour inverser la matrice (I.20i) et la forme (I.21ii) pour inverser la matrice (I.20ii), on aura les expressions suivantes :

$$R_{N+1,t}^{-1} = \begin{bmatrix} 0 & 0 \\ 0 & R_{N,t-1}^{-1} \end{bmatrix} + \begin{bmatrix} 1 \\ -R_{N,t-1}^{-1}P_{N,t}^a \end{bmatrix} (P_{0,t}^a - P_{N,t}^{aT}R_{N,t-1}^{-1}P_{N,t}^a)^{-1} \begin{bmatrix} 1 & -P_{N,t}^{aT}R_{N,t-1}^{-1} \end{bmatrix} \quad (\text{I.22i})$$

$$R_{N+1,t}^{-1} = \begin{bmatrix} R_{N,t}^{-1} & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} -R_{N,t}^{-1}P_{N,t}^r \\ I \end{bmatrix} (P_{0,t}^r - P_{N,t}^{rT}R_{N,t}^{-1}P_{N,t}^r)^{-1} \begin{bmatrix} -P_{N,t}^{rT}R_{N,t}^{-1} & I \end{bmatrix} \quad (\text{I.22ii})$$

La forme (I.22i) fait apparaître un prédicteur aller optimal au sens des moindres carrés (le terme aller désigne, la modélisation de l'échantillon x_t par une combinaison linéaire de son passé) :

$$a_{N,t} = R_{N,t-1}^{-1}P_{N,t}^a \quad (\text{I.23})$$

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

$$J_{N,t}^a = \sum_{i=1}^t \lambda^{t-i} (x_i - a_{N,i}^T X_{N,i-1})^2 \quad (\text{I.24})$$

Les versions récursives du prédicteur aller (I.23) faisant intervenir soit le gain de Kalman (I.14), soit le gain de Kalman dual (I.18), s'obtiennent de la même façon que les versions récursives (I.13) et (I.16) du filtre transverse $H_{N,t}$:

$$a_{N,t} = a_{N,t-1} - C_{N,t-1} \bar{e}_{N,t} \quad (\text{I.25i})$$

$$a_{N,t} = a_{N,t-1} - \tilde{C}_{N,t-1} e_{N,t} \quad (\text{I.25ii})$$

Où $\bar{e}_{N,t}$ et $e_{N,t}$ désignent respectivement l'erreur de prédiction aller à priori et l'erreur de prédiction aller à posteriori, qui sont données par :

$$\bar{e}_{N,t} = x_t - a_{N,t-1}^T X_{N,t-1} \quad (\text{I.26i})$$

$$e_{N,t} = x_t - a_{N,t}^T X_{N,t-1} \quad (\text{I.26ii})$$

De la même façon, la forme (I.22ii) fait apparaître un prédicteur retour optimal au sens des moindres carrés (le terme retour désigne la modélisation de l'échantillon x_{t-N} par une combinaison linéaire des échantillons futurs) :

$$b_{N,t} = R_{N,t}^{-1} P_{N,t}^r \quad (\text{I.27})$$

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

$$J'_{N,t} = \sum_{i=1}^t \lambda^{t-i} (x_{t-N} - b_{N,t}^T X_{N,t})^2 \quad (\text{I.28})$$

Les versions récursives de $b_{N,t}$ s'obtiennent en substituant dans la solution (I.27) des versions récursives de $R_{N,t}$ et de $P_{N,t}$. Elles s'écrivent :

$$b_{N,t} = b_{N,t-1} - C_{N,t} \bar{r}_{N,t} \quad (\text{I.29i})$$

$$b_{N,t} = b_{N,t-1} - \tilde{C}_{N,t} r_{N,t} \quad (\text{I.29ii})$$

Où $\bar{r}_{N,t}$ et $r_{N,t}$ désignent respectivement l'erreur de prédiction retour a priori et l'erreur de prédiction retour a posteriori, qui sont données par :

$$\bar{r}_{N,t} = x_{t-N} - b_{N,t-1}^T X_{N,t} \quad (\text{I.30i})$$

$$r_{N,t} = x_{t-N} - b_{N,t}^T X_{N,t} \quad (\text{I.30ii})$$

Les termes entre parenthèses dans les expressions (I.22i) et (I.22ii) représentent respectivement les variances des erreurs de prédiction aller et retour (minima des critères (I.24) et (I.28)).

Leurs versions récursives s'écrivent [11] :

$$\alpha_{N,t} = \lambda \alpha_{N,t-1} + \bar{e}_{N,t} e_{N,t} \quad (\text{I.31})$$

$$\beta_{N,t} = \lambda \beta_{N,t-1} + \bar{r}_{N,t} r_{N,t} \quad (\text{I.32})$$

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

Les équations qui vont permettre de propager les gains de KALMAN d'ordre $(N+1)$ s'obtiennent en multipliant à droite les expressions (I.22i) et (I.22ii) par le vecteur $[-X_{N+1,t}]$ convenablement partitionné :

$$C_{N+1,t} = \begin{bmatrix} 0 \\ C_{N,t-1} \end{bmatrix} - \frac{e_{N,t}}{\alpha_{N,t}} \begin{bmatrix} 1 \\ -a_{N,t} \end{bmatrix} \quad (\text{I.33i})$$

$$C_{N+1,t} = \begin{bmatrix} C_{N,t} \\ 0 \end{bmatrix} - \frac{r_{N,t}}{\beta_{N,t}} \begin{bmatrix} -b_{N,t} \\ 1 \end{bmatrix} \quad (\text{I.33ii})$$

De la même façon en multipliant à droite les expressions (I.22i) et (I.22ii) prises aux instants $t-1$ par $[-\lambda^{-1}X_{N+1,t}]$ convenablement partitionné, on obtient les deux expressions du gain de KALMAN dual d'ordre $N+1$:

$$\tilde{C}_{N+1,t} = \begin{bmatrix} 0 \\ \tilde{C}_{N,t-1} \end{bmatrix} - \frac{\bar{e}_{N,t}}{\lambda\alpha_{N,t-1}} \begin{bmatrix} 1 \\ -a_{N,t-1} \end{bmatrix} \quad (\text{I.34i})$$

$$\tilde{C}_{N+1,t} = \begin{bmatrix} \tilde{C}_{N,t} \\ 0 \end{bmatrix} - \frac{\bar{r}_{N,t}}{\lambda\beta_{N,t-1}} \begin{bmatrix} -b_{N,t-1} \\ 1 \end{bmatrix} \quad (\text{I.34ii})$$

Il faut noter que les expressions (I.33) et (I.34) font apparaître des erreurs de prédiction qui peuvent être calculées, sans les relations de convolutions (I.26) et (I.30), si on dispose de la première ou de la $(N+1)$ ème composante du gain de Kalman ou du gain de Kalman dual :

$$e_{N,t} = -\alpha_{N,t} C_{N+1,t}^1 \quad (\text{I.35i})$$

$$\bar{e}_{N,t} = -\lambda\alpha_{N,t-1} \tilde{C}_{N+1,t}^1 \quad (\text{I.35ii})$$

$$r_{N,t} = -\beta_{N,t} C_{N+1,t}^{N+1} \quad (\text{I.36i})$$

$$\bar{r}_{N,t} = -\lambda\beta_{N,t-1} \tilde{C}_{N+1,t}^{N+1} \quad (\text{I.36ii})$$

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 l'erreur de filtrage a posteriori (I.17) et remplacer le filtre $H_{N,t}$ par son équation d'adaptation (I.13) il vient :

$$e_{N,t} = \bar{e}_{N,t} \left(1 + C_{N,t}^T X_{N,t} \right) \quad (\text{I.37})$$

Le terme entre parenthèse dans cette expression définit ce que l'on appelle la variable de vraisemblance

$$\gamma_{N,t} = 1 + C_{N,t}^T X_{N,t} = 1 - X_{N,t}^T R_{N,t}^{-1} X_{N,t} \quad (\text{I.38})$$

En théorie, la matrice $R_{N,t}$ et son inverse sont définis positifs. La variable $\gamma_{N,t}$ sera donc comprise entre 0 et 1 pour un fonctionnement normal de l'algorithme.

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

$$e_{N,t} = \gamma_{N,t-1} \bar{e}_{N,t} \tag{I.39}$$

$$\bar{r}_{N,t} = \gamma_{N,t} \bar{r}_{N,t} \tag{I.40}$$

D'autre part, les équations de mise à jour des vecteurs $H_{N,t}$, $\alpha_{N,t}$ et $b_{N,t}$ utilisant soit les erreurs a priori, soit les erreurs a posteriori, sont strictement équivalentes. Ceci conduit, en égalant deux d'entre elles, à la relation qui lie le gain de Kalman et le gain de Kalman dual :

$$C_{N,t} = \gamma_{N,t} \tilde{C}_{N,t} \tag{I.41}$$

En substituant cette relation dans (I.38) on obtient la relation suivante :

$$\gamma_{N,t} = \frac{1}{1 - \tilde{C}_{N,t}^T X_{N,t}} \tag{I.42}$$

Enfin, au lieu d'utiliser les définitions (I.38) ou (I.42) de la variable de vraisemblance, on utilise des versions plus rapides pour réduire la complexité. Ceci s'obtient en appliquant à la définition (I.38) les formes (I.33).

En multipliant (I.33i) à gauche par le vecteur $[x_t, X_{N,t-1}^T]$, on trouve l'expression de la variable de vraisemblance d'ordre N+1 :

$$\gamma_{N+1,t} = \frac{\lambda \alpha_{N,t-1}}{\alpha_{N,t}} \gamma_{N,t-1} \tag{I.43}$$

De la même manière, en multipliant (I.33ii) à gauche par le vecteur $[X_{N,t}^T, x_{t-N}]$ il vient :

$$\gamma_{N+1,t} = \frac{\lambda \beta_{N,t-1}}{\beta_{N,t}} \gamma_{N,t} \tag{I.44}$$

et si on pose

$$\theta_{N,t} = \frac{\lambda \beta_{N,t-1}}{\beta_{N,t}} \tag{I.45}$$

et en manipulant les expressions (I.32) et (I.36) on aura d'autres formes d'écritures

$$\begin{aligned} \theta_{N,t} &= 1 + \bar{r}_{N,t} C_{N+1,t}^{N+1} \\ &= 1 + \bar{r}_{N,t} \gamma_{N+1,t} \tilde{C}_{N+1,t}^{N+1} \end{aligned} \tag{I.45i}$$

On calcule alors la variable de vraisemblance d'ordre N par la relation suivante :

$$\gamma_{N,t} = \frac{\gamma_{N+1,t}}{\theta_{N,t}} \tag{I.46}$$

On montre aussi, en utilisant l'expression (I.15), la définition de la variable de vraisemblance (I.38), la propriété $\det[I+XY] = \det [I+YX]$ (det : déterminant d'une matrice) et les liens entre les déterminants des matrices $R_{N+1,t}$ et $R_{N,t}$ que la variable de vraisemblance peut se mettre sous la forme suivante [11] :

$$\begin{aligned} \gamma_{N,t} &= \lambda^N \frac{\det[R_{N,t-1}]}{\det[R_{N,t}]} \\ &= \lambda^N \frac{\beta_{N,t}}{\alpha_{N,t}} \end{aligned} \tag{I.47}$$

1.6 Algorithme FTF 7N

A partir du paragraphe précédent (I.5.2.2), 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 N. 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 à priori. Leur complexité arithmétique est de l'ordre de 7N multiplications par itération.

Une version obtenue par [12], appelé FTF (Fast Transversal Filter), propage la variable de vraisemblance $\gamma_{N,t}$ et le gain de Kalman dual. Cet algorithme FTF 7N est résumé par la Table I.1 (7N pour la complexité).

- Variables disponibles à l'instant t :	
$a_{N,t-1}; b_{N,t-1}; \tilde{C}_{N,t-1}; \gamma_{N,t-1}; \alpha_{N,t-1}; \beta_{N,t-1}; H_{N,t-1}$	
- Nouvelles informations : x_t et x_{t-N}	
- Modélisation de x_t et x_{t-N}	
$\bar{e}_{N,t} = x_t - a_{N,t-1}^T X_{N,t-1}$ $\alpha_{N,t} = \lambda \alpha_{N,t-1} + \gamma_{N,t-1} \bar{e}_{N,t}^2$ $\gamma_{N+1,t} = \frac{\lambda \alpha_{N,t-1}}{\alpha_{N,t}} \gamma_{N,t-1}$ $\tilde{C}_{N+1,t} = \begin{bmatrix} 0 \\ \tilde{C}_{N,t-1} \end{bmatrix} - \frac{\bar{e}_{N,t}}{\lambda \alpha_{N,t-1}} \begin{bmatrix} I \\ -a_{N,t-1} \end{bmatrix}$ $a_{N,t} = a_{N,t-1} - \bar{e}_{N,t} \gamma_{N,t-1} \tilde{C}_{N,t-1}$ $\bar{r}_{N,t} = -\lambda \beta_{N,t-1} \tilde{C}_{N+1,t}^T$ $\gamma_{N,t} = \frac{\gamma_{N+1,t}}{1 + \gamma_{N+1,t} \bar{r}_{N,t} \tilde{C}_{N+1,t}^T}$ $\begin{bmatrix} \tilde{C}_{N,t} \\ 0 \end{bmatrix} = \tilde{C}_{N+1,t} - \tilde{C}_{N+1,t}^T \begin{bmatrix} -b_{N,t-1} \\ I \end{bmatrix}$ $b_{N,t} = b_{N,t-1} - \bar{r}_{N,t} \gamma_{N,t} \tilde{C}_{N,t}$ $\beta_{N,t} = \lambda \beta_{N,t-1} + \gamma_{N,t} \bar{r}_{N,t}^2$	
- filtrage de y_t	
$\bar{e}_{N,t} = y_t - H_{N,t-1}^T X_{N,t}$ $H_{N,t} = H_{N,t-1} - \bar{e}_{N,t} \gamma_{N,t} \tilde{C}_{N,t}^T$	

Table I.1 : algorithme FTF 7N

1.7 Stabilisation numérique de l'algorithme FTF7N

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

Les erreurs numériques se propagent d'une manière non bornée au cours du temps, pour un facteur d'oubli inférieur à 1, ce qui conduit à une solution instable à plus au moins long terme. Des efforts importants ont été faits pour expliquer l'origine de cette instabilité numérique. Plusieurs méthodes ont été proposées pour combattre le problème de l'instabilité numérique [8], [13].

En ce qui concerne l'algorithme FTF7N, nous avons utilisé la sixième version de stabilisation numérique qui a été proposé dernièrement par M.AREZKI dans [14].

Cette dernière qui est basée 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 $\xi_{N,i}$ » est choisie et donnée par la relation suivante :

$$\xi_{N,i} = \bar{r}_{N,i}^c - [(1 - \mu_x) \bar{r}_{N,i}^{f_0} + \mu_x \bar{r}_{N,i}^{f_1}]$$

avec

$$\mu_x = 0.75 \quad [14]$$

$$\bar{r}_{N,i}^c = x_{i-N} - b_{N,i-1}^T X_{N,i}$$

$$\bar{r}_{N,i}^{f_0} = -\lambda \beta_{N,i-1} \tilde{C}_{N+1,i}^{N+1}$$

$$\bar{r}_{N,i}^{f_1} = -\lambda^{-N+1} \gamma_{N,i-1} \alpha_{N,i-1} \tilde{C}_{N+1,i}^{N+1}$$

Pour assurer la stabilité de l'algorithme, il faut satisfaire la condition suivante [14].

$$\lambda > 1 - \frac{1}{2N + 3.5} \quad (III.10)$$

1.8 Initialisation de l'algorithme FTF 7N

Dans le cas de la fenêtre antérieure, les algorithmes MCR supposent que les signaux x_i et y_i sont nuls avant l'instant initial $t = 0$. Par conséquent, les erreurs de prédiction retour et le prédicteur retour doivent être nuls avant l'instant $t = N$.

Citons ici la procédure d'initialisation la plus simple et la plus sûre du point de vue numérique

pour $t = 0$:

$$\begin{aligned}
 a_{N,t} &= \mathbf{0}_N^\dagger ; \tilde{C}_{N,t} = \mathbf{0}_N^\dagger ; \\
 \bar{e}_{N,t} &= \mathbf{0} ; \alpha_{N,t} = E_0 \lambda^N ; \beta_{N,t} = E_0 ; \gamma_{N,t} = 1 \\
 H_{N,t} &= H_{N,0}
 \end{aligned}$$

$H_{N,0}$: vecteur arbitraire

pour $t < N$:

$$\begin{aligned}
 b_{N,t} &= \mathbf{0}_N^\dagger \\
 \bar{r}_{N,t} &= \mathbf{0}
 \end{aligned}$$

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 l'algorithme sont étroitement liées au choix de la constante.

En pratique, il faut assurer le démarrage de l'algorithme on pourra choisir par exemple la constante E_0 qui vérifie l'inégalité suivante [12] :

$$E_0 \geq \frac{N}{100} \sigma_x^2$$

Où σ_x^2 est l'énergie du signal x_t .

Des valeurs de E_0 assurant le bon fonctionnement initial sont données dans [14].

1.9 Conclusion

Dans ce chapitre nous avons présenté le modèle de l'annulation de l'écho acoustique comme étant un problème d'identification adaptative d'un filtre à réponse impulsionnelle finie (RIF) modélisant le chemin d'écho (canal acoustique de couplage entre le haut-parleur et le microphone) suivi par l'onde acoustique, ainsi que deux algorithmes d'identification adaptative : L'algorithme du gradient stochastique normalisé (NLMS) de complexité $2N$ et l'algorithme des moindres carrés transversaux rapides (FTF) de complexité $7N$, aussi nous avons donné les différents critères objectifs sur lesquelles se base l'étude comparative faite dans le chapitre III entre les différentes versions d'algorithmes d'identification adaptative à complexité réduite [3].

Chapitre II

**REDUCTION DE LA COMPLEXITE
DES CALCULS DES ALGORITHMES
D'IDENTIFICATION ADAPTATIVE**

II.1 Introduction

L'identification en temps réel de systèmes linéaires à réponses impulsionnelles longues au moyen d'algorithmes de filtrage adaptatif est un problème très général que l'on rencontre dans de nombreuses applications de traitement du signal et de commandes des systèmes. Une application typique dans le domaine des télécommunications est l'annulation de l'écho acoustique en téléphonie main-libres dont un exemple typique est la téléconférence.

Plusieurs algorithmes de filtrages adaptatifs transverses ont été développés, et qui sont de performances supérieures à l'algorithme du gradient stochastique (LMS)[5][15], classiquement utilisé dans cette application en raison de sa faible complexité de calcul, bien que sa convergence soit lente sur les signaux de parole.

Les algorithmes de types moindres carrés transversaux rapides (Fast Transversal Filter : FTF)[5] convergent beaucoup plus vite que le gradient stochastique sur la parole, mais leur complexité est sensiblement plus élevée (de l'ordre de $8L$ pour les versions numériquement stabilisées).

Dans ce chapitre on présentera deux familles d'algorithmes d'identification adaptative, la famille des algorithmes de type NEWTON rapide est dérivée des algorithmes des moindres carrés transversaux rapides, et la famille d'algorithme appelée "par blocs".

Trois nouvelles versions d'algorithmes sont proposées, ces dernières sont [3] : La version de l'algorithme des moindres carrés transversaux rapides numériquement stabilisées par bloc (BFTF8L), la version de Newton rapide par bloc (BFTF12N2L) et la version d'algorithme de Newton rapide simplifié par bloc (BFNS). Ces nouvelles versions d'algorithmes sont inspirées de la version de l'algorithmes du gradient stochastique par bloc (BNLMS)

On propose aussi une nouvelle version de l'algorithme de Newton rapide FTF12N2L en utilisant la nouvelle méthode de stabilisation qui a été proposée dernièrement par M. AREZKI [14]. Entre outre, on présente une autre simplification de l'algorithme de Newton dite simplifiée (FNS) qui permet de réduire encore la complexité des calculs.

II.2 Algorithme FTF12N2L

Les algorithmes NLMS et FTF peuvent être considérés comme des versions stochastiques de l'algorithme de Newton qui cherche itérativement la solution H_{opt} de l'équation $\nabla J(t, H(t)) = 0_L$ (0_L est le vecteur nul de taille L). Le critère J minimisé est les moindres carrés moyens $E[\bar{\varepsilon}^2(t)]$ dans le cas du NLMS et les moindres carrés pondérés (avec préfenêtrage) dans le cas du FTF :

$$J(t) = \sum_{i=1}^t \lambda^{t-i} (y(i) - H^T(t)X(i))^2. \quad (II.1)$$

Dans le cas du NLMS le signal d'entrée $x(t)$ est un bruit blanc (c'est à dire un processus autoregressif (AR) d'ordre zéro), alors que dans le cas du FTF ce signal est un processus AR d'ordre L , on notera que cette dernière hypothèse est utilisée explicitement dans les formes

transverses rapides des moindres carrés qui mettent en œuvre des prédicteurs d'ordre L. L'idée de base de la nouvelle famille des algorithmes de fast Newton est que si le signal d'entrée peut être modélisé comme un processus AR d'ordre N inférieur à L, alors on peut construire une approximation optimale de la matrice de covariance $R_L(t)$ d'ordre L par un mécanisme d'extrapolation simple à partir des N premières valeurs de la covariance estimée de l'entrée.

Il en résulte que la mise à jour du gain de Kalman de taille L peut être obtenue à partir des prédicteurs d'ordre N correspondant à un algorithme FTF du même ordre N [16]. Ceci conduit à une importante réduction du volume des calculs lorsque N est très inférieur à L.

La matrice de covariance extrapolée est associée aux prédicteurs aller et retour d'ordre L dont N composantes seulement sont non nulles et égales à celles des prédicteurs associés au problème de prédiction optimale d'ordre N.

Les variances des erreurs de prédiction d'ordre L, sont identiques à celles d'ordre N. Ceci est traduit par les équations suivantes :

$$a_{L,t} = \begin{bmatrix} a_{N,t} \\ \mathbf{0}_{L-N} \end{bmatrix} \quad (\text{II.2a}) \quad \alpha_{L,t} = \alpha_{N,t} \quad (\text{II.2b})$$

$$b_{L,t} = \begin{bmatrix} \mathbf{0}_{L-N} \\ b_{N,t_D} \end{bmatrix} \quad (\text{II.3a}) \quad \beta_{L,t} = \beta_{N,t_D} \quad (\text{II.3b})$$

Où $\mathbf{0}_{L-N}$ est le vecteur nul de taille L-N.

On notera qu'un retard intervient dans l'observation des variables de prédiction retour, en effet $t_D = t - L + N$. Ceci est le caractère non stationnaire de la covariance qui est estimée à partir des données courantes.

On note que l'algorithme FTF propage le gain de Kalman dual défini par :

$$\tilde{C}_{L,t} = -\frac{1}{\lambda} \bar{R}_L^{-1} (t-1) X(t) = \frac{1}{\gamma_{L,t}} C_{L,t} \quad (\text{II.4})$$

Où $\gamma_{L,t}$ est la variable de vraisemblance.

Si on remplace les prédicteurs d'ordre L du FTF par les prédicteurs tronqués (II.2a) et (II.3a), on obtient les équations suivantes pour la mise à jour du gain de Kalman dual :

$$\tilde{C}_{L+1,t} = \begin{bmatrix} \mathbf{0} \\ \tilde{C}_{L,t-1} \end{bmatrix} - \begin{bmatrix} \tilde{S}_{N+1,t} \\ \mathbf{0}_{L-N} \end{bmatrix} \quad (\text{II.5})$$

$$\begin{bmatrix} \tilde{C}_{L,t} \\ \mathbf{0} \end{bmatrix} = \tilde{C}_{L,t-1} - \begin{bmatrix} \mathbf{0}_{L-N} \\ \tilde{U}_{N+1,t} \end{bmatrix} \quad (\text{II.6})$$

La mise à jour de la variable de vraisemblance extrapolée d'ordre L qui intervient dans l'équation de mise à jour du filtre par l'intermédiaire de (II.4) est obtenu par :

$$\delta_{L,t} = S_{L,t-1} + \tilde{S}_{N+1,t}^1 \tilde{e}_{N,t} + \tilde{U}_{N+1,t}^{N+1} \tilde{r}_{N,t_D} \quad (\text{II.7})$$

$$\gamma_{L,t} = \frac{1}{1 - \delta_{L,t}} \quad (\text{II.8})$$

Les vecteurs $\tilde{S}_{N+1,t}$ et \tilde{U}_{N+1,t_D} ainsi que les erreurs de prédiction directe et retour sont fournies par la partie prédiction de l'algorithme FTF d'ordre N.

Cela nous permet d'avoir un algorithme appelé FTF12N2L [16] qui comporte deux parties de prédictions d'ordre N extraites de l'algorithme FTF, l'une utilisant les données disponibles à l'instant t et l'autre les données retardées disponibles à l'instant t_D .

La liste des équations correspondantes à l'algorithme FNTF12N2L stabilisé est donnée par la table II.1. Nous avons modifié l'algorithme originale de [16] en lui appliquant la procédure de stabilisation d'écrite dans [14]. Cette dernière est basée sur la rétroaction des erreurs numériques dans le calcul des variables de prédiction rétrogrades.

Dans l'algorithme FTF12N2L nous avons utilisé la sixième version de stabilisation numérique [14], cette méthode de stabilisation est gérée par les équations suivantes.

La variable «indicateur de divergence $\xi_{N,t}$ » est choisie et donnée par la relation suivante :

$$\xi_{N,t} = \tilde{r}_{N,t}^e - \left[(1 - \mu_s) \tilde{r}_{N,t}^{f_c} + \mu_s \tilde{r}_{N,t}^{f_i} \right] \quad (\text{II.9})$$

avec

$$\mu_s = 0.75 \quad (\text{II.9a})$$

$$\tilde{r}_{N,t}^e = x_{t-N} - b_{N,t-1}^T X_{N,t} \quad (\text{II.9b})$$

$$\tilde{r}_{N,t}^{f_c} = -\lambda \beta_{N,t-1} \tilde{C}_{N+1,t}^{N+1} \quad (\text{II.9c})$$

$$\tilde{r}_{N,t}^{f_i} = -\lambda^{-N+1} \gamma_{N,t-1} \alpha_{N,t-1} \tilde{C}_{N+1,t}^{N+1} \quad (\text{II.9d})$$

Pour assurer la stabilité de l'algorithme, il faut satisfaire la condition suivante [14].

$$\lambda > 1 - \frac{1}{2N + 3.5} \quad (\text{II.10})$$

Ces équations qui permettent la stabilité de l'algorithme sont repérées dans la table II.1 par (S1, S2, S3)

Cet algorithme (FTF12N2L), nécessite $12N+2L$ multiplications (en négligeant les opérations sur les scalaires), soit un gain théorique en complexité de l'ordre d'un facteur 4 sur le FTF lorsque $N \ll L$, la complexité théorique est donc approximativement celle du NLMS. La complexité de calcul est donnée par le tableau récapitulatif n° II.1 (paragraphe II.4).

• **Partie de prédiction de l'algorithme FTF12N2L à l'instant t.**

Les équations repérées (S1, S2, S3) correspondent à la technique de stabilisation numérique [14]. On pose $X_{N,t}^T = [x_t, x_{t-1} \dots x_{t-N+1}]$; $\tilde{C}_N, \tilde{C}_{N+1}$: intermédiaire de calcul.

$\begin{aligned} \bar{e}_{N,t} &= x_t - a_{N,t-1}^T X_{N,t-1} \\ e_{N,t} &= \gamma_{N,t-1} \bar{e}_{N,t} \\ \alpha_{N,t} &= \lambda \alpha_{N,t-1} + \bar{e}_{N,t} e_{N,t} \\ \gamma_{N+1,t} &= \frac{\lambda \alpha_{N,t-1}}{\alpha_{N,t}} \gamma_{N,t-1} \\ \tilde{S}_{N+1,t} &= \frac{\bar{e}_{N,t}}{\lambda \alpha_{N,t-1}} \begin{bmatrix} 1 \\ -a_{N,t-1} \end{bmatrix} \\ \tilde{C}_{N+1,t} &= \begin{bmatrix} 0 \\ \tilde{C}_{N,t-1} \end{bmatrix} - \tilde{S}_{N+1,t} \\ a_{N,t} &= a_{N,t-1} - e_{N,t} \tilde{C}_{N,t-1} \\ \bar{r}_{N,t}^{conv} &= x_{t-N} - b_{N,t-1}^T X_{N,t} \quad (S1) \end{aligned}$	$\begin{aligned} \xi_{N,t} &= \bar{r}_{N,t}^{conv} + \lambda \beta_{N,t-1} \tilde{C}_{N+1,t}^{N+1} \quad (S2) \\ \bar{r}_{N,t} &= \bar{r}_{N,t}^{conv} + \xi_{N,t} \quad (S3) \\ \gamma_{N,t} &= \frac{\gamma_{N+1,t}}{1 + \gamma_{N+1,t} \bar{r}_{N,t} \tilde{C}_{N+1,t}^{N+1}} \\ r_{N,t} &= \gamma_{N,t} \bar{r}_{N,t} \\ \tilde{U}_{N+1,t} &= \tilde{C}_{N+1,t}^{N+1} \begin{bmatrix} -b_{N,t-1} \\ 1 \end{bmatrix} \\ \begin{bmatrix} \tilde{C}_{N,t} \\ 0 \end{bmatrix} &= \tilde{C}_{N+1,t} - \tilde{U}_{N+1,t} \\ b_{N,t} &= b_{N,t-1} - r_{N,t} \tilde{C}_{N,t} \\ \beta_{N,t} &= \lambda \beta_{N,t-1} + \bar{r}_{N,t} r_{N,t} \end{aligned}$
---	---

Table II.1 : Partie de prédiction de l'algorithme FTF12N2L à l'instant t.

II.2.1 Initialisation de l'algorithme FTF12N2L

L'initialisation de l'algorithme suit la procédure optimale adaptée au préfenêtrage. Les prédicteurs $a_{N,0}, b_{N,0}, a_{N,-L+N}, b_{N,-L+N}$ et le gain de KALMAN dual, $\tilde{C}_{N,-L+N}, \tilde{C}_{L,0}$ sont mis à zéro, tandis que les variables de vraisemblances, $\gamma_{N,-L+N}, \gamma_{L,t}$ sont mis à un.

La table II.2 résume l'initialisation des variances des erreurs de prédictions.

$\alpha_{L,0} = \lambda^L e_0$	$\beta_{L,0} = \lambda^{L-N} e_0$	$\alpha_{L,-L+N} = \lambda^N e_0$	$\beta_{N,-L+N} = e_0$
--------------------------------	-----------------------------------	-----------------------------------	------------------------

Table II.2 : Initialisation des variances des erreurs de prédiction

Où e_0 est une estimation a priori de la variance σ_x^2 de l'entrée.

II.3 Algorithme Fast Newton Simplifié (FNS)

L'algorithme de 'fast Newton simplifié' (FNS) [13] est le résultat de la combinaison de l'algorithme du gradient stochastique LMS avec l'algorithme de 'Fast Newton Transversal filters' FTF12N2L.

Cette combinaison a pour but d'exploiter les performances des deux types d'algorithmes, afin de donner un algorithme qui possède une complexité de calcul comparable à celle du gradient stochastique est une vitesse de convergence proche de celle de l'algorithme FTF12N2L.

Ceci est réalisé en utilisant la structure de l'algorithme LMS, dans la partie de prédiction de l'algorithme FTF12N2L, ce qui donne naissance à un nouvel algorithme appelé 'Fast Newton Simplifié FNS.

Les erreurs de prédictions directes et retour sont données par les équations suivantes :

$$e_{M,i}^f = x_i - A_{M,i-1}^T X_{M,i-1} \tag{II.11}$$

$$e_{M,i}^b = x_{i-M} - B_{M,i-1}^T X_{M,i} \tag{II.12}$$

Où $A_{M,i-1}$ et $B_{M,i-1}$ sont les deux prédicteurs aller et retour, et sont donnés par :

$$A_{M,i} = A_{M,i-1} + \mu^f e_{M,i}^f X_{M,i-1} \tag{II.13}$$

$$B_{M,i} = B_{M,i-1} + \mu^b e_{M,i}^b X_{M,i} \tag{II.14}$$

Les deux nouveaux paramètres μ^f et μ^b qui sont les pas d'adaptation des prédicteurs aller et retour sont sélectionnés par simulation.

Les deux parties de filtrage et de la mise à jour des coefficients du filtre transverse adaptatif sont données par les équations suivantes.

$$\varepsilon_{N,i} = d_i - H_{N,i-1}^T X_{N,i} \tag{II.15}$$

$$H_{N,i} = H_{N,i-1} + \mu \varepsilon_{N,i} \tilde{C}_{N,i} \tag{II.16}$$

avec :

$\varepsilon_{N,i}$: est l'erreur de filtrage adaptatif

d_i : est le signal désiré

$\tilde{C}_{N,i}$: est le gain de Kalman dual

μ : est le pas d'adaptation de la partie de filtrage

Le pas d'adaptation de la partie de filtrage est donné par :

$$\mu = \frac{2\sigma_{ms}^2}{\sigma_x^2 N + K\alpha_{M,i-N+M}^d} \tag{II.17}$$

où :

σ_x^2 : est la variance du signal d'entrée $x(n)$.

α_M^d : est la puissance de l'erreur de prédiction directe/indirecte d'ordre M des prédicteurs aller/retour.

K : est un paramètre de l'algorithme ($k=100$)[13].

σ_{mis}^2 : est le disadjustement des paramètres qui est donné par :

$$\sigma_{mis}^2 = \frac{\sigma_{ex}^2(\infty)}{\sigma_{min}^2} \quad (\text{II.18})$$

Où :

$\sigma_{ex}^2(\infty)$: est la variance de l'erreur de filtrage supplémentaire.

σ_{min}^2 : est la variance de l'erreur de filtrage minimale (erreur quadratique moyenne minimale).

Les nouveaux vecteurs de la prédiction aller/retour d'ordre $M+1$ sont donnés comme suit :

$$D_{M+1,t-1} = \begin{bmatrix} 1 \\ -A_{M,t-1} \end{bmatrix} \quad (\text{II.19})$$

$$D_{M+1,t-1} = \begin{bmatrix} -B_{M,t-1} \\ 1 \end{bmatrix} \quad (\text{II.20})$$

Les équations qui vont permettre de calculer le gain de Kalman sont données sous formes deux versions différentes, la première version utilise les deux prédicteurs aller et retour pour la mise à jour du gain de Kalman, cette dernière est donnée par la relation suivante.

Version 1 :

$$\begin{bmatrix} \tilde{C}_{N,t} \\ * \end{bmatrix} = \begin{bmatrix} x_t \\ C_{N,t-1} \end{bmatrix} + Ke_{M,t}^d \begin{bmatrix} D_{M+1,t-1} \\ 0_{N-M} \end{bmatrix} - Ke_{M,n_N}^d \begin{bmatrix} 0_{N-M} \\ D_{M+1,n_M-1} \end{bmatrix} \quad (\text{II.21})$$

Où

$$e_{M,t}^d = e_{M,t}^f$$

$$e_{M,n_M}^d = e_{M,t}^b$$

et

$$n_M = t - N + M$$

Une autre version de cet algorithme qui permet encore une fois la réduction de la complexité des calculs est la version deux. Cette dernière consiste à éliminer toute la partie de prédiction retour, en utilisant seulement le prédicteur aller et l'erreur de prédiction aller dans la mise à jour du gain de Kalman. Cette version est illustrée par la relation suivante :

Version 2 :

$$\begin{bmatrix} \tilde{C}_{N,t} \\ * \end{bmatrix} = \begin{bmatrix} x_t \\ C_{N,t-1} \end{bmatrix} + Ke_{M,t}^d \begin{bmatrix} D_{M+1,t-1} \\ 0_{N-M} \end{bmatrix} \quad (\text{II.22})$$

Les équations de l'algorithme FNS sont listées dans la **table II.3** suivante. La complexité de calcul est donnée par le tableau récapitulatif n° II.1 (**paragraphe II.4**).

<p><i>Variables à l'instant t-1:</i></p> <p>$W_{N,t-1}, \tilde{C}_{N,t-1}, X_{N,t-1}$</p> <p>$A_{M,t-1}, B_{M,t-1}, A_{M,t_M-1}, B_{M,t_M-1} \quad (t_M = t - N + M)$</p>			
<p><i>Nouvelle information:</i></p> <p>d_t, x_t</p>			
<p><i>Adaptation du prédicteur directe et indirecte en utilisant LMS :</i></p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; border-right: 1px solid black; padding: 5px;"> $e_{M,t}^f = x_t - A_{M,t-1}^T X_{M,t-1}$ $A_{M,t} = A_{M,t-1} + \mu^f e_{M,t}^f X_{M,t-1}$ $D_{M+1,t-1} = \begin{bmatrix} I \\ -A_{M,t-1} \end{bmatrix}$ $e_{M,t}^d = e_{M,t}^f$ </td> <td style="width: 50%; padding: 5px;"> $e_{M,t}^b = x_{t-M} - B_{M,t-1}^T X_{M,t}$ $B_{M,t} = B_{M,t-1} + \mu^b e_{M,t}^b X_{M,t}$ $D_{M+1,t-1} = \begin{bmatrix} -B_{M,t-1} \\ I \end{bmatrix}$ $e_{M,t}^d = e_{M,t}^b$ </td> </tr> </table>		$e_{M,t}^f = x_t - A_{M,t-1}^T X_{M,t-1}$ $A_{M,t} = A_{M,t-1} + \mu^f e_{M,t}^f X_{M,t-1}$ $D_{M+1,t-1} = \begin{bmatrix} I \\ -A_{M,t-1} \end{bmatrix}$ $e_{M,t}^d = e_{M,t}^f$	$e_{M,t}^b = x_{t-M} - B_{M,t-1}^T X_{M,t}$ $B_{M,t} = B_{M,t-1} + \mu^b e_{M,t}^b X_{M,t}$ $D_{M+1,t-1} = \begin{bmatrix} -B_{M,t-1} \\ I \end{bmatrix}$ $e_{M,t}^d = e_{M,t}^b$
$e_{M,t}^f = x_t - A_{M,t-1}^T X_{M,t-1}$ $A_{M,t} = A_{M,t-1} + \mu^f e_{M,t}^f X_{M,t-1}$ $D_{M+1,t-1} = \begin{bmatrix} I \\ -A_{M,t-1} \end{bmatrix}$ $e_{M,t}^d = e_{M,t}^f$	$e_{M,t}^b = x_{t-M} - B_{M,t-1}^T X_{M,t}$ $B_{M,t} = B_{M,t-1} + \mu^b e_{M,t}^b X_{M,t}$ $D_{M+1,t-1} = \begin{bmatrix} -B_{M,t-1} \\ I \end{bmatrix}$ $e_{M,t}^d = e_{M,t}^b$		
<p><i>calcul du gain:</i></p> <p><i>version 1:</i></p> $\begin{bmatrix} \tilde{C}_{N,t} \\ * \end{bmatrix} = \begin{bmatrix} x_t \\ C_{N,t-1} \end{bmatrix} + Ke_{M,t}^d \begin{bmatrix} D_{M+1,t-1} \\ 0_{N-M} \end{bmatrix} - Ke_{M,t_M}^d \begin{bmatrix} 0_{N-M} \\ D_{M+1,t_M-1} \end{bmatrix}$ <p><i>version 2:</i></p> $\begin{bmatrix} \tilde{C}_{N,t} \\ * \end{bmatrix} = \begin{bmatrix} x_t \\ C_{N,t-1} \end{bmatrix} + Ke_{M,t}^d \begin{bmatrix} D_{M+1,t-1} \\ 0_{N-M} \end{bmatrix}$			
<p><i>Partie de filtrage:</i></p> <p>$\varepsilon_{N,t} = d_t - H_{N,t-1}^T X_{N,t}$</p> <p>$H_{N,t} = H_{N,t-1} + \mu \varepsilon_{N,t} C_{N,t}$</p>			

Table II.3 : listing de l'algorithme "Fast Newton" simplifié.

II.4 Complexité des calculs

Le tableau II.1 suivant donne la complexité de calcul (nombre d'opération numérique : les multiplications et les divisions) des algorithmes d'identification adaptative suivants :

- Algorithme des moindres carrés rapide FTF,
- Algorithme de "Fast Newton" FNTF12N2L,
- Algorithme de "Fast Newton Simplifié" FNS,
- Algorithme du gradient stochastique normalisé NLMS.

Ces complexités des calculs sont données en fonction de N et de L où L représente la taille du filtre transversal et N représente la taille de la partie de prédiction.

(dans ce tableau une division est prise égale à 16multiplications).

Type d'algorithmes	Nombre de Multiplications	Nombre de Divisions	Complexité (MADPR)
FTF8L	8L	3	8L+48
FTF12N2L	2L+12N	7	2L+12N+112
FNS(version :1)	2L+6N	0	2L+6N
FNS(version :2)	2L+3N	0	2L+3N
NLMS	2L	1	2L+16

Tableau II.1 : complexité des calculs des algorithmes : FTF, FTF12N2L, FNS et NLMS.

Le tableau II.2 donne un exemple typique de la complexité des calculs des algorithmes d'identification adaptatif cités au-dessus par rapport à celle du NLMS pour une taille du filtre transverse égale à (L=256) et un ordre de prédiction égale à (N=16).

Types d'algorithmes	Nombre des Multiplications	Par rapport à /NLMS
FTF8L	2096	4
FTF12N2L	816	1.54
FNS(version :1)	608	1.15
FNS(version :2)	560	1.06
NLMS	528	1

Tableau II.2 : la complexité pour L= 256, N= 16

II.5 Algorithmes de filtrage adaptatif par blocs

Plusieurs familles d'algorithmes par blocs ont été développées avec comme principal objectif la réduction de la complexité des calculs. Le principe de base de ces algorithmes par blocs est l'actualisation du filtre adaptatif qui est effectuée une fois par bloc de taille M échantillons.

Nous présentons dans les paragraphes suivants, trois nouvelles versions d'algorithmes par bloc que nous avons publiés dans [3]. Ces nouvelles versions sont : la version de l'algorithme des moindres carrés transversaux rapides numériquement stabilisés par bloc BFTF8L, la version de l'algorithme de Newton rapide par bloc BFTF12N2L et la version de l'algorithme de Newton rapide simplifié par bloc BFNS. Ces trois nouvelles versions sont inspirées de l'algorithme du gradient stochastique par bloc BNLMS qu'on va le présenter dans le paragraphe suivant.

II.5.1 Algorithme NLMS par blocs (Bloc-NLMS)

L'algorithme Bloc-NLMS est construit à partir du NLMS en imposant que les coefficients du filtre adaptatif restent fixes sur chaque bloc de taille M . On définit la matrice d'échantillons (de taille $M \times L$) :

$$\underline{X}_t^T = [X_{t-M+1}, X_{t-M+2}, \dots, X_t] \quad (\text{II.23})$$

Le filtre H étant maintenu fixe entre les instants $t-M+1$ et t , le vecteur des M erreurs successives (différentes de celles du NLMS) est donné par :

$$\tilde{E}_t^T = [\tilde{e}_{t-M+1}, \tilde{e}_{t-M+2}, \dots, \tilde{e}_t] \quad (\text{II.24})$$

satisfait à :

$$\tilde{E}_t = Y_t - \underline{X}_t^T H_{t-M} \quad (\text{II.25})$$

avec :

$$Y_t^T = [y_{t-M+1}, y_{t-M+2}, \dots, y_t]. \quad (\text{II.26})$$

On impose le pas d'adaptation μ_B fixe sur tout le bloc, l'équation d'adaptation par bloc du filtre s'écrit :

$$H_t = H_{t-M} + \frac{\mu_B}{\sigma_x^2 M} \underline{X}_t^T \tilde{E}_t \quad (\text{II.27})$$

Où

σ_x^2 : représente l'énergie du signal d'entrée.

II.5.2 Algorithmes (Bloc-FTF, Bloc-FTF12N2L et Bloc-FNS)

De la même manière que le bloc-NLMS, l'adaptation des coefficients du filtre adaptatif se fera une fois par bloc de taille M échantillons consécutifs sauf que les équations d'adaptations des trois algorithmes par Bloc diffèrent de celles du Bloc-NLMS. Ces équations sont résumées dans le tableau II.3 suivant :[3].

Type d'algorithmes par Bloc	Equations d'adaptations du filtre transverse	
Bloc-FTF8L.	$H_t = H_{t-M} + \tilde{C}_{L,t} \tilde{E}_t$	(II.28)
Bloc-FTF12N2L	$H_t = H_{t-M} + \tilde{C}_{N,t} \tilde{E}_t$	(II.29)
Bloc-FNS	$H_t = H_{t-M} + \mu_B \tilde{E}_t \tilde{C}_{N,t}$	(II.30)

Tableau II.3 : équations d'adaptations du filtre transverse des trois algorithmes par blocs.

avec :

$\tilde{C}_{L,t}$: est le vecteur du gain de Kalman dual d'ordre L.

$\tilde{C}_{N,t}$: est le vecteur du gain de Kalman dual d'ordre N.

μ_B : est le pas d'adaptation de la partie de filtrage de l'algorithme Bloc-FNS.

\tilde{E}_t : est le vecteur des M erreurs successives.

Le tableau II.4 suivant donne la complexité des calculs de ces algorithmes. Ce tableau montre que ces versions d'algorithmes proposées permettent une importante réduction de la complexité des calculs par rapport à leurs versions originales qu'on peut les retrouver en mettant la taille du bloc M égale à 1.

Type d'algorithmes	Nombre de Multiplications	Nombre de Divisions	Complexité (MADPR)
BFTF8L	7L+L/M	3	7L+L/M+48
BFTF12N2L	L+12N+L/M	7	L+12N+L/M+112
BFNS(version :1)	L+6N+L/M	0	L+6N+L/M
BFNS(version :2)	L+3N+L/M	0	L+3N+L/M
BNLMS	L+L/M	1	L+16+L/M

Tableau II.4 : Complexité des calculs des algorithmes : BFTF8L, BFTF12N2L, BFNS et BNLMS.

Le tableau II.5 suivant donne un exemple de la complexité des calculs relative des algorithmes d'identification adaptative cité au-dessus par rapport à celle du NLMS et par rapport leurs versions originales pour une taille du filtre transverse égale à (L=256), un ordre de prédiction égale à (N=16) et une taille du bloc d'échantillons (M=16).

Type d'algorithmes	Complexité. (MADPR)	Par rapport à /NLMS	Amélioration en complexité
BFTF8L	1856	3.625	11.45 %
BFTF12N2L	576	1.125	29.41 %
BFNS(version : 1)	368	0.71875	39.47 %
BFNS(version : 2)	320	0.625	42.85 %
BNLMS	272	0.53125	48.48 %

Tableau II.5 : Complexités des calculs relatives des algorithmes par blocs :
BFTF8L, BFTF12N2L, BFNS et BNLMS.
 $L=256, N=16, M=16.$

II.5.3 Stabilisation numérique des algorithmes FTF et FTF12N2L par Bloc

La stabilisation numérique ainsi que l'initialisation de ces deux algorithmes restent la même que celles de leurs versions originales.

La seule différence réside au niveau de la constante d'initialisation E_0 qui change en fonction de la taille M du bloque d'échantillon dans lequel les coefficients du filtre adaptatif restent constants. La structure par bloc nous a obligé d'augmenter la valeur de la constante d'initialisation des variances de prédiction E_0 à chaque fois qu'on augmente dans la taille du bloc d'échantillons M . Des valeurs expérimentales correspondantes à E_0 pour un fonctionnement stable des algorithmes BFTF8L et BFTF12N2L sont montrées dans les tableaux contenant les résultats des simulations faites dans le chapitre III.

II.6 Conclusion

Nous avons présenté des algorithmes adaptatifs transverses récemment développés avec une complexité des calculs réduites pour l'identification des réponses impulsionnelles longues.

Trois nouvelles versions d'algorithmes appartenant à la famille dite "par bloc" sont proposées et présentées [3]. Ces algorithmes permettent une importante réduction de la complexité des calculs par rapport aux versions originales qu'on peut les retrouver avec une taille du bloc égale à 1. Ces algorithmes sont les moindres carrés transversaux rapides numériquement stabilisés par bloc BFTF8L, l'algorithme de Newton rapide par bloc BFTF12N2L et l'algorithme de Newton rapide simplifié par bloc BFNS. Ces algorithmes sont inspirés à partir de la structure par bloc de l'algorithme du gradient stochastique par bloc BNLMS.

On a proposé aussi une nouvelle forme de l'algorithme de Newton rapide FTF12N2L en lui appliquant une méthode de stabilisation qui c'est apparue dernièrement [14]. Une forme simplifiée de l'algorithme de Newton rapide FNS a été présentée dans le but de réduire encore la complexité des calculs, cette simplification est présentée sous formes de deux versions.

Ces différentes versions d'algorithmes à complexité réduite vont être testées et comparées entre elles afin d'évaluer leurs performances dans le contexte de l'annulation de l'écho acoustique et cela dans le cas stationnaire et non stationnaire. Cette étude comparative est présentée en détaille dans le chapitre III.

Chapitre III

**ETUDE COMPARATIVE DES
ALGORITHMES
D'IDENTIFICATION ADAPTATIVE
A COMPLEXITE REDUITE**

III.1 Introduction

Dans ce chapitre, nous allons tester et comparer tous les algorithmes d'identification adaptative à complexité réduite présentés en détail dans le chapitre II. Cette étude va nous permettre de choisir des algorithmes d'identification adaptative présentant les meilleurs rapports qualité/coût. Ces derniers vont être implantés sur la carte OROS-AU32 au chapitre IV.

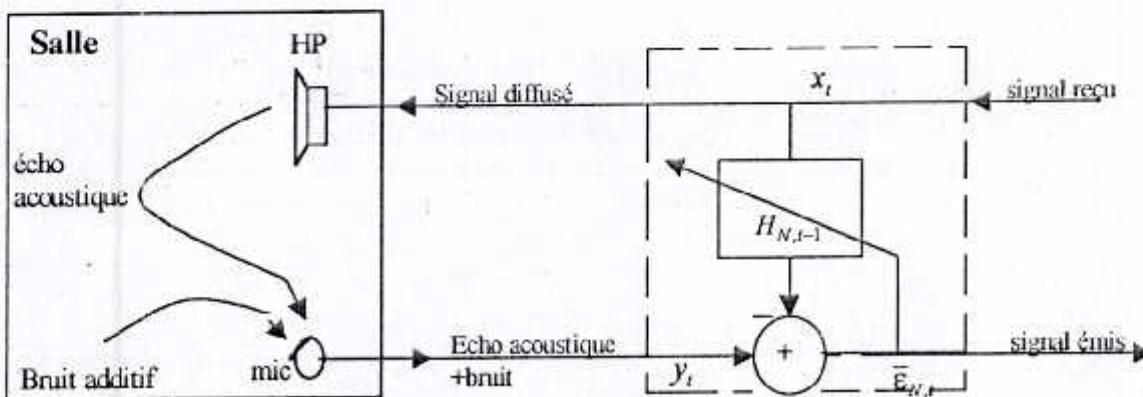
Pour les tests et les comparaisons, nous nous plaçons dans le cas de l'annulation d'écho acoustique dans un milieu acoustique sonorisé (salle). Cet écho acoustique a pour origine la propagation du son depuis le haut-parleur vers le microphone d'un terminal (poste « mains libres » par exemple) ; cette propagation inclut les réflexions sur les parois de la salle et sur les obstacles présents dans cette salle.

L'écho acoustique est gênant pour le correspondant distant dès que le retard de transmission dans le réseau de télécommunication dépasse 30 ms (communication longue distance, transmission par satellite) ; il doit donc être atténué le plus possible.

La figure III.1 représente l'annuleur d'écho acoustique et la configuration du milieu acoustique. Le rôle des algorithmes d'identification du chapitre II est de fournir à ce système le filtre transverse adaptatif H estimé à partir des signaux x et y . Le signal y capté par le microphone de prise de son peut éventuellement être perturbé par un bruit additif.

Les principales contraintes d'un tel système sont :

- Taille du canal acoustique : elle varie de 128 points pour les petites salles (cabine téléphonique) jusqu'à environ 5000 points pour les salles de tailles importantes (théâtre, salle cinéma).
- Canal acoustique variable : les mouvements d'objets (fenêtre) ou de personnes dans la salle rendent le canal acoustique non-stationnaire.
- Parole : le signal d'identification de l'identification x (parole) n'est pas un bon signal d'excitation pour l'identification de systèmes. L'identification utilise généralement un bruit blanc comme signal d'excitation.



HP : Haut-parleur
mic : microphone

Figure III.1 : Modèle de terminal audio avec écho acoustique.
La partie encadrée schématise le principe de l'annulation d'écho.

Dans cette étude, on se place dans le cas d'un canal acoustique stationnaire. Pour simuler le milieu acoustique, nous disposons de trois réponses impulsionnelles de couplage acoustique mesurées en régime stationnaire (Figure III.2). La première est une réponse impulsionnelle mesurée dans l'habitacle d'une voiture, la deuxième dans une salle de visioconférence et la troisième dans une salle d'audioconférence. Ces trois réponses impulsionnelles contiennent respectivement 1024, 4096 et 8192 points.

Cette étude comparative est divisée en deux parties, dans la première partie on présentera les résultats obtenus dans le cas d'un signal d'entrée stationnaire (bruits), alors que la deuxième partie sera consacrée au cas d'un signal d'entrée non stationnaire (parole). Les performances auxquelles nous nous intéressons dans cette étude sont

- Le niveau de l'erreur de filtrage en sortie ou l'erreur quadratique moyenne (EQM).
- L'effet de la taille du filtre transversale.
- L'effet du bruit additif sur l'erreur en sortie.
- La vitesse de convergence des algorithmes.
- La stabilité numérique.

Par stabilité numérique, nous entendons un fonctionnement à long terme des algorithmes de type, FTF rapide et celle de la famille de Newton rapide [3], pendant lequel les variables restent proches de leurs valeurs optimales.

L'indice de performance est l'évolution temporelle de l'erreur de filtrage normalisée, par celles du signal à modéliser (ou signal d'écho), exprimé en (dB), il est donné par :

$$EQMN(t) = 10 \log \left(\frac{\sigma_{e_{N,t}}^2}{\sigma_{y_t}^2} \right) \quad (\text{III.1})$$

où σ symbolise une moyenne temporelle de L échantillons consécutifs, et $\log(\cdot)$ représente le logarithme à base 10, avec :

$$\sigma_{e_{N,t}}^2 = \frac{1}{L} \sum_{i=1}^L \bar{e}_{N,i}^2$$

$$\sigma_{y_t}^2 = \frac{1}{L} \sum_{i=1}^L y_i^2$$

Avec le signal de parole comme signal d'entrée, l'indice de performance (III.1) privilégie les zones du signal de forte énergie. Il n'est pas très significatif dans les zones de très faible énergie (silences) où l'erreur de filtrage et le signal ont des niveaux comparables.

Pour cette raison on utilise dans la suite, le critère qui donne l'évolution temporelle de l'énergie de filtrage à priori donnée par :

$$EQM(t) = 10 \log(\sigma_{N,t}^2)$$

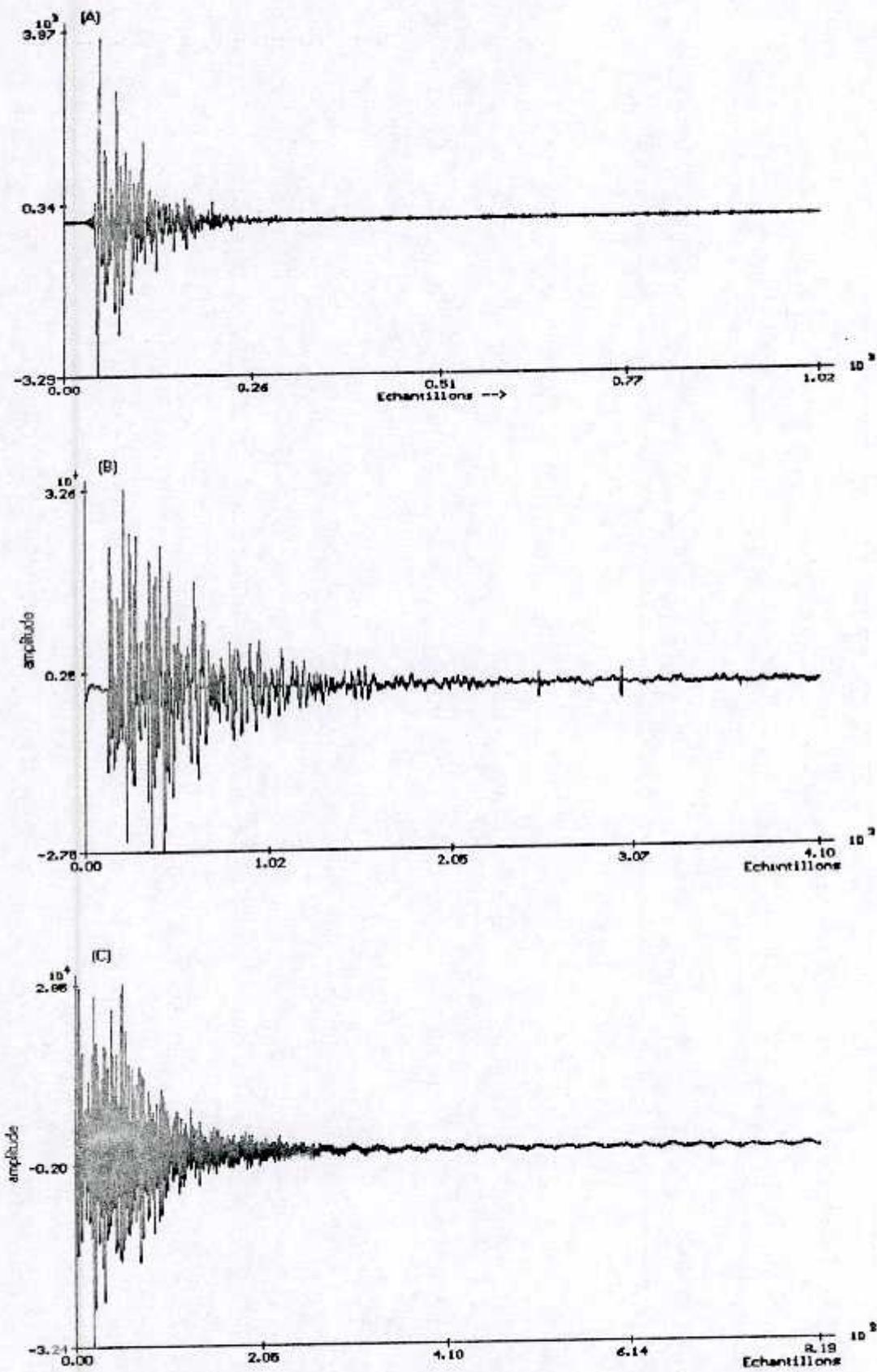


Figure III.2 : Réponses impulsionnelles de canaux de couplage acoustiques
 (A) habitacle de voiture
 (B) salle de visioconférence
 (C) salle d'audioconférence

III.2 Résultats de simulation

A - cas stationnaire

III.2.1 Simulation des algorithmes de filtrage adaptatif à complexité réduite

Les signaux x_i utilisés dans cette partie sont soit un bruit blanc gaussien, soit un bruit stationnaire à spectre moyen de la parole (bruit USASI). Le premier sert surtout à vérifier les conditions de la stabilité numérique obtenues, le deuxième est plus intéressant à cause de son spectre, 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. Ces signaux sont échantillonnés à la fréquence $f_e=16\text{kHz}$.

Le signal de sortie y_i est le signal x_i transformé par le canal acoustique de couplage d'un habitacle de voiture (section A de la figure III.2) de taille réelle $L=1024$ points. Nous avons utilisé juste les 256 premiers points de ce canal pour la constitution des signaux de tests.

III.2.1.1 Algorithme NLMS

Dans cette partie on va tester l'algorithme NLMS avec des signaux simples qui ont été décrits dans le paragraphe précédent. les paramètres de cette simulation sont :

L : taille du filtre adaptatif.

μ : pas d'adaptation de l'algorithme.

E_0 : énergie d'initialisation minimale de l'algorithme.

• Effet de la taille du filtre adaptatif sur l'algorithme NLMS

Les résultats de cette simulation sont présentés sur le tableau III.1 et la figure III.3

Le tableau III.1 montre que l'identification de la réponse impulsionnelle d'un système donné nécessite que la taille du filtre adaptative transverse soit égale ou supérieure (taille surestimée) à la taille réelle du système à identifier. Aussi on remarque une dégradation de l'énergie de l'erreur de filtrage normalisée (EQMN) lorsqu'on a à l'entrée un bruit USASI, ce qui montre que la vitesse de l'adaptation de l'algorithme dépend de la nature du signal d'excitation.

L	EQMN(dB)	
	Signal test 1	Signal test 2
32	-1.25	-2.15
64	-4.21	-5.35
128	-14.25	-16.2
256	-86.28	-87.0
512	-86.14	-69.5
1024	-86.15	-46.9

Tableau III.1 : Effet de la taille du filtre adaptatif sur l'algorithme, $\mu = 1, E_0 = 1$.

Signal test 1 : bruit blanc.

Signal test 2 : bruit USASI.

La figure III.3 correspond au cas où il n'y a pas de bruit ajouté en sortie. Pour tester la vitesse de convergence de l'algorithme NLMS en régime initial et en régime permanent, nous avons simulé un non-stationnarité en régime permanent en changeant l'amplification de la salle. On remarque que la vitesse de convergence de cet algorithme est plus importante lorsqu'on identifie des systèmes de taille faible, ce qui montre donc, que la vitesse de convergence de l'algorithme NLMS est inversement proportionnelle à la taille du filtre transverse.

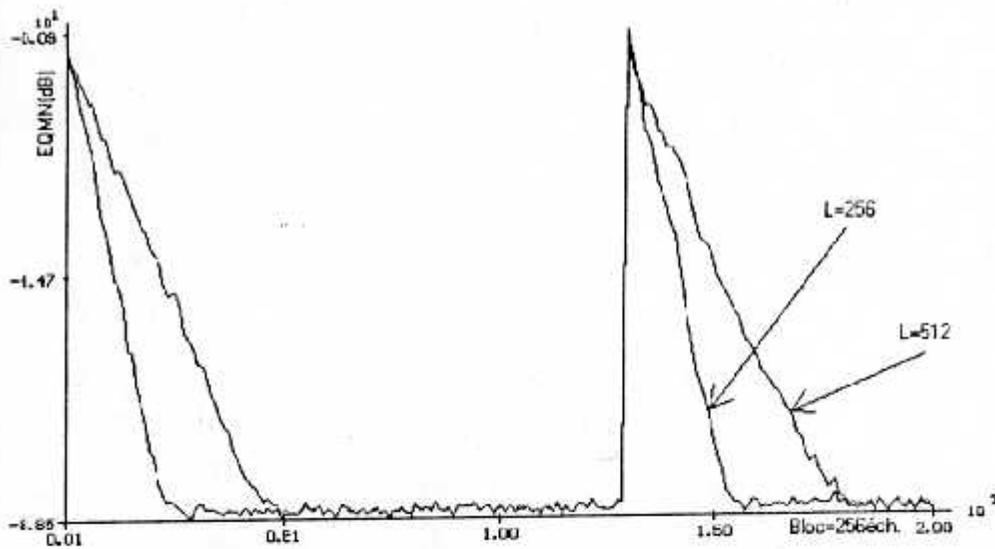


Figure III.3 : convergence du NLMS pour $L = 256, L = 512$.
Signal d'entrée : bruit blanc.

• Effet du bruit additif sur les performances de l'algorithme NLMS

En présence du bruit additif le signal écho est décrit comme suit :

$$y(n) = h(n) * x(n) + b(n) \quad (\text{III.2})$$

Où $b(n)$ est un bruit blanc.

Le RSB entre le signal $y(n)$ et le bruit additif $b(n)$ est calculé suivant la formule suivante :

$$RSB_y(y/b) = \frac{\sum_{i=0}^{\infty} y^2(i)}{\sum_{i=0}^{\infty} b^2(i)} \quad (\text{III.3})$$

Où la taille du filtre $L = 256$.

L'identification de $h(n)$ à partir des signaux $y(n)$ et $x(n)$ a donné les résultats du tableau III.2 et de la figure III.4.

Le tableau III.2 montre bien une dégradation importante de l'énergie de l'erreur de filtrage normalisée (EQMN) de l'algorithme NLMS, cette dernière (EQMN) reste de l'ordre de grandeur du bruit ajouté si le RSB est supérieure à 30dB, dans le cas contraire, l'algorithme est fortement perturbé et on peut même dire qu'il ne converge pas.

La figure III.4 correspond à un rapport signal sur bruit de l'ordre de 40dB en sortie, on note une dégradation importante de l'erreur asymptotique normalisée de l'algorithme qui reste de l'ordre de grandeurs du bruit ajouté en sortie.

$RSB_y (y/b)dB$	$EQMN(dB)$
60	-59.05
40	-40.50
30	-30.10
20	-7.00
10	-6.90

Tableau III.2 : Effet du bruit additif sur l'algorithme
Signal d'entrée : bruit USASI bruité. $L = 256, \mu = 0.5$

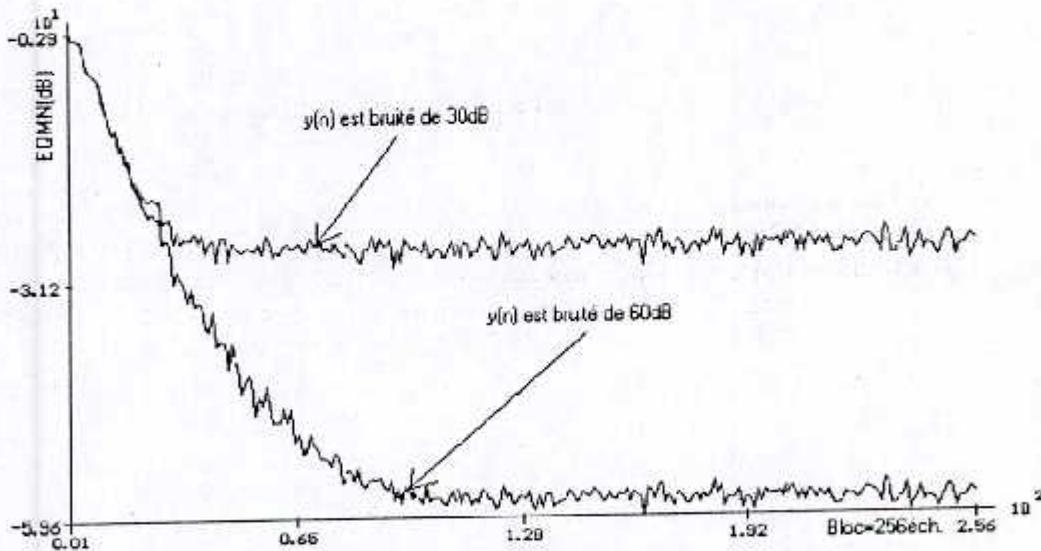


Figure III.4 : Convergence de l'algorithme NLMS dans le cas de la sortie bruité.

III.2.1.2 Algorithme FTF8L

Les paramètres de cette simulation sont :

- L : taille du filtre transverse.
- λ : facteur d'oubli exponentiel.
- E_0 : constante d'énergie minimale.

• Effet de la taille du filtre adaptatif sur l'algorithme

Cette simulation a permis d'avoir les résultats classés dans le tableau III.3. Ces résultats montrent bien que l'identification des systèmes avec l'algorithme FTF8L nécessite que la taille du filtre transverse soit égale ou supérieure (surestimée) à la taille réelle du système à identifier. On remarque aussi que l'algorithme a donné pratiquement les mêmes résultats de l'énergie de l'erreur de filtrage normalisée (EQMN) en utilisant deux signaux de tests qui sont différents entre eux, ce qui montre que la vitesse de l'adaptation de cet algorithme est indépendante de la nature du signal d'entrée. Cela est la propriété principale qui caractérise la famille des algorithmes des moindres carrés rapides.

Pour tester la vitesse de convergence de l'algorithme FTF8L en régime initial et en régime permanent nous avons procédé de la même manière que précédemment. Les résultats sont présentés sur la figure III.5. On note que cet algorithme a une très grande vitesse de convergence en phase initiale et même en régime permanent, ce qui lui permet d'atteindre l'optimum juste après 550 itérations environ (30ms) pour le cas du filtre de taille ($L=256$). Cette propriété importante est due essentiellement à l'indépendance du transitoire (vitesse de convergence) de cet algorithme de la nature du signal d'excitation.

Paramètres de l'algorithme.			EQMN (dB)	
L	λ	E_0	Signal test 1	Signal test 2
32	0.9853	0.1	-1.8	-1.87
64	0.9924	0.1	-2.95	-2.53
128	0.9965	0.1	-14.58	-13.51
256	0.9983	0.5	-88.32	-88.38
512	0.9992	1.0	-87.77	-87.12
1024	0.9998	1.0	-88.19	-88.16

Tableau III.3 : Effet de la taille du filtre adaptatif sur l'algorithme
Signal test 1 : bruit blanc, Signal test 2 : bruit USASI

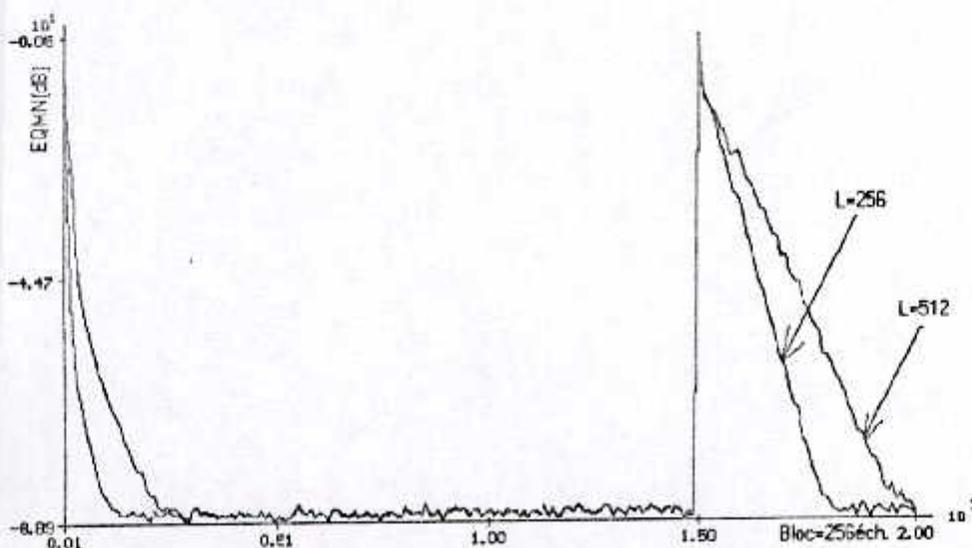


Figure III.5 : Convergence du FTF8L pour : $L = 256$ et 512
Signal d'entrée : bruit blanc.

• Effet du bruit additif sur l'algorithme

Nous avons utilisé les mêmes signaux de tests des simulations précédentes, cela nous a permis d'obtenir les résultats des différents RSB qui sont donnés le tableau III.4. la figure III.6 représente le cas d'un RSB=60dB et le cas sans bruit. On observe une dégradation importante de la performance asymptotique du FTF8L, mais l'erreur asymptotique normalisée de l'algorithme reste de l'ordre de grandeur du bruit rajouté à la sortie.

$RSB(y/b)dB$	$EQMN(dB)$
10	-7.05
20	-7.00
30	-31.51
40	-41.70
60	-61.00

Tableau III.4 : Effet du bruit additif sur l'algorithme
Signal d'entrée : bruit USASI bruité. $L = 256$

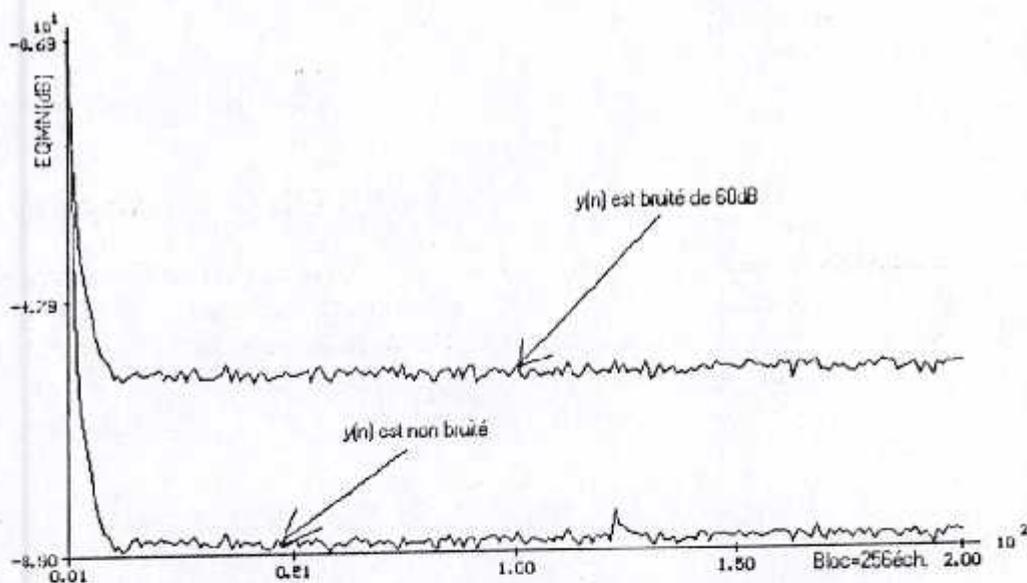


Figure III.6 : bruitage de $y(n)$. Un autre signal qui est non bruité est montré en référence.

III.2.1.3 Algorithme FNTF12N2L

Les paramètres de cette simulation sont :

L : La taille du filtre

N : La taille des prédicteurs aller et retour :

λ : Facteur d'oubli exponentiel

E_0 : Constante d'initialisation minimale (variance minimale)

• Effet de la taille des prédicteurs sur l'algorithme FTF12N2L

Pour évaluer l'effet de la taille des prédicteurs sur les performances de l'algorithme FTF12N2L, nous avons effectué plusieurs tests sur des différentes tailles des prédicteurs aller et retour de l'algorithme. On note que dans tous ces tests, la taille du filtre adaptatif est conserver constante et égale à $L=256$ coefficients.

Ces différents tests ont permis de tracer la courbe présentée par la figure III.7 qui exprime l'évolution de l'EQMN (énergie de l'erreur quadratique moyenne normalisée) en fonction des différentes tailles des prédicteurs aller et retour de l'algorithme adaptatif FTF12N2L. Cette étude nous a permis de voir que l'algorithme FTF12N2L donne des résultats identiques à ceux de l'algorithme FTF8L lorsque la taille des prédicteurs est supérieur ou égale à 140 points pour une taille de $L=256$.

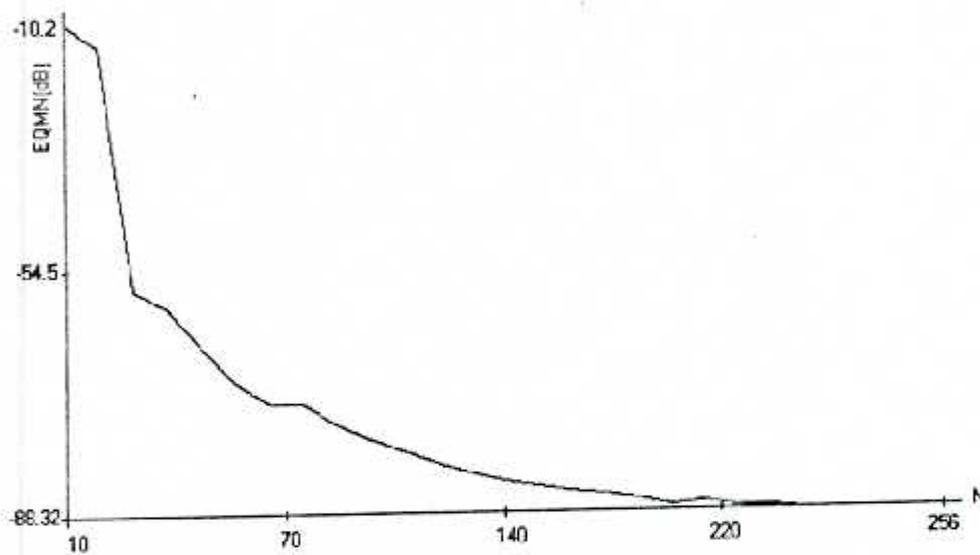


Figure III.7 : EQMN est fonction de la taille des prédicteurs N
Signal d'entrée est un bruit USASI (spectre moyen de la parole).
 $L=256$, $\lambda=0.9987$, $E_0=10$.

• Effet du bruit additif sur l'algorithme FTF12N2L

L'algorithme FTF12N2L avec une taille du filtre égale à $L=256$, un facteur d'oubli $\lambda=0.9987$ et une taille des prédicteurs $N=140$ donne les mêmes résultats que l'algorithme FTF8L dans le cas bruité.

La figure III.8 présente un exemple d'identification avec un signal à la sortie bruitée. Même conclusion que l'algorithme FTF8L, ce qui prouve qu'ils ont les mêmes performances de la convergence en régime permanent sur les signaux de sorties qui sont bruités.

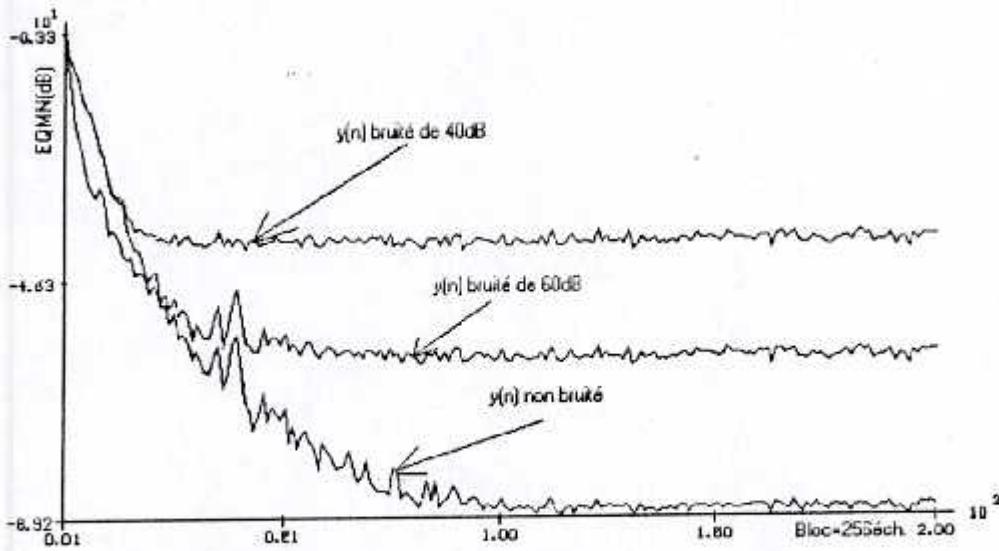


Figure III.8 : convergence du FTF12N2L pour : $L = 256$, $N = 32$
 Signal d'entrée : bruit USASI.
 (un signal non bruité est montré en référence)

• Cas d'un signal AR en entrée

Pour mieux comprendre le comportement de l'algorithme FTF12N2L par rapport à l'algorithme FTF8L nous avons tracé la réponse de ces deux algorithmes pour une entrée d'un signal auto régressif (AR) d'ordre 10 d'un bruit blanc. Le signal x_t est construit par la convolution du bruit blanc par un système tout pôle d'ordre 10. Les pôles de ce système sont :

$$0.98e^{+j0.02\pi}, 0.96e^{+j0.21\pi}, 0.8e^{+j0.96\pi}, 0.75e^{+j0.45\pi}, 0.7e^{+j0.44\pi}$$

Le tracé de la figure III.9 est le résultat de la simulation faite avec différents ordres de prédiction qui sont choisis au tour de la taille réel du modèle autoregressif utilisé.

On observe que lorsque l'ordre de prédiction N est inférieur à 10 la convergence du FTF12N2L est inférieure à celle du FTF8L classique ($N=L$) et que pour $N \geq 10$ les performances sont identiques (voir la figure III.9). Ceci démontre que le FTF12N2L se comporte bien si l'ordre de prédiction est adapté à l'ordre du modèle du signal d'entrée, (les ordres de prédictions surestimés donnent de meilleurs résultats que les ordres sous-estimés). Ce résultat est conforme à la théorie et au principe même d'obtention des algorithmes de Fast Newton.

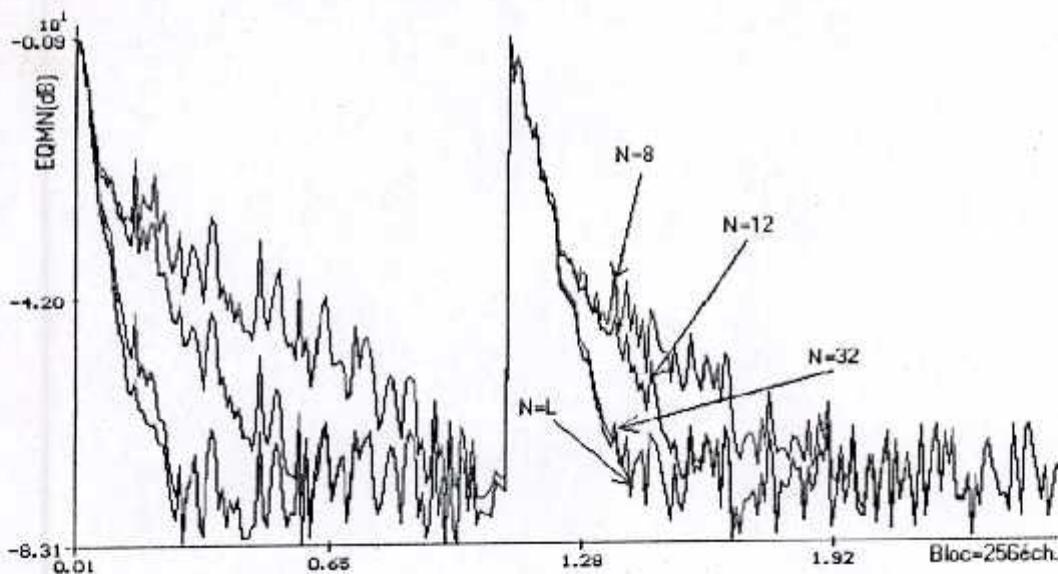


Figure III.9 : convergence du FTF12N2L $L = 256, \lambda = 0.9987$
Signal d'entrée AR d'ordre 10.

III.2.1.4 Algorithme FNS

Les paramètres de cette simulation sont :

L : Taille du filtre adaptatif,

N : Taille des prédicteurs,

$\mu^f = \mu^b$: Pas d'adaptation des prédicteurs directe et retour,

μ : Pas d'adaptation du filtre adaptatif,

• Effet de la taille du filtre adaptatif sur l'algorithme FNS

Pour évaluer l'effet de la taille du filtre sur l'algorithme, nous avons utilisé les mêmes signaux qui ont été décrits en dessus, cela nous a permis d'obtenir les résultats classés dans les tableaux suivants.

L	N	μ	$EQMN(dB)$
32	8	0.003	-0.94
128	16	0.00162	-14.54
512	16	0.000186	-87.56
1024	16	0.0001	-88.50

Tableau III.5 : Effet de la taille du filtre adaptatif sur l'algorithme
Signal d'entrée est un bruit blanc. $\mu^f = \mu^b = 0.00001$

L	N	μ	$EQMN(dB)$
32	8	0.000451	4.5
128	16	0.00069	-10.25
512	16	0.000174	-80.45
1024	16	0.000087	-85.88

Tableau III.6 : Effet de la taille du filtre adaptatif sur l'algorithme
Signal d'entrée est un bruit USASI. $\mu^f = \mu^b = 0.00001$

La figure III.10 suivante représente un échantillon des résultats résumés dans le tableau III.5. Ce résultat montre que l'algorithme FNS identifie mieux les réponses impulsionnelles lorsque le signal d'excitation est un bruit blanc.

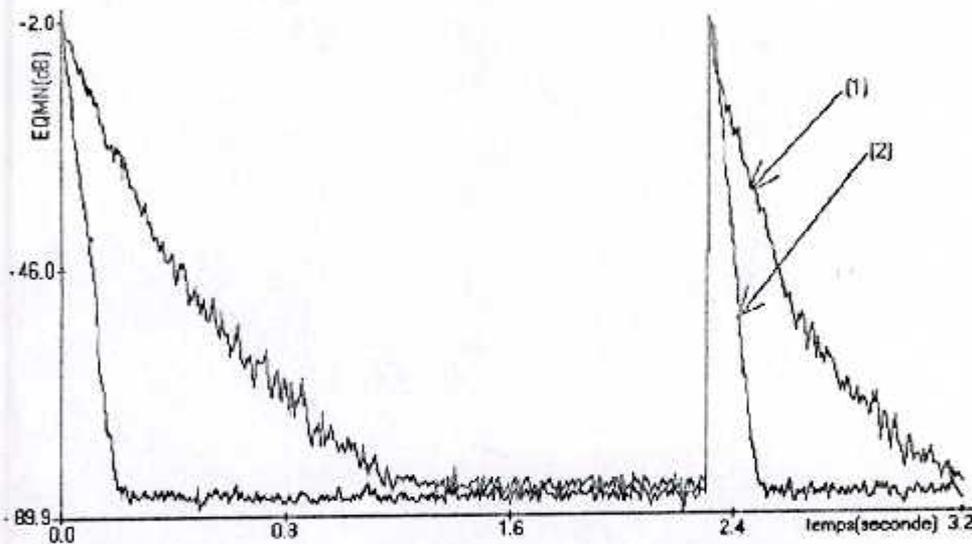


Figure III.10 : Convergence de l'algorithme FNS pour :
(1) : $L = 256$, (2) : $L = 512$, Signal d'entrée : bruit Blanc.

• Effet du bruit additif sur l'algorithme FNS

Nous avons utilisé les mêmes signaux de simulation que précédemment avec l'algorithme NLMS, cela nous a permis d'obtenir les résultats classés dans le tableau suivant.

$RSB(y/b)dB$	$EQMN(dB)$
10	-5.45
20	-6.32
30	-29.68
40	-39.70
60	-59.89

Tableau III.7 : Effet de la taille du filtre sur l'algorithme
Signal d'entrée : bruit USASI. $L = 256$, $N = 16$,
 $\mu^f = \mu^b = 0.00001$, $\mu = 0.000348$

Les figures III.11 et III.12 sont la représentation graphique de quelques résultats du tableau III.7, ces résultats montrent que l'algorithme FNS identifie mieux les réponses impulsionnelles dans les milieux bruités, si et seulement si le RSB est supérieur à $30dB$.

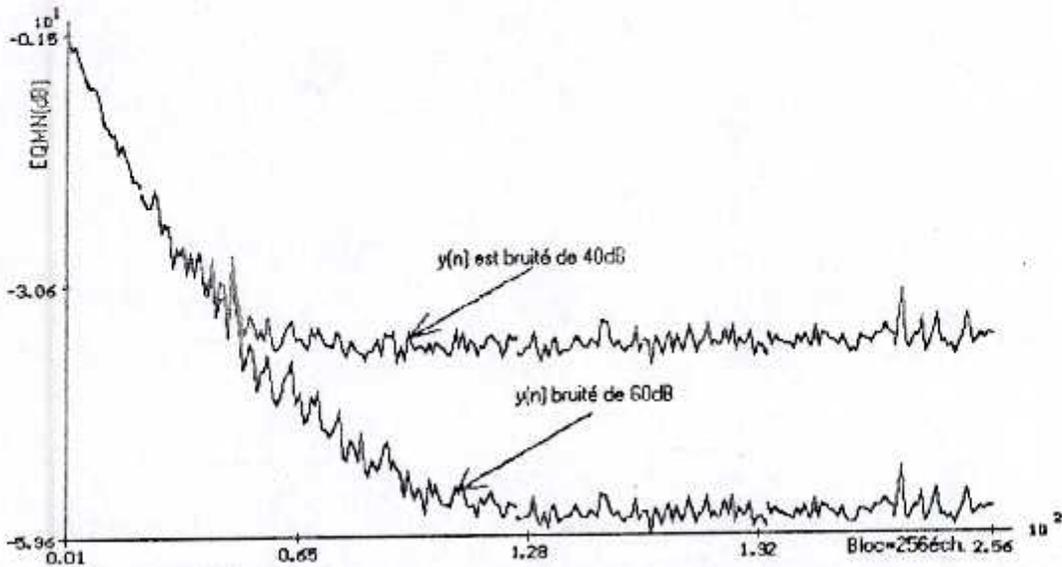


Figure III.11 : Convergence du FNS lorsque le signal de sortie est bruité.
 $L = 256, N = 32$

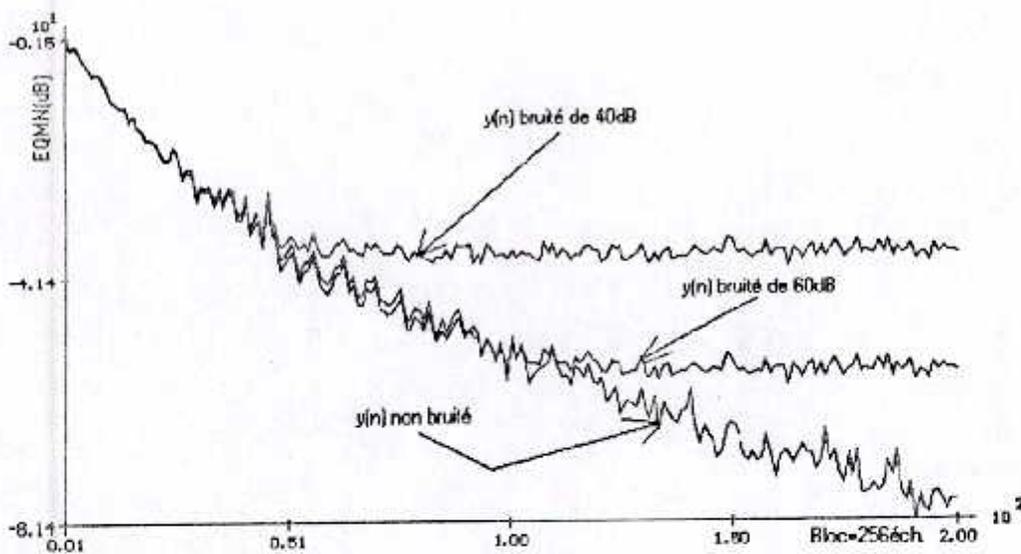


Figure III.12 : Convergence du FNS lorsque le signal de sortie est bruité.
 (Un signal de sortie non bruité est montré en référence).

La figure III.13 suivante représente les résultats obtenus par la comparaison de l'algorithme FNS avec l'algorithme NLMS. Nous avons utilisé le même signal décrit précédemment. On observe que l'algorithme FNS converge plus rapidement que le NLMS. Ceci confirme l'amélioration apportée par l'algorithme FNS [3].

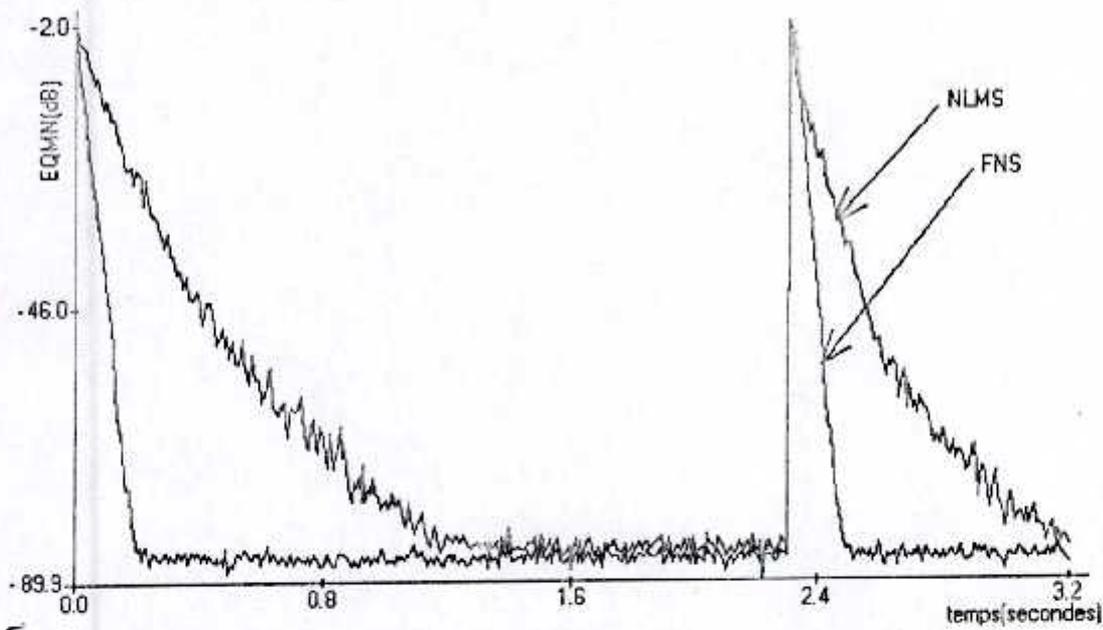


Figure III.13 : Comparaison de la vitesse de convergence entre l'algorithme FNS et NLMS. Signal d'entrée : bruit USASI

La figure III.14 représente le résultat de la comparaison de la version n°1 de l'algorithme FNS avec sa version n°2, ce résultat montre la grande similitude entre ces deux versions dans l'identification des réponses impulsionnelles.

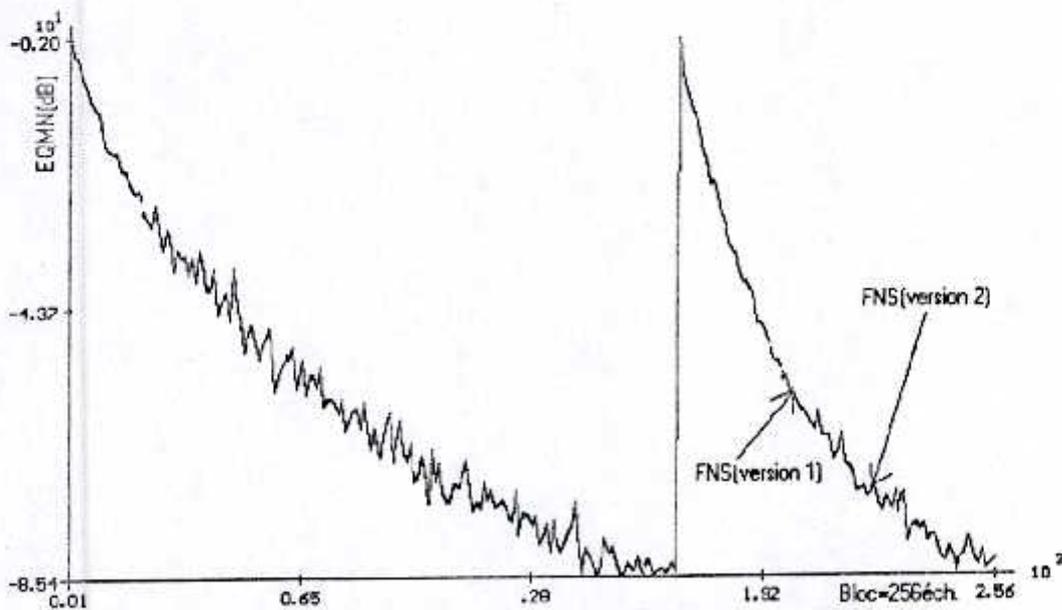


Figure III.14 : Les performances des deux versions d'algorithme FNS $L = 128, N = 16$, Signal d'entrée Bruit USASI.

III.2.2 Simulation des algorithmes de filtrage adaptatif par blocs à complexité réduite

Dans cette partie on va présenter les résultats de simulation des quatre versions d'algorithmes que nous avons proposer et présenter dans le chapitre II [3].

On va utiliser les même signaux de test qui ont été utilisés précédemment, ainsi que le même indice de performance qui est l'erreur de filtrage adaptatif normalisée par le signal à modéliser (signal écho). Le paramètre M désigne la taille du bloc.

III.2.2.1 Algorithme NLMS par bloc (BNLMS)

• Effet de la taille du filtre adaptatif sur l'algorithme

Cette simulation a permet d'obtenir les résultats classés dans les tableaux III.8 et III.9. Le premier tableau correspond aux résultats de simulation qui a été réalisée avec comme signal d'entrée le bruit blanc, alors que le deuxième tableau correspond à celle qui a été faite avec le bruit USASI. D'après ces résultats on remarque que l'algorithme identifie mieux les réponses impulsionnelles lorsqu'on utilise un bruit blanc comme un signal d'excitation, donc l'algorithme BNLMS est sensible à la nature du signal d'excitation $x(n)$.

La figure III.15 suivante présente le tracé de quelques résultats de l'algorithme BNLMS, comparé avec sa version originale (NLMS) que l'on retrouve avec ($M=1$). On remarque que la dégradation de la vitesse de convergence augmente avec l'augmentation de la taille du bloc M . Par ailleurs, on note que l'erreur en régime permanent est indépendante de la taille M du bloc.

L	M	$EQMN(dB)$
32	8	1.58
128	16	-13.54
256	16	-88.67
512	32	-87.45
1024	32	-87.53

Tableau III.8 : Effet de la taille du filtre sur l'algorithme
Signal d'entrée : bruit blanc, $E_0 = 1, \mu = 1$

L	M	μ	$EQMN(dB)$
32	8	0.3	1.48
128	16	0.2256	-12.00
256	16	0.2256	-37.58
512	32	0.1625	-23.20
1024	32	0.1625	-16.50

Tableau III.9 : Effet de la taille du filtre sur l'algorithme
Signal d'entrée : bruit USASI, $E_0 = 1$

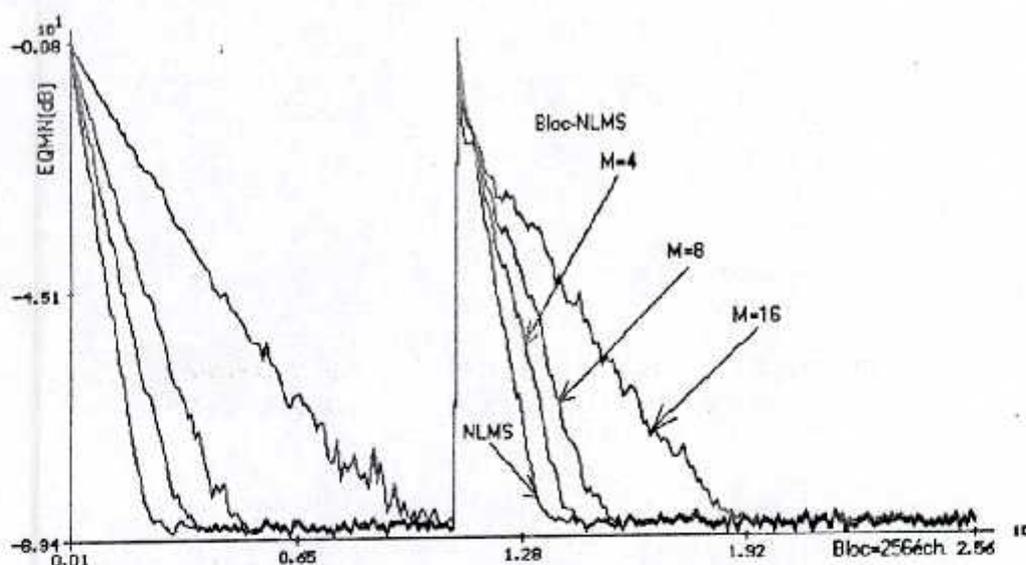


Figure III.15 : Convergence de l'algorithme BNLMS pour $M=4, 8, 16$ comparé avec sa version originale $NLMS (M=1)$.
Signal d'entrée : un bruit blanc. $\mu = 0.5$

• Effet du bruit additif sur l'algorithme

Dans cette simulation nous avons obtenu les résultats classés dans le tableau III.10. On note une grande sensibilité de l'algorithme BNLMS aux bruits additifs. Ceci est dû au ralentissement de vitesse de convergence de ce type d'algorithmes.

$RSB(y/b)dB$	$EQMN(dB)$
10	-05.50
20	-05.82
30	-30.00
40	-36.52
60	-38.55

Tableau III.10 : Effet de la taille du filtre sur l'algorithme
Signal d'entrée : est un bruit USASI,
 $L = 256, \mu = 0.225, M = 8$.

III.2.2.2 Algorithme FTF8L par bloc (BFTF8L)

• Effet de la taille du filtre adaptatif sur l'algorithme BFTF8L

Lors de cette simulation nous avons obtenu les résultats qui sont donnés dans les tableaux III.11 et III.12. D'après ces résultats, on peut dire que cette version d'algorithme par blocs présente une dégradation de la vitesse de convergence lorsque la taille du filtre transverse est supérieure à 256. Aussi on remarque que cette version est moins sensible à la nature du signal d'excitation que la version de l'algorithme BNLMS.

L	N	E_0	λ	$EQMN(dB)$
32	8	1	0.987	2.08
64	8	1	0.997	-2.51
128	16	1	0.99864	-13.95
256	16	1	0.9984	-81.21
512	32	3	0.9995	-54.16
1024	32	4	0.99981	-50.01

Tableau III.11 : Effet de la taille du filtre sur l'algorithme BFTF8L
Signal d'entrée : bruit blanc.

L	N	E_0	λ	$EQMN(dB)$
32	8	28	0.987	1.05
64	8	28	0.997	-3.56
128	16	26	0.99864	-13.00
256	16	60	0.9984	-85.64
512	32	256	0.9995	-60.48
1024	32	500	0.99981	-56.00

Tableau III.12 : Effet de la taille du filtre sur l'algorithme BFTF8L
Signal d'entrée : bruit USASI.

• Effet du bruit additif sur l'algorithme

Lors de cette simulation nous avons obtenu les résultats classés dans le tableau III.13.

$RSB(y/b)dB$	$EQMN(dB)$
10	-5.00
20	-5.50
30	-29.85
40	-39.92
60	-55.85

Tableau III.13 : Effet de la taille du filtre sur l'algorithme Signal
d'entrée : bruit USASI. $L = 256, \lambda = 0.9987, E_0 = 60$

Même conclusion que l'algorithme FTF8L.

La figure III.16 représente le résultat de la comparaison de cet algorithme avec sa version originale ($M=1$) pour une taille du bloc égale à $M=16$. Ce résultat montre une légère dégradation dans la vitesse de convergence initiale, qui devient très faible en régime permanent.

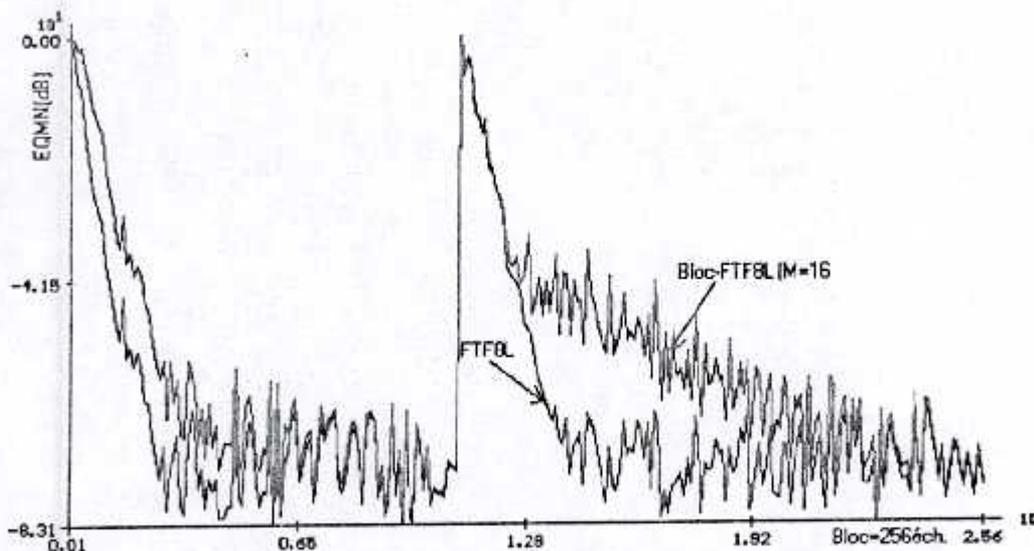


Figure III.16 : Convergence du BFTF8L pour $M=16$ comparé avec sa version originale FTF8L, $L = 256, \lambda = 0.9987$
Signal d'entrée AR d'ordre 10.

III.2.2.3 Algorithme FTF12N2L par bloc (BFTF12N2L)

- Effet de la taille du filtre adaptatif sur l'algorithme

Lors de cette simulation nous avons obtenu les résultats suivants. Même conclusion que l'algorithme FTF8L par bloc (BFTF8L).

L	M	E_0	N	λ	$EQMN(dB)$
32	8	1	8	0.9870	1.50
128	16	1	16	0.9974	-14.80
256	16	1.75	16	0.9985	-70.58
512	32	8	16	0.9995	-50.20
1024	32	8	16	0.9996	-42.19

Tableau III.14 : Effet de la taille du filtre sur l'algorithme
Signal d'entrée est un bruit blanc.

L	M	E_0	N	λ	$EQMN(dB)$
32	8	1	8	0.9870	1.80
128	16	25	16	0.9974	-14.50
256	16	32	16	0.9985	-75.77
512	32	128	16	0.9998	-48.29
1024	32	128	16	0.9999	-35.56

Tableau III.15 : Effet de la taille du filtre sur l'algorithme
Signal d'entrée est un bruit USASI.

• Effet du bruit additif sur l'algorithme

Nous avons obtenu le même résultat que l'algorithme FTF8L.

La figure III.17 représente le résultat de la comparaison de cet algorithme avec sa version originale ($M=1$) pour une taille du bloc égale à $M=16$. Même remarque avec l'algorithme FTF8L par bloc (BFTF8L).

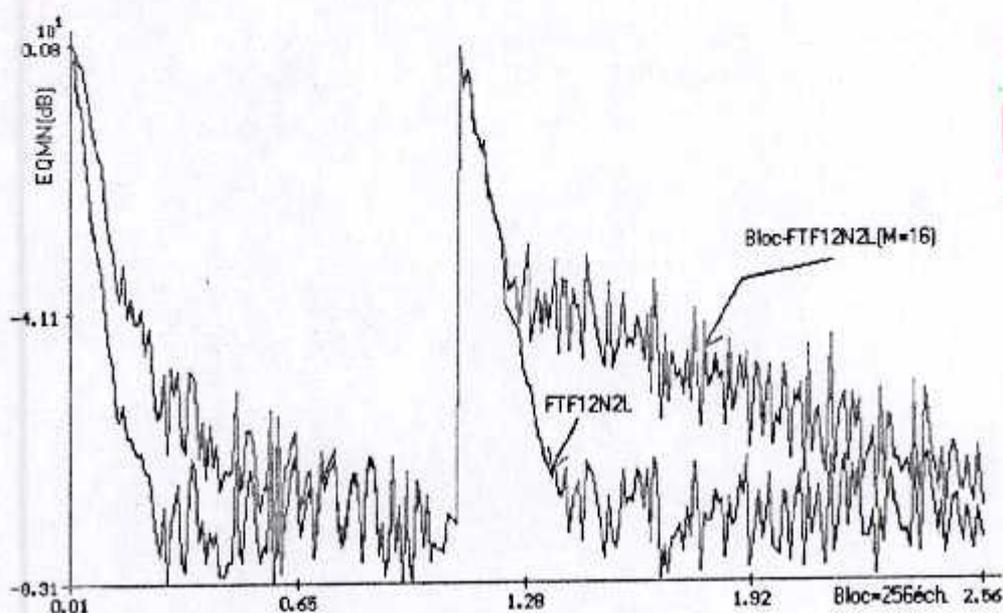


Figure III.17 : Comparaison de la vitesse de convergence du BFTF12N2L pour $M=16$, avec sa version originale (FTF12N2L). Signal d'entrée AR d'ordre 10.
 $L = 256, \lambda = 0.9987$

III.2.2.4 Algorithme FNS par bloc (BFNS)

• Effet de la taille du filtre adaptatif sur l'algorithme

Lors de cette simulation nous avons obtenu les résultats suivants. Ces résultats montrent que l'algorithme BFNS est très sensible à la nature du signal d'excitation $x(n)$.

L	M	N	μ	EQMN (dB)
32	8	8	0.0030000	2.50
64	8	16	0.0016250	-2.30
128	16	16	0.0008600	-10.51
256	16	16	0.0005125	-83.52
512	32	16	0.0001860	-71.68
1024	32	16	0.0001000	-67.38

Tableau III.16 : Effet de la taille du filtre sur l'algorithme
 Signal d'entrée : bruit blanc. $\mu^f = \mu^b = 0.0001$

L	M	N	μ	$EQMN(dB)$
32	8	8	0.000025	0.50
64	8	8	0.00003125	-2.50
128	16	16	0.00002125	-13.50
256	16	16	0.0000125	-32.70
512	32	16	0.000005125	-21.50
1024	32	16	0.000002125	-15.42

Tableau III.17 : Effet de la taille du filtre sur l'algorithme
Signal d'entrée : bruit USASI. $\mu^f = \mu^b = 0.00001$

• Effet du bruit additif sur l'algorithme

Cette simulation nous a permis d'obtenir les résultats classés dans le tableau III.18.

$RSB(y/b)dB$	$EQMN(dB)$
10	-4.5
20	-5.51
30	-26.0
40	-20.52
60	-29.56

Tableau III.18 : Effet de la taille du filtre sur l'algorithme
Signal d'entrée : bruit USASI. $L = 256, \lambda = 0.9987, E_0 = 64$

La figure III.18 représente le résultat de la comparaison de cet algorithme avec sa version originale ($M=1$) pour des différentes tailles du bloc M , ($M=4,8,16$). Ce résultat montre que la dégradation de la vitesse de convergence, en régime permanent, augmente avec l'augmentation de la taille du bloc M .

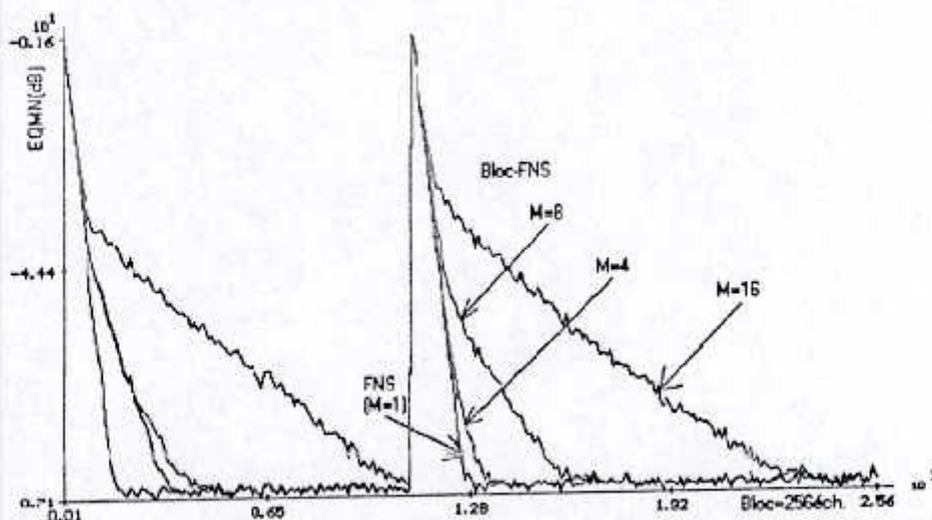


Figure III.18 : Comparaison de la vitesse de convergence de l'algorithme BFNS avec sa version originale FNS ($M=1$), pour $M=4,8,16$
Signal d'entrée : un bruit blanc. $L = 256$ et $N = 16$.

B - Cas non stationnaire (Parole)

III.2.3 Description des signaux de parole

La parole est un signal réel, d'énergie finie, non stationnaire. Sa structure est complexe et variable dans le temps :

Tantôt périodique (plus exactement quasi périodique) pour les sons voisés, ils sont très riches en harmonique d'une fréquence fondamentale. Ce qui leur donne un caractère assez facilement prévisible. Ils sont de forte énergie. La fréquence fondamentale, varie en moyenne de 70 Hz à 150 Hz chez les hommes et de 100Hz à 400Hz chez les femmes et les enfants.

Tantôt aléatoire pour les signaux non voisés, ils ne présentent pas de structure périodique, ils ont les caractéristiques spectrales d'un bruit légèrement corrélé, ils sont d'énergie moyenne et peu prévisible. Tantôt des silences qui ont une énergie quasi nulle.

La classification qui vient d'être esquissée est forcément un peu sommaire. Cependant, du fait de l'inertie de l'appareil phonatoire, le signal de parole peut être considéré comme stationnaire pendant des intervalles de temps de l'ordre de 20 ms.

Le signal d'entrée utilisé est constitué de deux phrases phonétiquement équilibrées, échantillonnées à 16 KHz et codées sur 16 bits.

La première phrase est prononcée par un locuteur masculin (signal parole1) et la deuxième par un locuteur féminin (signal parole2).

Signal parole1 : «un loup se jeta immédiatement sur la petite chèvre» (figure III.19 (A))

Signal parole2 : «il se garantira du froid avec ce bon capuchon» (figure III.19 (B))

III.2.4 Description des canaux de couplage acoustique

Nous disposons de deux réponses impulsionnelles de couplages acoustiques mesurées en régime stationnaire (figure III.2 du chapitre III).

La première est une réponse impulsionnelle mesurée dans une salle de visioconférence, cette réponse impulsionnelle contient 4096 points.

La deuxième est une réponse impulsionnelle mesurée dans une salle d'audioconférence, cette réponse impulsionnelle contient 8192 points.

Ces deux réponses impulsionnelles sont données par la figure III.2.

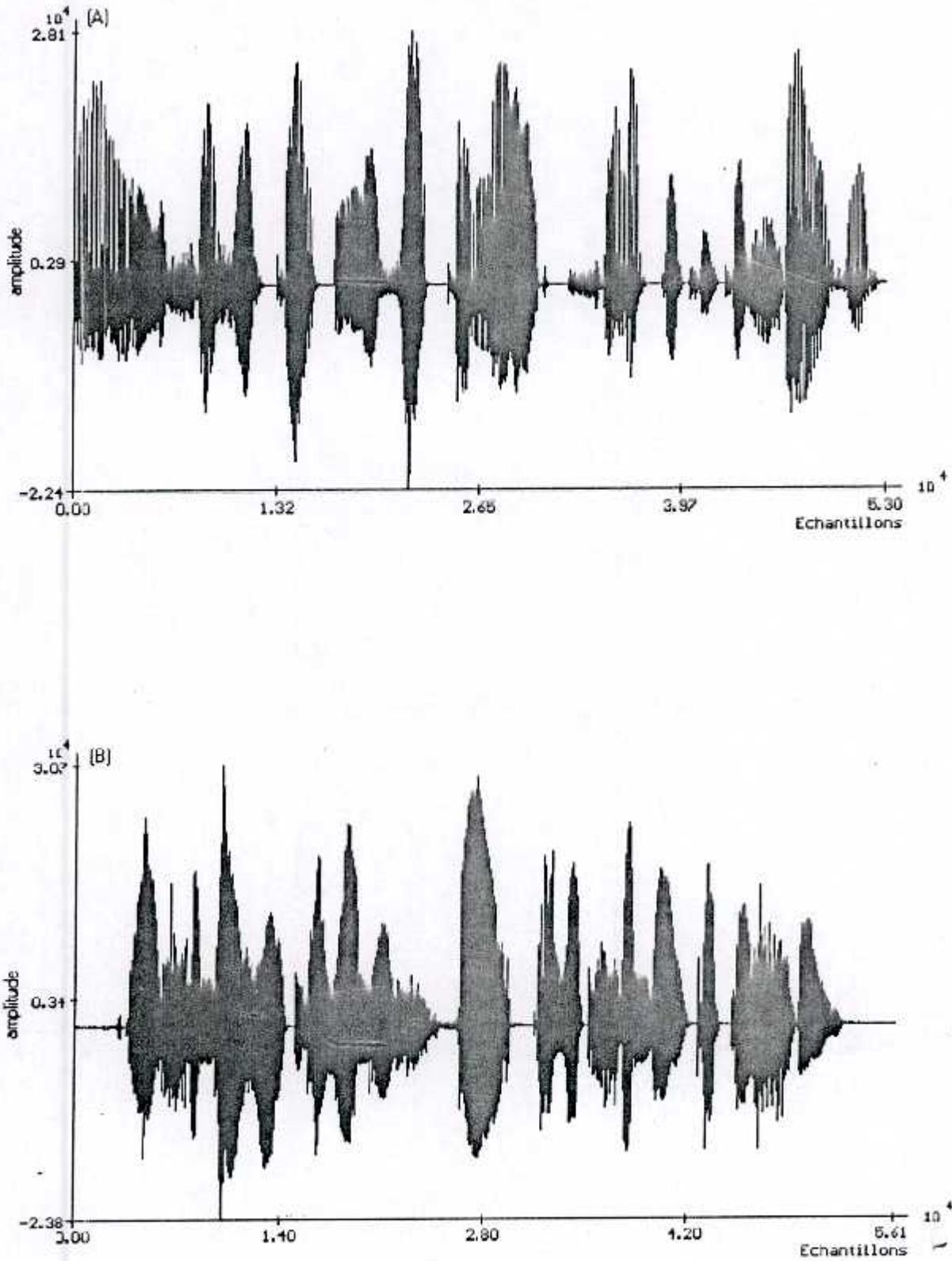


Figure III.19 : Signaux de parole 1 et 2.
 (A) signal de parole 1
 (B) signal de parole 2
 fréquence d'échantillonnage $F_e = 16 \text{ KHz}$

III.2.5 Simulation des algorithmes de filtrage adaptatif à complexité réduite

Dans les paragraphes suivants, nous allons simuler l'effet de la longueur significative de la réponse impulsionnelle d'un canal de couplage donné sur l'erreur quadratique moyenne. Les résultats de deux applications réalisées pour chaque algorithme d'identification adaptative à complexité réduite sont présentées.

Pour les signaux de test, le signal y_i est le signal d'entrée x_i (parole1 ou parole2) transformé par les deux canaux de couplage acoustique mesurés en régime stationnaire qui sont donnés par la figure III.2.

Les EQM (l'énergie de l'erreur de filtrage) de simulations sont calculées sur une moyenne de 128 points consécutifs.

III.2.5.1 Algorithme FTF8L

Nous avons réalisé 2 applications. La première application correspond à l'identification de la réponse impulsionnelle de la salle visioconférence et la deuxième est celle d'une salle d'audioconférence.

Les paramètres de cette simulation sont :
 Taille du filtre : L
 Facteur d'oubli exponentiel : λ
 Constante d'énergie minimale : E_0

- **Première application**

Le canal stationnaire est formé de 2048 premiers points d'une réponse impulsionnelle de couplage acoustique mesurée dans une salle visioconférence.

Taille filtre L	Signal parole 1			Signal parole 2		
	E_0	λ	$EQM(dB)$	E_0	λ	$EQM(dB)$
512	0.5	0.996000	-94.0	0.5	0.999600	-95.3
1024	1.0	0.999800	-93.1	1.0	0.999800	-92.9
2048	1.0	0.999987	-75.9	1.0	0.999987	-82.6

Tableau III.20 : $EQM(dB)$ pour des N donnés.
 (cas d'une salle visioconférence)

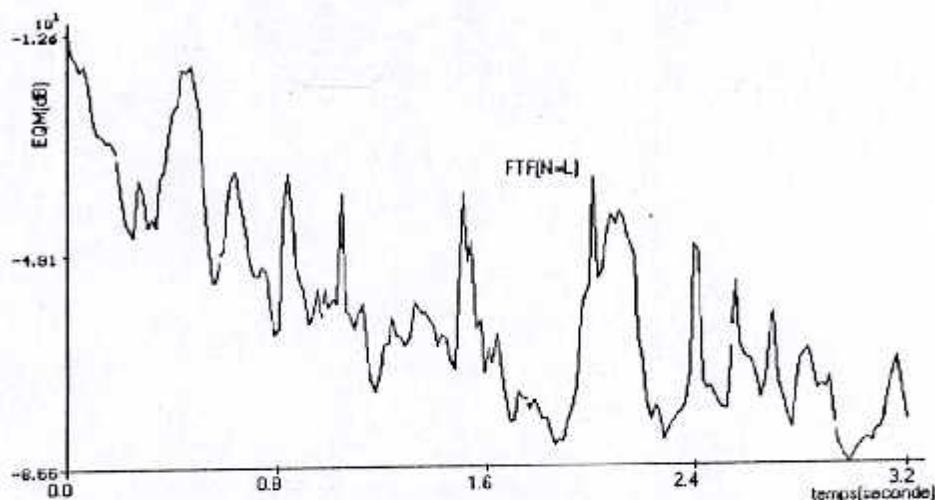


Figure III.20 : Evolution temporelle de l'EQM (énergie de l'erreur de filtrage) pour l'algorithme FTF8L. $L = 1024$, $\lambda = 0.99987$, pas de bruit en sortie. (cas d'une salle visioconférence)

• Deuxième application

Le canal stationnaire est formé des 4096 premiers points d'une réponse impulsionnelle de couplage acoustique mesurée dans une salle audioconférence.

Taille filtre L	Signal parole 1			Signal parole 2		
	E_0	λ	EQM(dB)	E_0	λ	EQM(dB)
512	0.5	0.996000	-92.2	0.5	0.999600	-92.7
1024	1.0	0.999800	-91.0	1.0	0.999800	-87.4
2048	1.0	0.999987	-69.0	1.0	0.999987	-68.4

Tableau III.21 : EQM(dB) pour des N donnés. (cas d'une salle audioconférence)

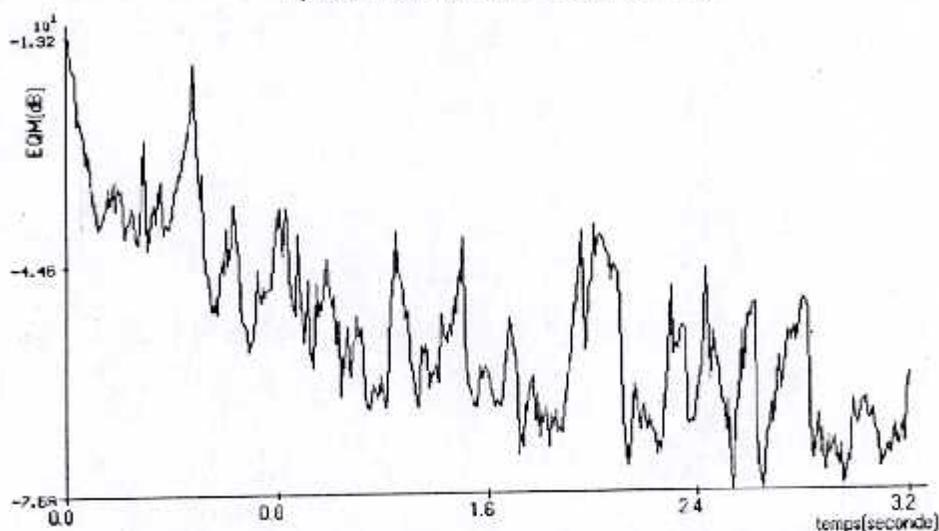


Figure III.21 : Evolution temporelle de l'EQM (énergie de l'erreur de filtrage) pour l'algorithme FTF8L. $L = 2048$, $\lambda = 0.999987$, pas de bruit en sortie. (Cas d'une salle audioconférence).

Les deux tableaux III.20 et III.21 montrent bien les bonnes performances de l'algorithme des moindres carrés rapides numériquement stables avec la parole comme signal excitant. Surtout sur les tailles du filtre transverse 512 et 1024 points avec une légère dégradation de l'EQM pour la taille de la réponse impulsionnelle de 2048 points.

Aussi ces résultats montrent que l'algorithme FTF8L a presque le même comportement envers deux signaux de parole qui sont différents entre eux (signal parole 1 et 2). Ceci montre son indépendance de la nature du signal excitant.

Les deux figures III.20 et III.21 illustrent deux exemples qui correspondent aux cas où il n'y a pas de bruit ajouté en sortie. Dans ces deux cas l'optimum est atteint juste après 28000 itérations environ (soit 1.7 secondes). Ce qui montre sa grande vitesse de convergence qui est due essentiellement à son indépendance de la nature du signal d'entrée.

Remarque importante :

Dans toutes les simulations suivantes nous avons procédé à tracer l'évolution de l'EQM de l'algorithme FTF8L de complexité 8L (L est la taille du filtre) pour le même signal excitant afin de le comparer avec l'algorithme qui est en cours de simulation.

III.2.5.2 Algorithme NLMS

- Les paramètres de cette simulation sont :
- Taille du filtre : L
- Pas d'adaptation : μ
- Constante d'énergie minimale : E_0

• **Première application**

Le canal stationnaire est formé de 2048 premiers points d'une réponse impulsionnelle de couplage acoustique mesurée dans une salle visioconférence.

L	Signal parole 1			Signal parole 2		
	E_0	μ	EQM (dB)	E_0	μ	EQM (dB)
512	1.0	1.0	-60.2	1.0	1.0	-80.1
1024	1.0	1.0	-48.2	1.0	1.0	-54.7
2048	1.0	1.0	-41.0	1.0	1.0	-51.4

Tableau III.22 : EQM (dB) pour des L donnés.
(cas d'une salle visioconférence)

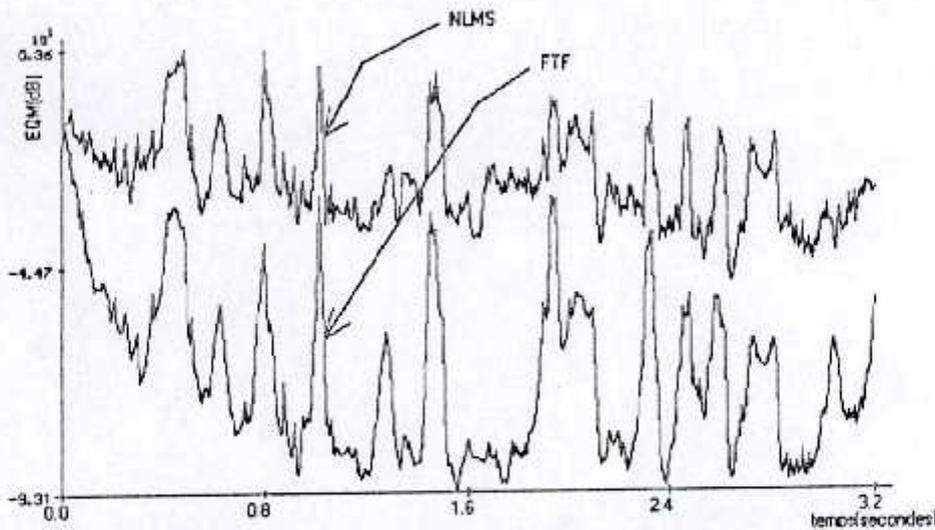


Figure III.22 : Evolution temporelle de l'EQM (énergie de l'erreur de filtrage) pour les algorithmes NLMS et FTF. $L = 1024$, $\mu = 1.0$, $\lambda = 0.9998$, pas de bruit en sortie (Cas d'une salle visioconférence).

• Deuxième application

Le canal stationnaire est formé des 4096 premiers points d'une réponse impulsionnelle de couplage acoustique mesurée dans une salle audioconférence.

L	Signal parole 1			Signal parole 2		
	E_0	μ	EQM (dB)	E_0	μ	EQM (dB)
512	1.0	1.0	-57.0	1.0	1.0	-73.8
1024	1.0	1.0	-53.3	1.0	1.0	-54.0
2048	1.0	1.0	-30.0	1.0	1.0	-31.4

Tableau III.23 : EQM (dB) pour des L donnés. (cas d'une salle audioconférence)

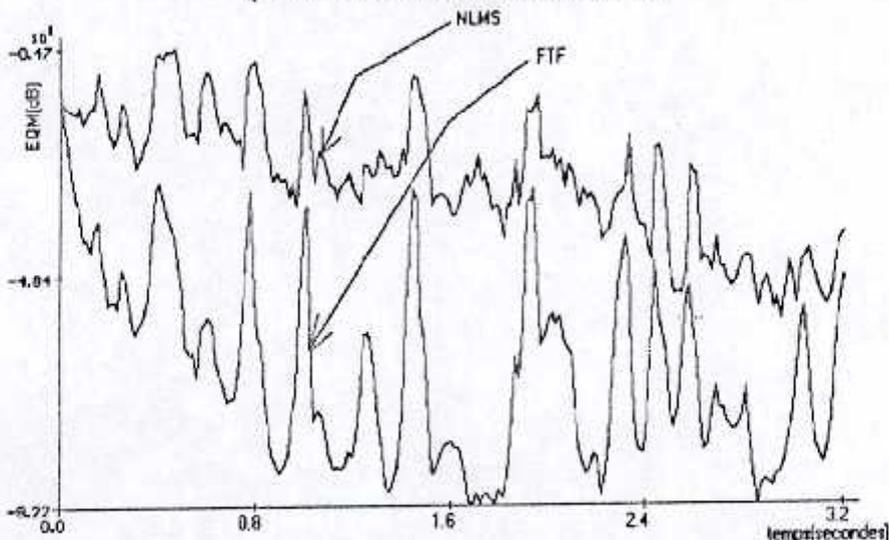


Figure III.23 : Evolution temporelle de l'EQM (énergie de l'erreur de filtrage) pour les algorithmes NLMS et FTF. $L = 512$, $\mu = 1.0$, $\lambda = 0.9998$ pas de bruit en sortie (Cas d'une salle audioconférence).

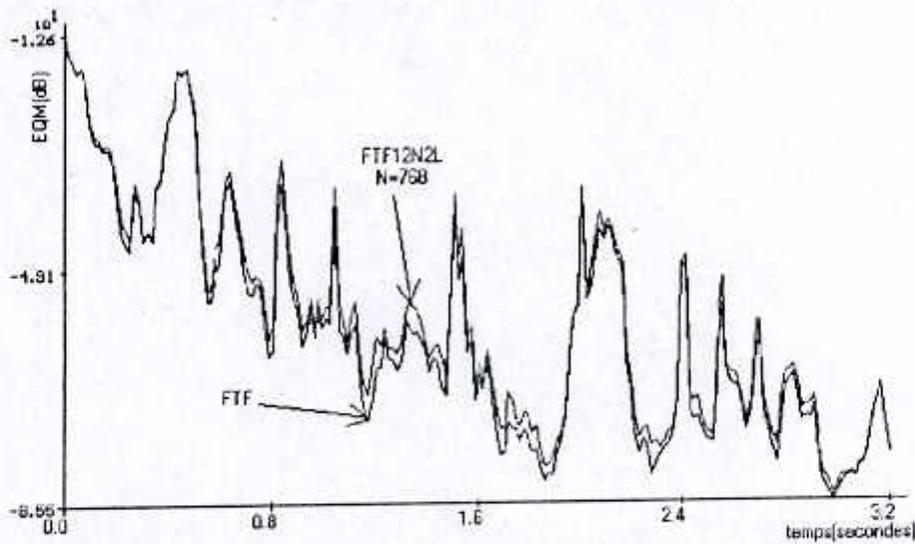


Figure III.24 : Evolution temporelle de l'EQM (énergie de l'erreur de filtrage) pour les algorithmes FNTF12N2L et FTF. $L = 1024$, $\lambda = 0.9998$, pas de bruit en sortie (Cas d'une salle visioconférence).

• Deuxième application

Le canal stationnaire utilisé est la réponse impulsionnelle de couplage acoustique mesurée dans une salle audioconférence de taille 4096 points.

L	Signal parole 1				Signal parole 2			
	N	λ	E_0	EQM (dB)	N	λ	E_0	EQM (dB)
512	384	0.9996	1.0	-93.6	384	0.9996	1.0	-94.1
1024	768	0.9998	1.0	-92.0	768	0.9998	1.0	-88.6
2048	1536	0.999987	1.0	-75.7	1536	0.999987	1.0	-63.4

Tableau III.25 : EQM (dB) pour des L donnés. (cas d'une salle audioconférence)

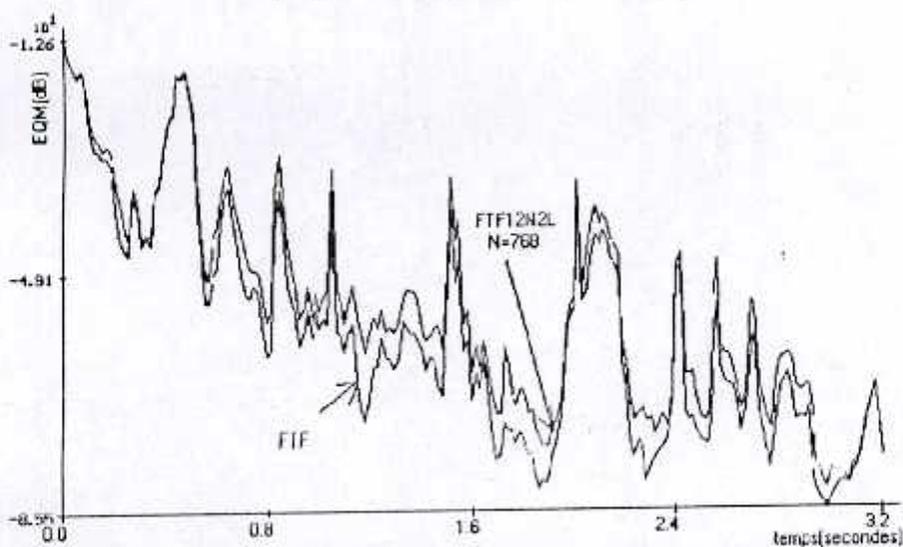


Figure III.25 : Evolution temporelle de l'EQM (énergie de l'erreur de filtrage) pour les algorithmes FNTF12N2L et FTF. $L = 1024$, $\lambda = 0.9998$, pas de bruit en sortie (Cas d'une salle audioconférence).

Les deux tableaux III.24 et III.25 montrent bien la grande similitude entre les résultats obtenus avec l'algorithme FTF8L est ceux obtenus avec l'algorithme FTF12N2L. Ce qui montre la grande ressemblance entre les performances des deux algorithmes s'ils s'opèrent sur les signaux de parole tels que le signal de parole 1 et 2.

De cela on peut conclure que tout ce qui est dit sur l'algorithme FTF8L reste, à quelques erreurs près, vraie pour l'algorithme FTF12N2L, ce qui montre les grands avantages offerts par l'algorithme FTF12N2L sur son ancienne version surtout sur le côté de la complexité qui est considérablement réduite (2L : ordre de la complexité de l'algorithme NLMS) lorsque la taille des prédicteurs est faible devant celle du filtre transverse.

Les deux figures III.24 et III.25 montrent bien la superposition presque totale des tracés des deux algorithmes pour la taille du filtre de 1024 points, ce qui démontre encore une fois les performances excellentes du nouvel algorithme malgré la réduction de la complexité qu'il l'a offert.

III.2.5.4 Algorithme FNS

Les paramètres de cette simulation sont :

Taille du filtre adaptatif : L

Taille des prédicteurs : N

Pas d'adaptation des prédicteurs directe et retour : $\mu^f = \mu^b$

Pas d'adaptation du filtre adaptatif : μ

- **Première application**

Le canal stationnaire utilisé est la réponse impulsionnelle de couplage acoustique mesurée dans une salle visioconférence de taille 2048 points.

L	N	Signal parole 1			Signal parole 2		
		$\mu^f = \mu^b$	μ	$EQM(dB)$	$\mu^f = \mu^b$	μ	$EQM(dB)$
512	15	0.05	0.0006	-77.0	0.05	0.0006	-82.4
1024	32	0.025	0.0003	-69.9	0.0125	0.000125	-49.1
2048	64	0.02	0.000175	-53.5	0.0115	0.0001	-42.2

Tableau III.26 : $EQM(dB)$ pour des L donnés.
(cas d'une salle visioconférence)

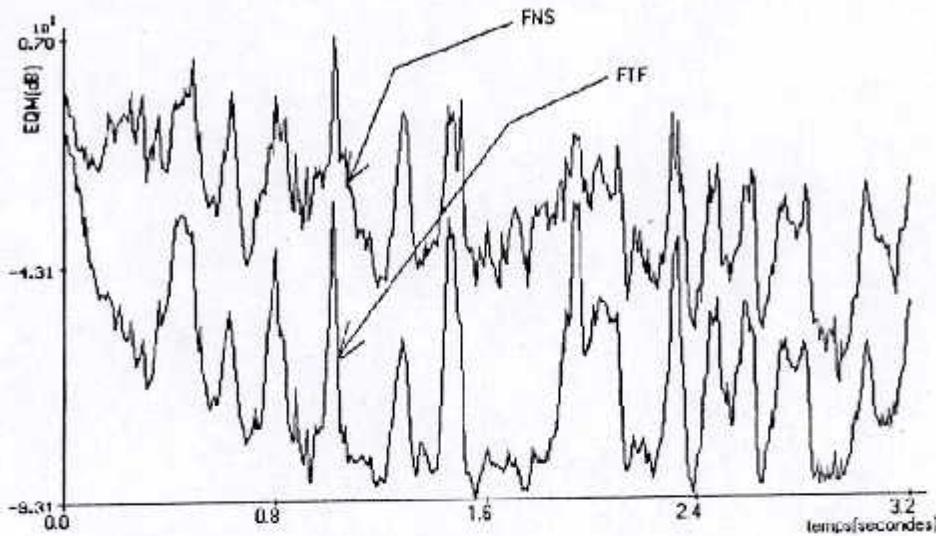


Figure III.26 : Evolution temporelle de l'EQM (énergie de l'erreur de filtrage) pour les algorithmes FNS et FTF8L. $L = 1024$, $\lambda = 0.9998$, pas de bruit en sortie. (cas d'une salle visioconférence)

• Deuxième application

Le canal stationnaire est formé des 4096 premiers points d'une réponse impulsionnelle α couplage acoustique mesurée dans une salle audioconférence.

L	N	Signal parole 1			Signal parole 2		
		$\mu^f = \mu^b$	μ	EQM (dB)	$\mu^f = \mu^b$	μ	EQM (dB)
512	15	0.05	0.0006	-72.3	0.05	0.0006	-84.9
1024	32	0.025	0.0003	-62.0	0.025	0.000125	-54.9
2048	64	0.0115	0.000125	-45.3	0.0115	0.0001	-42.6

Tableau III.27 : EQM (dB) pour des L donnés. (cas d'une salle audioconférence)

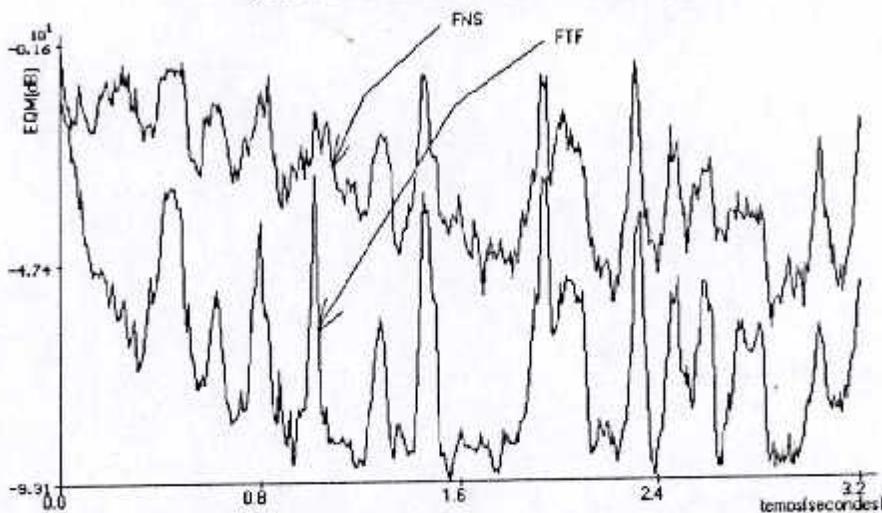


Figure III.27 : Evolution temporelle de l'EQM (énergie de l'erreur de filtrage) pour les algorithmes FNS et FTF. $L = 1024$, $\lambda = 0.9998$, pas de bruit en sortie. (cas d'une salle audioconférence)

Les deux tableaux III.26 et III.27 montrent bien le bon comportement de l'algorithme FNS (fast Newton Simplifié) avec les signaux de parole 1 et 2, si la taille du filtre transverse est égale à 512 points.

Une dégradation plus importante et croissante est remarquée dans le comportement de l'algorithme avec le signal de parole 2 si la taille du filtre transverse est égale soit à 1024 ou 2048 points. Cette dégradation devient moins importante avec le signal de parole 1. Ceci nous permet de dire que le comportement de l'algorithme FNS dépend de la nature du signal d'entrée. Cela est dû essentiellement, est pour la simple raison, à sa partie de prédiction et de ses deux prédicteurs aller et retour qui sont d'une structure LMS très simple, cette partie de prédiction contrôle indirectement la partie de filtrage adaptatif par le biais du gain de Kalman que son calcul est lié intimement aux calculs des prédicteurs aller et retour. De ce fait, et puisque la structure LMS est sensible à la nature du signal d'entrée, cela a rendu le comportement global de l'algorithme sensible à la nature et l'énergie du signal excitant, mais avec un degré moins important que l'algorithme NLMS.

La comparaison entre les deux figures III.26 et III.27 avec les figures III.22 et III.25 montrent clairement l'amélioration que la partie de prédiction et la normalisation du gain des prédicteurs aller et retour par l'énergie du signal d'entrée (structure NLMS), a apporté au comportement global de l'algorithme par rapport à l'algorithme NLMS. Surtout sur la vitesse de convergence qui reste toujours limitée par sa dépendance à la nature du signal d'entrée (mais moins dépendante que l'algorithme NLMS).

III.2.6 Simulation des algorithmes de filtrage adaptatif par blocs à complexité réduite

Nous avons simulé 4 algorithmes par blocs à complexité réduite, ces algorithmes sont : NLMS par bloc (BNLMS), FTF par bloc (BFTF8L), FTF12N2L par bloc (BFTF12N2L), et FNS par bloc (BFNS).

Les simulations des algorithmes par blocs ont été réalisées pour deux types de canaux acoustiques (salle d'audioconférence de visioconférence) (figure III.2) convoluées avec les deux signaux de paroles (voir figure III.19).

Remarque :

Dans toutes les simulations suivantes, nous avons opté à tracer pour chaque algorithme par bloc, sa version originale afin de mieux les comparer.

III.2.6.1 Algorithme BNLMS

Les paramètres de cette simulation sont :

Taille du filtre adaptatif : L

Taille du Bloc : M

Pas d'adaptation : μ

Constante d'énergie minimale : E_0

• Première application

Le canal stationnaire est formé de 2048 premiers points d'une réponse impulsionnelle de couplage acoustique mesurée dans une salle visioconférence.

L	Signal parole 1				Signal parole 2			
	M	μ	E_0	$EQM(dB)$	M	μ	E_0	$EQM(dB)$
512	16	0.125	1.0	-38.6	16	0.125	1.0	-42.2
1024	16	0.125	1.0	-34.4	16	0.125	1.0	-39.9
2048	16	0.125	1.0	-31.9	16	0.125	1.0	-30.4

Tableau III.28 : $EQM(dB)$ pour des L donnés.
(cas d'une salle visioconférence)

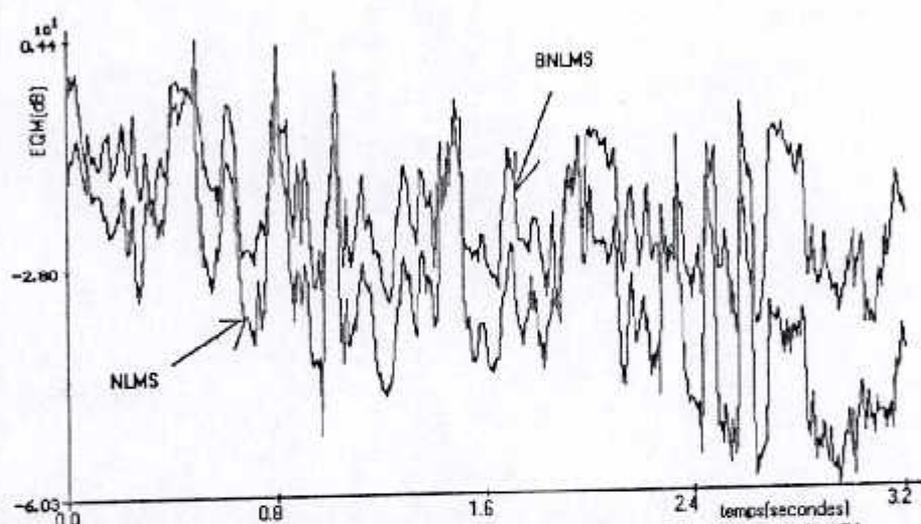


Figure III.28 : Evolution temporelle de l' EQM (énergie de l'erreur de filtrage) pour deux algorithmes : NLMS et BNLMS. $L = 512$, $M = 16$, pas de bruit en sortie.
(Cas d'une salle visioconférence).

• Deuxième application

Le canal stationnaire est formé des 4096 premiers points d'une réponse impulsionnelle de couplage acoustique mesurée dans une salle audioconférence.

L	Signal parole 1				Signal parole 2			
	M	μ	E_0	$EQM(dB)$	M	μ	E_0	$EQM(dB)$
512	16	0.125	1.0	35.2	16	0.125	1.0	-42.0
1024	16	0.125	1.0	-27.5	16	0.125	1.0	-32.1
2048	16	0.125	1.0	-21.5	16	0.125	1.0	-28.0

Tableau III.29 : $EQM(dB)$ pour des L donnés.
(Cas d'une salle audioconférence).

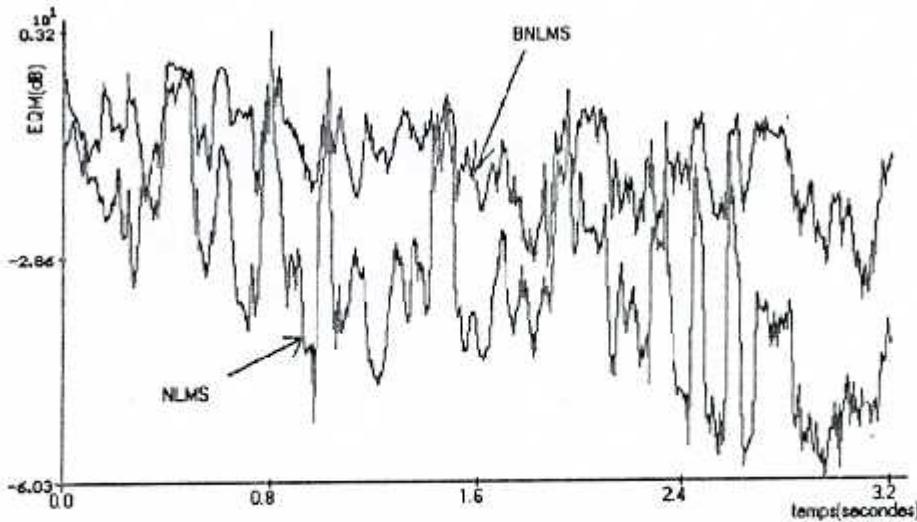


Figure III.29 : Evolution temporelle de l'EQM (énergie de l'erreur de filtrage) pour deux algorithmes : NLMS et BNLMS. $L = 512$, $M = 16$, pas de bruit en sortie. (Cas d'une salle audioconférence).

Les deux tableaux III.28 et III.29 montrent bien que l'algorithme du gradient stochastique normalisé par bloc (BNLMS), ne conserve pas le comportement de l'algorithme du gradient NLMS avec les signaux de la parole, mais on remarque une dégradation supplémentaire de l'EQM, cette dégradation est liée directement à la taille du bloc M , dont lequel les coefficients du filtre adaptatif transverse restent inchangés.

Les figures III.28 et III.29 correspondent au cas où il n'y a pas de bruit ajouté à la sortie. On notera que durant l'expérience, l'algorithme NLMS par bloc (NLMS) reste en phase transitoire ce qui signifie encore une fois que sa vitesse de convergence est dépendante de la nature du signal d'entrée.

Nous pouvons conclure que ce nouvel algorithme nous permis de réduire significativement la complexité des calculs qui se paye par une dégradation de la vitesse de convergence. Cette dégradation dépend directement de la taille du bloc qu'en le mettant égale à 1 on trouve sa version originale (NLMS). Pour $M=1$ on retrouve les résultats de l'algorithme NLMS

III.2.6.2 Algorithme BF8L

Les paramètres de cette simulation sont :

Taille du filtre adaptatif	: L
Taille du Bloc	: M
Facteur d'oubli exponentiel	: λ
Constante d'énergie minimale	: E_0

• Première application

Le canal stationnaire est formé de 2048 premiers points d'une réponse impulsionnelle de couplage acoustique mesurée dans une salle visioconférence.

		Signal parole 1			Signal parole 2		
L	M	E_0	λ	$EQM(dB)$	E_0	λ	$EQM(dB)$
512	16	25	0.996000	-79.8	50	0.999600	-71.3
1024	16	50	0.999800	-72.1	100	0.999800	-77.7
2048	16	150	0.999987	-45.3	250	0.999987	-45.7

Tableau III.30 : $EQM(dB)$ pour des L donnés.
(cas d'une salle visioconférence)

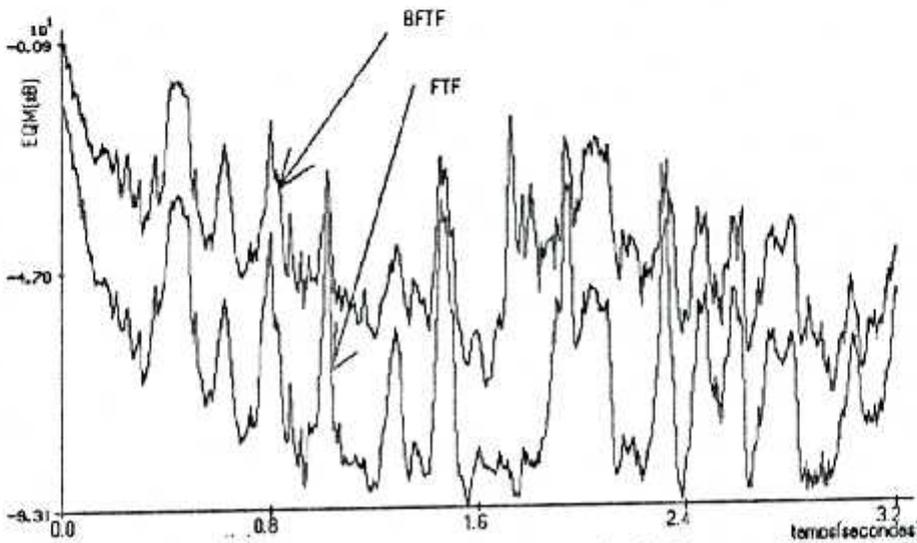


Figure III.30 : Evolution temporelle de l' EQM (énergie de l'erreur de filtrage) pour deux algorithmes : FTF8L et BFTF8L. $L = 1024$, $\lambda = 0.99987$, $M = 16$, pas de bruit en sortie.
(cas d'une salle visioconférence)

☛ Deuxième application

Le canal stationnaire est formé des 4096 premiers points d'une réponse impulsionnelle de couplage acoustique mesurée dans une salle audioconférence.

		Signal parole 1			Signal parole 2		
L	M	E_0	λ	$EQM(dB)$	E_0	λ	$EQM(dB)$
512	16	25	0.996000	-73.0	50	0.999600	-66.6
1024	16	50	0.999800	-57.0	100	0.999800	-67.3
2048	16	150	0.999987	-41.1	250	0.999987	-42.8

Tableau III.31 : $EQM(dB)$ pour des L donnés.
(cas d'une salle audioconférence)

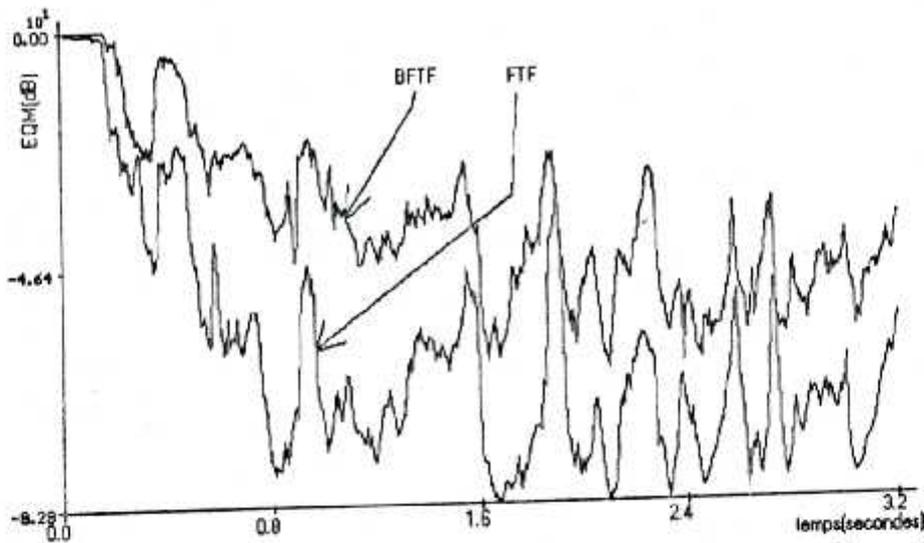


Figure III.31 : Evolution temporelle de l'EQM (énergie de l'erreur de filtrage) pour deux algorithmes : FTF8L et BFTF8L. $L = 1024$, $\lambda = 0.99987$, $M = 16$, pas de bruit en sortie. (cas d'une salle audioconférence)

Les résultats obtenus montrent qu'il y a une dégradation de cet algorithme avec la parole par rapport à l'algorithme FTF8L. Les tableaux III.30 et III.31 montrent bien que l'algorithme BFTF8L a donné des bons résultats pour les tailles du filtre transverse moyennes (512 points) et a représenté une dégradation croissante de l'EQM en augmentant dans la taille du filtre transverse tel que le cas de 1024 et 2048 points. On remarque aussi que l'algorithme a nécessité une augmentation importante dans les valeurs des constantes d'initialisations pour un fonctionnement stable et correct.

On note aussi que cette fois l'algorithme a montré une certaine dépendance au signal d'entrée, car il a donné des résultats pour le signal de parole1 mieux que celle du signal de parole2 avec le canal de couplage acoustique de la salle de visioconférence. Cela est dû essentiellement à :

- 1- La nature des tranches du signal excitant surtout les tranches des silences qui dégradent significativement l'adaptation.
- 2- Le blocage et le déblocage aléatoire de l'adaptation dans des zones inconnues du signal de parole excitant.

Cette manière aléatoire de réaliser les blocs dans l'algorithme implique des adaptations de l'algorithme durant des zones de silence et bloque l'adaptation dans certaines zones de signal utile. Cette alternance aléatoire de bloquer et débloquer l'adaptation entraîne une dégradation importante des performances de l'algorithme BFTF8L par rapport à celles de FTF8L.

Les figures III.30 et III.31 correspondent au cas où il n'y a pas de bruit ajouté à la sortie. On notera que l'algorithme BFTF8L atteint l'optimum après 39000 itérations environ (2.4 secondes), tandis que l'algorithme FTF8L l'atteint après juste 15000 itérations environ (1.2 secondes). La bonne vitesse de convergence de l'algorithme FTF8L par bloc (qui reste faible devant sa version originale) par rapport aux autres versions par blocs est due à son faible dépendance à la nature du signal d'entrée malgré le blocage de l'adaptation durant M échantillons consécutifs.

III.2.6.3 Algorithme BFTF12N2L

Les paramètres de cette simulation sont :

- Taille du filtre adaptatif : L
- Taille des prédicteurs : N
- Taille du Bloc : M
- Facteur d'oubli exponentiel : λ
- Constante d'énergie minimale : E_0

• **Première application**

Le canal stationnaire utilisé est la réponse impulsionnelle de couplage acoustique mesurée dans une salle visioconférence de taille 2048 points.

		Signal parole 1				Signal parole 2			
L	M	N	λ	E_0	$EQM(dB)$	N	λ	E_0	$EQM(dB)$
512	16	384	0.9996	1.0	-78.8	384	0.9996	1.0	-74.3
1024	16	768	0.9998	1.0	-76.3	768	0.9998	1.0	-77.3
2048	16	1536	0.999987	1.0	-45.1	1536	0.999987	1.0	-45.0

Tableau III.32 : $EQM(dB)$ pour des L donnés.
(cas d'une salle visioconférence)

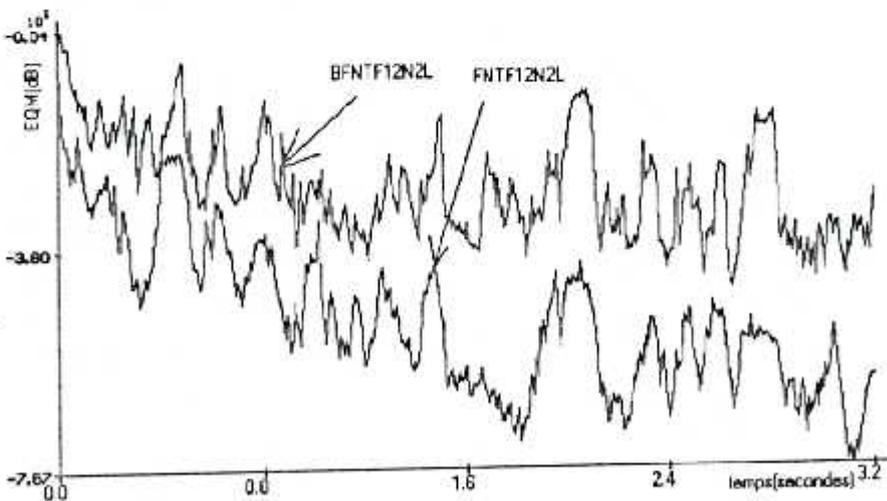


Figure III.32 : Evolution temporelle de l' EQM (énergie de l'erreur de filtrage) pour deux algorithmes : FTF12N2L et BFTF12N2L. $L = 2048$, $\lambda = 0.999987$, $M = 16$ pas de bruit en sortie. (cas d'une salle visioconférence)

• **Deuxième application**

Le canal stationnaire est formé des 4096 premiers points d'une réponse impulsionnelle de couplage acoustique mesurée dans une salle audioconférence.

L	M	Signal parole 1				Signal parole 2			
		N	λ	E_0	$EQM(dB)$	N	λ	E_0	$EQM(dB)$
512	16	384	0.9996	1.0	-72.4	384	0.9996	1.0	-74.8
1024	16	768	0.9998	1.0	-56.2	768	0.9998	1.0	-65.9
2048	16	1536	0.999987	1.0	-40.8	1536	0.999987	1.0	-42.3

Tableau III.33 : $EQM(dB)$ pour des L donnés.
(cas d'une salle audioconférence)

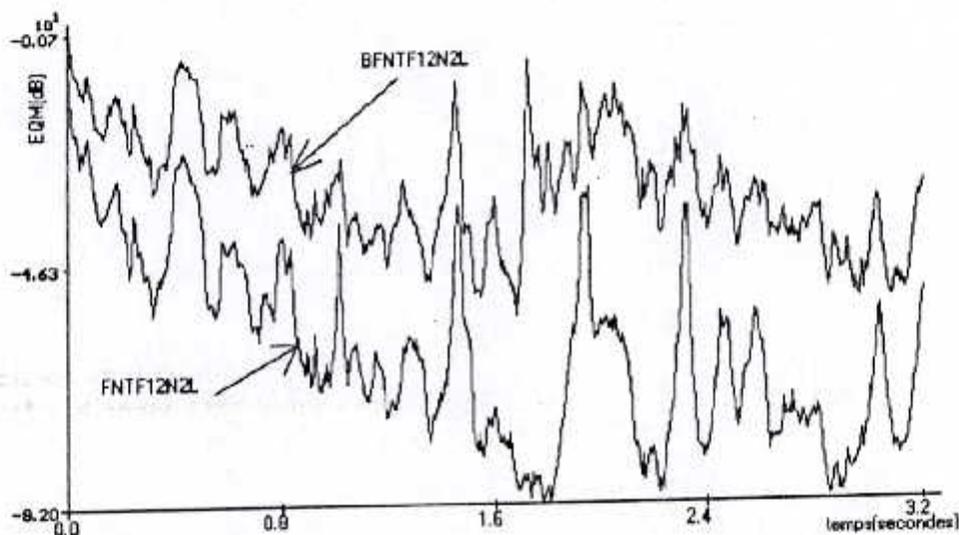


Figure III.33 : Evolution temporelle de l'EQM (énergie de l'erreur de filtrage) pour deux algorithmes : FTF12N2L et BFTF12N2L. $L = 1024$, $\lambda = 0.99987$, $M = 16$ pas de bruit en sortie. (cas d'une salle audioconférence)

Les tableaux III.31 et III.32 montrent bien le comportement similaire entre l'algorithme FTF12N2L par bloc (BFTF12N2L), et l'algorithme FTF8L par bloc (BFTF8L) dans toutes les expériences faites avec les trois tailles réelles du filtre transverse. On conclut que l'algorithme FTF12N2L et sa version par blocs ont exactement les mêmes performances sur les signaux de parole que l'algorithme FTF8L et sa version par bloc.

Les figures III.32 et III.33 montrent que dans ces deux cas et pour une taille du bloc $M=16$ (avec $L=1024$ points), l'optimum est atteint presque à la fin de l'expérience (3.2 secondes) pour la version par bloc est de 26000 itérations environ (1.65 secondes) pour sa version originale. Ceci montre la dégradation de la vitesse de convergence de l'algorithme étudié. On note que cette dégradation est proportionnelle à la taille du bloc M .

III.2.6.4 Algorithme BFNS

Les paramètres de cette simulation sont :

Taille du filtre adaptatif : L

Taille des prédicteurs : N

Taille du Bloc : M

Pas d'adaptation des prédicteurs directe et retour : $\mu^f = \mu^b$

Pas d'adaptation du filtre adaptatif : μ

- **Première application**

Le canal stationnaire utilisé est la réponse impulsionnelle de couplage acoustique mesurée dans une salle visioconférence de taille 2048 points.

			Signal parole 1			Signal parole 2		
L	N	M	$\mu^f = \mu^b$	μ	$EQM(dB)$	$\mu^f = \mu^b$	μ	$EQM(dB)$
512	15	16	0.05	0.0001	-55.0	0.05	0.0001	-37.3
1024	32	16	0.025	0.000085	-47.2	0.025	0.000035	-33.4
2048	64	32	0.0115	0.00002	-36.9	0.01	0.00001	-28.6

Tableau III.34 : $EQM(dB)$ pour des L donnés.
(cas d'une salle visioconférence)

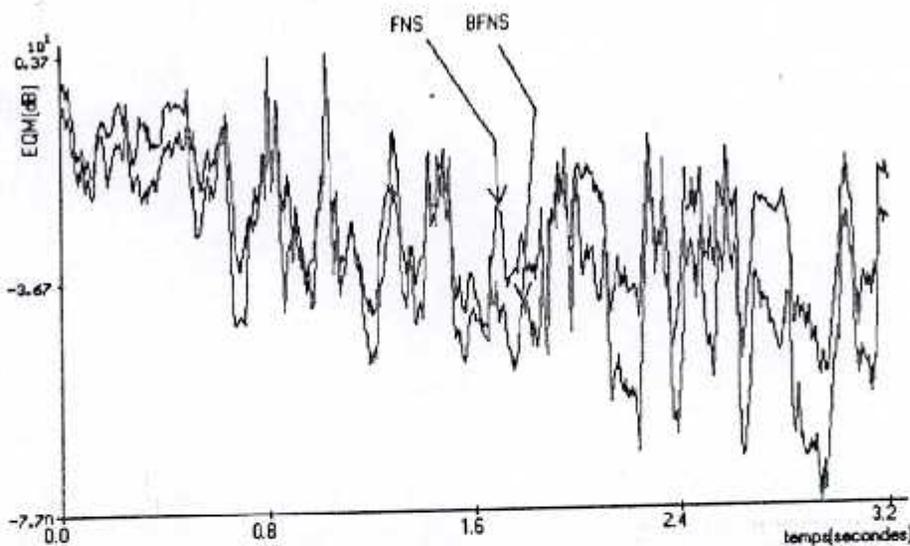


Figure III.34 : Evolution temporelle de l'EQM (énergie de l'erreur de filtrage) pour deux algorithmes : FNS et BFNS. $L = 512$, $M = 16$, pas de bruit en sortie.
(cas d'une salle visioconférence)

- **Deuxième application**

Le canal stationnaire est formé des 4096 premiers points d'une réponse impulsionnelle de couplage acoustique mesurée dans une salle audioconférence.

L	N	M	Signal parole 1			Signal parole 2		
			$\mu^f = \mu^b$	μ	$EQM(dB)$	$\mu^f = \mu^b$	μ	$EQM(dB)$
512	15	16	0.05	0.0002	-52.4	0.05	0.00008	-41.2
1024	32	16	0.05	0.0001	-45.3	0.025	0.000025	-33.1
2048	64	32	0.01	0.00002	-21.8	0.01	0.00001	-27.0

Tableau III.35 : $EQM(dB)$ pour des L donnés.
(cas d'une salle audioconférence)

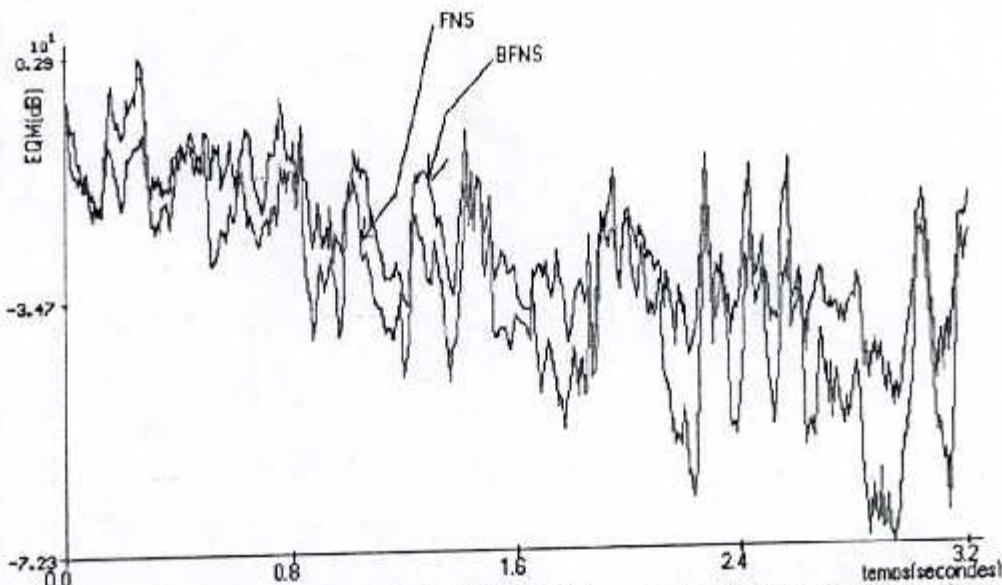


Figure III.35 : Evolution temporelle de l' EQM (énergie de l'erreur de filtrage) pour deux algorithmes : FNS et BFNS. $L = 512$, $M = 16$, pas de bruit en sortie.
(cas d'une salle visioconférence)

Les tableaux III.34 et III.35 montrent bien le bon comportement de l'algorithme FNS par bloc avec le signal parole1, si la taille du filtre transverse est de 512 points. Avec une dégradation croissante de son comportement dans les autres expériences faites avec les deux signaux de paroles. Cette dégradation est liée directement à la taille du bloc M durant lequel les coefficients du filtre transverse restent constants.

Les figures III.34 et III.35 montrent que même à la fin de l'expérience de durée de 3.2 secondes, l'algorithme FNS par bloc est encore en phase transitoire. Ce qui signifie que sa vitesse de convergence est dépendante de la nature du signal d'entrée.

III.3 Comparaison des performances des algorithmes de filtrage adaptatif à complexité réduite

Les résultats du tableau III.36 montrent les performances relatives de tous les algorithmes d'identification adaptative que nous avons présentés dans le chapitre II. Les abréviations qui apparaissent dans ce tableau sont données comme suit :

Dep : La dépendance de la vitesse de convergence de l'algorithme à la nature du signal d'entrée.

EQM : Erreur quadratique moyenne en régime permanent.

8 : Le meilleur algorithme.

1 : L'algorithme le plus mauvais.

Algorithmes	Vitesse de convergence	Dep.	EQM	Taille importante	Robustesse aux bruits	Stabilité numérique	Complexité des calculs
NLMS	5	Oui	5	5	5	8	6
FTF8L	8	Non	8	8	8	7	1
FTF12N2L	7	Non	7	7	7	7	3
FNS	6	Oui	6	6	6	8	5
BNLMS	1	Oui	1	1	1	8	8
BFTF	4	Non	4	4	4	7	2
BFTF12N2L	3	Non	3	3	3	7	4
BFNS	2	Oui	2	2	2	8	7

Tableau III.36 : Comparaison des performances relatives des algorithmes de filtrage adaptatif à complexité réduite avec des tailles optimales de L, N et M. (L : taille du filtre, N : taille des prédicteurs, M : taille du bloc.)

Dans cette étude comparative entre les performances des algorithmes adaptatifs, nous avons basé sur les points suivants :

- **Vitesse de convergence**

Nous avons remarqué que les deux algorithmes FTF8L et FNTF12N2L sont les meilleurs algorithmes de point de vue vitesse de convergence en phase initiale et en régime permanent. Les deux versions par bloc de ces deux algorithmes ont montré aussi une convergence plus ou moins bonne lorsque la taille du bloc est faible. Cela fait une bonne caractéristique des algorithmes appartenant à la famille des moindres carrés.

L'algorithme FNS vient dans la troisième position de point de vue vitesse de convergence qui est plus meilleure que celle de l'algorithme NLMS qui se trouve dans la quatrième position. Les deux versions par blocs de ces deux algorithmes ont montré une dégradation de la vitesse de convergence. Ceci est dû au blocage de l'adaptation des coefficients du filtre durant le bloc. Cette dégradation est proportionnelle à la taille du bloc.

- **Dépendance de la nature du signal excitant**

Nous avons remarqué que les deux algorithmes de type moindres carrés transversaux rapides FTF8L et FTF12N2L ont donné de bons résultats quelque soit la nature du signal

d'entrée (bruit blanc, bruit USASI, parole). Ces résultats montrent l'indépendance de ces algorithmes du signal d'excitation. Ce résultat reste valable pour leurs versions par blocs lorsque la taille de ces derniers est faible.

Les deux algorithmes de type gradient stochastique NLMS et BNLMS et ceux qui utilisent la structure LMS dans leurs parties de prédiction FNS et BFNS ont montré soit une faible ou une forte indépendance de la nature du signal excitant, ce qui a dégradé leurs performances avec les signaux de parole.

- **Erreur quadratique moyenne (EQM)**

Nous avons remarqué que tous les algorithmes de filtrage adaptatif présentés au-dessus ont donné de bonnes valeurs pour les EQMN en régime permanent lorsqu'on a un bruit blanc à l'entrée. Les algorithmes FTF8L et FNTF12N2L sont les seuls qui conservent cette propriété avec les signaux de parole sur une taille égale à 64k échantillons; ceci est un autre avantage de la famille des algorithmes des moindres carrés et de Newton rapides par rapport à la famille du gradient.

- **Taille importante du filtre transverse**

Les algorithmes de type moindres carrés transversaux rapides (FTF8L et FNTF12N2L) présentés dans le tableau III.36 se comportent bien avec des filtres de taille importante (supérieure à 1000 points). Alors que les autres algorithmes ont présenté une dégradation proportionnelle à la taille du filtre transverse.

- **Robustesse aux bruits**

Les quatre versions d'algorithmes par blocs ont montré une grande sensibilité aux bruits par rapport à leurs versions originales qui se sont avérées plus robustes. Les résultats obtenus (les EQMN) avec ces dernières sont de l'ordre de grandeur du bruit ajouté en sortie si le RSB est supérieur ou égale à 30dB. Tous les autres algorithmes présentés dans le tableau III.36 se dégradent d'une manière significative et on peut même dire qu'ils ne convergent pas avec les signaux bruités d'un niveau inférieur à 30dB.

- **Stabilité numérique**

Lors de toutes les simulations faites avec les algorithmes adaptatifs présentés dans le tableau III.36, nous n'avons rencontré aucun problème d'instabilité numérique. Dans le cas des algorithmes des moindres carrés on doit bien choisir le facteur d'oubli exponentiel et la constante d'initialisation des variances d'erreurs de prédiction pour assurer un bon fonctionnement de ces algorithmes à long terme sans rupture causée par l'accumulation des erreurs numériques.

- **Complexité des calculs**

Les quatre versions d'algorithmes par bloc ont présenté une complexité très réduite par rapport à leurs versions originales lorsque la taille du bloc est grande. Un compromis dans le choix de la taille du bloc et de la taille des prédicteurs aller et retour (dans le cas de l'algorithme BFNTF12N2L) doit être effectué afin d'avoir un algorithme qui a les performances des algorithmes des moindres carrés rapides et la complexité du gradient.

III.4 Conclusion

En conclusion, on peut dire que tous les algorithmes de filtrage adaptatif que nous avons présenté se comportent bien avec le bruit blanc comme signal d'excitation lorsque la taille du filtre à identifier est faible. Une dégradation des performances de la vitesse de convergence des algorithmes par blocs est remarquée dans leurs comportements avec des tailles du filtre adaptatif grandes (exemple 2048 points). Cette dégradation devient plus importante avec des valeurs grandes de la taille M du bloc. Nous avons remarqué également que l'identification des systèmes avec ces algorithmes nécessite que la taille du filtre transverse soit supérieure ou égale à la taille réelle du modèle à identifier.

Tous les algorithmes présentés ont montré une grande sensibilité aux bruits additifs. Cette dernière dégrade significativement leurs performances lorsque le RSB du milieu est inférieure à 30 dB. Une grande ressemblance dans les performances des algorithmes FTF12N2L et FTF8L est remarquée lorsque la taille des prédicteurs aller et retour est supérieur ou égale à la taille réelle du modèle à identifier. Ce qui montre l'intérêt de cet algorithme dans ses applications de l'annulation d'écho.

Pour un signal non stationnaire à l'entrée comme la parole, les deux algorithmes FTF8L et FTF12N2L sont avérés plus robuste aux non stationnarité du signal d'entrée, et ils ont montré la plus grande vitesse de convergence par rapport à tous les autres algorithmes d'identification adaptative. Cela est dû essentiellement à l'indépendance des performances des deux algorithmes, de la famille des moindres carrés, de la nature du signal d'entrée. Tous les autres algorithmes et même ceux par blocs, ont montré soit une faible ou forte indépendance de la nature du signal excitant, ce qui a dégradé sensiblement leurs performances de l'identification avec des signaux de la parole.

Une légère dégradation de la vitesse d'adaptation est remarquée avec les deux algorithmes (FTF8L et FTF12N2L) sur l'identification des réponses impulsionnelles de grandes tailles (exemple : 2048 points), cela est dû essentiellement et pour la simple raison aux choix du facteur d'oubli exponentiel qui doit être choisi très proche de 1, surtout pour l'identification de systèmes de grandes tailles, afin de répondre aux conditions de la stabilité numérique. Ce choix limite la vitesse de convergence.

Une grande similitude entre les deux algorithmes des moindres carrés FTF8L et FTF12N2L a été remarquée sur tous les côtés, et en plus, on note que l'algorithme FTF12N2L offre un avantage supplémentaire et qui est très important, cet avantage se présente dans sa faible complexité des calculs qui peut devenir de l'ordre de celle de l'algorithme du gradient stochastique normalisé NLMS si on choisit la taille des prédicteurs aller et retour, très faible devant la taille du filtre transverse à identifier.

De ce fait, on peut dire que le meilleur algorithme d'identification adaptative qui présente le meilleur rapport qualité/coût est le FTF12N2L avec les signaux stationnaire ou non stationnaire, ces performances peuvent être résumées comme suit :

- Complexité très réduite : $(2L+12N)$ MADPR. (multiplication et division par pas récursif).
- Stable numériquement (grâce à la sixième version de stabilisation numérique [14]).
- Robuste aux bruits.
- Très bon comportement avec les signaux de la parole.

Chapitre IV

IMPLANTATION DES ALGORITHMES D'IDENTIFICATION ADAPTATIVE A COMPLEXITE REDUITE SUR LA CARTE OROS-AU32

IV.1 Introduction

Le premier microprocesseur spécialement conçu pour le traitement de signaux analogiques était le 2920, présenté en 1976 par INTEL. Dessiné par le père du microprocesseur (en 1971) M.E.Hoof, ce circuit intégrait une unité de calcul spécialisée ainsi qu'un convertisseur A/N et N/A [17].

Malgré son intérêt, son circuit n'a connu qu'un succès d'estime, à cause de ses limitations technologiques :

- Alimentation bitensions,
- Absence de multiplieur intégré,
- Entrée/sortie analogique,...etc.

C'est au début des années 1980 que sont apparus les premiers microprocesseurs spécialisés réellement exploitables, intégrant sur une même puce :

- Une unité arithmétique en virgule fixe (au format 16bits),
- Un multiplieur câblé,
- Et des entrées/sorties numériques.

Les représentants les plus connus de cette catégorie sont les μ PD 7720 (NEC) et les TMS 32010 (TEXAS INSTRUMENT) qui ont longtemps servi de référence.

La principale différence entre ces deux processeurs réside dans leur architecture, le 32010 possédant encore une architecture de Von Neumann tandis que le 77230 opte déjà pour la structure Harvard [18].

Bénéficiant des progrès de la technologie, une deuxième génération de matériel est apparue en 1984/1985, adoptant systématiquement une architecture à bus séparés et reprenant en les améliorant les performances de la précédente génération :

- Arithmétique sur 32 bits avec des opérations en virgule fixe et en virgule flottante. —
- Capacité de mémoire accrue,
- Nombre important de registres généraux.

En exécutant des algorithmes en traitement de signal et en télécommunication où il y a des sommes et des produits répétitifs, un processeur DSP gagne 10 à 100 fois en performance que le processeur conventionnel. Parmi les produits de cette catégorie, on peut citer le μ PD 77230 (NEC) et le TMS32010 (TEXAS INSTRUMENT). —

Pendant un certain temps les supports logiciels étaient pauvres ou difficiles à mettre en œuvre. L'utilisation des processeurs était limitée aux processeurs conventionnels.

Enfin, la fin de l'année 1987 a vu l'apparition d'une troisième génération de processeur et des outils logiciels qui ont permis de faciliter l'utilisation des processeurs DSP au même titre que les processeurs conventionnels. Les performances ont été améliorées par :

- L'innovation architecturale avec la suppression de mémoires intégrées et la présence de caches,

- La multiplication des bus (bus parallèles) et l'amélioration des facultés de communication avec l'extérieur en augmentant la dynamique d'adressage (bus externes plus grands) et la mémoire interne
- Division par deux du temps de cycle grâce à l'utilisation de :
 - Générateurs d'adresse indépendants
 - Registres à décalage rapide (barrel shifters)
 - Registres multitâches
 - Jeux d'instructions plus larges
- adoption de boîtier évolué (PGA ou LCC)
- consommation minimale due à la technologie C-MOS

Les processeurs les plus performants, existant sur le marché actuellement sont les processeurs suivants :

- ADSP-2100 d'ANALOG DEVICES
- DSP-32 d'ATT
- MB-8764 de FUJITSU
- DSP-56000 de MOTOROLA
- LM-32900 de NATIONAL
- μ PD-77230 de NEC
- MSM-6992 de OKI SEMICONDUCTOR
- PCB-5011 de PHILIPS et SIGNETICS
- TS-68930 (a) et TS-68931 de THOMSON (MOSTEK)
- TMS 320C31 de TEXAS INSTRUMENT
- ZR-34161 de ZORAN

Dans ce chapitre nous allons présenter les résultats obtenus sur l'implantation des algorithmes de filtrage adaptatif sur la carte OROS-AU32 du microprocesseur TMS320C31 de Texas Instruments. Cette implantation se portera sur l'algorithme du gradient stochastique normalisé NLMS et l'algorithme de Newton rapide FNTF12N2L.

IV.2 Méthodologie de l'implantation sur la carte OROS AU-32

La carte OROS-AU32 est une carte format PC comportant un processeur de traitement du signal 32 bits flottant et 2 voies analogiques complètes d'entrée et sortie sur 16 bits. Elle est destinée à toutes les applications de traitement du signal sur PC exigeant une puissance de calcul importante et une grande qualité analogique.

Elle complète la gamme de cartes proposées par OROS dans le sens d'une augmentation des performances de calcul principalement et d'une plus grande souplesse de développement. Elle bénéficie totalement des qualités analogiques qui ont fait le succès des cartes OROS-AU21 et OROS-AU22, et qui ont même été encore améliorées. L'utilisation d'un circuit spécifique ASIC permet de rassembler sur une carte au format PC tout l'environnement nécessaire au développement d'application scientifique et industrielle performante.

La carte OROS-AU32 est divisée en deux parties, une partie analogique qui contient les chaînes d'entrées/sorties et la gestion des convertisseurs, et une partie numérique qui se compose principalement d'un processeur, la mémoire, et l'interface PC.

Une présentation détaillée du matériel et du logiciel de la carte est donnée en **annexe A**. La méthodologie de l'implantation sur la carte **OROS-AU32** est faite suivant les étapes suivantes :

- Elaboration de l'algorithme à implanter sur la carte en langage évolué (soit Microsoft C ou Borland C version 2.0 ou supérieure)
- Faire la compilation avec les outils de développement soft de la carte **OROS-AU32**, compilateur Texas Instruments version 5.0
- Elaboration d'un programme qui assiste le chargement du code exécutable de l'algorithme sur la carte **OROS-AU32**.
- Lancement de l'exécution de l'algorithme sur la carte à partir du PC
- Récupération des résultats sur le PC.

Trois types d'implantations seront présentés, la première est l'implantation en temps différé sur le processeur **TMS320C31**, la deuxième sur la carte et en fin on présentera une implantation qui utilise deux processeurs, celui du PC et du **TMS320C31**

IV.3 Acquisition des données analogiques par la carte OROS-AU32

Afin de vérifier le fonctionnement des chaînes d'acquisition et maîtriser les instructions propres aux canaux, nous avons commencé par réaliser un programme qui fait l'acquisition des signaux utiles à notre application et le stockage sur le disque dur du PC sans effectuer de traitement, puis nous présentons l'implantation des deux algorithmes de filtrage adaptatif NLMS et FNTF12N2L sur la carte **OROS-AU32**.

a) Initialisation de l'acquisition

Les différentes étapes de l'acquisition sont les suivantes.

- Initialisation des variables globales.
- Chargement du programme (partie DSP) dans le mémoire de la carte (Loadifnot_loaded). Ceci réalise aussi la mise en route du **TMS320C31**.
- Initialisation des caractéristiques de la chaîne analogique.
- (configuration, gain, correction d'offset, fréquence d'échantillonnage).
- Connexion du canal analogique de la carte.
- (connexion du canal mémoire de la carte (CH_Connect).
- Allocation de l'espace mémoire réservé au canal mémoire de la carte (BOARD_BU_CHANNEL).
- Allocation de l'espace mémoire réservé au canal analogique de la carte (BOARD_AN_CHANNEL).
- Ouverture des fichiers sur disque dur en écriture.
- Ouverture des canaux précédemment créés (CH_open).

- Connexion du transfert temps réel entre les canaux analogiques et la mémoire de la carte (BoardRealTimeConnect, voir **Annexe C**).
- Démarrage des convertisseurs (CH_Start).
- Lancement de la boucle d'acquisition côté PC.

b) boucle d'acquisition du PC

Il s'agit de la fonction qui réalise le transfert des échantillons de la carte vers le disque. Les différentes étapes d'acquisition sont les suivantes.

- Vérifié qu'il n'y a pas d'erreur de synchronisation sur la carte (BoardRealTimeConnect, voir **annexe C**).
- Observer le nombre d'échantillons disponibles dans la mémoire tampon (T2) de la carte (CH_Available).
- Minorer le nombre par le nombre d'échantillons que peut contenir la mémoire tampon (T1) du PC.
- Lire le bloc d'échantillons défini précédemment dans la mémoire de la carte et le stocker dans la mémoire tampon intermédiaire (T1) du PC (CH_Read).
- Si le nombre d'échantillons à acquérir est atteint, alors le processeur s'arrête.
- Si non, on recommence à partir du point 1.

b) Terminaison de l'acquisition

Quelle que soit la façon dont on détecte la fin de l'acquisition il faut arrêter les convertisseurs et libérer les ressources allouées. Ceci peut être fait dans une fonction de fin d'acquisition.

Les étapes sont les suivantes.

- arrêter les convertisseurs (CH_Stop).
- Une fois que le canal analogique ne contient plus d'échantillon, déconnecter le transfert «temps réel» entre ce canal et la mémoire de la carte et récupérer les échantillons acquis (BoardRealTimeDisconnect, voir **Annexe C**).
- Fermeture du canal analogique (CH_Close).
- Une fois qu'il n'y a plus d'échantillons dans la mémoire de la carte, arrêt de la boucle d'acquisition côté PC.
- Fermeture du second canal de la carte (CH_Close).
- Fermeture des fichiers ouverts sur le disque dur.
- Libération de l'espace mémoire réservé aux canaux de la carte (CH_Dispose).
- Libération de l'espace mémoire utilisé dans le PC.
- Déconnexion des canaux de la carte (CH_Disconnect).

Le principe de l'acquisition des signaux analogiques avec la carte OROS-AU32 est schématisé par la **figure IV.1**.

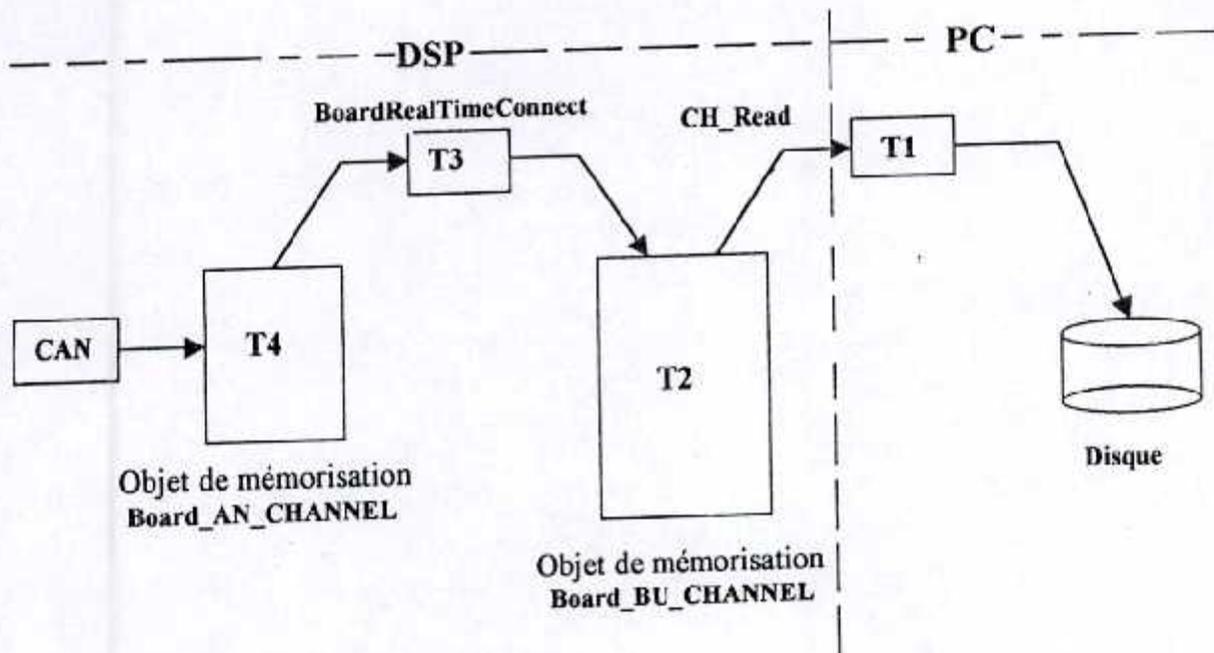


Figure IV.1 : Structure des canaux et objets de mémorisation dans une acquisition par la carte OROS-AU32 sur disque.

IV.4 Implantation temps différé.

a) principe

Afin de vérifier le bon fonctionnement de l'algorithme de filtrage adaptatif en cours de simulation (soit NLMS ou FNTF12N2L) sur la carte OROS-AU32, nous avons commencé par charger le code exécutable de l'algorithme sur la carte en utilisant l'instruction LoadIfNot_Loaded(nomfich). Cette instruction charge le fichier qui contient le code exécutable dans la mémoire de carte après avoir vérifié que ce fichier n'est pas présent dans sa mémoire. Le lancement de la fonction se fait à partir du PC. (Voir Annexe A).

Le test du fonctionnement de l'algorithme en cours de simulation se fait avec deux fichiers qui contiennent les signaux utiles au fonctionnement de l'algorithme en cours de simulation sur la carte OROS-AU32.

Le PC après avoir chargé le code exécutable du programme côté DSP dans la mémoire de la carte ouvre les deux fichiers de tests puis lit deux blocks de N échantillons de chaque fichier et les transfère vers le DSP qui effectue le traitement de cette trame de données. Une fois le traitement effectué le TMS320C31 renvoie un autre block de données qui représente les valeurs de l'énergie de l'erreur de filtrage normalisée par l'énergie du signal écho au PC qui les enregistre dans un troisième fichier, et cette boucle se répète jusqu'à la fin des deux fichiers de test. Les valeurs de l'indice de performance utilisé (EQMN) est récupéré dans le troisième fichier au niveau du PC.

Le principe de cette implantation avec les deux algorithmes de filtrage adaptatif NLMS et FNTF12N2L est schématisé par la figure IV.2 suivante. Les opérations arithmétiques sont réalisées en virgule flottante 32 bits.

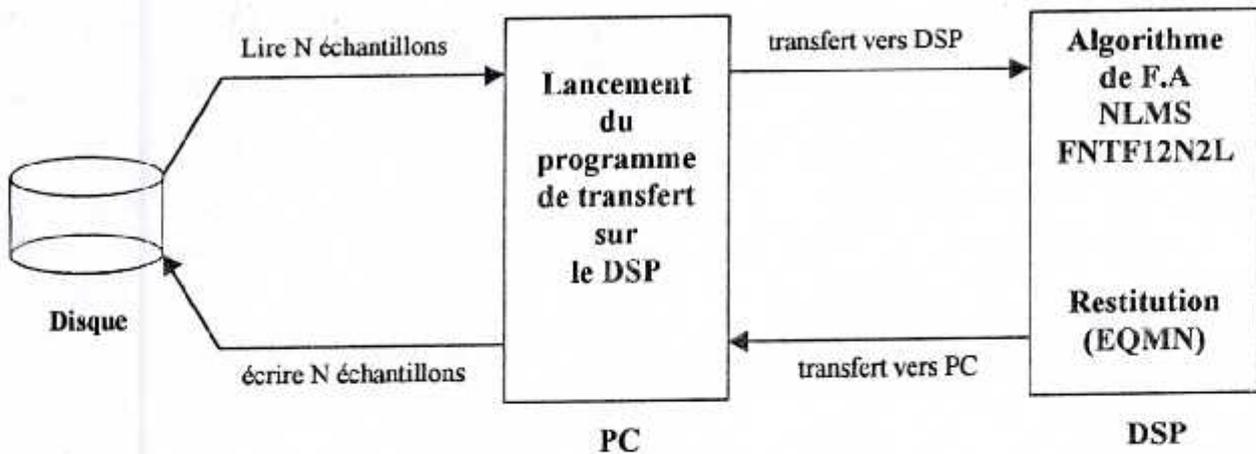


Figure IV.2 : Schéma de l'implantation des algorithmes NLMS et FNTF12N2L Sur la carte OROS-AU32 en temps différé.
EQMN : l'énergie de l'erreur de filtrage adaptatif normalisée par le signal écho.

b) Résultats de l'implantation

Pour le signal d'entrée x_t on prendra un bruit blanc gaussien de variance $\sigma_x^2 = 0.15$. Le signal de sortie y_t est le signal x_t transformé par le canal acoustique d'un habitacle de voiture comportant 1024 points. (voir figure III.2).

Les résultats qui ont été obtenus par simulation sur le PC ainsi que ceux obtenus sur le processeur du traitement de signal numérique le TMS320C31 en temps différé avec les deux algorithmes NLMS et FNTF12N2L sont donnés par les deux tableaux suivants.

Les paramètres de cette simulation sont :

Tableau IV.1 : (cas de l'algorithme NLMS)

L : taille du filtre transversal de l'algorithme.

μ : pas d'adaptation de l'algorithme.

E_0 : constant d'énergie de normalisation minimale de l'algorithme.

Tableau IV.2 : (cas de l'algorithme NLMS)

L : taille du filtre transversal de l'algorithme.

N : taille des prédicteurs aller et retour.

E_0 : constante d'énergie minimale de l'algorithme.

λ : facteur d'oubli exponentiel qui assure la stabilité de l'algorithme.

Les deux tableaux IV.1 et IV.2 montrent bien la grande similitude entre les résultats obtenus avec les deux algorithmes NLMS et FNTF12N2L sur le PC et sur le processeur TMS320C31, cela est dû essentiellement à la précision de calcul qui est la même sur les deux processeurs et à la nature de l'implantation qui est faite en virgule flottante qui affirme l'obtention du même résultat.

L	$EQMN(dB)$	
	Résultats du PC	Résultats du TMS320C31
32	-87.2	-87.1
128	-87.5	-87.2
1024	-84.3	-84.5

Tableau IV.1 : Comparaison entre les résultats du PC et du TMS320C31 ($\mu = 1.0$, $E_0 = 0.0025$)

L	N	E_0	λ	$EQMN(dB)$	
				Résultats du PC	Résultats du TMS320C31
32	8	1.0	0.987	-87.20	-87.25
128	64	1.0	0.9985	-87.40	-88.41
256	128	1.0	0.9989	-85.16	-85.25

Tableau IV.2 : Comparaison entre les résultats du PC et du TMS320C31 avec l'algorithme FNTF12N2L.

Les deux figures IV.3 et IV.4 montrent le cas où il n'y a pas de bruit ajouté en sortie, on remarque la superposition presque totale des deux tracés de l'énergie de l'erreur de filtrage normalisée obtenue sur le PC et sur le TMS320C31 pour les deux algorithmes NLMS et FNTF12N2L pour la taille 128 points. Ce qui montre encore que les vitesses de convergence des algorithmes ne dépendent pas de la nature du processeur utilisé en simulation. Dans ce type d'implantation et pour une fréquence d'échantillonnage égale à 16kHz, on a réalisé l'implantation des deux algorithmes avec une taille du filtre transversal maximale égale à 1024.

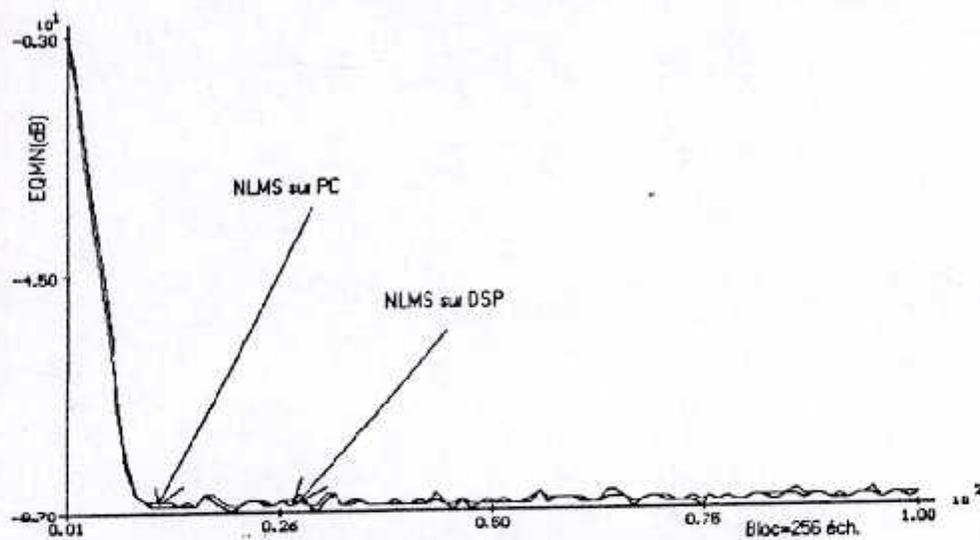


Figure IV.3 : Comparaison de la vitesse de convergence entre l'EQMN obtenue sur le PC et sur le TMS320C31 avec l'algorithme NLMS, $L = 128$, $\mu = 1.0$.

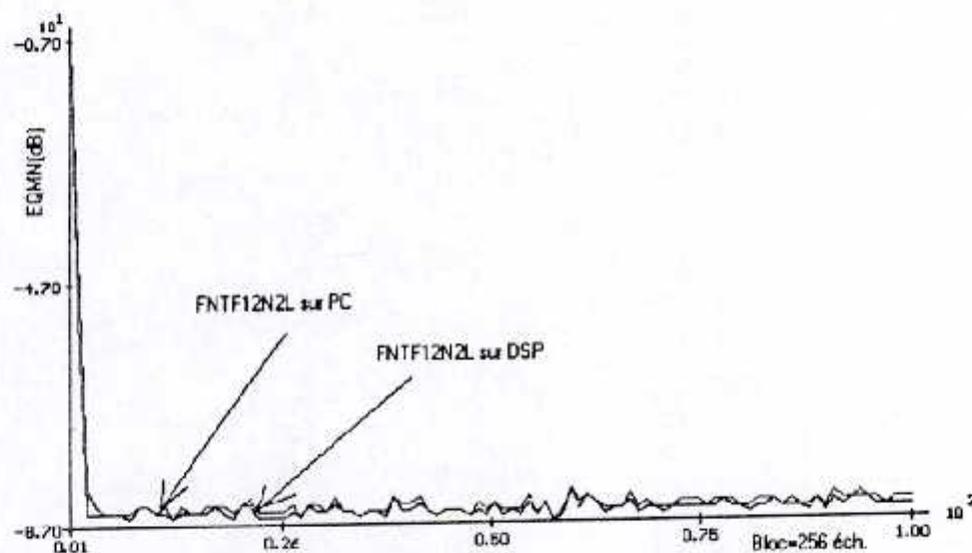


Figure IV.4 : Comparaison de la vitesse de convergence entre l'EQMN obtenue sur le PC et sur le TMS320C31 avec l'algorithme FNTF12N2L, $L = 128$, $\mu = 1.0$.

IV.5 Acquisition et calculs sur DSP

a) Principe

Pour l'acquisition des signaux de simulation en temps réel nous avons suivi le même principe de gestion des buffers qui a été présenté dans l'annexe A. On utilise un canal de type analogique en mode producteur, en définissant un tel canal, les échantillons convertis par la chaîne d'entrée sont directement transmis vers l'objet de mémorisation associé à ce canal. Cet objet de mémorisation réside sur la carte (BOARD_AN_LOCAL). Cette mémoire tampon

est de type FIFO pour éviter l'écrasement des échantillons. Une connexion temps réel (BoardRealTimeConnect) entre le canal analogique et un deuxième canal de type consommateur résidant sur la carte (Board_BU_LOCAL), permet le transfert des échantillons mémorisés par le premier canal.

Cette opération nous permis de récupérer les échantillons directement sur le canal consommateur, la taille de l'objet de mémorisation associé à ce dernier canal doit être supérieure au double de la taille du bloc d'échantillons traités par l'algorithme en cours de simulation ($2 \times N$). Dès que les N premiers échantillons sont présents dans le buffer, on procédera au traitement de ce bloc tandis que ce même buffer se charge grâce à la connexion temps réel avec le canal analogique. Et ceci se répète jusqu'à la fin de l'opération de traitement de ces flots continus des échantillons d'entrée. Et à la fin du traitement on remplit un autre buffer défini en producteur par les valeurs de l'indice de performance utilisé (EQMN), ces valeurs vont être récupérés au niveau du PC.

Le principe de cette implantation avec les deux algorithmes de filtrage adaptatif est schématisé et donné par la figure IV.5.

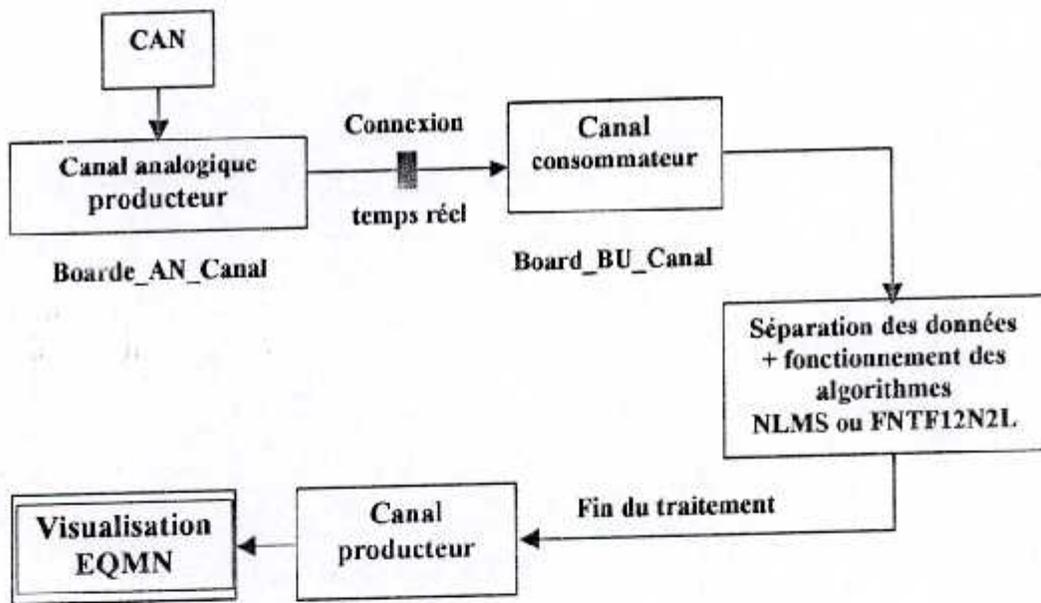


Figure IV.5 : Schéma de l'implantation en temps réel sur la carte OROS-AU32

b) Résultats de l'implantation

pour l'acquisition des signaux utiles au fonctionnement de l'algorithme en cours de simulation, nous avons procédé de la même manière que dans l'Annexe A.

Après l'acquisition des signaux on procède à la séparation des échantillons de la voie1 et de la voie2 qui correspondent aux échantillons $x(\text{voie1})$ et $y(\text{voie2})$. Ces deux échantillons sont mis sur un mot de 32 bits en mémoire par le TMS320C31. Les 16 bits du poids fort de ce mot

représentent l'échantillon x de la voie 1 et les 16 bits du poids fort représentent l'échantillon y de la voie 2, ces deux échantillons sont en complément à deux.

Les résultats qui ont été obtenus sur le TMS320C31 avec les deux algorithmes d'identification adaptative NLMS et FNTF12N2L sont donnés par les deux tableaux IV.3 et IV.4. (même paramètres que dans IV.2).

Les deux tableaux IV.3 et IV.4 montrent bien la dégradation importante des valeurs de l'énergie de l'erreur de filtrage normalisé (EQMN). Cette dégradation est plus importante avec l'algorithme NLMS qu'avec l'algorithme FNTF12N2L. Cette dégradation est due essentiellement au bruits présents sur les convertisseurs analogique/numérique, c'est-à-dire le bruit électronique qui existe sur les deux voies analogiques de la carte. Cette méthode d'implantation des algorithmes adaptatifs est réalisée avec une taille du filtre maximale égale à 256 points avec une fréquence d'échantillonnage des convertisseurs égale à 16kHz.

Taille du filtre L	E_0	μ	EQMN(dB)
32	0.0025	1.0	-49.26
64	0.0025	1.0	-52.30
128	0.0025	1.0	-53.02

Tableau IV.3 : l'EQMN obtenues avec l'algorithme NLMS pour des L donnés.

Taille du filtre L	Taille des prédicteurs N	E_0	λ	EQMN(dB)
32	8	1.0	0.987	-53.26
64	32	1.0	0.993	-49.30
128	64	1.0	0.9985	-54.10

Tableau IV.4 : l'EQMN obtenues avec l'algorithme FNTF12N2L pour des L données.

Les deux figures IV.6 et IV.7 montrent une dégradation importante de la performance asymptotique des deux algorithmes NLMS et FNTF12N2L, cette dégradation est plus importante avec l'algorithme du gradient stochastique normalisé (NLMS), ce qui montre encore une fois sa sensibilité aux bruits.

On remarque aussi que l'énergie de l'erreur de filtrage normalisée reste de l'ordre de grandeur du bruit qui existe sur les convertisseurs. Avec une fréquence d'échantillonnage égale à 16kHz, nous avons implanté les deux algorithmes en temps réel avec une taille du filtre transverse maximale égale à 256 points.

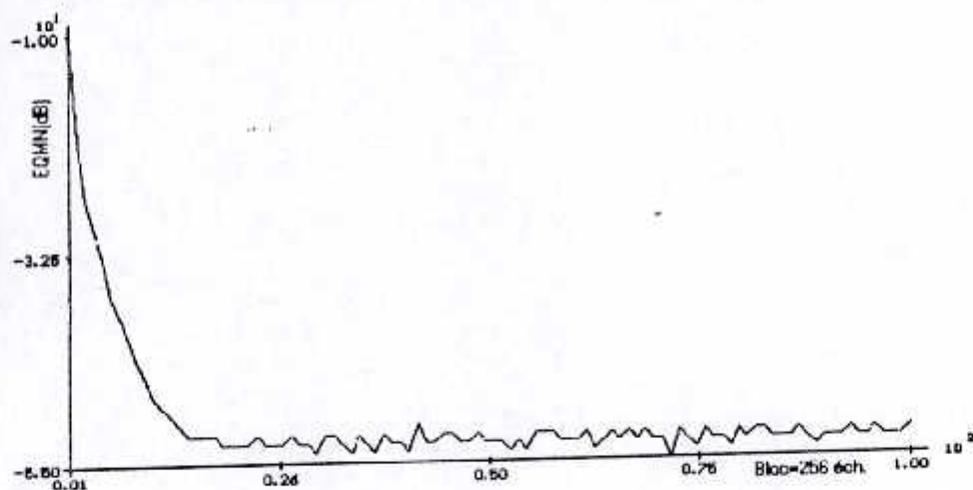


Figure IV.6 : Evolution temporelle de l'EQMN obtenue avec l'algorithme NLMS sur le TMS320C31, $L = 128$, $\mu = 1.0$, $E_0 = 0.0025$.

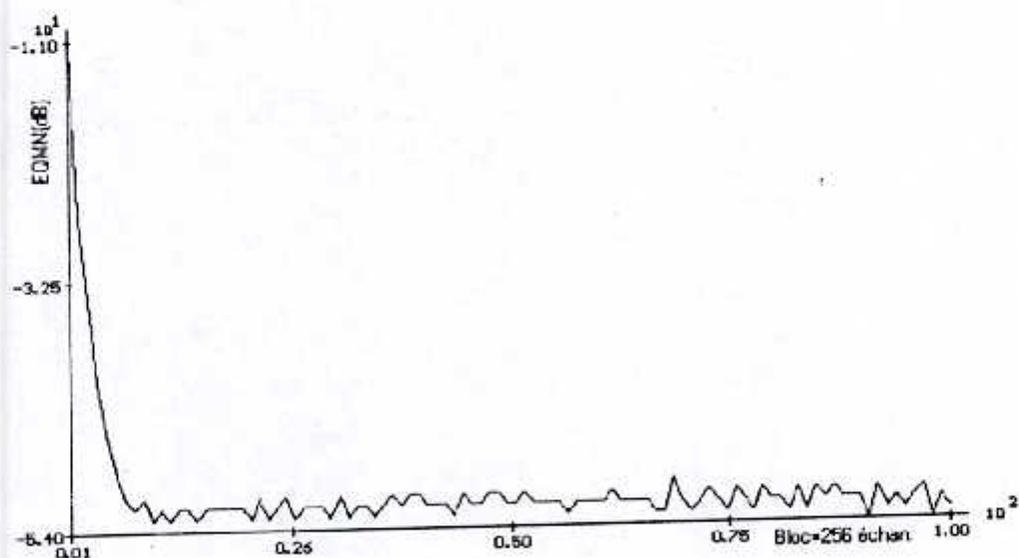


Figure IV.7 : Evolution temporelle de l'EQMN obtenue avec l'algorithme FNTF12N2L sur le TMS320C31, $L = 128$, $N = 64$, $\lambda = 0.9989$.

IV.6 Filtrage en temps réel sur PC

a) Principe

Le principe de cette méthode d'implantation est basé sur l'utilisation de deux processeurs. Le premier processeur est celui du DSP TMS320C31, qui permet l'acquisition des signaux nécessaires aux fonctionnements des algorithmes de filtrage adaptatif en cours de simulation à partir des convertisseurs analogiques/numériques d'entrée de la carte OROS-AU32.

Le deuxième processeur est celui du PC qui permis de son côté de faire fonctionner l'algorithme en cours de simulation avec les signaux issus de la carte OROS-AU32.

Pour l'exécution de ces deux algorithmes selon ce principe, on suivra le même schéma que celui utilisé pour une exécution sur la carte OROS-AU32 (voir figure IV.5).

Les modifications à introduire se situent au niveau des canaux qui seront déclarés en "Remote" : c'est-à-dire que l'accès aux canaux s'effectuera par le PC. Les instructions d'écriture ou de lecture se situeront dans le programme côté PC.

Dans ce programme on commencera par charger une fonction sur la carte, cette fonction ne contient pas forcément d'instructions à exécuter, son rôle est de déterminer la répartition de la mémoire de la carte et de mettre le processeur TMS320C31 en marche. La suite du programme est semblable au programme implanté sur la carte OROS-AU32 : (réglage de la chaîne analogique, initialisation et connexion des canaux, mise en marche des convertisseurs, lecture d'un bloc d'échantillons, séparation des données et traitement sur PC).

b) Résultats de l'implantation sur le PC.

Pour l'acquisition des signaux utiles au fonctionnement de l'algorithme en cours de simulation (soit NLMS ou FNTF12N2L), nous avons procédé de la même manière que dans le cas de l'acquisition sur la carte.

Après l'acquisition des signaux on procède à la séparation des échantillons de la voie1 et de la voie2 qui correspondent aux échantillons $x(\text{voie1})$ et $y(\text{voie2})$ (voir paragraphe III.2). Cette fois la séparation aura lieu sur le PC. Les 16 bits du poids fort d'un mot de 32 bits sur le TMS3230C31 représentent l'échantillon x de la voie 1 et les 16 bits du poids fort représentent l'échantillon y de la voie 2, ces deux échantillons sont en complément à deux.

Les résultats qui ont été obtenus sur le TMS320C31 avec les deux algorithmes d'identification adaptative NLMS et FNTF12N2L sont donnés par les deux tableaux IV.5 et IV.6. (même paramètres que précédemment).

Les deux tableaux IV.5 et IV.6 montrent bien que l'identification avec ce type d'implantation est possible avec une dégradation importante des valeurs de l'énergie de l'erreur de filtrage normalisée (EQMN) qui est utilisé comme indice de performance. Cette dégradation est remarquée dans les deux cas que ce soit avec l'algorithme NLMS ou avec l'algorithme FTF12N2L. Ces résultats sont très proches des résultats obtenus avec l'implantation sur la carte.

Taille du filtre L	E_0	μ	EQMN(dB)
32	0.0025	1.0	-52.0
64	0.0025	1.0	-59.30
128	0.0025	1.0	-53.02

Tableau IV.5 : l'EQMN obtenues avec l'algorithme NLMS pour des L donnés.

Taille du filtre L	Taille des prédicteurs N	E_0	λ	$EQMN(dB)$
32	8	1.0	0.987	-54.20
64	32	1.0	0.993	-61.30
128	64	1.0	0.9985	-57.10

Tableau IV.6 : l'EQMN obtenues avec l'algorithme FNTF12N2L pour des L données

Les deux figures IV.8 et IV.9 montrent une dégradation importante de la performance asymptotique des deux algorithmes NLMS et FNTF12N2L. On remarque aussi que l'énergie de l'erreur de filtrage normalisée reste de l'ordre de grandeur du bruit présent sur les convertisseurs (bruit électronique). Ce type d'implantation a permis l'exécution de l'algorithme NLMS et FTF12N2L en temps réel sur le PC avec des tailles maximales égales à 30 points avec l'NLMS et 15 points avec le FTF12N2L avec une fréquence d'échantillonnage égale à 16 kHz.

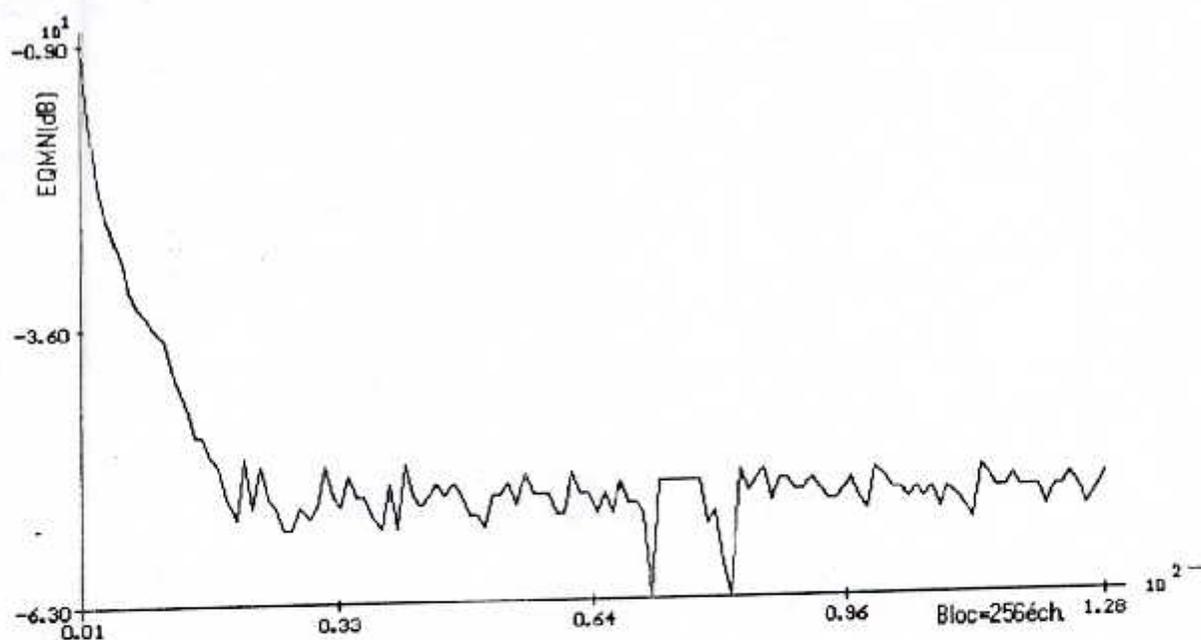


Figure IV.8 : Evolution temporelle de l'EQMN obtenue avec l'algorithme NLMS sur le PC +TMS320C31, $L = 128$, $\mu = 1.0$

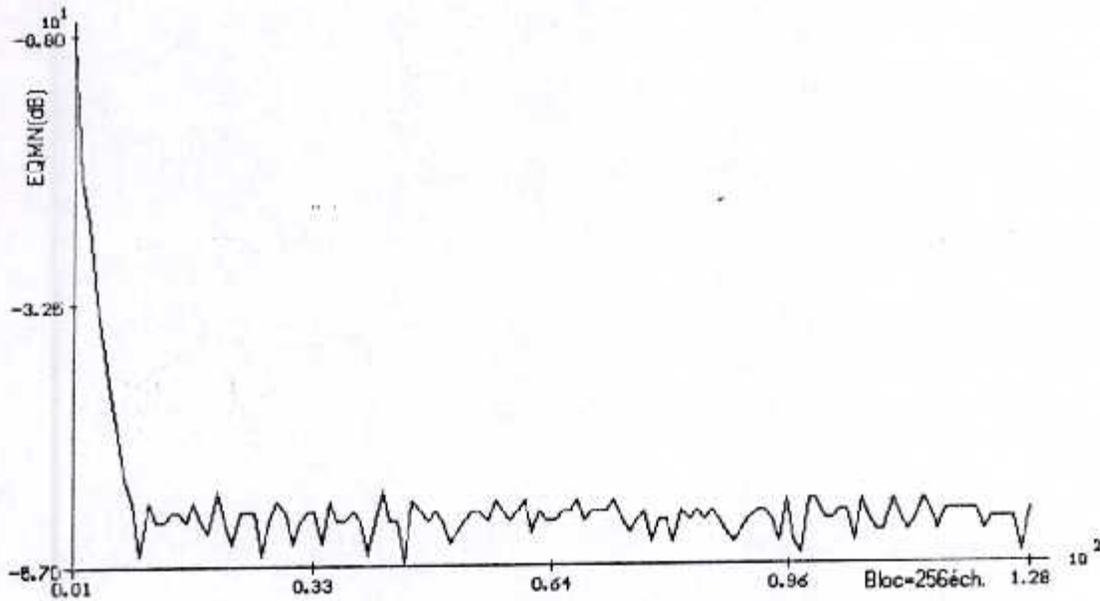


Figure IV.9 : Evolution temporelle de l'EQMN obtenue avec l'algorithme FNTF12N2L sur le PC + TMS320C31, $L=128$, $N=64$, $\lambda = 0.9989$

IV.7 Conclusion

En conclusion on peut dire qu'avec une fréquence d'échantillonnage de 16 kHz et un temps de cycle de 60 ns pour le TMS320C31, nous avons réalisé trois types d'implantation des deux algorithmes de filtrage adaptatif (algorithme du gradient stochastique normalisé NLMS et de Newton rapide FTF12N2L) sur le processeur TMS320C31 de Texas Instruments en temps différé et en temps réel et sur les deux processeurs TMS320C31 et celui du PC (pentium) en temps réel.

Les résultats de l'implantation en temps différé sont exactement identiques à ceux obtenus sur le PC. Cela est dû à la nature de l'implantation qui est faite en virgule flottante et à la précision des calculs qui est la même sur les deux processeurs.

Le deuxième type d'implantation a été présenté dans le but de faire fonctionner les deux algorithmes cités au-dessus en temps réel sur la carte OROS-AU32, dans ce type d'implantation le processeur TMS320C31 fait à la fois l'acquisition des signaux analogiques à partir des convertisseurs analogique ainsi que l'exécution des algorithmes adaptatifs. Cette implantation a permis l'exécution des deux algorithmes avec une taille maximale de 256 points du filtre transverse à la fréquence d'échantillonnage de 16kHz.

Le troisième type d'implantation des algorithmes adaptatifs que nous avons proposé et présenté est basé sur l'utilisation de deux processeurs dans le fonctionnement de notre application, l'un est le TMS320C31 qu'on l'utilise pour l'acquisition des signaux analogiques et l'autre est celui du PC qui sert à l'exécution des deux algorithmes. Ce type d'implantation a permis d'exécuter les deux algorithmes en temps réel sur le PC et sur la carte OROS-AU32 avec des tailles du filtre transverse appropriée. Ces dernières sont liées intimement à la fréquence d'échantillonnage des convertisseurs analogique/numérique et à la vitesse d'exécution des algorithmes adaptatifs sur le PC.

**CONCLUSION GENERALE
&
PERSPECTIVES**

Conclusion générale et perspectives

L'étude que nous avons présentée concerne l'utilisation pratique des algorithmes de filtrage adaptatif pour l'identification des filtres à réponses impulsionnelles acoustiques. Ces filtres apparaissent dans des applications d'annulation d'écho acoustique et de débruitage de la parole pour certains nouveaux terminaux des télécommunications, comme la téléconférence et le téléphone "mains libres".

Dans ce contexte, nous avons proposé trois nouvelles versions d'algorithmes de filtrage adaptatif transverse qui appartiennent à la famille d'algorithmes dite "par bloc". Cette dernière qui est basée sur l'actualisation des coefficients du filtre adaptatif transverse une fois par bloc de taille M échantillons, permet une importante réduction de complexité des calculs.

Ces trois versions sont : la version d'algorithme des moindres carrés transversaux rapides numériquement stabilisée par bloc (BFTF8L), la version d'algorithmes de Newton rapide par bloc (BFTF12N2L) et la version de Newton rapide simplifiée (BFNS). Ces versions sont inspirées de la structure par bloc de l'algorithme du gradient stochastique (BNLMS).

L'étude par simulation de ces trois versions d'algorithmes a montré leurs bonnes performances sur les signaux stationnaires ou non stationnaire (parole). Une légère dégradation de la vitesse de convergence de ces algorithmes est remarquée. Cette dernière devient importante lorsque la taille du bloc d'échantillons M devient grande.

Nous avons également proposé une nouvelle structure de la partie de prédiction de l'algorithme de Newton rapide FTF12N2L en utilisant la nouvelle version de stabilisation numérique qui a été proposée dernièrement par M. AREZKI [14]. Cette dernière qui est basée sur la rétroaction des erreurs numériques dans le calcul des variables de prédiction retour a permis un fonctionnement à long terme de l'algorithme sans aucun problème de divergence.

Cet algorithme (FTF12N2L) a montré à la fois des performances identiques à celles de l'algorithme des moindres carrés (FTF8L) lorsque l'ordre de prédiction est supérieur à l'ordre réel du modèle à identifier, et une complexité des calculs proche de celle du gradient lorsque la taille des prédicteurs est faible devant celle du filtre transverse. Cet algorithme a été sélectionné avec l'algorithme NLMS pour une implantation sur le processeur spécialisé TMS320C31 de Texas Instruments de la carte OROS-AU32.

L'annulation d'écho avec les différents algorithmes cités au-dessus n'a posé aucun problème particulier en pratique avec la parole comme signal d'excitation.

Enfin, nous avons présenté les résultats obtenus avec trois types d'implantation sur le processeur spécialisé TMS320C31 des deux algorithmes d'identification adaptative NLMS et FTF12N2L.

La première implantation est en temps différé sur le TMS320C31. Les résultats obtenus sont identiques à ceux obtenus sur le PC avec les deux algorithmes. Ce qui montre que les performances des deux algorithmes ne dépendent pas de la nature du processeur utilisé.

La deuxième implantation est sur la carte, c'est-à-dire le processeurs TMS320C31 fait à la fois l'acquisition des signaux analogiques depuis le convertisseur analogique/numérique et l'exécution de l'algorithme qui se trouve dans sa mémoire. Ce type d'implantation a permis l'exécution des deux algorithmes en temps réel avec une taille du filtre transverse maximale égale à 256 points et une fréquence d'échantillonnage des convertisseurs égale à 16kHz. On note que la taille maximale du filtre transverse de l'algorithme qu'on veut exécuter en temps réel sur des processeurs de traitement numérique du signal spécialisés est inversement proportionnelle à la fréquence d'échantillonnage et à la complexité de calcul de l'algorithme.

Le troisième type d'implantation est basé sur l'utilisation du processeur TMS320C31 pour l'acquisition des signaux analogiques, alors que l'exécution de l'algorithme s'effectuera sur le PC. Ce type d'implantation a permis l'exécution en temps réel des deux algorithmes avec des tailles du filtre transverse appropriée. On note que ce type d'implantation est limité par la vitesse de l'exécution de l'algorithme sur le PC qui dépend de sa complexité et par la fréquence d'échantillonnage des signaux d'entrée.

Comme perspectives de recherches dans le prolongement du travail décrit dans cette thèse, nous proposons l'implantation en virgule fixe de tous les algorithmes que nous avons proposés et présentés sur le processeur spécialisé en traitement du signal TMS320C31 de la carte OROS-AU32.

Bibliographie

- [1] A.Gilloire et J.P.Julien "L'acoustique des salles dans les télécommunications", L'écho des recherches, No 127. 1^{er} trimestre 1987.
- [2] J.P.Julien, A.Gilloire et A.Saliou, "Mesure de réponses impulsionnelles en acoustique", Note technique CENT NT/LAA/TSS181, Juillet 1984.
- [3] M.DJENDI, A.BENALLAL "Réduction de la Complexité des Calculs des Algorithmes d'Identification adaptative", Conférence Maghrébine en Génie Electrique CMGE'99, Université Mentouri, Constantine, Algérie 4-6 Décembre1999.
- [4] J.P.Julien, "Acoustique des salles, prise et restitution du son, analysées à partir de la notion de canal acoustique", Note technique CENT NT/LAA/TSS/128, Août 1982.
- [5] A Benallal, "Etude des algorithmes MCR et application à l'identification de réponses impulsionnelles acoustique", Thèse de Doctorat, université de Rennes I, France Jan. 1989.
- [6] A.Benveniste, M.Metivier et P.Priouret, "Algorithmes adaptatifs et approximations stochastiques: Théorie et Applications", Dans Techniques stochastiques, MASSON 1987.
- [7] P.Huyes and Al, "Adaptative filters - a review of techniques", BT Technology Journ. Vol.10 No 1 Jan. 1996.
- [8] H.Schutze, Z.Ren, "Numerical characteristics of Fast Least Squares Transversal adaptation algorithms - A comparative study", Signal processing Vol.44, No 27, 1992.
- [9] M.G.Bellanger, "Engineering aspects of fast RLS algorithms in transversal adaptative filters", Proc. of IEEE ICASSP 1987, Dallas.
- [10] P.Fabre and C.Gueguen, "Fast RLS algorithms : Preventive divergence", Proc. of IEEE ICASSP 1985, Tampa.
- [11] G.Carayannis, D.Manolakis and N.Kalouptsidis, "A Unified View of Parametric Processing Algorithms for Prewindowed Signals", Signal Processing, vol-10, No 4, June 1986.
- [12] J.M.Cioffi and Th.Kailath, "Fast recursive least squares transversal filters for adaptative filtering", IEEE Trans. On Acoust. Speech and Signal Process. Vol-32, No 2, Apr.1984.
- [13] Panagiotis P. Mavridis and George V. Moustakidiss "Simplified Newton-Type Adaptive Estimation Algorithms", IEEE Transaction on signal processing, Vol.44, NO.8, August 1996.
- [14] M.AREZKI "Etude des Algorithmes des Moindres Carrés Transversaux Rapides en Présence de la Parole", Thèse de Magistère, Université de BLIDA, Algérie, Décembre 1997.

-
- [15] G.Moustakides, S.Theodoris "Fast Newton Transversal Filters – A New class of Adaptive Estimation Algorithms", IEEE Trans. Signal Processing, vol. ASSP-39, n°10,PP.2184-2193,Oct.1991
- [16] P.Petillon, A.Gilloire, S.Theodoridis "The Fast Newton Transversal Filter : An Efficient Scheme for Acoustic Echo Cancellation in Mobile Radio", IEEE Transaction on signal Processing, Vol.42,NO.3,March 1994
- [17] L. MEKALICHE "Annuleur d'écho acoustique", Mémoire de magistère, Université d' ORAN, Algérie 1993.
- [18] Vijary K. Madisetti "Digital Signal Processors, an introduction to rapid prototyping and design synthesis", Georgia Institute of Technologie", IEEE PRESS, Copyright 1995 by Buterworth-Heineman.
- [19] OROS-AU32 architecture, OROS-Mnuel AU32, Document 80 600 22F, juin 1996.
- [20] OROS-AU32lib Bibliothèque de fonctions, OROS-AU32lib, Document n°80 620 34F, mise à jour, 21juin 1996.
- [21] TMS320C3x/C4x Optimizing C compiler, User's Guide, Texas Instruments, juin 1992.
- [22] TMS320C3x/C4x Assembly language Tools, User's Guide, Texas Instruments, juin 1992.

ANNEXES

ANNEXE A

• **Présentation générale de la carte OROS-AU32**

La carte **OROS-AU32** est une carte format **PC** comportant un processeur de traitement du signal 32 bits flottant et 2 voies analogiques complètes d'entrée et sortie sur 16 bits. Elle est destinée à toutes les applications de traitement du signal sur **PC** exigeant une puissance de calcul importante et une grande qualité analogique.

Elle complète la gamme de cartes proposées par **OROS** dans le sens d'une augmentation des performances de calcul, principalement et d'une plus grande souplesse de développement. Elle bénéficie totalement des qualités analogiques qui ont fait le succès des cartes **OROS-AU21** et **OROS-AU22**, et qui ont même été encore améliorées.

L'utilisation d'un circuit spécifique **ASIC** permet de rassembler sur une carte au format **PC** tout l'environnement nécessaire au développement d'applications scientifiques et industrielles performantes.

La carte **OROS-AU32** est divisée en deux parties, une partie analogique qui contient les chaînes d'entrées/sorties et la gestion des convertisseurs, et une partie numérique qui se compose principalement processeur, de la mémoire, et l'interface **PC**.

• **Partie analogique**

La partie analogique de la carte **OROS-AU32** est un système complet d'acquisition et de restitution bivoie, avec une résolution de 16 bits pour des signaux dans la bande de fréquence 0-20kHz [19]. Le schéma synoptique présenté à la **figure 1**, décrit l'architecture générale de cette partie analogique.

a) **Chaque voie d'entrée comporte :**

- Une sélection d'entrée (continue/alternative) avec une tension maximale de 3.16V_{eff}.
- Un amplificateur de gain principal programmable par pas de 20 dB.
- Un système de correction d'offset qui permet de réinjecter à la sortie de l'amplificateur principale une tension de par pas de 0.5mv.

Un amplificateur de gain fin programmable dans la gamme 0-30 dB par pas de 1dB.

Un convertisseur analogique/numérique de type « DELTA-SIGMA » bivoie avec une résolution de 16 bits. Il inclut un filtre numérique dont la fréquence de coupure s'adapte automatiquement à la fréquence d'échantillonnage.

b) **Chaque voie de sortie comporte :**

- Un convertisseur numérique/analogique également de « DELTA-SIGMA ».

Un amplificateur adaptateur de gain fixe.

La fréquence de base de l'horloge d'échantillonnage est de 48kHz et peut être divisée par 1, 2, 3, 4, 5, 6, 8, 10, 2 ou 16.

L'utilisation de convertisseurs « DELTA-SIGMA » entraîne une adaptation automatique du filtrage anti-repliement quel que soit le facteur de division utilisé.

(10 dBV correspond à un niveau de 3,16 Volt efficace)

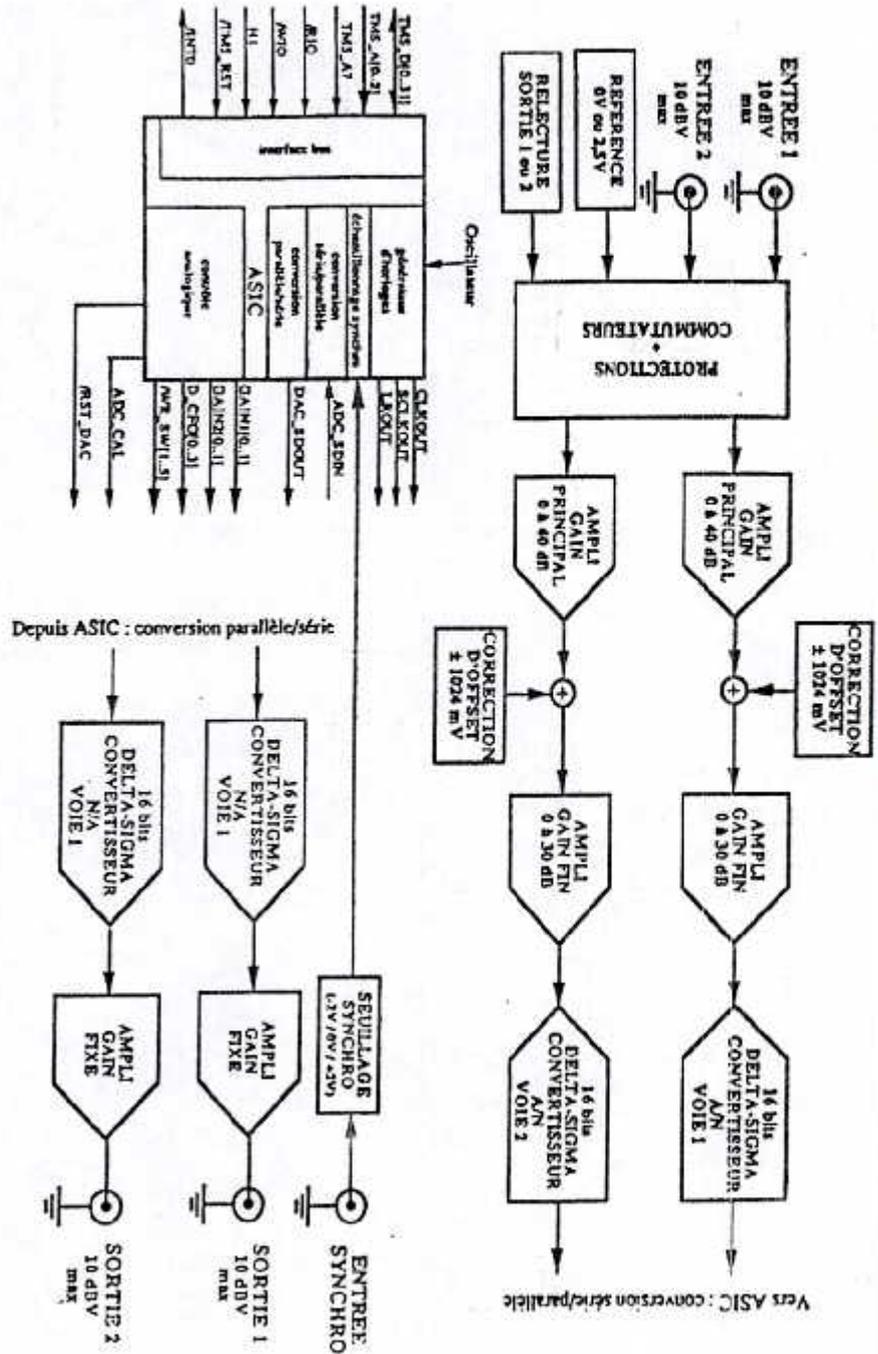


Figure 1 : Schéma synoptique de la partie analogique de la carte OROS-AU32

• Commande de la chaîne analogique

Quatre registres de 16 bits, implanté dans l'espace d'entrée/sortie du TMS320C31, permettent de contrôler la chaîne analogique [19].

Ces quatre registres sont les suivants :

IO_REG1 : registre d'échantillons de la voie 1 (16 bits signés).

IO_REG2 : registre d'échantillons de la voie 2 (16 bits signés).

REGTCT : registre de contrôle des convertisseurs : initialisation, saturation, fréquence.

REGCTA : registre de contrôle de la chaîne analogique : gains, offsets, fréquence, calibration.

Remarque : les registres analogiques sont cadrés sur les 16 bits de poids fort du bus de données du TMS320C31 (TMS_D[31.....16].)

La description détaillée des registres de l'interface analogique est donnée dans le paragraphe suivant.

A) Les registres d'entrées/sorties des échantillons IO_REG1 et IO_REG2

Les échantillons de signal sont codés sur 16 bits, en complément à deux. Ces deux registres peuvent être lus ou écrits par le TMS320C31

Le registre d'échantillon de la voie 1 est implanté à l'adresse IO_REG1(400084h).

Le registre d'échantillon de la voie 2 est implanté à l'adresse IO_REG2(400085h).

Lorsqu'un couple d'échantillons d'entrée est disponible, une interruption est générée vers le TMS320C31 (ligne d'interruption la plus prioritaire). Le processeur vient alors lire ce couple et charge un nouveau couple d'échantillons dans les deux registres du convertisseur de sortie. Ces lectures et écritures effacent la demande d'interruption.

B) Le registre de contrôle des convertisseurs REGTCT

Ce registre REGTCT de 16 bits, implanté à l'adresse REGTCT (400086h) gère la mise en route des convertisseurs, le choix de la fréquence d'échantillonnage et permet de connaître un éventuel débordement du signal à l'entrée des convertisseurs analogique/numérique.

C) Le registre de contrôle de la chaîne analogique REGCTA

Ce registre REGCTA de 16bits, implanté à l'adresse 400087h permet de contrôler sur chaque voie :

- les interruptions programmables de la chaîne d'entrée.
- le gain de l'amplificateur de gain principal.
- la correction d'offset.
- le gain de l'amplificateur de gain fin

IV.2.3 Partie numérique

La carte OROS-AU32 a été bâtie autour d'un microprocesseur de traitement de signal (DSP) de Texas Instrument TMS320C31 à virgule flottante sur mots de 32 bits. Elle a une fréquence d'horloge égale à 33Mhz, un cycle de base 60ns et une puissance de calcul de 33Mflops. [19] (figure 2)

Elle est vue comme un ensemble de quatre registres consécutifs de 16 bits (uniquement) implantés dans l'espace d'entrées-sorties du PC hôte. Ces registres permettent au PC de contrôler tous les périphériques de la carte et d'accéder à la totalité de la mémoire. L'adresse de base de livraison est : 310h.

• Implantation mémoire

Le TMS320C31 accède sans état d'attente à 64 kmots de 32 bits en mémoire statique permettant la vitesse d'accès maximale au programme et aux données. Cette mémoire est située dans les adresses basses du TMS320C31, permettent en particulier de charger directement un code d'initialisation dans la zone de démarrage (boot) du microprocesseur. La mémoire disponible sur la carte est organisée en deux banques de mots de 32 bits. Ces deux banques peuvent être équipées soit de circuit mémoire 32Koctets, soit de circuits 128 koctets [19].

• Les registres d'interfaces de la carte

Les quatre registres d'interfaces permettent :

- Le contrôle de l'état de la carte par le PC.
- Le transfert de données 32 bits par accès 16 bits successifs.
- Le positionnement des données à transférer dans l'espace d'adressage TMS.

Une présentation détaillée du fonctionnement de ces registres est donnée dans le paragraphe suivant.

• Accès à la carte par le PC

A) principe

Le bloc de communication PC permet d'assurer un canal bidirectionnel 32 bits de données (en mode half duplex) entre le bus PC ISA 16 bits et le bus 32 bits de données. Ce bloc de communication permet d'assurer les transferts quelles que soient les vitesses respectives de chaque bus.

Le PC est maître des transferts et possède un accès direct par vol de cycle du TMS en écriture et en lecture des données dans tout l'espace mémoire du TMS.

Un système d'auto-incrémentation automatique permet au PC d'accéder à des adresses consécutives en cas d'accès répété dans la zone TMS [19].

L'accès aux adresses TMS par le PC s'effectue par segment de 64kmots consécutifs : un transfert est constitué d'une adresse de base sur 8 bits fixés par programmation (SEGMENT) s'ajoutant à une adresse de déplacement sur 16 bits (DEPLACEMENT) s'incrémentant à chaque transfert.

L'algorithme général de transfert de N mots à partir d'une adresse ADDR dans l'espace mémoire TMS est donc le suivant :

Calculer l'adresse de base des transferts : $ADDR = \text{SEGMENT} + \text{DEPLACEMENT}$

Initialiser la configuration générale de l'interface.

Programmer l'adresse TMS de base des transferts (SEGMENT).

Programmer l'adresse TMS de DEPLACEMENT dans le segment.

Faire écrire ou lire un mot de 32 bits (ou 2 x16) jusqu'à N mots transférés.

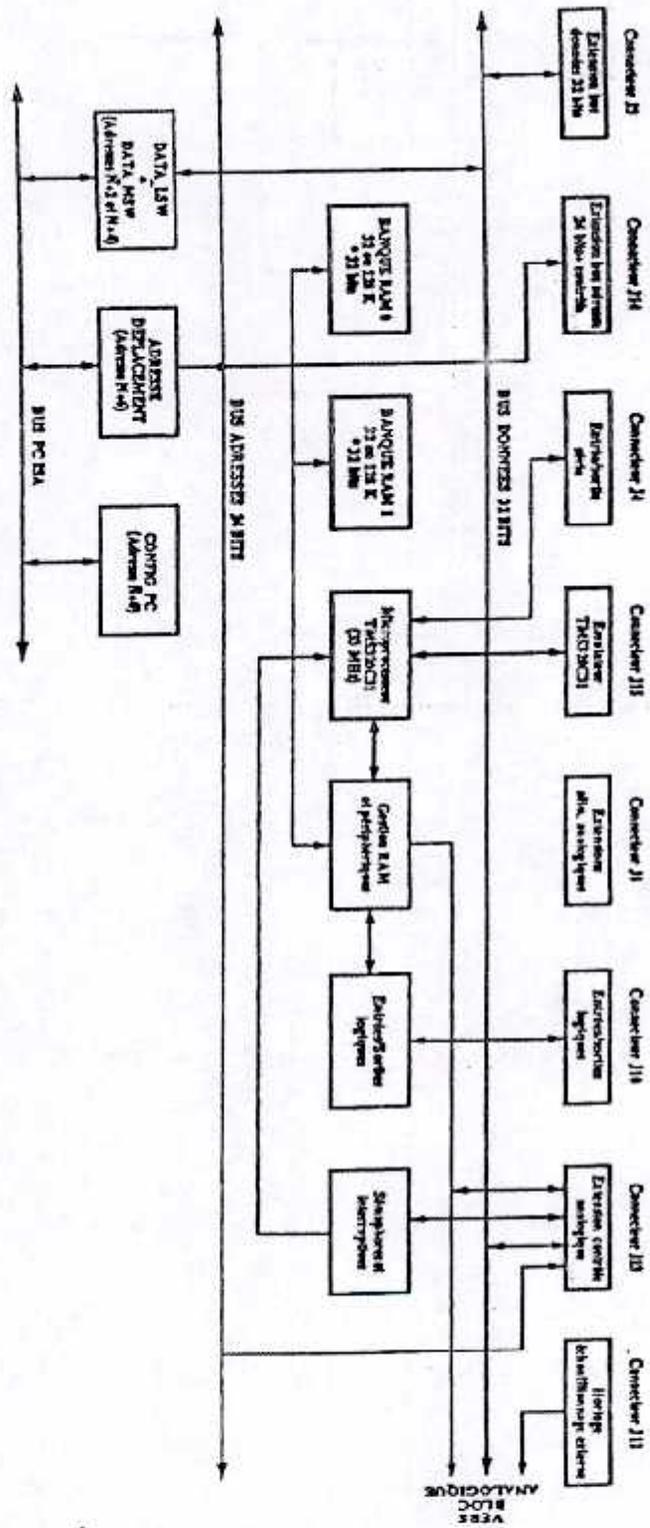


Figure 2 : Schéma synoptique de la partie numérique de la carte OROS-AU32

B) Fonctionnement

L'interface est découpée en trois fonctions différentes :

Un registre de configuration CONFIG-PC inscriptible par le PC

Ce registre est découpé en deux parties simultanément :

- Configuration des échanges :

En écriture :

Six bits de programmation de l'interface par le PC

- RESET du TMS.
- Autorisation de l'interruption du TMS vers le bus PC.
- Auto-incrémentation des adresses TMS.
- RESET matériel de la carte.
- ACCES 16 bits en deux accès identiques successifs (cas des PC 286).
- ACCES 32 bits sur deux adresses successives (cas des PC 386 ou 486 ou pentium).

En lecture :

Cinq bits de relecture de l'état de l'interface par le PC

- RESET du TMS.
- Autorisation de sortie de l'interruption sur le bus PC.
- Auto-incrémentation des adresses TMS.
- Etat de l'automate de transfert (chien de garde dépassé).
- Etat de la demande d'interruption du PC par le TMS.

- Adresse de base (SEGMENT) des échanges :

En écriture le PC doit programmer sur le poids fort de CONFIG_PC les 8 bits constituant le segment de base des transferts : ces 8 bits doivent être programmés avec les 8 bits de poids fort de l'adresse de transfert TMS.

Par exemple, une adresse TMS ADRESSE [23..0] sera découpée ainsi :

SEGMENT[7..0] = ADRESSE[23..16]
 DEPLACEMENT[15..0] = ADRESSE[15..0]

Un registre ADRESSE-DEPLACEMENT : contenant l'adresse de départ d'un transfert dans un segment. Cette adresse (non-reliable) s'auto-incrémente après chaque transfert de mot de 32 bits si le bit correspondant de CONFIG_PC est activé.

Deux registres de données : DATA-LSW et DATA-MSW

Ces deux registres 16 bits, situés à deux adresses consécutives, stockent un mot de 32 bits en deux accès PC ou un cycle dans l'espace mémoire de la carte. Un cycle d'écriture ou de lecture à l'adresse TMS constituée des registres SEGMENT et ADRESSE-DEPLACEMENT est systématiquement déclenché après un cycle complet d'accès PC du registre DATA_MSW (ou DATA_LSW selon le mode 16 bits ou 32 bits d'accès).

Remarque

Les échanges côté PC sont effectués par mots de 16 bits (donc sur des adresses) ou par mots de 32 bits uniquement.