uOttawa

L'Université canadienne
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

uOttawa

L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

Simon Robidas

AUTEUR DE LA THÈSE / AUTHOR OF THESIS

M.A.Sc. (génie électrique)

GRADE / DEGREE

École d'ingéniere et de technologie de l'information

FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

Comparaison de méthodes pour la détection d'activité vocale à bande large sous différents bruits

TITRE DE LA THÈSE / TITLE OF THESIS

Martin Bouchard

DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

EXAMINATEURS (EXAMINATRICES) DE LA THÈSE / THESIS EXAMINERS

Wail Gueaieb

Rafik Goubran

Gary W. Slater

Le Doyen de la Faculté des études supérieures et postdoctorales / Dean of the Faculty of Graduate and Postdoctoral Studies

COMPARAISON DE MÉTHODES POUR LA DÉTECTION D'ACTIVITÉ VOCALE À BANDE LARGE SOUS DIFFÉRENTS BRUITS

Par

Simon Robidas

Cette thèse est soumise à la faculté
d'Études Supérieures et Postdoctorales
afin de répondre aux exigences du
programme de

Maîtrise ès sciences appliquées,
génie électrique

L'institut de génie électrique et de génie
informatique d'Ottawa-Carleton

École d'ingénierie et de technologie de
l'information (ÉITI)

Faculté de génie

Université d'Ottawa

Your file Votre référence
ISBN: 978-0-494-18463-9
Our file Notre référence
ISBN: 978-0-494-18463-9

UNIVERSITÉ D'OTTAWA

RÉSUMÉ

DÉTECTION D'ACTIVITÉ
VOCALE POUR BANDE LARGE
SOUS DIFFÉRENTS BRUITS

Par Simon Robidas

La détection d'activité vocale (VAD) s'applique à un signal sonore, qu'il soit en temps réel ou non. Cette technique consiste à différencier entre les segments du signal qui comportent une voix humaine (segment d'activité) de ceux qui n'en ont pas (segment de non activité). Le détecteur d'activité vocale permet, par exemple, à un ordinateur relayant une conversation voix sur IP de réduire son débit durant les périodes de non activité. Ceci permet une économie importante de la largeur de bande requise pour cette activité, tant au niveau de l'usager qu'au niveau du réseau[1]. D'autres exemples d'application sont le multimédia en général et la téléphonie cellulaire.

Les algorithmes VAD classiques ont été conçus pour la téléphonie traditionnelle échantillonnée à bande étroite (8 kHz). Cependant, de plus en plus d'applications requièrent maintenant des signaux à bande large (16 kHz), d'où l'utilité de comparer les algorithmes classiques[2] et de nouveaux algorithmes conçus pour bande large, soit commerciaux [G722] ou non [NEM01]. Cette thèse démontre et compare l'efficacité de trois algorithmes, soit le VAD faisant partie du codec

[1] Ceci suppose, bien sûr, que tous les utilisateurs du réseau utilisent un détecteur d'activité vocale.

[2] Lorsque utilisés avec des signaux à bande large.

G729.b, le VAD faisant partie du codec G722.2 et le VAD présenté par [NEM01], lorsqu'ils sont soumis à différents types de bruit.

TABLE DES MATIÈRES

TABLE DES MATIÈRES 1

LISTE DE FIGURES 4

LISTE DES TABLEAUX 5

REMERCIEMENTS 6

DÉFINITIONS D'ACRONYMES 7

ÉQUIVALENCE DE TERMES FRANÇAIS-ANGLAIS 8

1 INTRODUCTION 9

1.1 MOTIVATION 9
1.2 MÉTHODES EXISTANTES POUR BANDE ÉTROITE 11
1.2.1 Approche classique *11*
1.2.2 Approche fondamentale *12*
1.3 MÉTHODES EXISTANTES POUR BANDE LARGE 12
1.4 ALGORITHMES UTILISÉS DANS CETTE THÈSE POUR FIN DE COMPARAISON 13
1.4.1 G729b *13*
1.4.2 HOSVAD *13*
1.4.3 G722.2 ..... *14*
1.5 QUELQUES AUTRES ALGORITHMES VAD 14
1.5.1 A robust voice activity detector for wireless communications using soft computing *14*
1.5.2 A Multi-Channel Speech/Silence Detector based on Time Delay Estimation and Fuzzy Classification *14*
1.5.3 Robust Voice Activity Detection Algorithm Based On the Perceptual Wavelet Packet Transform *15*
1.5.4 Improved voice activity detection based on a smoothed statistical likelihood ratio *15*
1.5.5 Voice Activity Detection Over Multiresolution Subspaces *16*
1.5.6 A soft voice activity detector based on a Laplacian-Gaussian model *16*
1.5.7 Voice Activity Detection in Nonstationary Noise *16*
1.6 ORGANISATION DE CETTE THÈSE 16
1.7 CONTRIBUTION 17

2 STATISTIQUES D'ORDRE SUPÉRIEUR 18

2.1 INTRODUCTION 18
2.2 DÉFINITION DES MOMENTS 18
2.3 DÉFINITION DES CUMULANTS 19
2.4 QUELQUES PROPRIÉTÉS UTILES DES CUMULANTS 21
2.5 CONCLUSION 22

3 FONCTIONNEMENT DU HOS VAD 23

3.1 INTRODUCTION 23
3.2 UN TRÈS BREF SURVOL 23
3.3 PRÉTRAITEMENT DU SIGNAL 24

3.4 DÉFINITION DE QUANTITÉS STATISTIQUES UTILISÉES 25
3.4.1 Estimateur de moment *26*
3.4.2 La variance *26*
3.4.3 Le degré d'asymétrie *26*
3.4.4 Le degré d'aplatissement *30*
3.4.5 La probabilité du bruit *32*
3.4.6 Le rapport signal sur bruit pour les basses fréquences *33*
3.4.7 Le rapport signal sur bruit pour toutes les fréquences *34*
3.5 TRANSITION PAROLE/BRUIT 34
3.6 TRANSITION BRUIT/PAROLE 35
3.7 LA MACHINE À ÉTATS FINIS COMPLÈTE 37
3.8 MODIFICATIONS POUR BANDE LARGE 37
3.8.1 Filtre passe-bas *37*
3.8.2 Nombre d'échantillons par trame *38*
3.9 COMPLEXITÉ DE CALCUL 38
3.10 CONCLUSION 39

4 DESCRIPTION DES VADS COMMERCIAUX UTILISÉS DANS CETTE THÈSE 40

4.1 INTRODUCTION 40
4.2 G729B 40
4.2.1 Survol de l'algorithme *40*
4.2.2 Différences dans l'application en bande large *44*
4.2.3 Complexité de calcul *45*
4.3 G722.2 46
4.3.1 Survol de l'algorithme *46*
4.3.2 Complexité de calcul *48*
4.4 COMPARAISON DE COMPLEXITÉ 49
4.5 CONCLUSION 51

5 TEST EFFECTUÉS 52

5.1 INTRODUCTION 52
5.2 BRÈVE DESCRIPTION DES TESTS EFFECTUÉS 52
5.3 FICHIERS AUDIO PROPRES UTILISÉS 53
5.4 FICHIERS AUDIO CORROMPUS PAR LE BRUIT 54
5.5 MESURES UTILISÉES 54
5.6 RÉSULTATS 57
5.6.1 Fichiers audio propres *57*
5.6.2 Réverbération et signal propre *59*
5.6.3 Réverbération avec bruit blanc, SNR = 50 dB *61*
5.6.4 Bruit blanc *63*
5.6.5 Bruit coloré *65*
5.6.6 Bruit de murmure *67*
5.6.7 Bruit de rue *69*
5.6.8 Bruit de voiture *71*
5.7 SYNTHÈSE DES RÉSULTATS 73
5.8 CONCLUSION 75

6 CONCLUSION 76

6.1 RÉCAPITULATION 76
6.2 SOMMAIRE 76

6.3 TRAVAUX SUBSÉQUENTS 77

BIBLIOGRAPHIE 78

ARTICLES ET LIVRES 78
STANDARDS 80

ANNEXE A – BASE DE DONNÉES UTILISÉE 82

FICHIERS SOURCE 82
MODIFICATIONS APPORTÉES AUX FICHIERS SOURCE 82

ANNEXE B – PSD DES SIGNAUX DE PAROLE ET DE BRUIT 84

PSD DES SIGNAUX DE PAROLE 84
PSD DES SIGNAUX DE BRUIT 93

ANNEXE C – FONCTION GETCLOCKCYCLES 97

LISTE DE FIGURES

Figure 1: exemple d'activité et de non activité 10
Figure 2: exemple d'enveloppe d'activité vocale 15
Figure 3: mise en place pour le HOS VAD 23
Figure 4: HOS VAD (simplifié) 24
Figure 5: Exemple de degré d'asymétrie 28
Figure 6: Exemple de probabilité de bruit calculée à partir du degré d'asymétrie 29
Figure 7: Exemple du degré d'aplatissement 31
Figure 8: Exemple de probabilité du bruit à partir du degré d'aplatissement 32
Figure 9: machine à états finis, schéma adapté de [NEM01] 37
Figure 10: organigramme du VAD G729b, reproduit de [G729b] 42
Figure 11: gamme des fréquences utilisées pour calculer E_1 44
Figure 12: schéma fonctionnel simplifié de l'algorithme de décision VAD du G722.2 47
Figure 13: exemple de fichier audio propre 53
Figure 14: mesures utilisées, schéma adapté de [BER01] 55
Figure 15: Exemples de distributions d'erreur 56
Figure 16: statistiques pour fichiers audio propres 57
Figure 17: résultat typique pour fichier audio propre 58
Figure 18: statistiques pour réverbération et signal propre 59
Figure 19: résultat typique pour réverbération et signal propre 60
Figure 20: statistiques pour réverbération et bruit blanc 61
Figure 21: résultat typique pour réverbération et bruit blanc 62
Figure 22: statistiques pour bruit blanc 63
Figure 23: résultat typique pour bruit blanc, SNR = 10dB 64
Figure 24: statistiques pour bruit coloré 65
Figure 25: résultat typique pour bruit coloré, SNR = 10dB 66
Figure 26: statistiques pour bruit de murmure 67
Figure 27: résultat typique pour bruit de murmure, SNR = 10 dB 68
Figure 28: statistiques pour bruit de rue 69
Figure 29: résultat typique pour bruit de rue, SNR = 10 dB 70
Figure 30: statistiques pour bruit de voiture 71
Figure 31: résultat typique pour bruit de voiture, SNR = 10 dB 72

LISTE DES TABLEAUX

Tableau 1: algorithmes utilisés dans cette thèse 13
Tableau 2: Complexité de calcul pour le HOSVAD 39
Tableau 3: description des variables et constantes utilisées dans l'organigramme du G729b 43
Tableau 4: sous-bandes utilisées par le VAD du G722.2 46
Tableau 5: complexité des trois algorithmes VADs 49
Tableau 6: synthèse de la performance des trois VADs 75

REMERCIEMENTS

Je désire remercier ma merveilleuse épouse Zuwena de m'épauler dans chaque projet que j'entreprends, peu importe la taille ou la durée. Son appui moral est une grande source de force et de quiétude.

Je désire aussi remercier le ministère ontarien de la Formation et des Collèges et Universités pour leur appui financier à travers la Bourse d'Étude Supérieure de l'Ontario (BÉSO).

L'appui financier de la faculté des études supérieures et postdoctorales de l'Université d'Ottawa, à travers la bourse d'excellence, fut aussi très apprécié.

DÉFINITIONS D'ACRONYMES

ETSI	Institut européen des normes de télécommunication
FIR	Finite impulse response (filtre en temps discret)
HOS	Higher order statistics (statistiques d'ordre supérieur)
ITU	International telecommunications union (Union internationale de télécommunication)
LPC	Linear predictive coding (codage de prediction linéaire)
LSF	Line spectral frequencies (fréquences de raies spectrales)
PE	Prediction error (erreur de prédiction LPC)
SNR	Signal to noise ratio (rapport signal sur bruit)
SKR	Skewness to kurtosis ratio (rapport du degré d'asymétrie au degré d'aplatissement)
VAD	Voice activity detection (détecteur d'activité vocale)

ÉQUIVALENCE DE TERMES FRANÇAIS-ANGLAIS

Degré d'aplatissement	Kurtosis
Degré d'asymétrie	Skewness
Fréquences de raies spectrales	Line spectral frequencies
Hauteur tonale	Pitch
Intégrale de probabilité	Error function
Seuil de comparaison	Threshold
Statistiques d'ordre supérieur	Higher order statistics (HOS)

Chapitre un

INTRODUCTION

1 Introduction

1.1 Motivation

La détection d'activité vocale nous permet de faire la distinction entre les segments d'un signal audio qui comportent de la voix humaine (période d'activité) et les segments du même signal qui n'en ont pas (période de non activité). C'est une partie importante de tout système de communication audio. Par exemple, on retrouve un détecteur d'activité vocale dans la téléphonie cellulaire (codec UTI-T G.729b, GSM 06.32) et dans la téléphonie voix sur IP. Dans le premier cas, la pile d'un téléphone mobile peut avoir une meilleure durée de vie en évitant de transmettre lors des périodes de non activité. Dans le deuxième cas, un VAD[3] permet une économie importante de la largeur de bande requise pour cette activité, tant au niveau de l'usager qu'au niveau du réseau[4].

[3] L'acronyme anglais VAD (*voice activity detector*) du détecteur d'activité vocale est utilisé dans cette thèse car il est très répandu dans la littérature scientifique et technologique.

[4] Ceci suppose, bien sûr, que tous les utilisateurs du réseau utilisent un détecteur d'activité vocale.

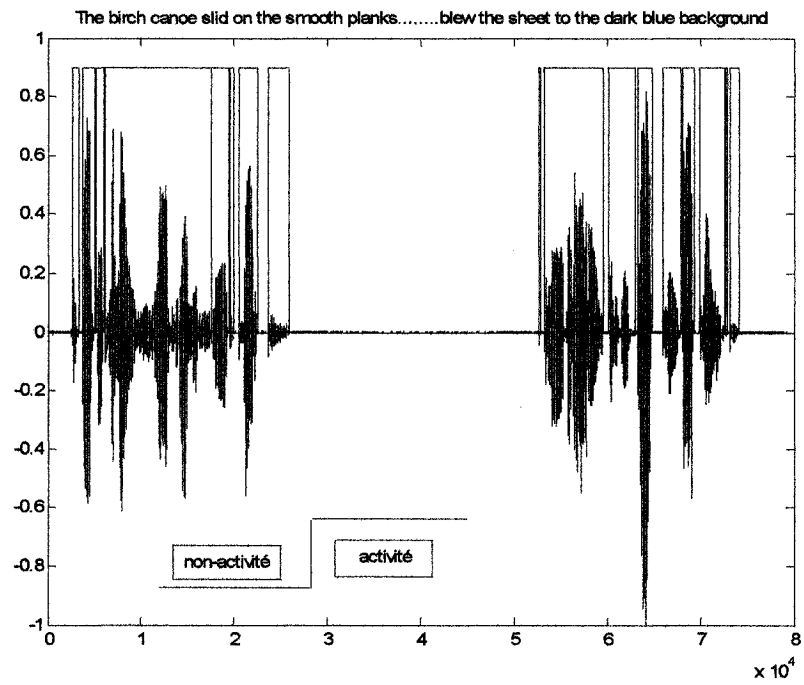


Figure 1: exemple d'activité et de non activité

Auparavant, la voix humaine était échantillonnée à 8 kHz (bande étroite) dans à peu près tous les systèmes de communications. Aujourd'hui, la voix humaine est échantillonnée à 16 kHz (bande large) dans de plus en plus de systèmes de communication. Par exemple, l'Union Internationale des Télécommunications (UIT) a déjà proposé quelques codecs, dont G722, G722.1 et G722.2. L'institut européen des normes de télécommunication (ETSI) a lui aussi proposé quelques codecs, comme le CDMA 2000.

L'efficacité du VAD de n'importe quel codec est limitée par les bruits environnants, que ce soit du bruit de parole indésiré en arrière-plan, du bruit blanc ou autre source de bruit. Cette thèse a pour but d'examiner et comparer l'efficacité de différents algorithmes VADs commerciaux et un algorithme trouvé en publication lorsque le signal de parole est corrompu avec divers types de bruit.

1.2 Méthodes existantes pour bande étroite

1.2.1 Approche classique

Les VAD commerciaux[5] et plusieurs VAD publiés[6] s'appuient sur le même principe de fonctionnement. Ils découpent un signal audio en trames de 10 à 30 ms, selon l'algorithme, afin de rendre le signal de chaque trame approximativement invariant dans le temps. Ensuite, ces algorithmes prennent différents types de mesures afin de déterminer si le segment audio sous observation est actif, c'est à dire contenant de la voix humaine, ou non. Les mesures les plus courantes sont :

1. **Niveau d'énergie du signal :** une comparaison de la puissance de la trame avec un certain seuil de comparaison, adaptatif ou non, peut révéler la présence de parole.

2. **Passage par zéro :** le nombre de fois dans une trame où le signal audio change de signe (+ / -) peut indiquer la présence de parole.

3. **Forme spectrale :** la distribution d'énergie dans diverses gammes de fréquences. Une certaine forme spectrale peut indiquer la présence de parole.

4. **Coefficients d'autocorrélation :** ces coefficients, utilisés dans la prédiction linéaire de parole (LPC), peuvent indiquer la présence de voix humaine, voisée ou non.

5. **Période de la hauteur tonale :** la période de la hauteur tonale peut indiquer la présence d'un segment de parole voisée.

---

[5] [G722], [G729.b]

[6] [BER98], [QI93], [BER99], [CHO01]

Après avoir évalué ces critères, ces algorithmes les comparent à un certain seuil, afin de déterminer si le segment sous observation est actif ou non. Ces seuils sont habituellement dynamiques et mis à jour pendant les périodes de non activité.

Ce sont les méthodes de classification des mesures vis-à-vis des seuils appropriés qui différencient ces algorithmes et non le type de mesure. Donc, ces algorithmes souffrent des mêmes maux, à différents degrés. Ils sont tous sensibles au bruit environnant, surtout si ce dernier n'est pas stationnaire [GOK00]. Un bruit de parole en arrière-plan est particulièrement problématique.

1.2.2 Approche fondamentale

Quelques algorithmes s'appuient sur d'autres mesures de comparaison. Cela prend habituellement la forme d'une transformée quelconque. La parole humaine ou le bruit environnant ont, dans le domaine transformé, une ou plusieurs propriétés à exploiter.

Par exemple, les cumulants du troisième ou quatrième ordre sont, en théorie, zéros pour les procédé aléatoires gaussiens[7]. Donc le bruit environnant, qui est souvent gaussien ou approximativement gaussien selon le théorème central limite [HOG97], peut être éliminé par cette transformation mathématique du signal. Les ondelettes [ERD00], la transformée cosinus [GAZ03] et l'opérateur Teager [SHI05] sont d'autres techniques fondamentales utilisées par certains.

1.3 Méthodes existantes pour bande large

L'algorithme commercial de référence est sans nul doute le VAD inclus avec le codeur de parole ITU G722.2.

[7] Voir chapitre deux.

1.4 Algorithmes utilisés dans cette thèse pour fin de comparaison

Algorithme	Type	Conçu pour	Source
UIT-T G729.b	Approche classique	Bande étroite	Le code source fourni avec [G729b] fut utilisé dans cette thèse.
HOS VAD	Approche fondamentale	Bande étroite	Implémenté par l'auteur à partir de [NEM01], avec de légères modifications pour bande large.
UIT-T G722.2	Approche classique	Bande large	Le code source fourni avec [G722] fut utilisé dans cette thèse.

Tableau 1: algorithmes utilisés dans cette thèse

1.4.1 G729b

Le G729b a été choisi car c'est le VAD de référence pour toute application à bande étroite. Il est donc intéressant de mesurer la performance de ce VAD en bande large, sans modification, afin de déterminer s'il est vraiment nécessaire de développer d'autres algorithmes pour la bande large.

1.4.2 HOSVAD

Cet algorithme a été choisi car il est présenté de façon très rigoureuse dans [NEM01], un article de quinze pages dans le prestigieux magazine *IEEE*

Transactions on Speech And Audio Processing. L'utilisation des statistiques de troisième et de quatrième ordre rend cet algorithme très attrayant, car ces statistiques sont un outil analytique très puissant lorsqu'on fait face à des bruits Gaussiens stationnaires ou non stationnaires.

1.4.3 G722.2

Le G722.2 a été choisi car c'est le VAD de référence pour toute application à bande large. Il est donc très intéressant de le comparer au HOSVAD, un algorithme développé pour la bande large mais utilisant des mesures tout à fait différentes.

1.5 Quelques autres algorithmes VAD

On peut trouver une multitude d'algorithmes VADs en publication. À titre d'information seulement, cette section en présente quelques uns.

1.5.1 A robust voice activity detector for wireless communications using soft computing

Cet algorithme, présenté dans [BER98], utilise les mêmes mesures que le G729b mais adopte une approche différente pour la classification des trames actives et non actives. En effet, [BER98] utilise plutôt des règles de logique floue afin de déterminer si la trame sous étude est active ou non.

1.5.2 A Multi-Channel Speech/Silence Detector based on Time Delay Estimation and Fuzzy Classification

Cet algorithme, présenté dans [BER99], est une amélioration de l'algorithme présenté en [BER98]. L'amélioration consiste à utiliser de nouvelles mesures obtenues dans un contexte multi canal, soit l'estimation du délai et de la l'énergie des différents canaux.

1.5.3 Robust Voice Activity Detection Algorithm Based On the Perceptual Wavelet Packet Transform

Cet algorithme, présenté dans [CHE05], utilise les ondelettes afin de découper le signal audio en dix-sept sous-bandes. Chaque sous-bande est ensuite soumise à une transformée Teager afin d'obtenir l'enveloppe d'activité vocale (VAS, ou *Voice Activity Shape* en anglais). Le VAS est ensuite comparé à un seuil adaptatif afin de déterminer si la trame sous études est active ou non.

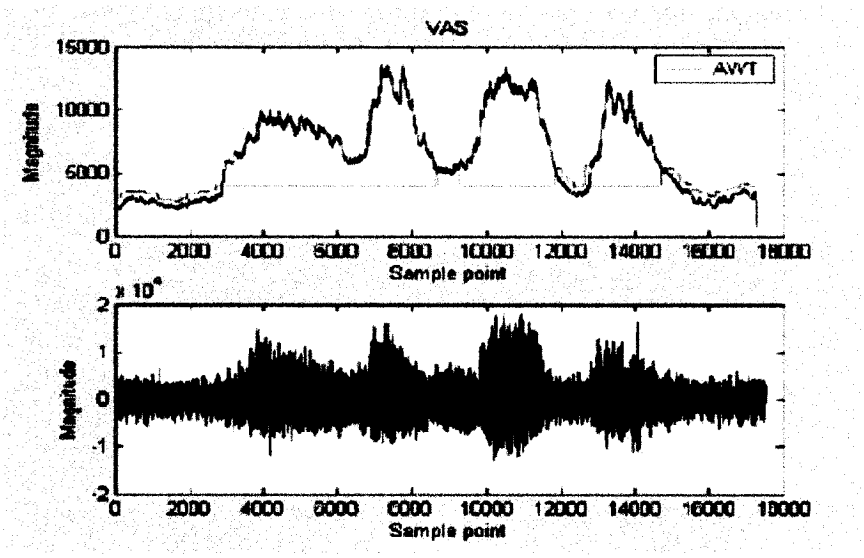


Figure 2: exemple d'enveloppe d'activité vocale

1.5.4 Improved voice activity detection based on a smoothed statistical likelihood ratio

La plupart des algorithmes VAD utilisent une période de traînage avant de passer de l'état parole à l'état bruit, ou inversement, afin de réduire les erreurs de décision lors des débuts et fins de parole. Cet algorithme, présenté dans [CHO01], étudie ce phénomène et propose une solution utilisant un test de plausibilité plutôt qu'une période de traînage.

1.5.5 Voice Activity Detection Over Multiresolution Subspaces

Cet algorithme, présenté dans [ERD00], utilise la transformée en ondelette discrète afin d'obtenir les racines caractéristiques du signal sous étude. La détermination de la présence de signal est faite à partir de ces racines.

1.5.6 A soft voice activity detector based on a Laplacian-Gaussian model

Cet algorithme, présenté dans [GAZ03], utilise une transformée en cosinus discrète (DCT) afin de décorréler le signal audio. L'auteur considère que les échantillons de paroles sont représentés par des variables aléatoires laplaciennes dans le domaine DCT. Les échantillons de bruits sont représentés par des variables aléatoires gaussiennes dans le même domaine. Un modèle de Markov caché est utilisé pour distinguer les trames ayant une distribution laplacienne (parole) des trames ayant des distributions gaussiennes (bruit).

1.5.7 Voice Activity Detection in Nonstationary Noise

Cet algorithme, présenté dans [GOK00], utilise des mesures classiques, telles le SNR, mais diffère du G729b en mettant à jour ses seuils de comparaison pendant les périodes d'activité.

1.6 Organisation de cette thèse

Au chapitre deux, quelques notions de base concernant les statistiques d'ordre supérieur sont abordées. Ces connaissances sont requises afin de bien comprendre le fonctionnement d'un algorithme étudié dans cette thèse, soit le HOSVAD.

Au chapitre trois, le fonctionnement du HOSVAD est expliqué en détail, en s'appuyant sur les notions présentées au chapitre deux. La complexité de calcul, en nombre de cycles d'horloges par trame de 10 ms, est aussi présentée.

Au chapitre quatre, les algorithmes VAD du G729b et du G722.2 sont expliqués. Leurs complexités de calcul, pour des trames de 5 ms et des trames de 20 ms, respectivement, sont présentées.

Au chapitre cinq, les tests pratiques effectués dans le cadre de cette thèse sont présentés. Les types de signaux audio ainsi que les types de bruit utilisés sont abordés. Les résultats détaillés de l'utilisation des trois VADs avec tous les signaux et tous les bruits sont fournis, ainsi qu'une synthèse des résultats.

Le chapitre six présente une conclusion basée sur les résultats du chapitre cinq et offre une suggestion de recherche reliée au travail de cette thèse.

1.7 Contribution

La contribution principale de cette thèse est de comparer différents algorithmes VAD pour bande large sous différents bruits, ce qui n'a pas été fait jusqu'à présent dans la littérature.

Chapitre deux

2 Statistiques d'ordre supérieur

2.1 Introduction

Il importe de décrire brièvement quelques notions de base concernant les statistiques d'ordre supérieur car un des VADs sous étude, soit le HOSVAD proposé par [NEM01], s'appuie sur ces notions mathématiques. Les notions de moment et cumulant sont expliquées dans ce chapitre, qui se termine par l'énoncé de quelques propriétés des cumulants utiles pour le HOSVAD.

2.2 Définition des moments

Les moments sont des quantités statistiques définies à partir de l'espérance d'un processus aléatoire quelconque. Supposons que x soit un signal réel, en temps discret et que ses moments existent jusqu'à l'ordre k. Dans ce cas, voici les définitions des moments du premier, deuxième, troisième et quatrième ordre :

$$m_{kx} = E\{x^k\}$$

Équation 1

$$m_{1x} = E\{x^1\}$$

Équation 2

$$m_{2x}(\tau_1) = E\{x(t)x(t+\tau_1)\}$$

Équation 3

$$m_{3x}(\tau_1,\tau_2) = E\{x(t)x(t+\tau_1)x(t+\tau_2)\}$$

Équation 4

$$m_{4x}(\tau_1,\tau_2,\tau_3) = E\{x(t)x(t+\tau_1)x(t+\tau_2)x(t+\tau_3)\}$$

Équation 5

Il est à noter que $E\{\ \}$ dénote l'espérance d'un signal. On peut tout de suite voir, en observant les équations 2 et 3, que le moment du premier ordre correspond à la moyenne du signal et que le moment du deuxième ordre n'est autre que l'autocorrélation du procédé aléatoire *x*.

2.3 Définition des cumulants

Les cumulants sont des quantités statistiques définies à partir des moments définis ci haut. Voici, d'après Nikias et Mendel [NIK93], les équations des cumulants du premier, deuxième, troisième et quatrième ordre :

$$C_{1x} = m_{1x}$$

Équation 6

$$C_{2x}(\tau_1) = m_{2x}(\tau_1) - m_{1x}m_{1x}$$

Équation 7

$$C_{3x}(\tau_1,\tau_2) = m_{3x}(\tau_1,\tau_2) - m_{1x}\left(m_{2x}(\tau_1) + m_{2x}(\tau_2) + m_{2x}(\tau_2-\tau_1)\right) + 2(m_{1x})^3$$

Équation 8

$$C_{4x}(\tau_1,\tau_2,\tau_3) = m_{4x}(\tau_1,\tau_2,\tau_3) - m_{2x}(\tau_1)m_{2x}(\tau_3-\tau_2) - m_{2x}(\tau_2)m_{2x}(\tau_3-\tau_1) - m_{2x}(\tau_3)m_{2x}(\tau_2-\tau_1)$$
$$- m_{1x}\big(m_{3x}(\tau_2-\tau_1,\tau_3-\tau_1) + m_{3x}(\tau_2,\tau_3) + m_{3x}(\tau_3,\tau_1) + m_{3x}(\tau_1,\tau_2)\big)$$
$$+ (m_{1x})^2\big(m_{2x}(\tau_1) + m_{2x}(\tau_2) + m_{2x}(\tau_3) + m_{2x}(\tau_3-\tau_1) + m_{2x}(\tau_3-\tau_2) + m_{2x}(\tau_2-\tau_1)\big)$$
$$- 6(m_{1x})^4$$

Équation 9

Supposons que le signal x ait une moyenne de zéro. Dans ce cas, le cumulant du premier ordre est aussi zéro, ce qui nous permet de simplifier les équations 6, 7, 8 et 9 comme suit :

$$C_{1x} = m_{1x} = 0$$

Équation 10

$$C_{2x}(\tau_1) = m_{2x}(\tau_1)$$

Équation 11

$$C_{3x}(\tau_1,\tau_2) = m_{3x}(\tau_1,\tau_2)$$

Équation 12

$$C_{4x}(\tau_1,\tau_2,\tau_3) = m_{4x}(\tau_1,\tau_2,\tau_3) - m_{2x}(\tau_1)m_{2x}(\tau_3-\tau_2) - m_{2x}(\tau_2)m_{2x}(\tau_3-\tau_1) - m_{2x}(\tau_3)m_{2x}(\tau_2-\tau_1)$$
$$C_{4x}(\tau_1,\tau_2,\tau_3) = m_{4x}(\tau_1,\tau_2,\tau_3) - C_{2x}(\tau_1)C_{2x}(\tau_3-\tau_2) - C_{2x}(\tau_2)C_{2x}(\tau_3-\tau_1) - C_{2x}(\tau_3)C_{2x}(\tau_2-\tau_1)$$

Équation 13

2.4 Quelques propriétés utiles des cumulants

Supposons que le signal $z(n) = x(n) + y(n)$, et que $x(n)$ et $y(n)$ sont des signaux indépendants. Dans ce cas, selon [SWA01] :

1. Les cumulants sont additifs

$$C_{kz} = C_{kx} + C_{ky}$$

Équation 14

2. Les cumulants d'un procédé aléatoire Gaussien[8] d'ordre $k>2$ sont zéros

$$C_{ky} = 0 \quad \text{si } y \text{ est Gaussien et } k > 2$$

Équation 15

Supposons maintenant que x est un signal non Gaussien, y est un signal de bruit Gaussien et que z est l'addition de x avec y. Dans ce cas :

$$C_{kz} = C_{kx} + C_{ky} = C_{kx} \quad \text{pour } k > 2$$

Équation 16

Étant donné que, en pratique, la plupart des bruits peuvent être estimés par un procédé aléatoire Gaussien grâce au théorème central limite [HOG97], ces deux propriétés deviennent particulièrement importantes dans le cadre de la détection d'activité vocale d'un signal de parole corrompu par le bruit.

[8] Le procédé Gaussien peut être blanc ou coloré [SWA01]

2.5 Conclusion

Les cumulants sont calculés à partir des moments qui, eux, sont définis à partir de l'espérance d'un processus aléatoire quelconque. L'intérêt porté pour les statistiques d'ordre supérieur réside dans l'additivité des cumulants et dans le fait qu'ils sont zéros pour tous processus gaussien, pourvu que l'ordre des cumulants soit supérieur à zéro.

3 Fonctionnement du HOS VAD

3.1 Introduction

Trois VADs sont comparés dans cette thèse : le G729b, le G722.2 et le HOS VAD proposé par [NEM01]. Les deux premiers comprennent des algorithmes VAD plutôt classiques, c'est-à-dire qu'ils utilisent des critères traditionnels, soit le niveau d'énergie, les passages par zéro, la forme spectrale, les coefficients d'autocorrélation et la hauteur tonale afin de décider s'il y a activité vocale ou non. Le HOSVAD, quant à lui, utilise de nouveaux critères afin de détecter l'activité vocale, ce qui rend intéressante sa comparaison aux deux autres algorithmes. Ce chapitre résume le fonctionnement du HOS VAD, mis en œuvre selon l'algorithme [NEM01] et légèrement modifié afin de traiter des fichiers sonores en bande large.

3.2 Un très bref survol

Supposons que y soit un signal de parole corrompu par w, un signal de bruit Gaussien. Dénotons z comme étant le signal résultant de l'addition du signal de parole avec le signal de bruit, i.e. $z = y + w$. Ensuite, filtrons z par un filtre LPC du dixième ordre. Le résidu du filtre sera nommé x.

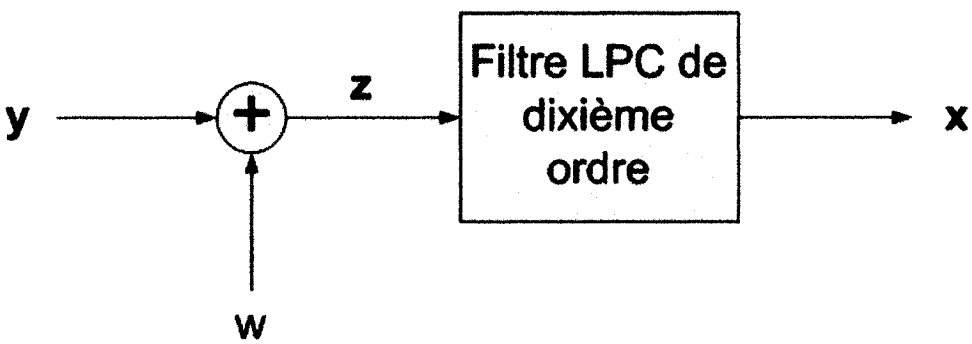


Figure 3: mise en place pour le HOS VAD

Si nous faisons une analyse de ces signaux en utilisant des statistiques des troisième et quatrième ordres, alors le bruit Gaussien w ne devrait, en théorie, avoir aucune incidence sur le résultat car les cumulants d'ordre $k > 2$ sont toujours zéros et ce, peu importe le rapport signal sur bruit. C'est donc une méthode très puissante.

Si on continue l'exemple précédant, le signal x est obtenu en passant le signal sonore composé de parole et de bruit par un filtre LPC de dixième ordre. Selon [NEM01], le résidu LPC, soit x dans notre exemple, est non gaussien lorsqu'il y a transition de parole à silence ou inversement. Donc, les transitions parole/bruit et bruit/parole produiront des pointes dans la valeur des cumulants du troisième et quatrième ordre. Si ces pointes sont l'entrée d'une machine à états fini, on a un VAD comme l'illustre le schéma suivant :

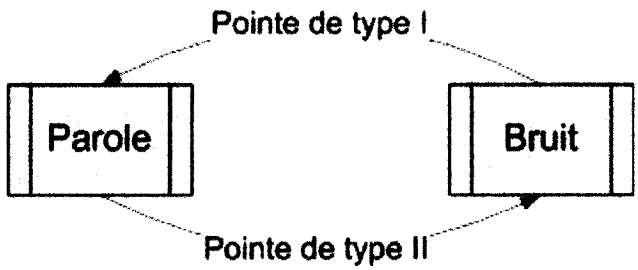


Figure 4: HOS VAD (simplifié)

3.3 Prétraitement du signal

Avant de faire l'analyse à partir de cumulants, le signal sonore est traité par un filtre LPC du dixième ordre afin d'enlever la corrélation à court terme du résidu. Selon [NEM01], ceci a pour effet d'enlever la corrélation à court terme, résultant ainsi en une enveloppe spectrale plate du résidu et rendant ainsi les caractéristiques spectrales de la parole moins sensibles au rapport signal sur bruit ainsi qu'au contenu spectral du bruit. Cependant, si une méthode utilisant des

statistiques du deuxième ordre est utilisée, comme la méthode d'autocorrélation, alors le résidu n'aura une enveloppe spectrale plate qu'en moyenne. Pour contrer ce problème, [NEM01] suggère d'utiliser des cumulants du troisième ordre dans l'analyse de prédiction linéaire, comme la méthode proposée par [PAL91], afin de rendre l'enveloppe spectrale de chaque trame réellement plate.

Deux copies du signal de sortie du filtre LPC seront conservées. La première sera filtrée par un filtre FIR passe-bas à soixante coefficients, dont la fréquence de coupure est de 3.6kHz. Le terme «résidu filtré» sera utilisé dans cette thèse pour y faire référence. La deuxième copie sera gardée intacte, aucun autre filtre n'y sera appliquée. Le terme «résidu non filtré» sera utilisé dans cette thèse pour y faire référence.

3.4 Définition de quantités statistiques utilisées

Note :

À moins d'indication contraire, toutes les quantités statistiques définies dans cette section sont calculées à partir de trames de 10 ms du résidu filtré.

Il n'est pas nécessaire de calculer $C_{3x}(\tau_1,\tau_2)$ et $C_{4x}(\tau_1,\tau_2,\tau_3)$ pour plusieurs valeurs de τ_1,τ_2,τ_3. Pour les processus gaussiens, $C_{3x}(\tau_1,\tau_2)$ et $C_{4x}(\tau_1,\tau_2,\tau_3)$ sont zéro pour toutes les valeurs de τ_1,τ_2,τ_3 (lorsque la moyenne statistique est considérée). Pour les processus non gaussiens, $C_{3x}(\tau_1,\tau_2)$ et $C_{4x}(\tau_1,\tau_2,\tau_3)$ sont non zéro (statistiquement, encore une fois). Donc, il est valable d'utiliser uniquement $\tau_1=\tau_2=\tau_3=0$. Ça rend l'estimation de $C_{3x}(0,0)$ et de $C_{4x}(0,0,0)$ très facile.

3.4.1 Estimateur de moment

Dans l'algorithme présenté par [NEM01], nous devons fréquemment calculer divers moments. Pour ce faire, nous utilisons l'approximation suivante lorsque $\tau_1 = \tau_2 = \tau_3 = 0$:

$$m_{kx} = \frac{1}{N}\sum_{n=0}^{N-1}(x(n))^k$$

Équation 17

3.4.2 La variance

La variance est estimée comme suit dans l'algorithme :

$$C_{2x} = C_{2x}(0) = m_{2x}(0) = \frac{1}{N}\sum_{n=0}^{N-1}(x(n))^2$$

Équation 18

3.4.3 Le degré d'asymétrie

La quantité C_{3x} est dénommée degré d'asymétrie. En anglais, on parle de *skewness*. Cette quantité est estimée comme suit :

$$SK = C_{3x} = C_{3x}(0,0) = m_{3x}(0,0) = \frac{1}{N}\sum_{n=0}^{N-1}(x(n))^3$$

Équation 19

Cette estimation non biaisée du degré d'asymétrie a une moyenne de zéro et une variance non unitaire dans le cas de bruit blanc Gaussien :

$$E[C_{3x}] = 0$$

Équation 20

$$\text{var}(C_{3x}) = \frac{15\,\text{var}^3(x)}{N}$$

Équation 21

Étant donné que l'estimation du degré d'asymétrie est la somme d'un grand nombre de variables aléatoires ayant une distribution statistique indépendante, le théorème central limite [HOG97] nous permet de définir une quantité ayant une moyenne de zéro et une variance unitaire :

$$SKa = SK * a = SK * \frac{1}{\sqrt{\text{var}^3(x)/N}}$$

Équation 22

Donc, *SKa* est le résultat de l'application d'un gain, *a*, à l'estimation non biaisée du degré d'asymétrie, *SK*. Il est à noter que *a* varie selon la variance du procédé aléatoire sous analyse. *SKa* tend vers zéro dans les trames sans activité vocale, comme le démontre le graphique suivant :

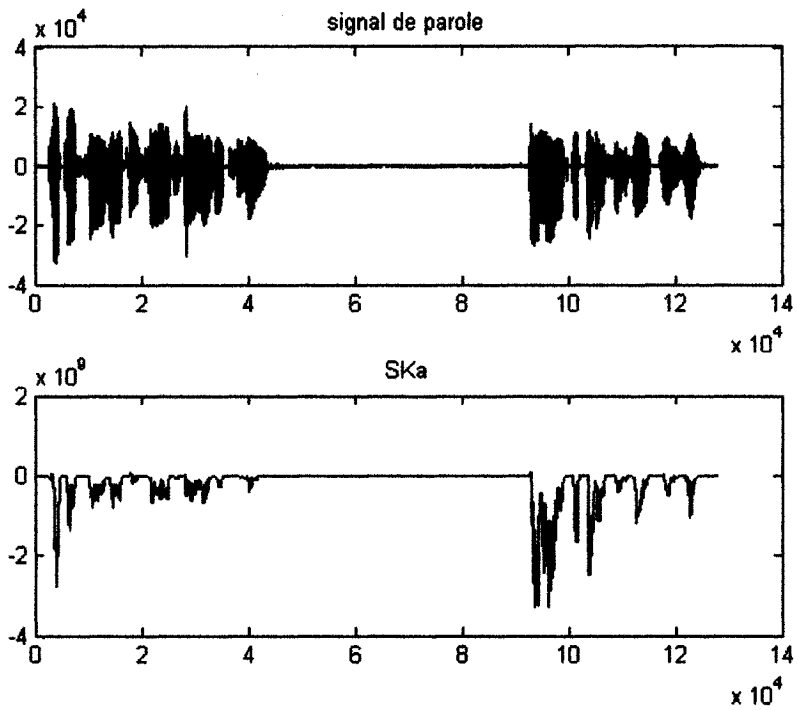


Figure 5: Exemple de degré d'asymétrie

La probabilité qu'une trame du signal est Gaussienne peut s'exprimer comme suit :

$$prob[bruit] = prob[|SKa| \geq a]$$

Équation 23

L'équation précédente équivaut à calculer l'aire sous la queue de la courbe Gaussienne de *SKa*. Nous pouvons aussi exprimer cette relation avec l'intégrale de probabilité, soit :

$$prob[bruit] = erfc(|a|)$$

Équation 24

Voici, graphiquement, ce que ça représente :

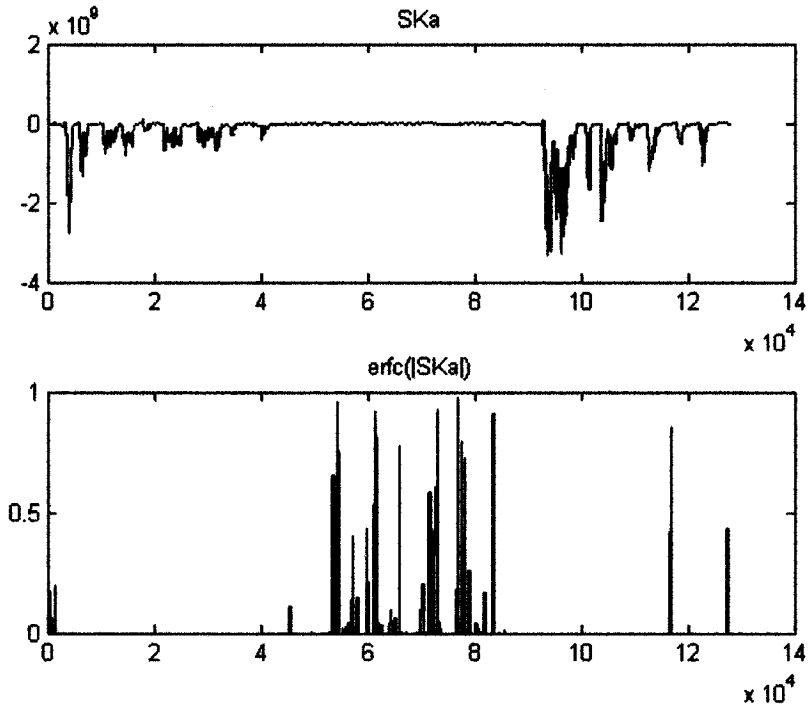


Figure 6: Exemple de probabilité de bruit calculée à partir du degré d'asymétrie

L'intégrale de probabilité tend vers zéro lorsqu'il y a de la parole, résultant en une faible probabilité de bruit dans ce cas, alors qu'elle tend vers 1 lorsqu'il y a absence de parole, donnant une forte probabilité de bruit. En utilisant l'intégrale de probabilité, le résultat est toujours borné entre zéro et un.

3.4.4 Le degré d'aplatissement

La quantité C_{4x} est dénommée degré d'aplatissement. En anglais, on parle de *kurtosis*. [NEM01] propose ceci afin d'obtenir une estimation non biaisée de cette quantité :

$$KUu = C_{4x} = C_{4x}(0,0,0) = \left(1+\frac{2}{N}\right)\frac{1}{N}\sum_{n=0}^{N-1}(x(n))^4 - 3(C_{2x})^2$$

Équation 25

Cette estimation non biaisée du degré d'aplatissement a une moyenne de zéro et une variance non unitaire. Afin d'obtenir une quantité avec une moyenne de zéro et une variance unitaire, [NEM01] propose ceci :

$$KUa = \frac{KUu}{\sqrt{\frac{3(\mathrm{var}(x))^4}{N}\left(104+\frac{452}{N}+\frac{596}{N^2}\right)}}$$

Équation 26

Donc *KUa* est le résultat de l'application d'un gain, *b*, à l'estimation non biaisée du degré d'aplatissement, *KUu*. *KUa* tend vers zéro dans les trames de silence, comme le démontre le graphique suivant :

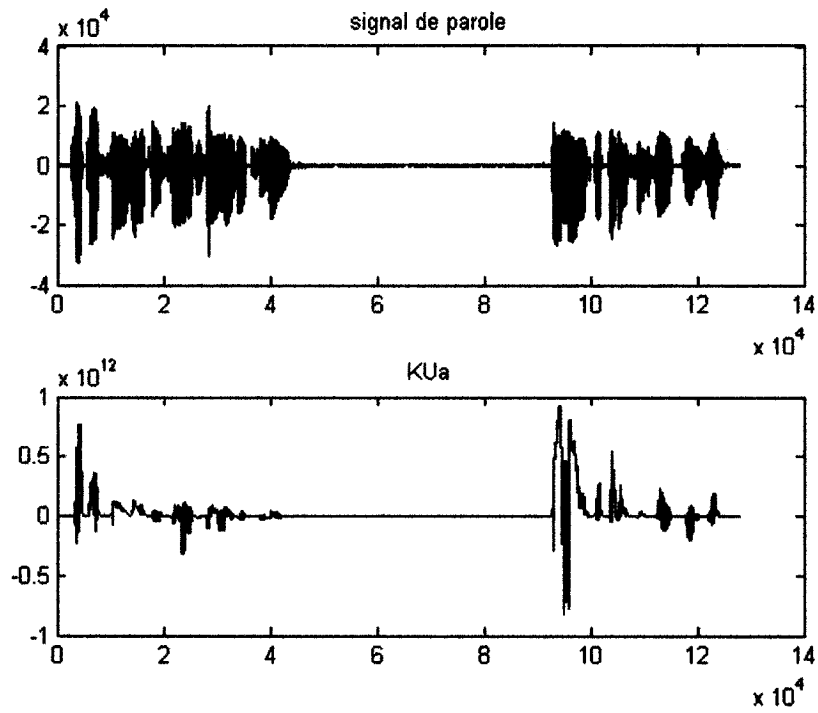


Figure 7: Exemple du degré d'aplatissement

Par un raisonnement similaire à la section précédente, nous avons :

$$prob[bruit] = prob[|KUa| \geq b] = erfc(|b|)$$

Équation 27

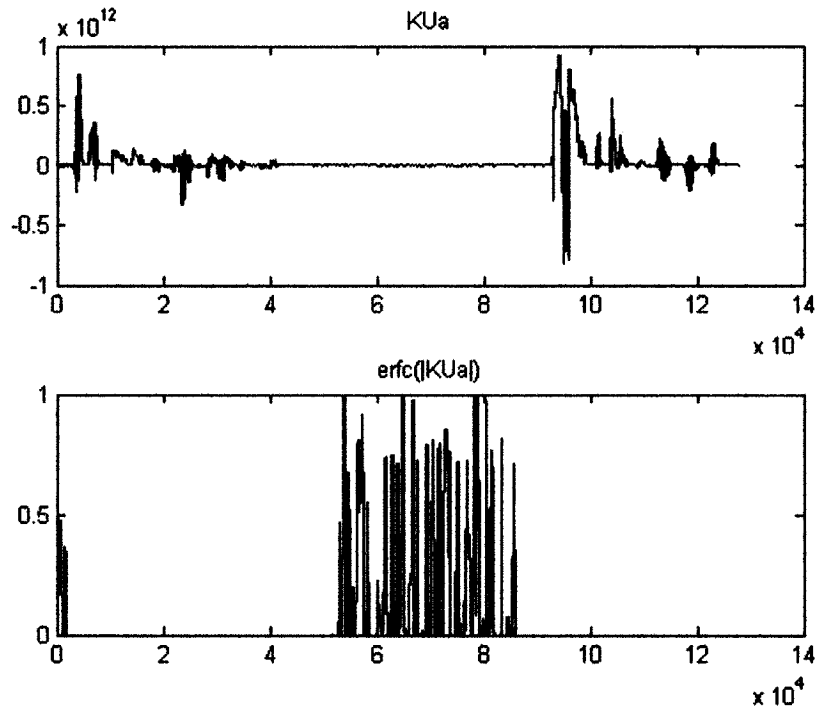


Figure 8: Exemple de probabilité du bruit à partir du degré d'aplatissement

Tout comme avec *SKa*, l' intégrale de probabilité tend vers zéro lorsqu'il y a de la parole, résultant en une faible probabilité de bruit dans ce cas, alors qu'elle tend vers 1 lorsqu'il y a absence de parole, donnant une forte probabilité de bruit. L'intégrale de probabilité nous permet ici aussi de borner la quantité entre zéro et un.

3.4.5 La probabilité du bruit

En combinant le degré d'aplatissement et le degré d'asymétrie, la probabilité du bruit se définit comme suit :

$$prob[bruit] = \frac{1}{2}\left(erfc(|\ SKa\ |) + erfc(|\ KUa\ |)\right)$$

Équation 28

3.4.6 Le rapport signal sur bruit pour les basses fréquences

En utilisant la puissance de la trame sous analyse ainsi que l'estimé de la puissance du bruit, [NEM01] propose cette méthode de calculer le rapport signal sur bruit pour les basses fréquences :

$$SNR = Pos\left[\frac{m_{2x}}{\mathrm{var}_{bruit}} - 1\right]$$

Équation 29

$$Pos[x] = \begin{cases} x & , \quad x > 0 \\ 0 & , \quad x \le 0 \end{cases}$$

Équation 30

Ici, m_{2x} est la puissance de la trame et var_{bruit} est l'estimé de la puissance du bruit, obtenu par :

$$\mathrm{var}_{bruit}(k) = (1-\beta)\mathrm{var}_{bruit}(k-1) + \beta m_{2x}$$

Équation 31

Quelques définitions :

var_{bruit} Estimation de la puissance du bruit

k	Index de la trame
β	0.1*prob[bruit]

3.4.7 Le rapport signal sur bruit pour toutes les fréquences

Ce rapport, SNR_{total}, est calculé de la même façon que le SNR, sauf que le résidu non filtré est utilisé pour re-calculer toutes les quantités utiles, soit m_{2x}, prob[bruit], etc. SNR_{total} représente donc le rapport signal sur bruit pour toutes les fréquences.

3.5 Transition parole/bruit

Dans la machine à états finis, il y a transition de l'état parole à l'état bruit lorsque trois conditions sont remplies, simultanément, pendant une période de trois trames. Elles sont :

1. La probabilité que la trame courante est seulement du bruit doit être supérieure à un seuil de comparaison, T_{Gaus}

$$prob[bruit] = \frac{1}{2}\left(erfc(|\,SKa\,|) + erfc(|\,KUa\,|)\right) > T_{Gaus}$$

Équation 32

2. Le degré d'asymétrie avec variance unitaire, lorsque normé avec la puissance du signal, doit être plus petit qu'un seuil de comparaison, $T_{\gamma 3}$.

$$\gamma_3 = \frac{SKa}{m_{2x}^{1.5}} < T_{\gamma 3}$$

Équation 33

3. Le degré d'aplatissement avec variance unitaire, lorsque normé avec la puissance du signal, doit être plus petit qu'un seuil de comparaison, $T_{\gamma 4}$.

$$\gamma_4 = \frac{KUa}{m_{2x}^2} < T_{\gamma 4}$$

Équation 34

L'article [NEM01] ne suggère aucune valeur pour les quantités T_{Gaus}, $T_{\gamma 3}$ et $T_{\gamma 4}$. Pour l'implémentation de cet algorithme, ces valeurs ont du être déterminées par essai et erreur. Pour T_{Gaus}, une valeur de 0.1 semble donner de bons résultats. Pour ce qui est de $T_{\gamma 3}$ et $T_{\gamma 4}$, γ_3 et γ_4 sont déterminés pour les trois premières trames du signal, qui sont considérées comme étant bruitées et dénuées de parole. $T_{\gamma 3}$ est donc égal à la valeur moyenne de γ_3 pour ces trois trames, multipliée par dix. $T_{\gamma 4}$ est égal à la valeur moyenne de γ_4 pour ces trois trames, multipliées par vingt.

3.6 Transition bruit/parole

Dans la machine à états finis, il y a transition de l'état bruit à l'état parole lorsqu'une de trois condition est remplie :

1. La probabilité de bruit doit être inférieure à un seuil de comparaison pour deux trames consécutives.

$$prob[bruit] = \frac{1}{2}\left(erfc(SKa) + erfc(KUa)\right) < T_{Gaus} \quad \text{pour deux trames}$$

Équation 35

2. Le rapport SKR doit être dans la gamme [0,1] et soit que le rapport signal sur bruit doit être plus élevé qu'un seuil de comparaison, soit que l'erreur de prédiction LPC[9] doit être plus petite qu'un seuil de comparaison.

$$(SKR \in [0,1]) \quad ET \quad ((SNR > T_{SNR\_1}) \quad OU \quad (PE < T_{PE}))$$

Équation 36

$$SKR = \frac{(SK)^2}{(KUu)^{1.5}}$$

Équation 37

3. Le rapport signal sur bruit total doit être plus élevé qu'un seuil de comparaison.

$$SNR_{total} > T_{SNR\_2}$$

Équation 38

Comme précédemment, les valeurs des quantités T_{SNR_1}, T_{SNR_2}, T_{PE}, T_{Gaus}, $T_{\gamma 3}$ et $T_{\gamma 4}$ ne sont pas mentionnées dans l'article. Il a donc fallu passer un certain temps à faire essai et erreur. Les valeurs choisies sont, respectivement, 5, 3, 0.25, 0.1, pour T_{SNR_1}, T_{SNR_2}, T_{PE} et T_{Gaus}. Les quantités $T_{\gamma 3}$ et $T_{\gamma 4}$ sont données dans la section précédente.

[9] Le signal sous analyse est le résidu d'un filtre LPC, donc chaque trame a une erreur de prédiction qui lui est associée.

3.7 La machine à états finis complète

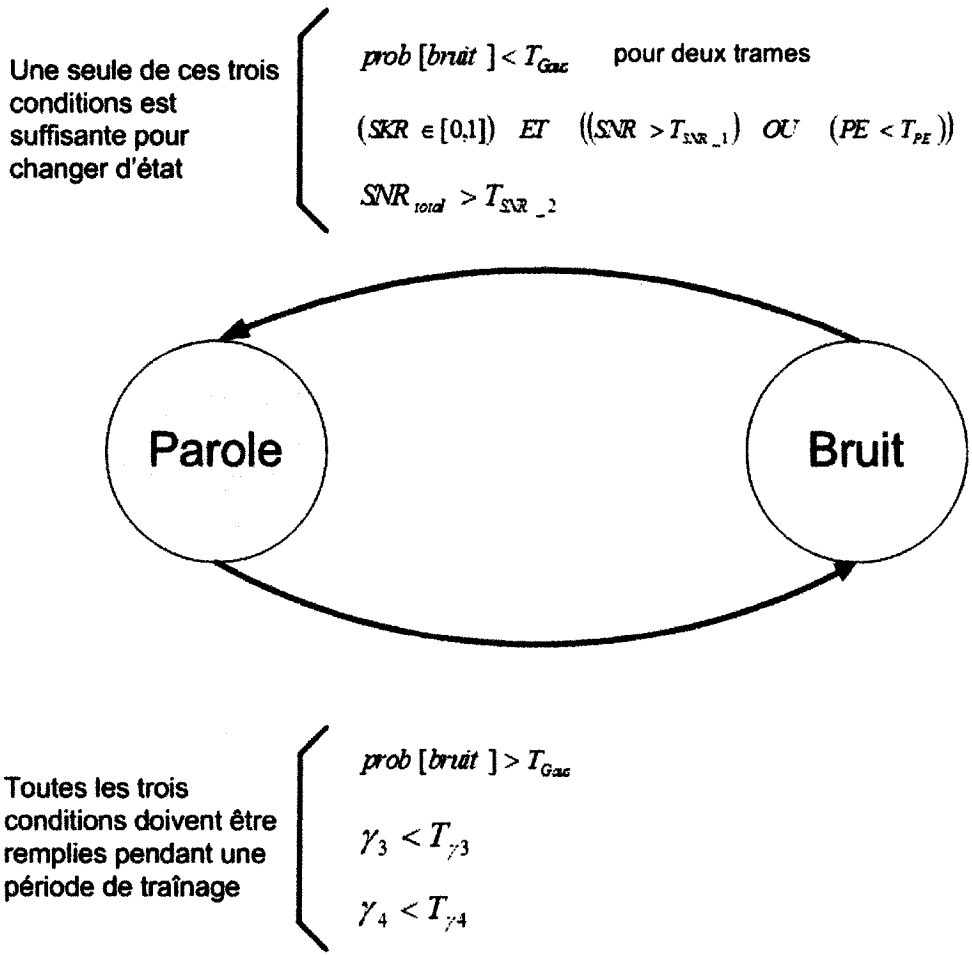


Figure 9: machine à états finis, schéma adapté de [NEM01]

3.8 Modifications pour bande large

L'algorithme présenté par [NEM01] fut développé pour des signaux à bande étroite. Afin de l'utiliser pour des signaux à bande large, les modifications suivantes furent jugées nécessaires.

3.8.1 Filtre passe-bas

La fréquence de coupure est passée de 1.8 kHz à 3.6 kHz.

3.8.2 Nombre d'échantillons par trame

Le nombre d'échantillons par trame fut augmenté afin de préserver des trames de 10 ms.

3.9 Complexité de calcul

La complexité de calcul a été mesurée en cycles d'horloge grâce à une fonction C nommée *getClockCycles( )*, invoquée par l'implémentation MATLAB du HOSVAD. Le code source de *getClockCycles( )* est inclus dans l'annexe C.

La complexité du HOSVAD a été mesurée en deux parties, soit la complexité du prétraitement de signal et la complexité du processus de décision. Cette approche a été retenue car le signal audio est découpé en trames de 20 ms pour le prétraitement et en trames de 10 ms pour le processus de décision.

En première partie, le nombre de cycles d'horloge requis pour le prétraitement du signal a été calculé pour chacune des 400 trames de 20 ms d'un signal audio de 8 secondes, afin d'en faire une moyenne. La fonction *getClockCycles( )* a été ensuite invoquée deux fois de suite afin de déterminer le temps système requis pour son exécution. La moyenne pour le prétraitement du signal a été ajustée en soustrayant cette valeur.

En deuxième partie, le nombre de cycles d'horloge requis pour le processus de décision a été calculé pour chacune des 800 trames de 10 ms d'un signal audio de 8 secondes, afin d'en faire une moyenne. La fonction *getClockCycles( )* a été ensuite invoquée deux fois de suite afin de déterminer le temps système requis pour son exécution. La moyenne pour le processus de décision a été ajustée en soustrayant cette valeur.

En dernier lieu, la complexité totale de l'algorithme pour le traitement d'une trame de 10 ms est obtenue en additionnant la moitié du nombre de cycles

d'horloge requis pour le prétraitement du signal avec le nombre de cycles d'horloge requis pour le processus de décision.

Étape	Moyenne du nombre de cycles d'horloge requis
Prétraitement du signal, trame de 20 ms	5.2561×10^7
Processus de décision, trame de 10 ms	1.5268×10^6
Moyenne pour une trame de 10 ms	2.78073×10^7

Tableau 2: Complexité de calcul pour le HOSVAD

Note :
La complexité de calcul a été mesurée pour un microprocesseur Pentium 4 cadencé à 3.0 GHz

3.10 Conclusion

Cet algorithme, présenté en [NEM01], utilise des mesures fondamentales basées sur les cumulants, tels que les degrés d'asymétrie et d'aplatissement, afin de déterminer la présence d'activité vocale. L'algorithme a été modifié pour bande large en doublant la fréquence de coupure du filtre passe-bas, lors du prétraitement du signal, et en doublant le nombre d'échantillons par trame afin de préserver des trames de 10 ms.

Chapitre trois

4 Description des VADs commerciaux utilisés dans cette thèse

4.1 Introduction

Trois VADs sont comparés dans cette thèse : le G729b, le G722.2 et le HOS VAD proposé par [NEM01]. Le HOSVAD ayant été décrit dans le précédant chapitre, il convient maintenant de décrire le G729b et le G722.2. La complexité mesurée des trois algorithmes est aussi présentée.

4.2 G729b

4.2.1 Survol de l'algorithme

Le VAD du G729b utilise quatre mesures afin de déterminer s'il y a activité vocale ou non. Ces mesures sont l'énergie de la pleine bande, l'énergie des fréquences basses, l'ensemble des fréquences de raies spectrales (LSF) et le nombre de passage par zéro.

Le VAD du G729b découpe le signal en trames de 80 échantillons, soit 10 ms lorsqu'un signal échantillonné à 8 kHz est utilisé, et classifie chaque trame comme étant active ou non active en cinq étapes :

1. Quatre mesures sont calculées :

    1.1. Énergie de la pleine bande

    1.2. Énergie des fréquences basses

    1.3. Ensemble des fréquences de raies spectrales

    1.4. Nombre de passage par zéro

Note :
Les 32 premières trames sont utilisées afin d'initialiser les moyennes à long terme de ces mesures. Pendant ces 32 trames, les trames ayant une énergie, obtenue à l'aide d'analyse de codage de prédiction linéaire, supérieure à 15 dB sont considérées actives. Autrement, les trames sont considérées non actives.
La 33ème trame est utilisée afin d'initialiser les énergies caractéristiques du bruit de fond.

2. Les paramètres de différence sont calculés, soit la différence entre les mesures calculées à l'étape 1 et leurs moyennes glissantes.

3. Une décision initiale est prise en tenant compte de quatorze frontières de décision. Chaque frontière compare un paramètre de différence avec soit une constante, soit une combinaison de constante et d'un autre paramètre. Si aucune des quatorze frontières de décision n'est vraie, alors la décision initiale est que la trame est non active[10].

4. Un lissage de décision est effectué en tenant compte de l'énergie de la trame et des décisions antérieures des trames voisines.

5. Si le VAD croit être en présence d'une trame non active, alors les moyennes glissantes sont mises à jour.

[10] Voir [G729b] section B.3.5 pour de plus amples détails sur les quatorze frontières de décision.

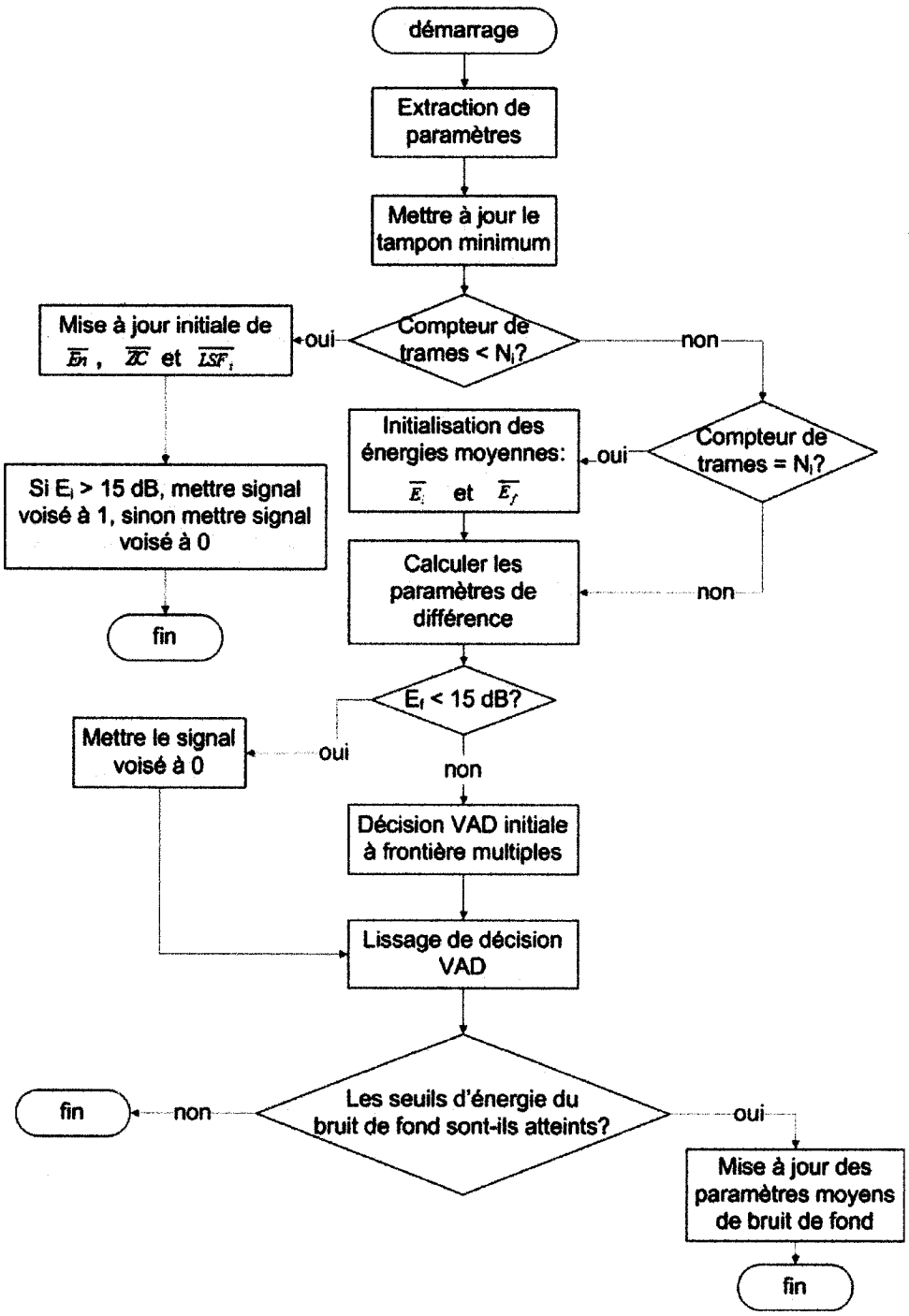


Figure 10: organigramme du VAD G729b, reproduit de [G729b]

Voici une description des variables et constantes utilisées dans l'organigramme du G729b :

Nom	Type	Valeur
$\overline{En}$	variable	Moyenne de l'énergie des N_i premières trames.
$\overline{ZC}$	variable	Moyenne du nombre de passage par zéro des N_i premières trames.
$\overline{LSF_i}$	variable	Moyenne des fréquences de raies spectrales des N_i premières trames.
N_i	constante	32
$\overline{E_l}$	variable	Moyenne glissante de l'énergie dans la bande des fréquences basses de bruit de fond.
$\overline{E_f}$	variable	Moyenne glissante de l'énergie de bruit de fond.
E_l	variable	Énergie dans la bande des fréquences basses de bruit de fond pour la trame courante.
E_f	variable	Énergie de bruit de fond pour la trame courante.

Tableau 3: description des variables et constantes utilisées dans l'organigramme du G729b

4.2.2 Différences dans l'application en bande large

L'algorithme VAD du G729b a été utilisé tel quel avec les signaux échantillonnés 16 kHz, ce qui veut dire que les trames analysées étaient en fait de 5 ms.

Une deuxième conséquence est que E_l et E_f comprennent le double de fréquence qu'en bande étroite.

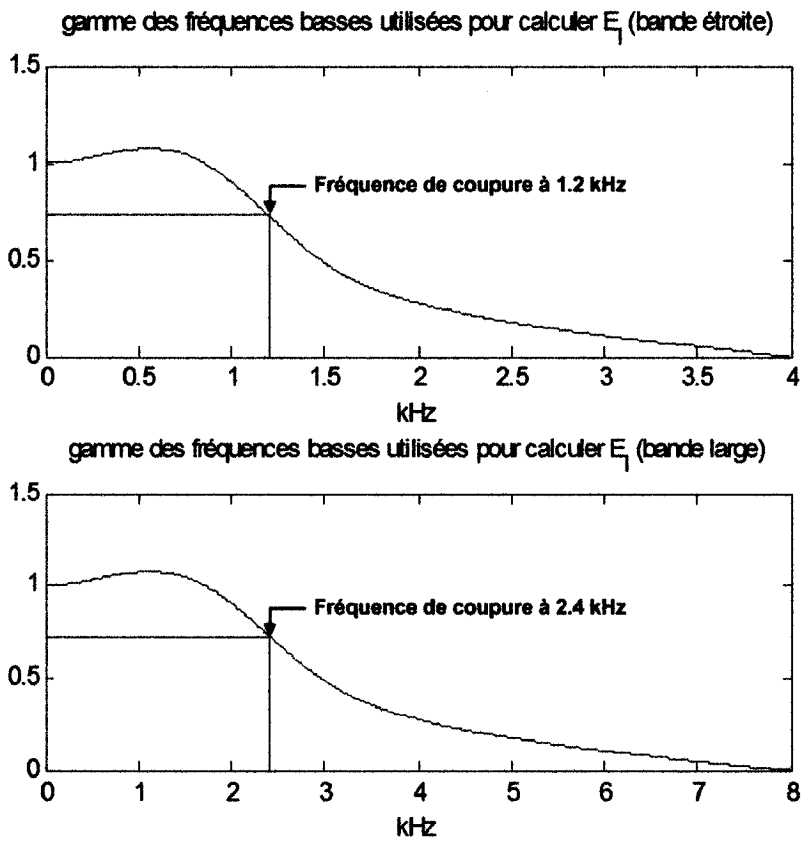


Figure 11: gamme des fréquences utilisées pour calculer E_l

4.2.3 Complexité de calcul

La complexité de calcul a été mesurée en cycles d'horloge grâce à une fonction C nommée *GetCC( )*, insérée dans le code du G729b. Le code source de *GetCC( )* est inclus dans l'annexe C.

Le nombre de cycles d'horloge requis pour l'exécution de l'algorithme vad du G729b a été déterminé pour chaque trame de 5 ms d'un signal audio de 8 secondes, afin d'en faire une moyenne. La fonction *GetCC( )* a été ensuite invoqué deux fois de suite afin de déterminer le temps système requis pour son exécution. La moyenne du nombre de cycles d'horloge requis pour le vad du G729b a été ajustée en soustrayant cette valeur. Le vad du G729b a donc besoin de 1.4604 x 10^5 cycles d'horloge afin de déterminer si une trame de 5 ms, échantillonnée à 16 kHz, est active ou non active.

Note :
La complexité de calcul a été mesurée pour un microprocesseur Pentium 4 cadencé à 3.0 GHz.
Aussi, le code C fourni par l'UIT-T a été utilisé dans cette thèse. C'est une implémentation de référence seulement : elle n'est pas optimisée. Les applications commerciales utilisant ce standard l'implémentent de façon plus efficace.

4.3 G722.2

4.3.1 Survol de l'algorithme

Le VAD du codeur UTI-T G722.2 utilise comme mesure principale le niveau d'énergie contenu dans diverses sous-bandes du signal d'entrée. Le signal est découpé en trames de 20 ms échantillonnées à 12,8 kHz qui, tour à tour, seront décomposées en ces douze sous-bandes :

Numéro de bande	Fréquences
1	0-200 Hz
2	200-400 Hz
3	400-600 Hz
4	600-800 Hz
5	800-1200 Hz
6	1200-1600 Hz
7	1600-2000 Hz
8	2000-2400 Hz
9	2400-3200 Hz
10	3200-4000 Hz
11	4000-4800 Hz
12	4800-6400 Hz

Tableau 4: sous-bandes utilisées par le VAD du G722.2

Chaque sous-bande est soumise à un détecteur de tonalité qui indique la présence de signal à périodicité élevée, tels une tonalité de signalisation ou un signal voisé. Ensuite, le niveau d'énergie est calculé pour chaque sous-bande alors que le niveau du bruit de fond n'est calculé que pour les trames classifiées non actives par le VAD, qui ne comprennent pas de tonalité de signalisation et dont le signal est stationnaire. La condition de stationnarité est requise afin d'éviter que des

augmentations transitoire du niveau de bruit de fond ne faussent les décisions du VAD.

Lorsque les mesures d'énergie du signal et du bruit de fond sont obtenues, le ratio du SNR du signal d'entrée avec le SNR du bruit de fond est calculé. Ce ratio est ensuite comparé à un seuil adaptatif afin d'obtenir une décision intermédiaire (trame active ou non-active). La décision finale du VAD est obtenue par traînage des décisions intermédiaires.

Le schéma qui suit illustre l'opération du VAD :

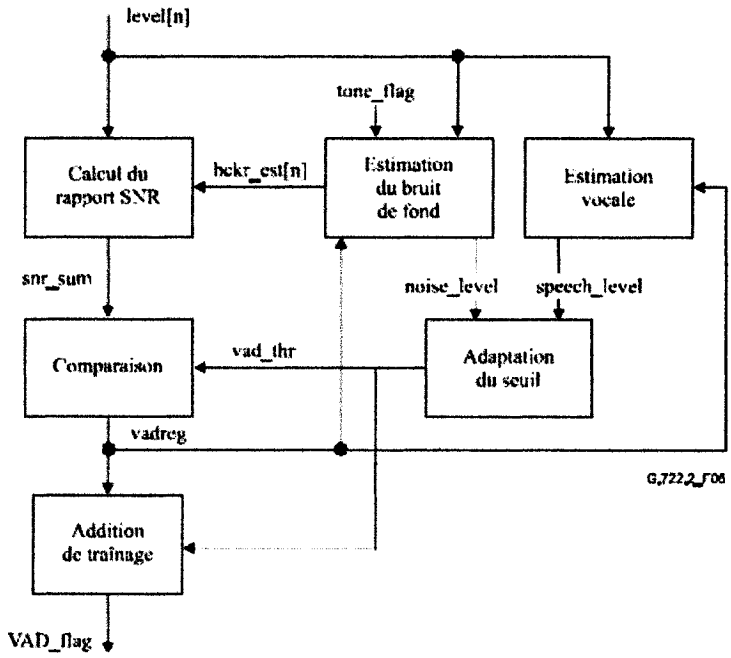


Figure 12: schéma fonctionnel simplifié de l'algorithme de décision VAD du G722.2[11]

[11] Schéma tiré de [G722], page 62

4.3.2 Complexité de calcul

La complexité de calcul a été mesurée en cycles d'horloge grâce à une fonction C nommée *GetCC( )*, insérée dans le code du G729b. Le code source de *GetCC( )* est inclus dans l'annexe C.

Le nombre de cycles d'horloge requis pour l'exécution de l'algorithme vad du G722.2 a été déterminé pour chaque trame de 20 ms d'un signal audio de 8 secondes, afin d'en faire une moyenne. La fonction *GetCC( )* a été ensuite invoquée deux fois de suite afin de déterminer le temps système requis pour son exécution. La moyenne du nombre de cycles d'horloge requis pour le vad du G722.2 a été ajustée en soustrayant cette valeur. Le vad du G722.2 a donc besoin de 4.2183 x 10^6 cycles d'horloge afin de déterminer si une trame de 20 ms, échantillonnée à 16 kHz, est active ou non active.

Note :
La complexité de calcul a été mesurée pour un microprocesseur Pentium 4 cadencé à 3.0 GHz.
Aussi, le code C fourni par l'UIT-T a été utilisé dans cette thèse. C'est une implémentation de référence seulement : elle n'est pas optimisée. Les applications commerciales utilisant ce standard l'implémentent de façon plus efficace.

4.4 Comparaison de complexité

Afin de mieux comparer la complexité des trois VADs, il est utile d'ajuster les valeurs mesurées afin qu'elles reflètent l'effort moyen nécessaire pour traiter une trame de 10 ms.

Algorithme VAD	**Complexité, en cycles d'horloge, pour une trame de 10 ms échantillonnée à 16 kHz**
HOSVAD	2.78073×10^7
G729b	2.92080×10^5
G722.2	2.10920×10^6

Tableau 5: complexité des trois algorithmes VADs

Le HOSVAD requiert cent fois plus de cycles d'horloge que le G729b pour une trame de 10 ms, et dix fois plus que le G722.2. Pour le HOSVAD, le prétraitement du signal, plus particulièrement l'analyse de codage de prédiction linéaire, est l'opération avec le coût le plus élevé. Cette analyse utilise des cumulants du troisième ordre afin de déterminer les coefficients LPC, selon la méthode suggérée par [PAL91].

$$\sum_{k=0}^{p} a_k C_k(i,j) = 0 \quad 1 \le i \le p, \quad 0 \le j \le i$$

Équation 39

Pour cette méthode, les cumulants sont définis comme suit :

$$C_k(i,j) = \sum_{n=p+1}^{N} s_{n-k} s_{n-i} s_{n-j}$$

Équation 40

Ici, p est l'ordre de prédiction, s est une trame de N échantillons et a_k sont les coefficients LPC désirés.

Selon la méthode de [PAL91], nous pouvons donc obtenir les coefficients a_k avec ces équations :

$$\begin{bmatrix} C_0(1,0) & C_1(1,0) & \text{K} & C_{10}(1,0) \\ C_0(1,1) & C_1(1,1) & \text{K} & C_{10}(1,1) \\ \text{M} & \text{M} & \text{M} & \text{M} \\ C_0(4,1) & C_1(4,1) & \text{K} & C_{10}(4,1) \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \text{M} \\ a_{10} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \text{M} \\ 0 \end{bmatrix}$$

Équation 41

Malheureusement, ceci n'est pas une matrice Toeplitz et donc l'algorithme Levinsin-Durbin ne peut être employé [PRO96]. Swami, Mendel et Nikias [SWA01] suggèrent d'utiliser la méthode récursive Trench afin de résoudre cette équation, mais des essais pratiques avec la méthode Trench se sont heurtés à des instabilités numériques, plus précisément des divisions par zéro. Pour résoudre la matrice ci haut et obtenir les coefficients a_k, la méthode d'élimination Gauss-Jordan a donc dû être employée. Cette méthode a une complexité de $O(n^3)$, où n est la dimension de la matrice carrée. Dans notre cas, n=11 ce qui donne donc une complexité de calcul de O(1331) à chaque trame de 20 ms, ce qui explique le coût élevé, en cycles d'horloge, du prétraitement du signal[12].

[12] Voir le tableau 2, à la section 3.8.

Pour ce qui est du G722.2, son coût en cycles d'horloges est dix plus élevé que le G729b car il divise le signal d'entrée en douze sous-bandes de fréquences, et donc répète les mêmes types de calculs douze fois. Il est aussi à noter que le code référence de l'UIT-T a été utilisé dans cette thèse pour le G729b et le G722.2. Les implémentations utilisées en industrie de ces algorithmes sont vraisemblablement plus efficaces du point de vue de la complexité de calcul.

4.5 Conclusion

Le fonctionnement des algorithmes VAD du G729b et du G722.2 a été présenté dans ce chapitre. La complexité de calcul, mesurée en cycles d'horloge, a été présentée pour le G729b, le G722.2 ainsi que le HOSVAD. Ce dernier est le plus complexe, suivi du G722.2. Le G729b requiert le moins de calcul par trame.

Chapitre cinq

5 Test effectués

5.1 Introduction

Plusieurs tests ont été effectués avec les trois VADs sous étude afin de comparer leur performance en présence de différents bruits et ce, à plusieurs rapports signal sur bruit. Ce chapitre décrit ces tests, présente les résultats pour chaque test et termine avec une synthèse des résultats.

5.2 Brève description des tests effectués

Chaque VAD sous étude a été utilisé pour l'analyse d'activité vocale de 16 fichiers audio comprenant de la parole propre. Ensuite, l'analyse de ces mêmes fichiers audio a été reprise en ajoutant tour à tour à chaque fichier propre:

1. Du bruit blanc avec six SNR allant de -5 dB à 20 dB, pour un total de 96 fichiers audio corrompus par ce type de bruit.

2. Du bruit coloré[13] avec six SNR allant de -5 dB à 20 dB, pour un total de 96 fichiers audio corrompus par ce type de bruit.

3. Du bruit de murmure, soit de la parole en arrière-plan, avec six SNR allant de -5 dB à 20 dB, pour un total de 96 fichiers audio corrompus par ce type de bruit.

4. Du bruit de réverbération correspondant à un long couloir, pour un total de 16 fichiers audio corrompus avec ce type de bruit.

5. Du bruit de blanc ajouté au bruit de réverbération de l'item précédent, avec un SNR de 50 dB, pour un total de 16 fichiers audio corrompus avec ce type de bruit.

6. Du bruit de rue avec six SNR allant de -5 dB à 20 dB, pour un total de 96 fichiers audio corrompus par ce type de bruit.

[13] Voir l'annexe A pour une description plus détaillée du bruit coloré

7. Du bruit provenant de l'intérieur d'une voiture en marche avec six SNR allant de -5 dB à 20 dB, pour un total de 96 fichiers audio corrompus par ce type de bruit.

Incluant les 16 fichiers non corrompus par le bruit, chaque VAD a ainsi analysé 528 fichiers audio, sous différents types de bruits et avec différents SNR.

5.3 Fichiers audio propres utilisés

Les 16 fichiers audio propres proviennent de [P23]. Ils proviennent de deux hommes et deux femmes, chacun ayant enregistré quatre fichiers audio distincts. Chaque fichier débute par un court silence, ensuite la première moitié de la phrase est énoncée, suivi d'un plus long silence pour se terminer par la deuxième moitié de la phrase. Chaque fichier audio a été échantillonné à 16 kHz[14].

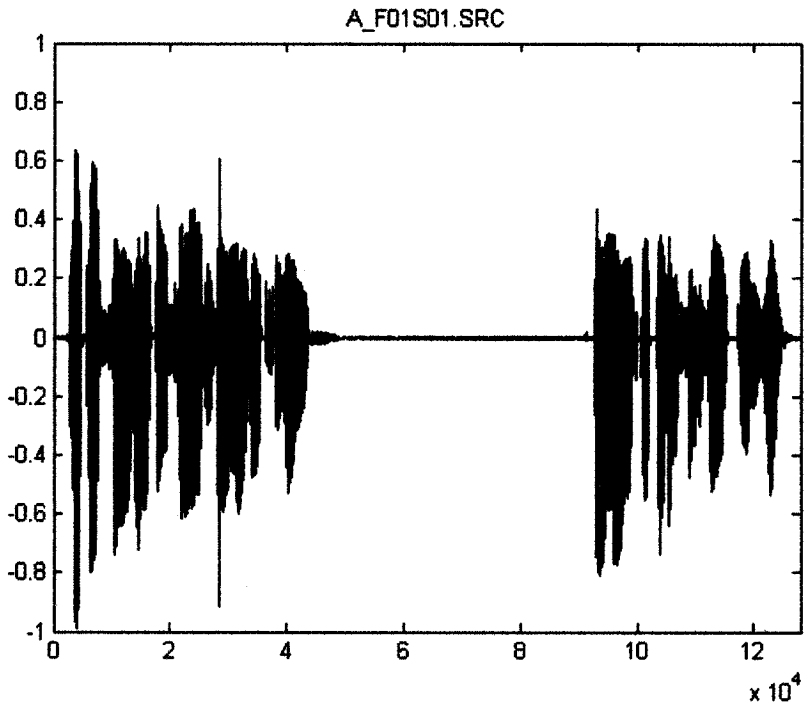


Figure 13: exemple de fichier audio propre

[14] Voir l'annexe B pour le PSD de tous les fichiers audio propres utilisés.

5.4 Fichiers audio corrompus par le bruit

Ces fichiers ont été créés par mixage de bruit échantillonné à 16 kHz[15] avec les fichiers propres, soit avec le logiciel SoundForge 7.0 pour le bruit de réverbération, soit avec MATLAB pour le reste.

5.5 Mesures utilisées

Tout comme dans [BER98], [BER01] et [BER02], quatre types de mesures sont utilisées afin de comparer l'efficacité des VADs, soit :

1. La mutilation en début de parole (**mdp**), soit des trames considérées comme non actives au début d'une période d'activité vocale.

2. La mutilation en milieu de parole (**mmp**), soit des trames considérées comme non active au milieu d'une période d'activité vocale.

3. Des trames actives en trop (**trop**), lorsque le VAD tarde à passer de l'état parole à l'état bruit, ce qui donne un surplus de trames considérées comme actives.

4. Le bruit faussement classifié en parole (**bep**), lorsque le VAD classifie faussement des trames en parole lors de périodes de non activité.

[15] Voir l'annexe B pour le PSD de tous les signaux de bruits utilisés.

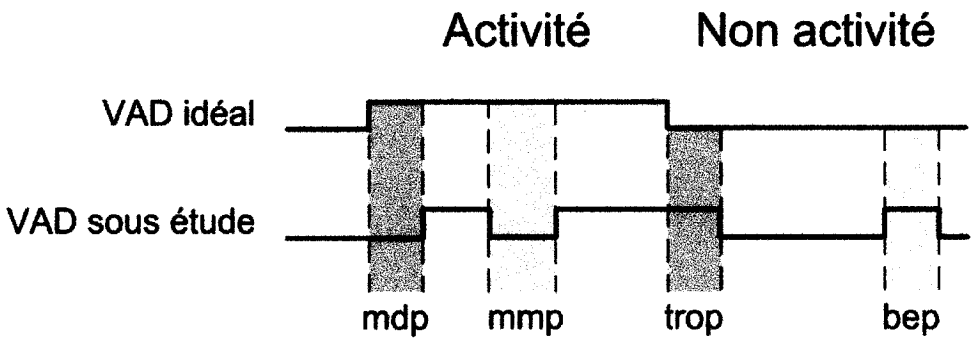


Figure 14: mesures utilisées, schéma adapté de [BER01]

Pour chaque type de bruit, la moyenne de ces mesures, calculée en pourcentage de toutes les trames, est présentée dans la section qui suit.

Note :

La distribution des erreurs n'est pas considérée, seulement l'erreur totale. Les deux VADs sous études de la Figure 15 seraient donc considérés comme équivalents. Des tests subjectifs pourraient donc être requis pour la sélection d'un VAD pour usage pratique.

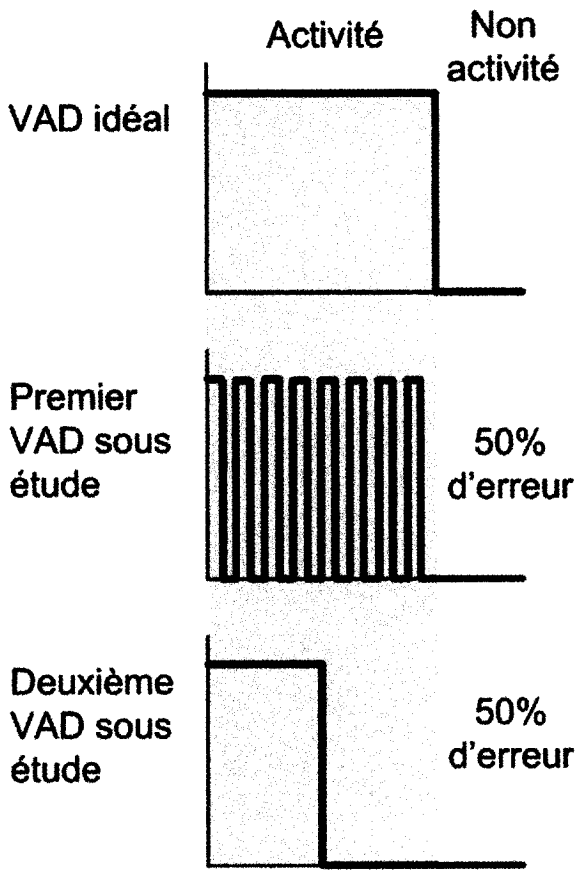


Figure 15: Exemples de distributions d'erreur

5.6 Résultats

5.6.1 Fichiers audio propres

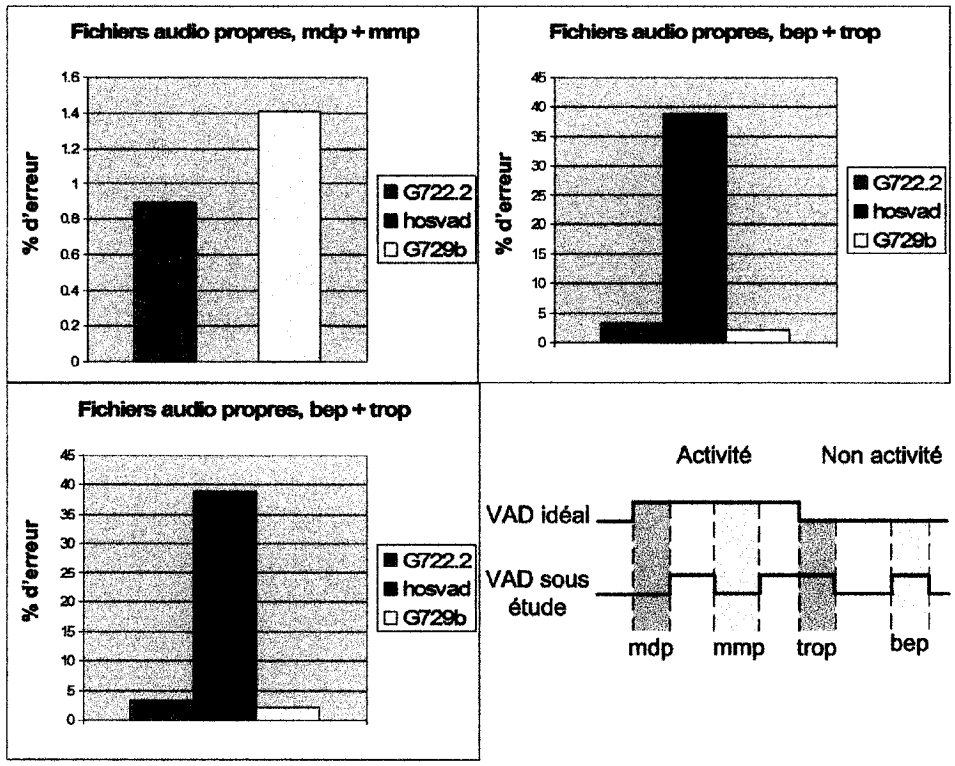


Figure 16: statistiques pour fichiers audio propres

Note :

Les taux de mutilation en début et en milieu de parole sont zéros pour le HOSVAD étant donné qu'il classifie la plupart des trames en parole.

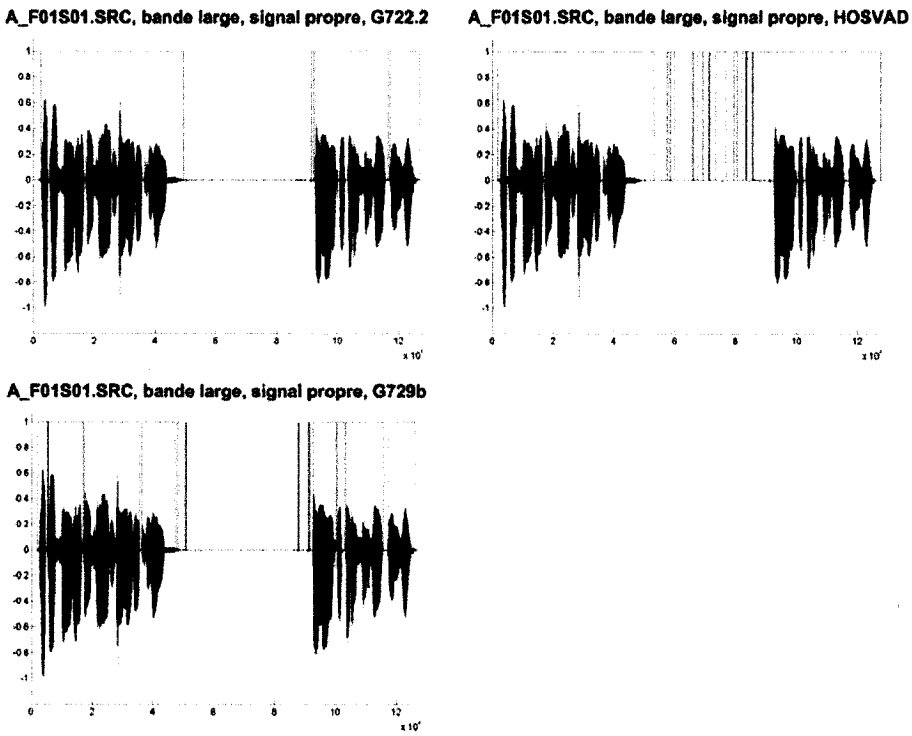


Figure 17: résultat typique pour fichier audio propre

Sans grande surprise, les VADs du G722.2 et du G729b sont très efficaces dans ce cas. Le HOSVAD, quant à lui, classifie beaucoup de trames de silence comme étant des trames de parole, ce qui peut être expliqué par le fait que les quantités γ_3 et γ_4, utilisées afin de transiger de l'état parole à l'état bruit, sont normalisées par la puissance de la trame sous analyse[16]. Pour un fichier audio propre, les trames sans activité vocale ont une puissance qui tend vers zéro, donc les quantités γ_3 et γ_4 tendent vers l'infini et les conditions $\gamma_3 = \frac{SKa}{m_{2x}^{1.5}} < T_{\gamma 3}$ et

[16] Voir section 3.4 de cette thèse.

$\gamma_4 = \frac{KUa}{m_{2x}^2} < T_{\gamma 4}$ ne sont pas remplies[17]. Il est à noter, cependant, que du bruit sera présent pour toute application où un VAD est utile, comme dans la téléphonie cellulaire, ce qui rend l'analyse des fichiers audio propre un exercice théorique, sans impact majeur sur le choix d'un algorithme VAD.

5.6.2 Réverbération et signal propre

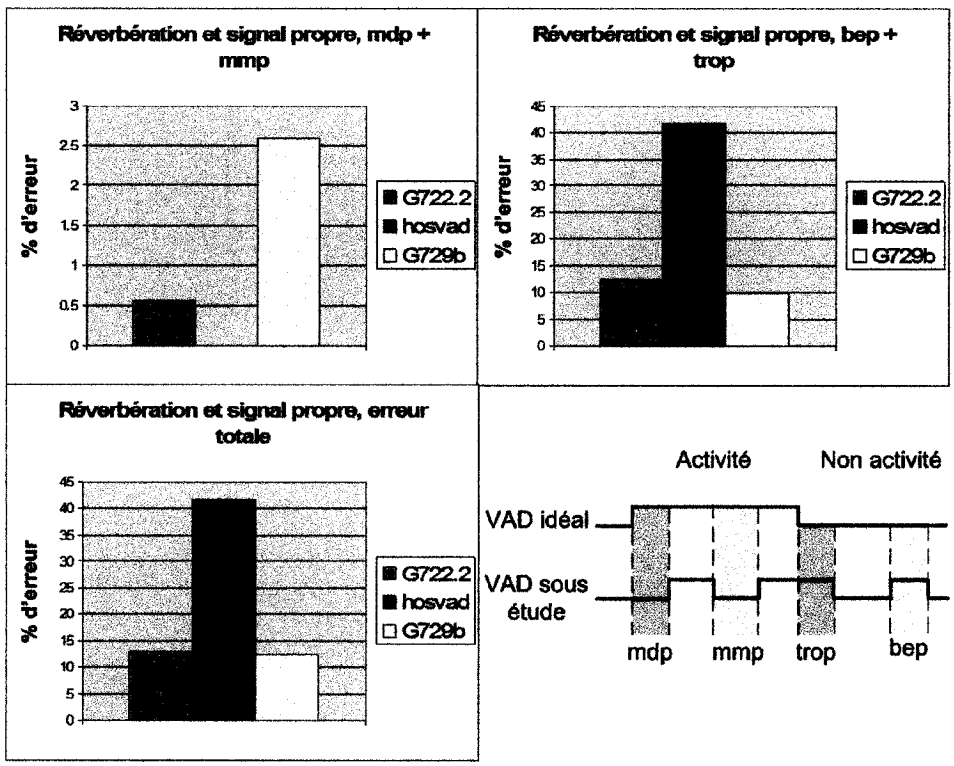


Figure 18: statistiques pour réverbération et signal propre

[17] Voir la machine à états finis du HOSVAD à la section 3.6 de cette thèse.

Note :

Les taux de mutilation en début et en milieu de parole sont zéros pour le HOSVAD étant donné qu'il classifie la plupart des trames en parole.

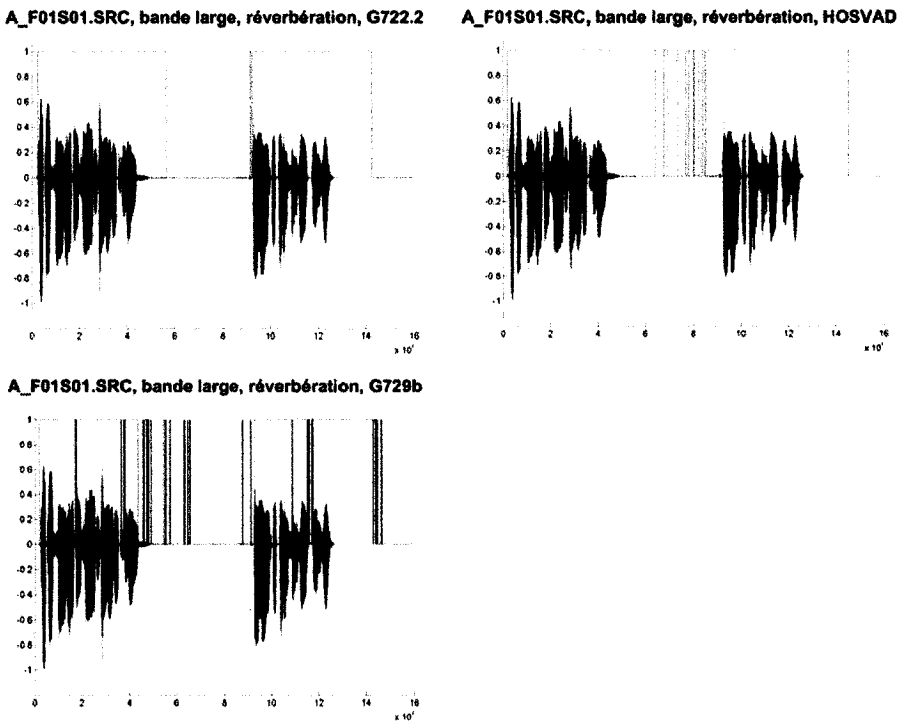


Figure 19: résultat typique pour réverbération et signal propre

Pour la réverbération, le G722.2 et le G729b ont des performances similaires, quoique le G722.2 ait des taux de mutilation en début et en milieu de parole légèrement inférieur au G729b. Le HOSVAD, quant à lui, classifie plusieurs trames de bruit en trames de parole. Étant donné que c'est le signal propre qui a été réverbéré, le HOSVAD souffre donc des mêmes maux qu'avec le fichier audio propre. Afin d'obtenir une meilleure appréciation de la performance du

HOSVAD dans le cas de la réverbération, un bruit blanc plutôt faible, soit avec un SNR de 50 dB, a été ajouté. Les résultats sont présentés à la section suivante.

5.6.3 Réverbération avec bruit blanc, SNR = 50 dB

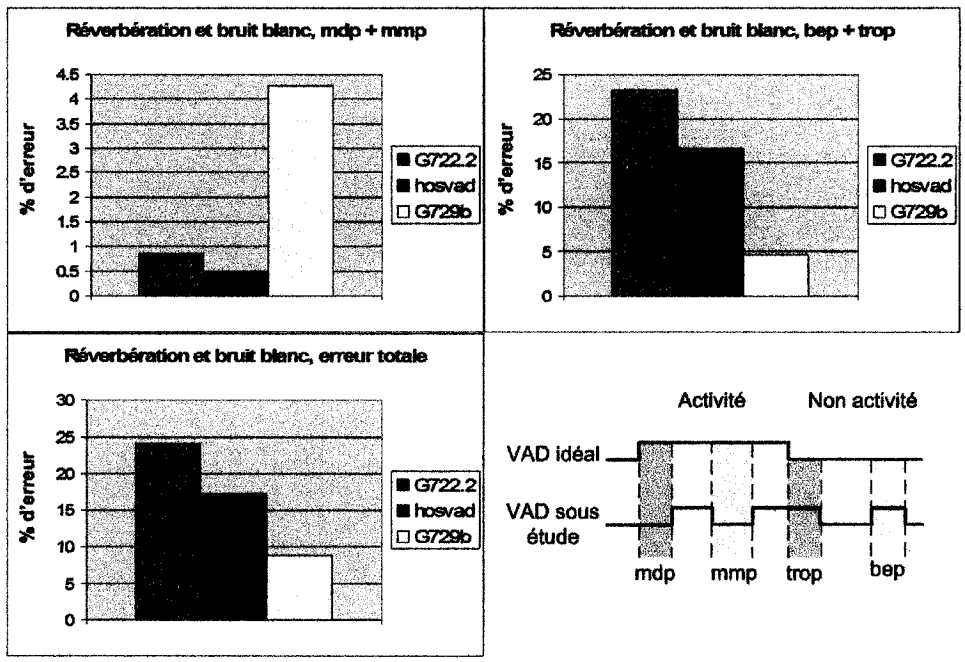


Figure 20: statistiques pour réverbération et bruit blanc

Note :

Pour l'ajout de bruit blanc, le SNR de 50 dB a été calculé à partir du signal propre non réverbéré. Aussi, le pourcentage d'erreur est calculé par comparaison avec le VAD idéal du signal propre non réverbéré.

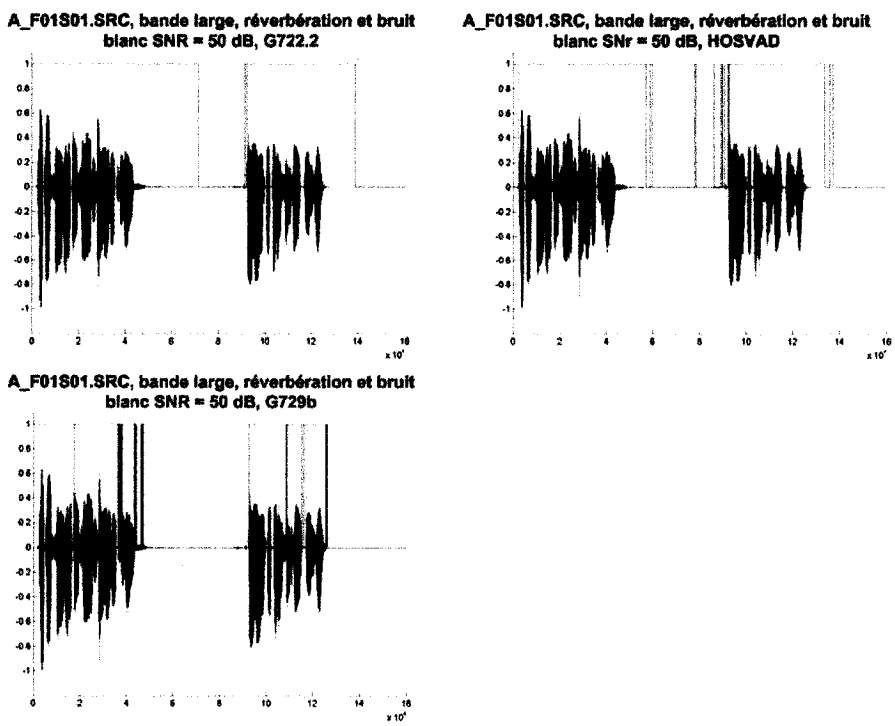


Figure 21: résultat typique pour réverbération et bruit blanc

En ajoutant du bruit blanc avec un SNR de 50 dB, la performance du HOSVAD s'améliore grandement car maintenant les trames non actives ont une puissance un peu plus élevée, ce qui borne les quantités γ_3 et γ_4 et ce qui permet ainsi de remplir les conditions $\gamma_3 = \frac{SKa}{m_{2x}^{1.5}} < T_{\gamma 3}$ et $\gamma_4 = \frac{KUa}{m_{2x}^{2}} < T_{\gamma 4}$ nécessaires pour transiger de l'état parole à l'état bruit[18]. Le HOSVAD se classe maintenant deuxième derrière le G729b.

[18] Voir la machine à états finis du HOSVAD à la section 3.6 de cette thèse.

5.6.4 *Bruit blanc*

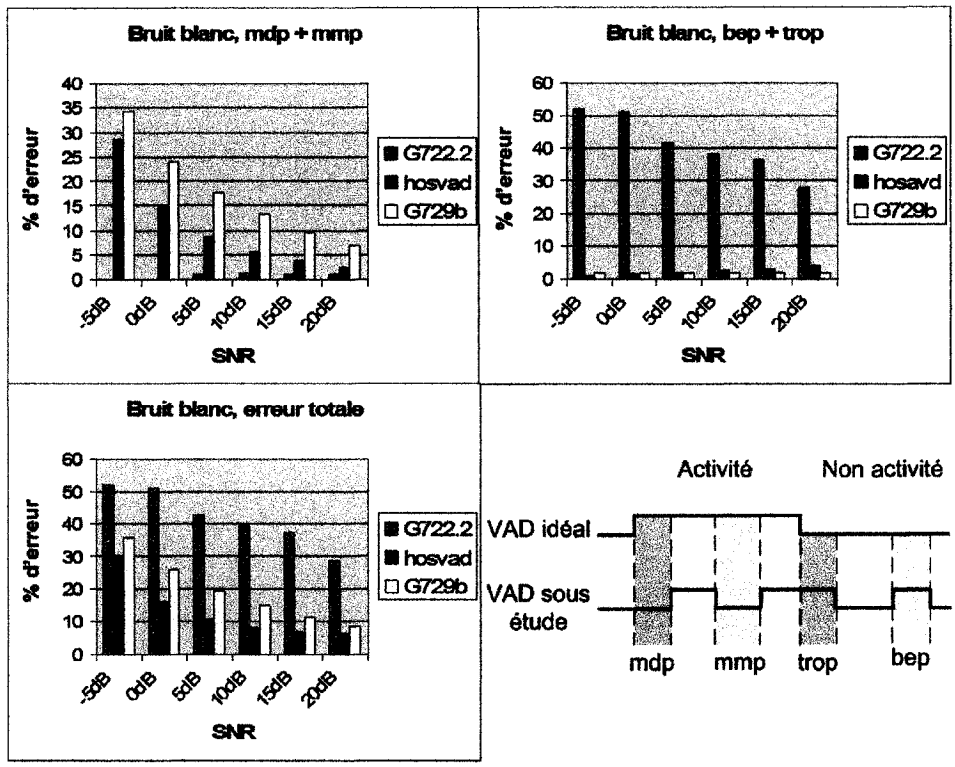


Figure 22: statistiques pour bruit blanc

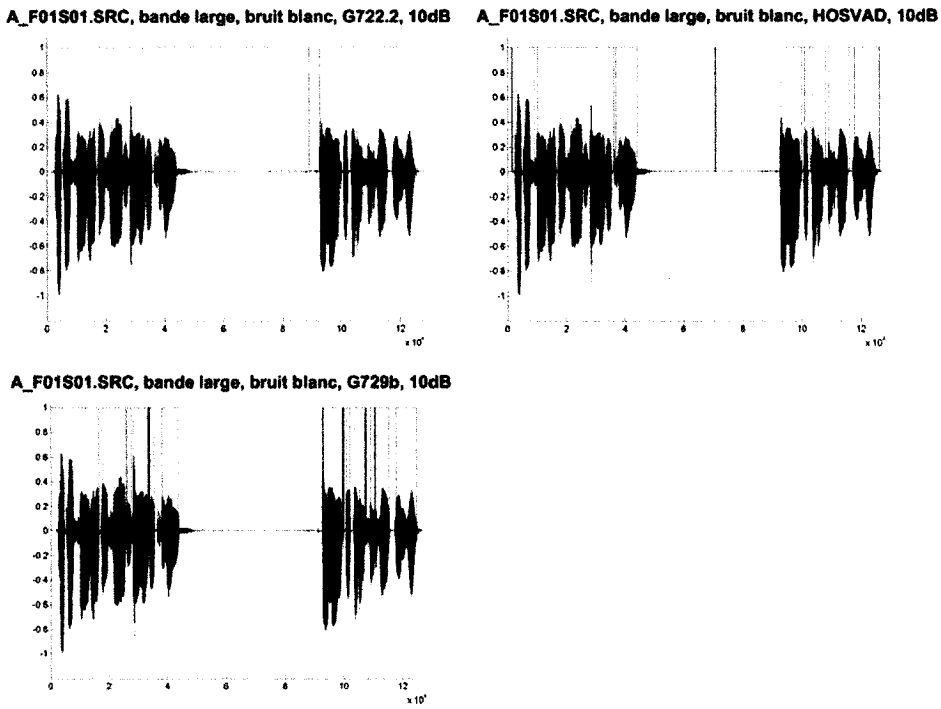


Figure 23: résultat typique pour bruit blanc, SNR = 10dB

Pour le bruit blanc, le HOSVAD a une performance nettement supérieure au G722.2 et il devance également le G729b. La bonne performance du HOSVAD peut être expliquée par le fait que cet algorithme utilise des statistiques du troisième et quatrième ordre qui sont, en théorie, insensibles aux procédés Gaussiens, comme le bruit blanc. Par contre, il est à noter que le HOSVAD utilise une statistique du deuxième ordre, soit la puissance de la trame, afin de normaliser certaines quantités, comme γ_3 et γ_4, ce qui a pour effet de rendre le HOSVAD un peu sensible à la puissance du bruit mais tout de même moins que le G722.2 et le G729b.

5.6.5 *Bruit coloré*

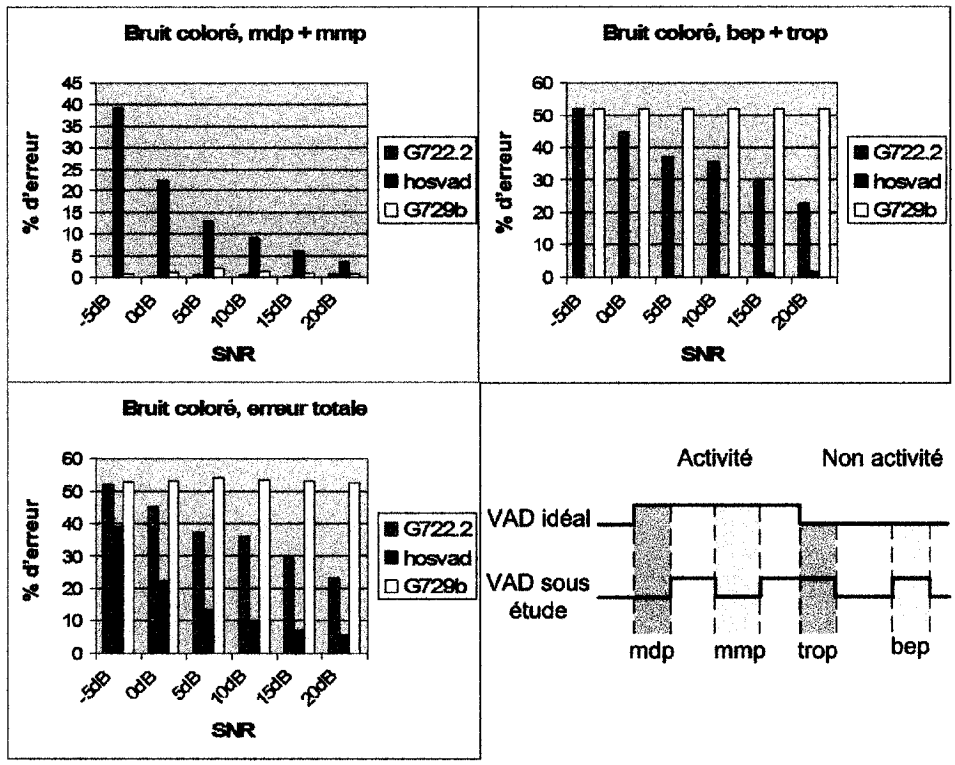


Figure 24: statistiques pour bruit coloré

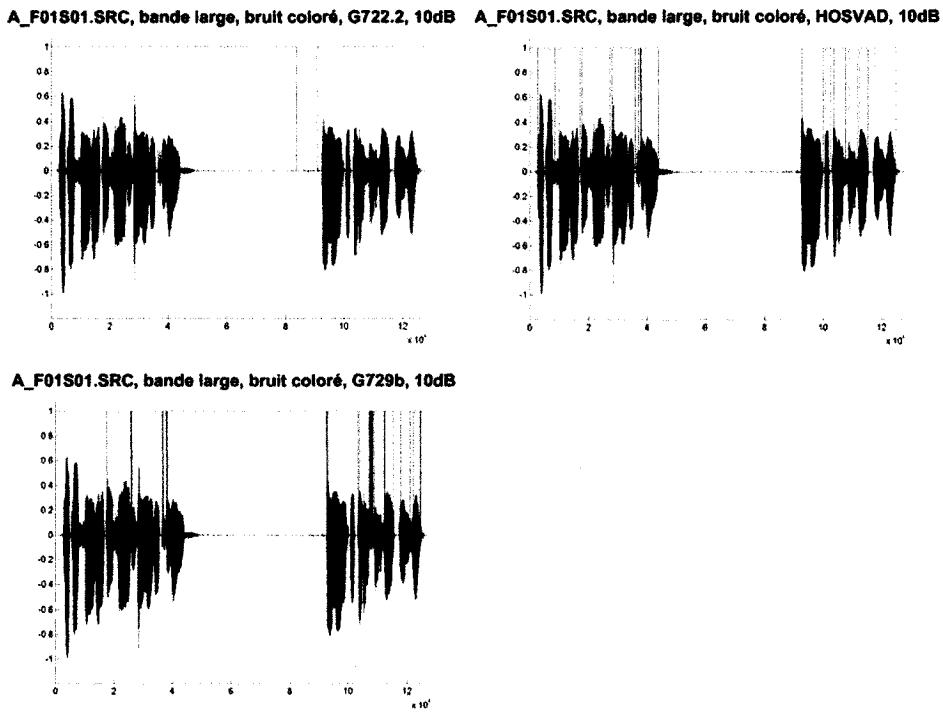


Figure 25: résultat typique pour bruit coloré, SNR = 10dB

Le HOSVAD réussit très bien à distinguer les trames actives et non actives en la présence de bruit coloré, ce que le G729b et le G722.2 peinent à faire. La bonne performance du HOSVAD peut être liée au fait que le bruit coloré est tout de même un procédé Gaussien, et donc, selon [SWA01] et [MEN91], les cumulants des troisième et quatrième ordres seront tout de même zéro pour ce bruit et donc le HOSVAD aura une performance similaire à lorsqu'il y a du bruit blanc. Une brève comparaison des figures 22 et 20 confirme la performance semblable du HOSVAD lorsque soumis à du bruit blanc et lorsque soumis à du bruit coloré.

Il est intéressant de noter que la plupart du contenu spectral du bruit coloré se retrouve en basse fréquence[19], ce qui explique la sensibilité du G722.2 et du G729b à ce bruit.

5.6.6 Bruit de murmure

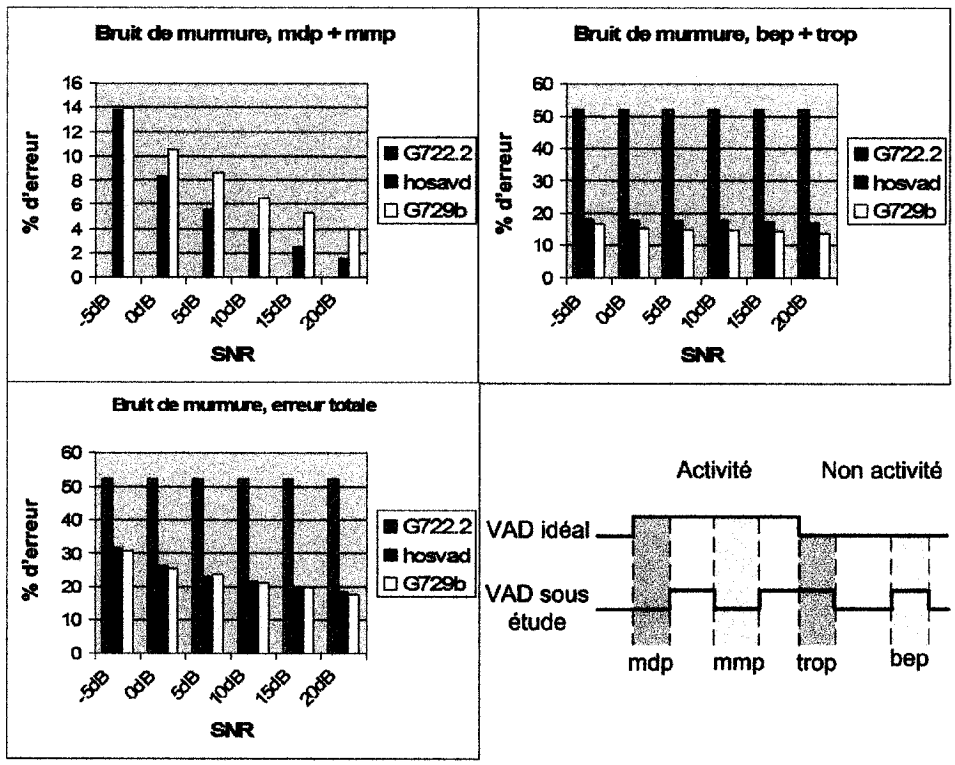


Figure 26: statistiques pour bruit de murmure

[19] Voir le PSD du bruit coloré dans l'annexe B.

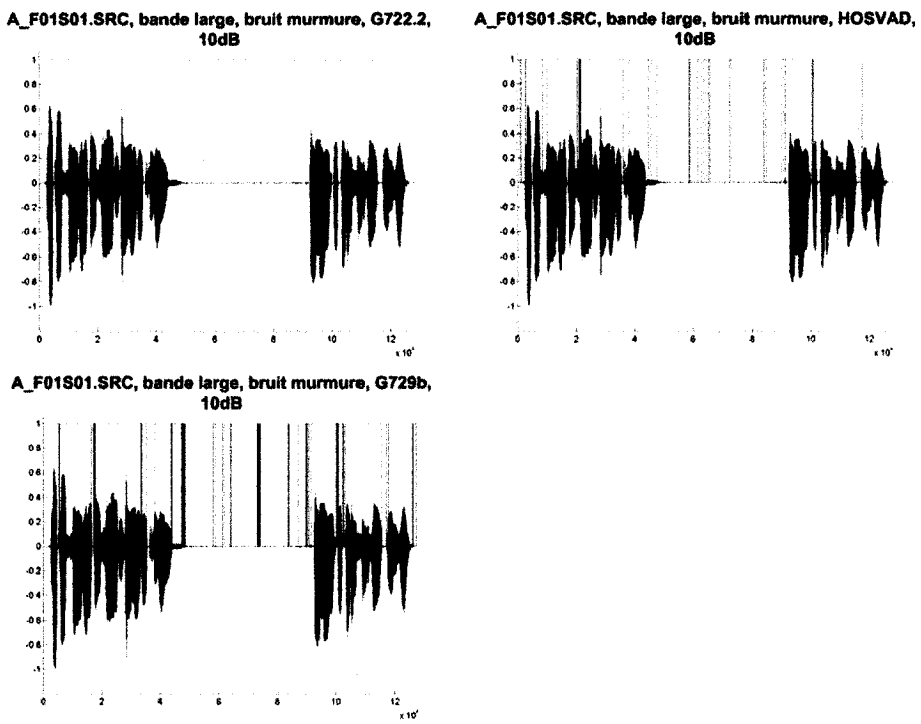


Figure 27: résultat typique pour bruit de murmure,
SNR = 10 dB

Bien que le G729b ait une erreur totale légèrement inférieure à celle du HOSVAD, ce dernier a des taux de mutilation en début et en milieu de parole plus avantageux.

Le G722.2 a de la difficulté à distinguer les trames de bruit parce que son algorithme VAD, qui est basé sur le SNR, ne calcule et met à jour le niveau du bruit de fond que pour les trames dont le signal est stationnaire. Or, le bruit de murmure est non stationnaire, ce qui explique les difficultés du G722.2.

5.6.7 *Bruit de rue*

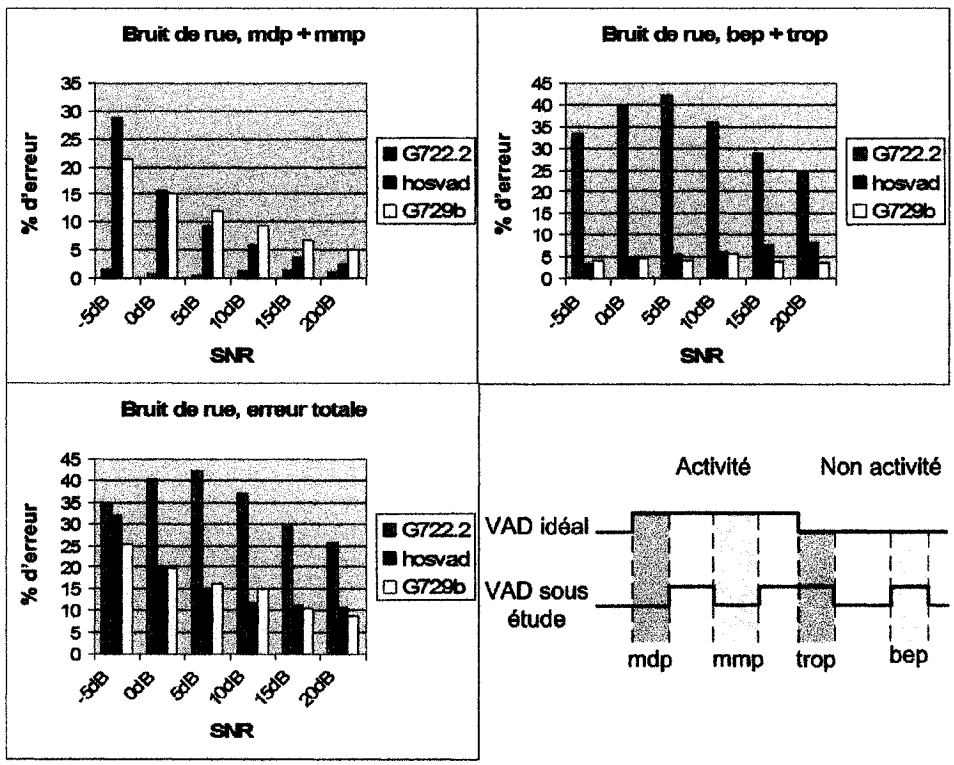


Figure 28: statistiques pour bruit de rue

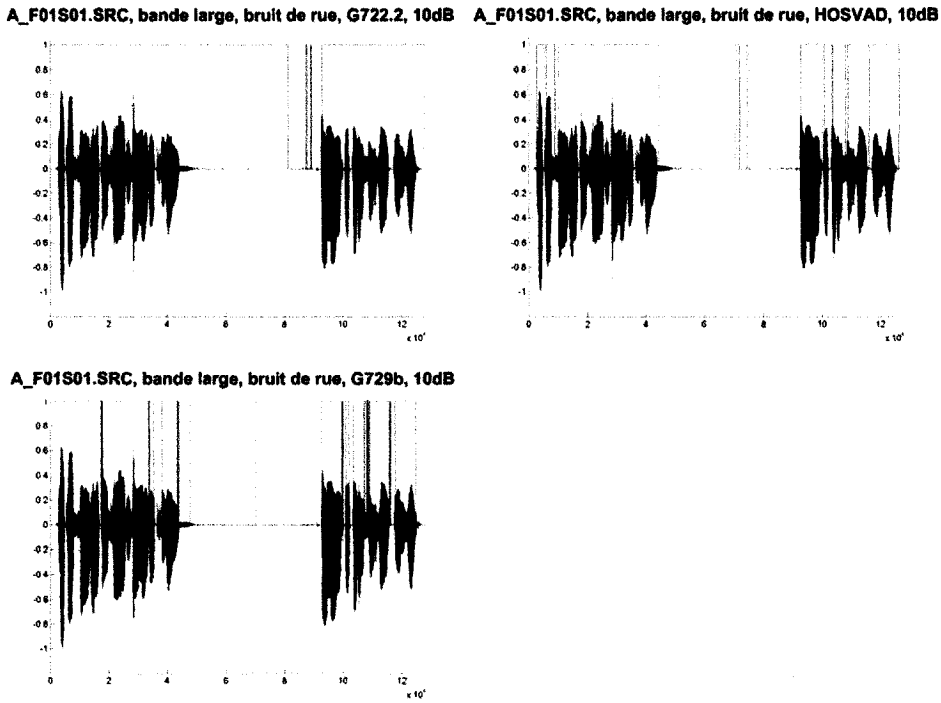


Figure 29: résultat typique pour bruit de rue, SNR = 10 dB

Pour le bruit de rue le HOSVAD a des taux de mutilation en début et en milieu de parole plus avantageux que les deux autres VADs ainsi qu'une erreur totale légèrement inférieure à celle du G729b et nettement inférieure à celle du G722.2.

5.6.8 *Bruit de voiture*

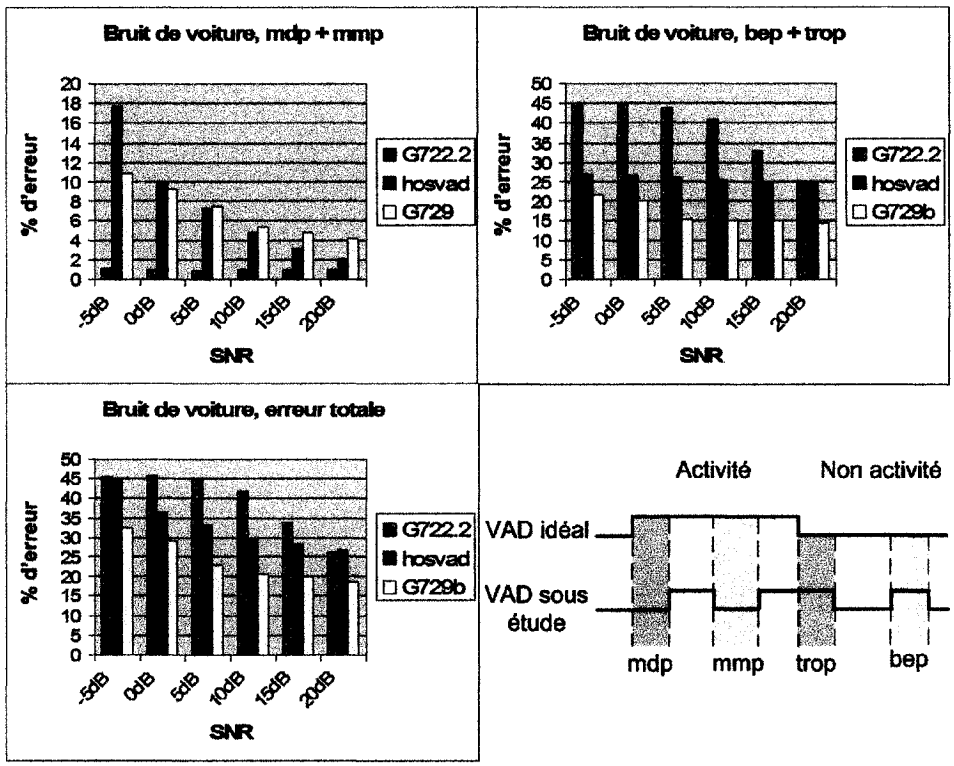


Figure 30: statistiques pour bruit de voiture

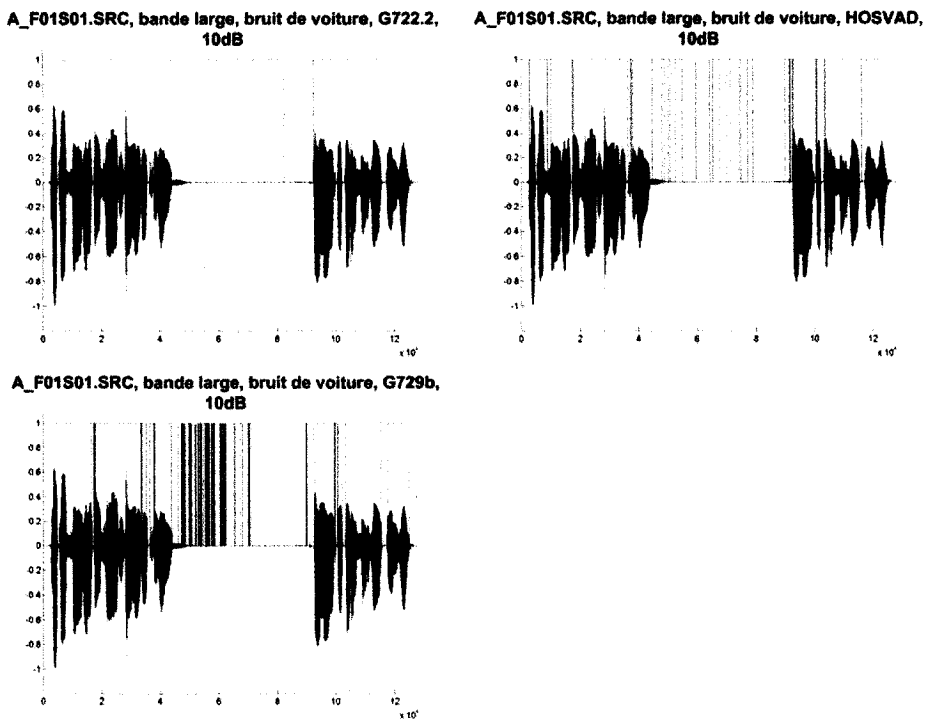


Figure 31: résultat typique pour bruit de voiture, SNR = 10 dB

Le G722.2 a de la difficulté à classifier les trames de bruit de voiture. Le HOSVAD a plus de succès, mais le G729b a une erreur totale inférieure aux deux autres algorithmes.

5.7 Synthèse des résultats

La section 5.6 démontre que tous les VADs ne sont pas égaux face à différents types de bruits. Le tableau suivant offre une synthèse de tous les résultats, en identifiant le VAD le plus performant face à chaque type de bruit étudié.

Type de bruit	**Meilleur VAD**	**Notes**
Fichier audio propre	G729b	Le G729b a la plus faible erreur totale et un taux de mutilation en début et au milieu de parole inférieur au HOSVAD et équivalent au G722.2
Réverbération et signal propre	G722.2	Le VAD du G722.2 a un taux de mutilation en début et en milieu de parole inférieur au G729b et une erreur totale inférieure au HOSVAD et presque aussi faible que le G729b.
Réverbération et bruit blanc	HOSVAD	Bien que le G729b ait une erreur totale inférieure mais comparable au HOSVAD, ce dernier a des taux de mutilation en début et en milieu de parole presque dix fois inférieurs à ceux du G729b.

Bruit blanc	HOSVAD	Le HOSVAD a l'erreur totale la plus faible et un taux de mutilation en début et en milieu de parole inférieur au G729b. Le G722.2 a le taux de mdp et mmp le plus faible, mais c'est parce qu'il classifie presque tout en parole, le rendant presque inutile.
Bruit coloré	HOSVAD	Le HOSVAD a une performance bien supérieure aux deux autres qui semblent classifier presque toutes les trames en parole. Le HOSVAD semble être le seul à bien pouvoir différencier entre parole et bruit coloré.
Bruit de murmure	HOSVAD	Le G722.2 peine à différencier entre parole et bruit. Le G729b y arrive bien, en ayant le plus bas taux d'erreur totale, mais souffre d'un haut taux de mutilation en début et milieu de parole. Le HOSVAD a un taux de mutilation en début et en milieu de parole inférieur au G729b et une erreur totale presque aussi basse que ce dernier.

Bruit de rue	HOSVAD	Le HOSVAD a le plus bas taux de mutilation en début et en milieu de parole, ainsi que la plus basse erreur totale.
Bruit de voiture	G729b	Le VAD du G729b a la plus basse erreur totale et un taux de mutilation équivalent ou inférieur au HOSVAD lorsque le SNR est égal ou inférieur à 10 dB. Le G722.2, quant à lui, parvient mal à identifier les trames de bruit.

Tableau 6: synthèse de la performance des trois VADs

Note :

Dans la plupart des tests, le G729b avait une meilleure performance que le G722.2, pourtant utilisé commercialement avec des signaux à large bande. Ceci peut être dû au fait que dans le cadre des tests effectués dans cette thèse, le G729b utilisait des trames de 5 ms alors que le G722.2 utilisait des trames de 20 ms. Ceci offre un avantage au G729b en ce qui a trait à la mutilation en début et en fin de parole.

5.8 Conclusion

En tout, 528 fichiers audio ont été soumis à chaque VAD afin d'en déterminer la performance moyenne sous certains bruits et ce, à différents SNR. Ce chapitre à présenté une description de ces tests ainsi que les résultats. Une synthèse décrivant le VAD le plus performant pour chaque type de bruit étudié a été aussi présentée.

Chapitre six

6 Conclusion

6.1 Récapitulation

Dans le premier chapitre, cette thèse explique l'utilité des algorithmes de détection d'activité vocale (VAD). Les approches les plus couramment utilisées pour la conception d'algorithmes VAD ont été présentées pour bande étroite et pour bande large.

Le second chapitre est une discussion de notions fondamentales de statistiques d'ordre supérieur. Ces notions sont nécessaires à la compréhension du HOSVAD, l'un des algorithmes sous études dans cette thèse.

Le troisième chapitre présente d'une façon détaillée le fonctionnement du HOSVAD, du prétraitement du signal jusqu'au processus de décision. Une mesure de la complexité de calcul est aussi donnée.

Le quatrième chapitre présente un survol des deux algorithmes VAD commerciaux sous étude. Une mesure de la complexité de calcul est donnée pour le G729b ainsi que pour le G722.2.

Le cinquième chapitre présente les fichiers audio propres utilisés ainsi que les bruits ajoutés à ces fichiers. Les mesures utilisées sont ensuite discutées, ainsi que les résultats pour chaque type de bruit sous forme de tableaux. Une synthèse des résultats est présentée en fin de chapitre.

6.2 Sommaire

Dans la majorité des types de bruit, le HOSVAD a une meilleure performance que le VAD du G722.2 et le VAD du G729b, pour les signaux à bande large. Il semble qu'une approche comprenant des statistiques des troisième et quatrième

ordres procure de nouvelles mesures, telles γ_3 , γ_4 et la probabilité de bruit calculée à partir des degrés d'aplatissement et d'asymétrie, soit efficace dans la distinction de la parole versus le bruit, surtout lorsque celui ci est Gaussien. Face à une application où les ressources informatiques sont limitées, le G729b est un bon deuxième choix.

6.3 Travaux subséquents

L'efficacité des mesures basées sur des statistiques des troisième et quatrième ordres étant démontrée, il serait intéressant de voir si on peut améliorer les décisions du HOSVAD en utilisant les même mesures avec d'autres méthodes de classification. Par exemple, un réseau de neurones pourrait être utilisé ou une autre méthode de soft computing.

BIBLIOGRAPHIE

Articles et livres

[BER98] **A robust voice activity detector for wireless communications using soft computing,** F. Beritelli, S. Casale, A. Cavallaero, *IEEE Journal on Selected Areas in Communications*, Vol. 16, Issue 9, Dec. 1998, pages 1818-1829

[BER99] **A Multi-Channel Speech/Silence Detector based on Time Delay Estimation and Fuzzy Classification**, F. Beritelli, S. Casale, A. Cavallaro, *IEEE Proceedings on Acoustics, Speech, and signal Processing (ICASSP'99)*, vol. 1, mars 1999, pages 93-96

[BER01] **Performance evaluation and comparison of ITU-T/ETSI voice activity detectors**, F. Beritelli, S. Casale, G. Ruggeri, *IEEE Proceedings on Acoustics, Speech, and signal Processing (ICASSP'01)*, Vol. 3, mai 2001, pages 1425-1428.

[BER02] **Performance evaluation and comparison of G.729/AMR/fuzzy voice activity detectors**, F. Beritelli, S. Casale, G. Ruggeri, S. Serrano, *IEEE Signal Processing Letters*, Vol. 9, Issue 3, March 2002, pages 85-88.

[CHE05] **Robust Voice Activity Detection Algorithm Based On the Perceptual Wavelet Packet Transform**, S. Chen, H. Wu, C. Chen, J. Ruan, and T.K. Truong, *IEEE Proceedings on Intelligent Signal Processing and Communications Systems*, décembre 13-16 2005, pages 45-48

[CHO01] **Improved voice activity detection based on a smoothed statistical likelihood ratio,** Y.D. Cho, K. Al-Naimi, A. Kondoz, *IEEE Proceedings on Acoustics, Speech, and signal Processing (ICASSP'01),* vol. 2, mai 2001, pages 737-740

[ERD00] **Voice Activity Detection Over Multiresolution Subspaces,** N. Erdol et R. Schultz, *IEEE Sensor Array and Multichannel Signal Processing Workshop,* 16 et 17 mars 2000, pages 217-220

[GAZ03] **A soft voice activity detector based on a Laplacian-Gaussian model,** S. Gazor et wei Zhang, *IEEE Transactions on Speech and Audio Processing,* vol. 11, no. 5, septembre 2003, pages 498-505

[GOK00] **Voice Activity Detection in Nonstationary Noise,** S. Gökhun Tanyer et H. Özer, *IEEE Transactions on Speech and Audio Processing,* vol. 8, no. 4, juillet 2000

[HOG97] **Probability and Statistical Inference, fifth edition,** R. V. Hogg and E. A. Tanis, Prentice Hall, New Jersey, 1997

[MEN91] **Tutorial on Higher-Order Statistics (Spectra) in Signal Processing and System theory: Theoretical Results and Some Applications,** J. M. Mendel, *Proceedings of the IEEE,* Vol. 79, No. 3, mars 1991.

[NEM01] **Robust Voice Activity Detection Using Higher-Order Statistics in the LPC Residual Domain,** E. Nemer, R. Goubran et S. Mahmoud, *IEEE Transactions on Speech And Audio Processing,* VOL. 9, NO. 3, mars 2001

[NIK93] **Signal Processing with Higher-Order Spectra**, C. L. Nikias and J. M. Mendel, *IEEE Signal Processing Magazine*, vol. 10, no. 3, pages 10-37, juillet 1993

[PAL91] **Recognition of noisy speech using cumulant-based linear prediction analysis**, K. Paliwal and M. Sondhi, *IEEE Proceedings International Conference on Acoustics, Speech, Signal Processing*, 1991, pages 429-432.

[PRO96] **Digital Signal Processing, Principles, Algorithms, and Applications, 3rd edition**, J. G. Proakis et D. G. Manolakis, Prentice Hall, 1996

[SWA01] **Higher Order Spectral Analysis Toolbox User's Guide, Version 2**, A. Swami, J. M. Mendel, C. L. Nikias, United Signals and Systems, Inc., 2001

[QI93] **Voiced-Unvoiced-Silence classification of Speech Using Hybrid Features and a Network Classifier**, Y. Qi et B. R. Hunt, *IEEE Transactions on Speech and Audio Processing*, vol. 1, no. 2, avril 1993

Standards

[G729b] **UIT-T Recommandation G.729, annexe B**, *Union internationale de télécommunication, secteur des normes de télécommunication*, novembre 1996

[G722] **UIT-T Recommandation G.722.2, annexe D : Codage vocal à bande large à 16 kbit/s environ par codage adaptatif multi débit à bande large (AMR-WB)**, *Union internationale de télécommunication, secteur des normes de télécommunication*, juillet 2003

[P23] **ITU-T Recommendation Supplement 23 to Series P (coded speech database)**, *Union internationale de télécommunication*, 1998

ANNEXE A – BASE DE DONNÉES UTILISÉE

Fichiers source

Pour la parole, les fichiers source utilisés dans cette thèse proviennent de [P23]. Les fichiers originaux contiennent des segments de parole longs de huit secondes. Ils sont en format binaire sans en-tête. Chaque échantillon sonore est représenté par 16 bits signé. La fréquence d'échantillonnage est de 16 kHz.

Pour le bruit blanc et de rue, les fichiers source utilisés proviennent également de [P23]. Ces fichiers ont le même format que les fichiers de parole.

Le bruit de murmure fut enregistré par l'auteur dans un restaurant d'Ottawa, le Greek Souvlaki House, bondé de gens, le 26 février 2006 vers midi.

Le bruit de voiture fur enregistré dans la voiture de l'auteur en se rendant au dit restaurant.

Le bruit coloré a été obtenu en filtrant le bruit blanc de [P23] par un résonateur numérique, selon l'équation 4.5.27 à la page 342 de [PRO96] reproduite ici :

$$H(z) = \frac{1 - z^{-2}}{1 - (2r\cos\omega_o)z^{-1} + r^2 z^{-2}}$$

Équation 42

Les paramètres $r = 0.95$ et $w_o = \pi/4$ ont été choisi arbitrairement.

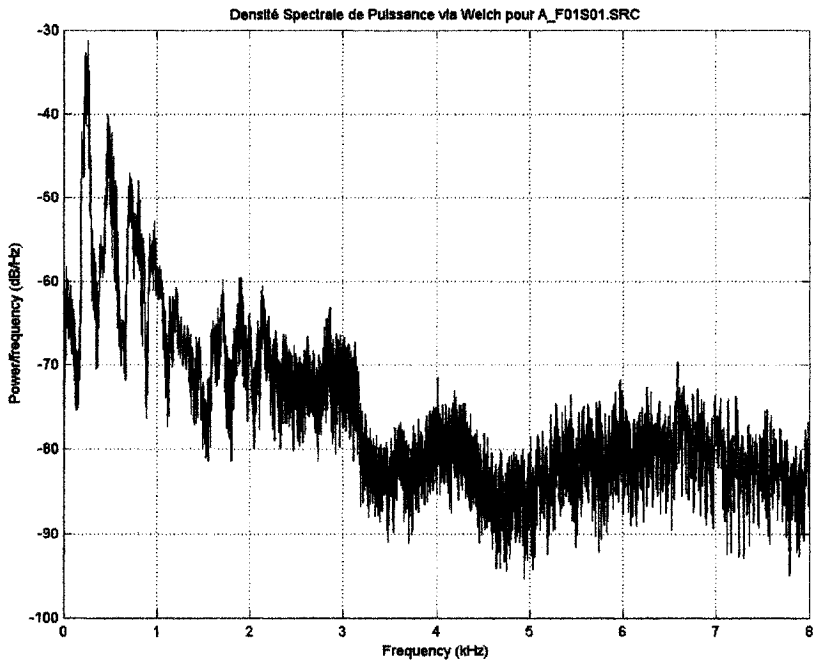
Modifications apportées aux fichiers source

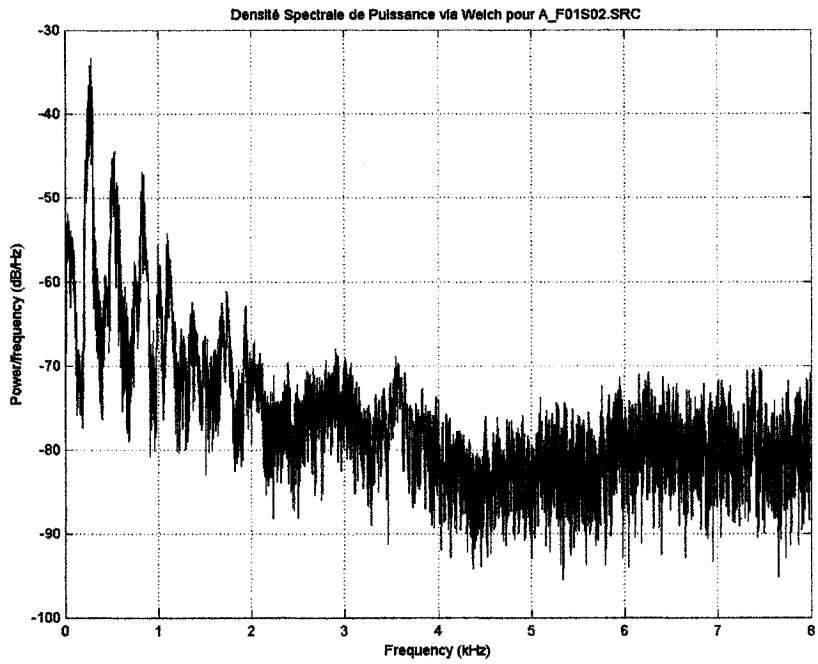
Les fichiers contenant de la parole réverbérée ont été obtenus à l'aide du logiciel SoundForge 7.0 de Sony. La réverbération ajoutée est typique d'un long corridor.

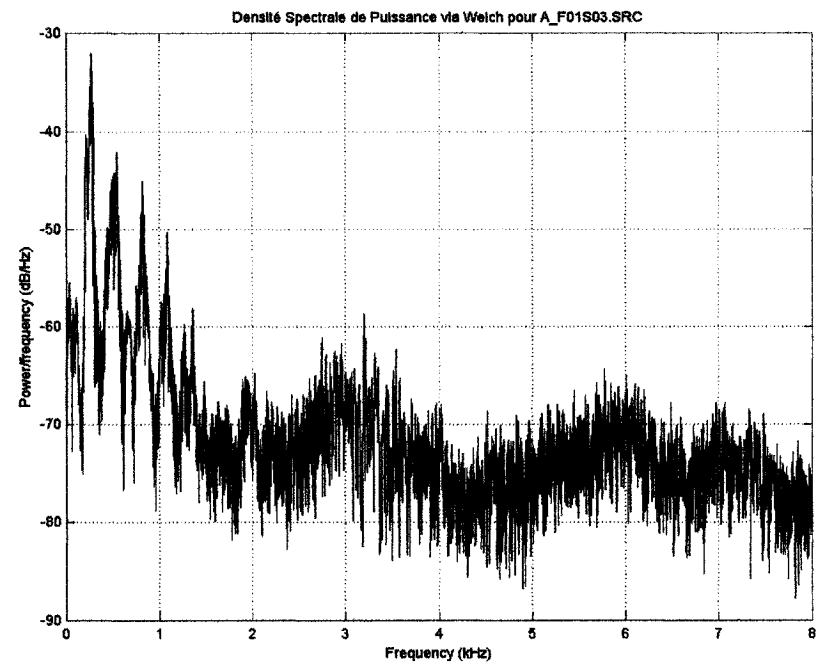
Les fichiers contenant parole et bruit on été obtenus par mixage numérique des fichiers source de parole et de bruit. Un script MATLAB a été utilisé pour ceci. La parole et le bruit on été mixés a différents ratios signal sur bruit, allant de -5 dB à 20 dB.

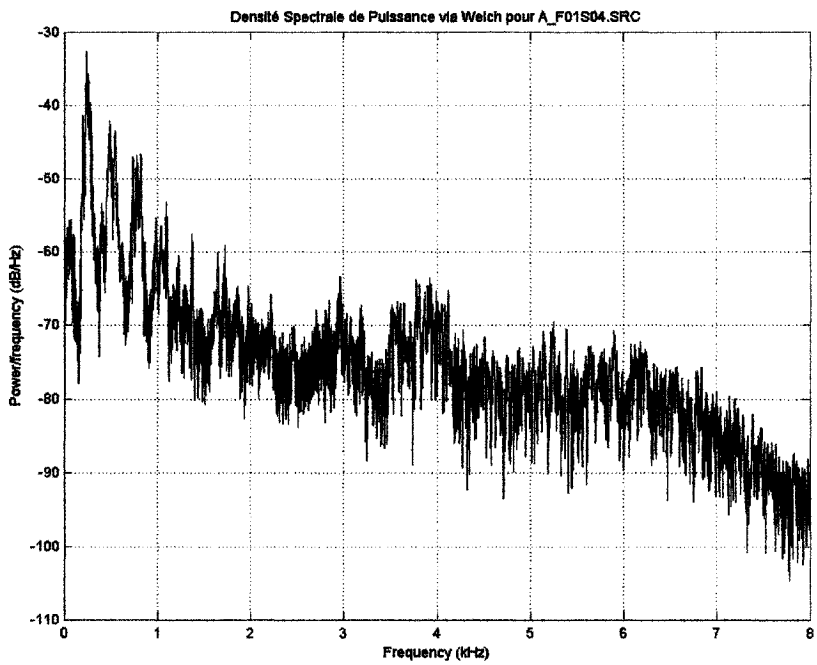
ANNEXE B – PSD DES SIGNAUX DE PAROLE ET DE BRUIT

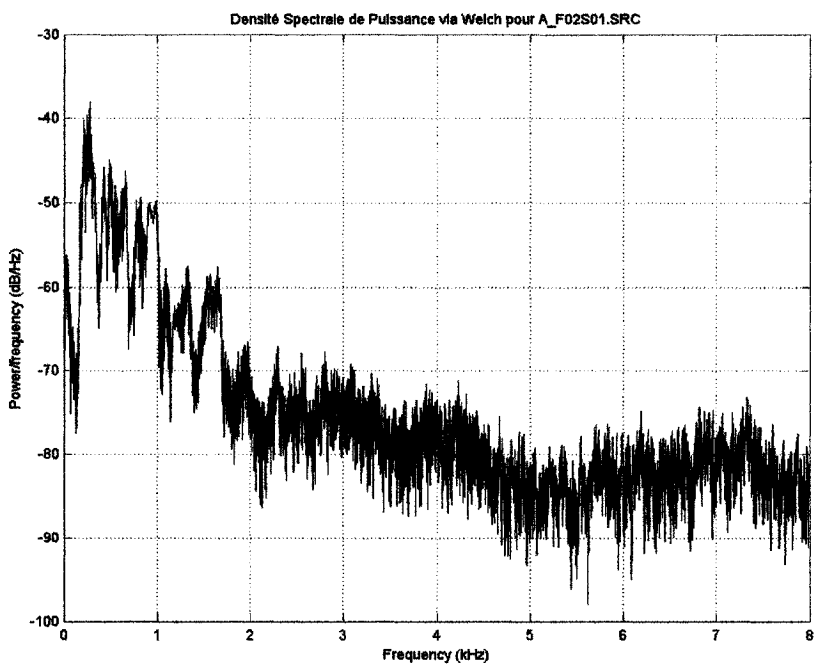
PSD des signaux de parole

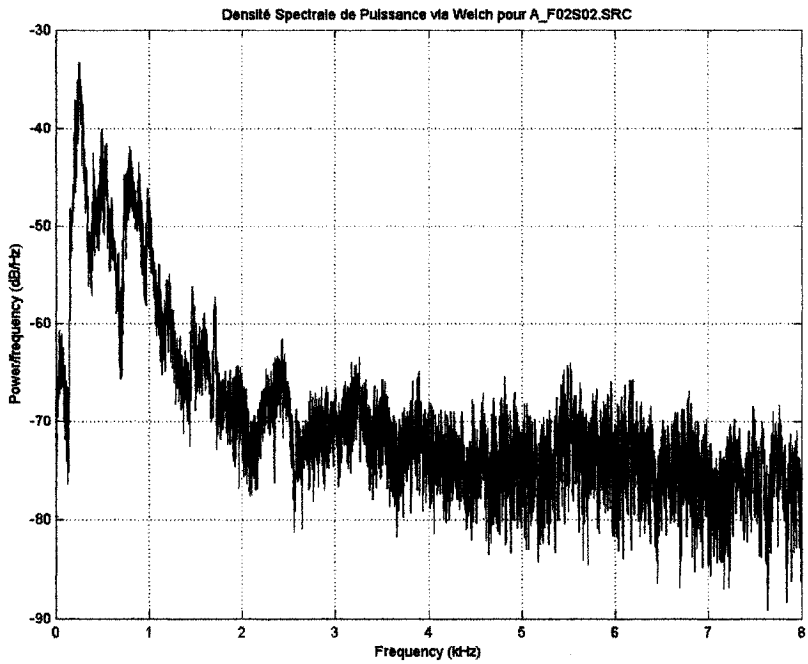


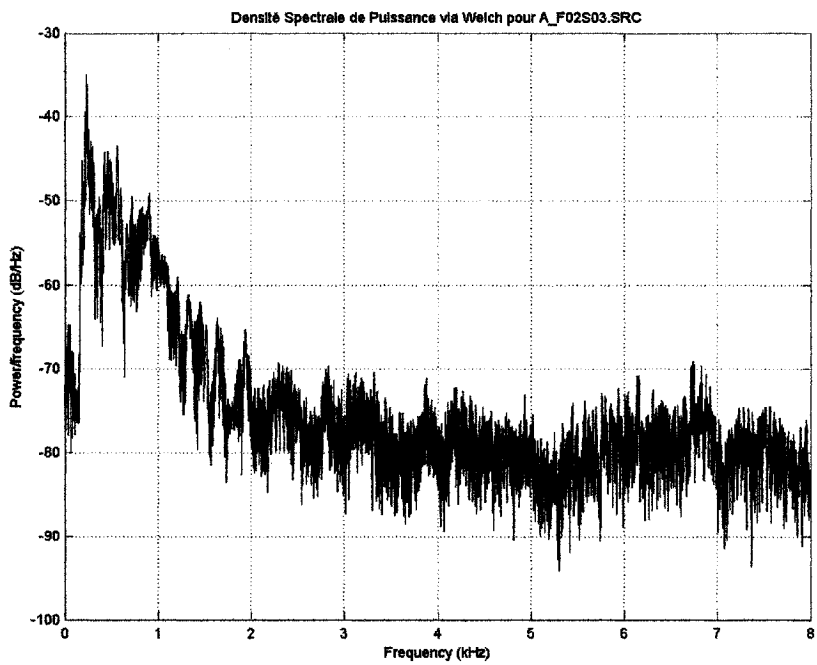

Densité Spectrale de Puissance via Welch pour A_F01S02.SRC
Power/frequency (dB/Hz)
Frequency (kHz)
-30
-40
-50
-60
-70
-80
-90
-100
0
1
2
3
4
5
6
7
8

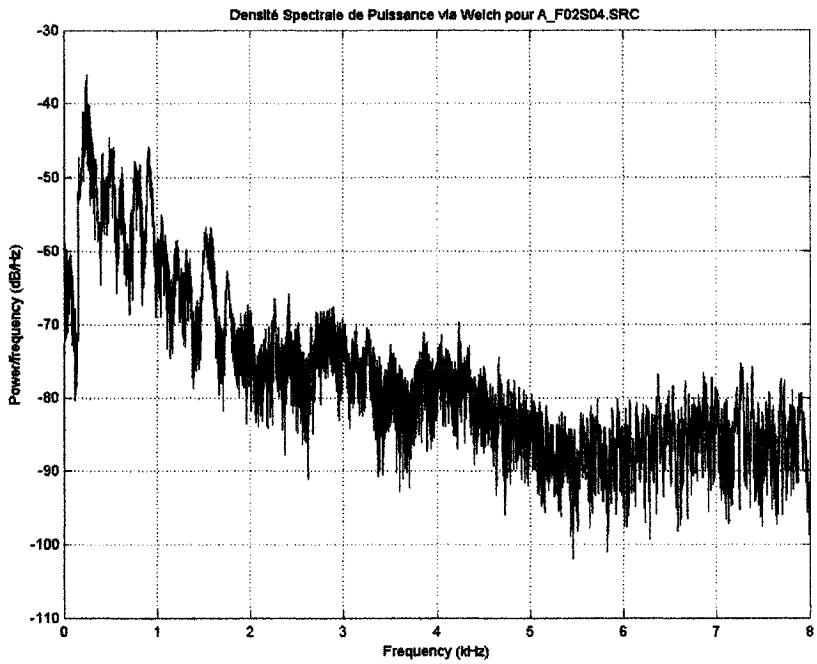

Densité Spectrale de Puissance via Welch pour A_F01S03.SRC
Power/frequency (dB/Hz)
Frequency (kHz)
-30
-40
-50
-60
-70
-80
-90
0
1
2
3
4
5
6
7
8

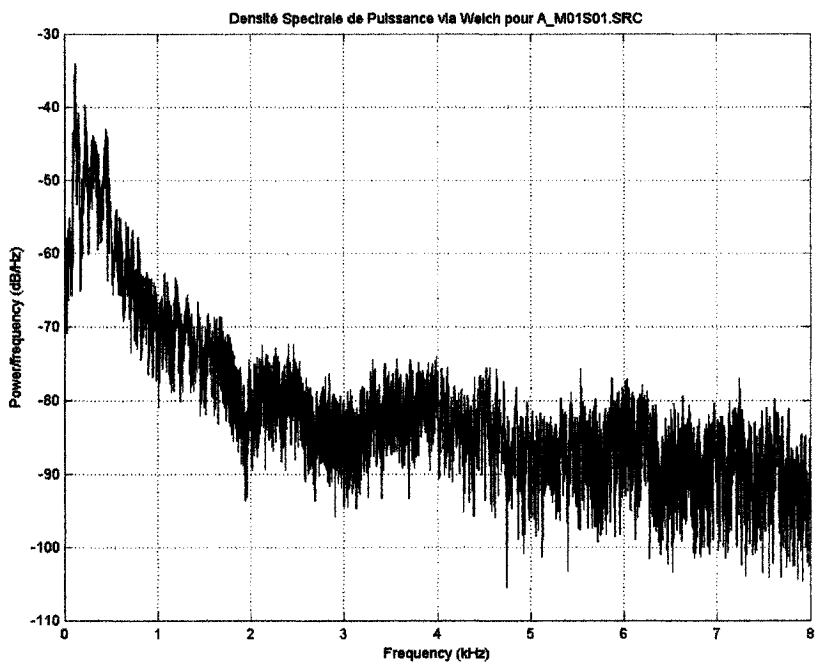

Densité Spectrale de Puissance via Welch pour A_F01S04.SRC
Power/frequency (dB/Hz)
Frequency (kHz)
-30
-40
-50
-60
-70
-80
-90
-100
-110
0
1
2
3
4
5
6
7
8

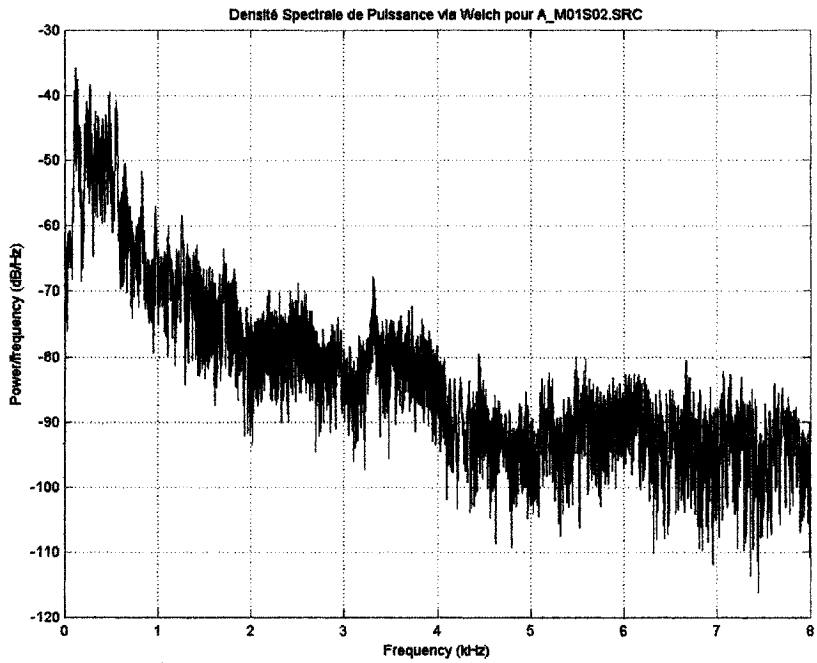

Densité Spectrale de Puissance via Welch pour A_F02S01.SRC
Power/frequency (dB/Hz)
Frequency (kHz)
-30
-40
-50
-60
-70
-80
-90
-100
0
1
2
3
4
5
6
7
8

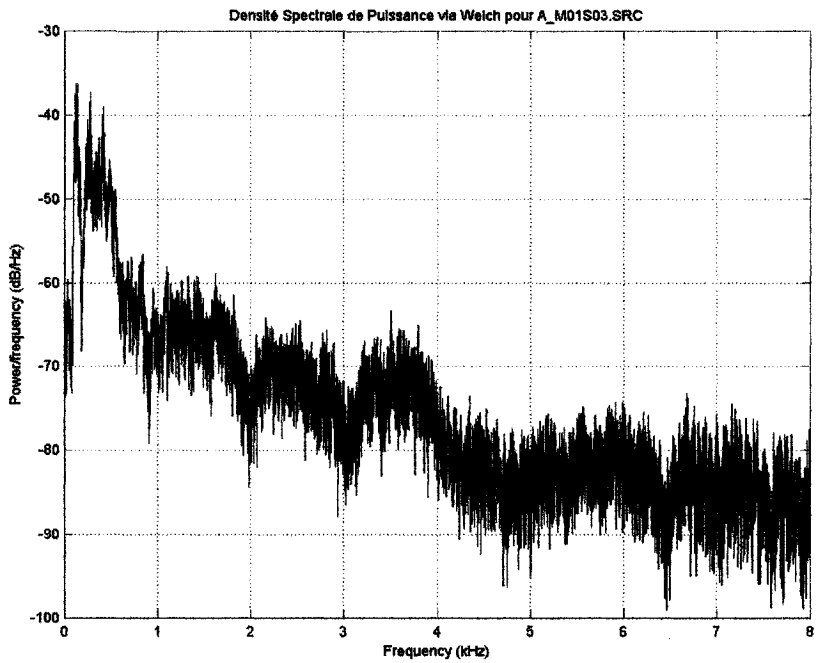

Densité Spectrale de Puissance via Welch pour A_F02S02.SRC
Power/frequency (dB/Hz)
Frequency (kHz)
-30
-40
-50
-60
-70
-80
-90
0
1
2
3
4
5
6
7
8

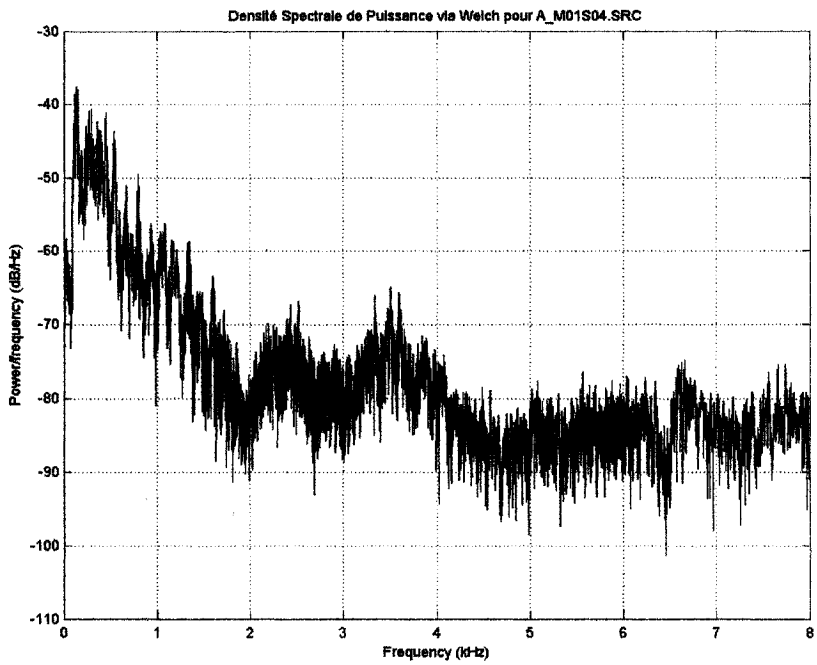

Densité Spectrale de Puissance via Welch pour A_F02S03.SRC
Power/frequency (dB/Hz)
Frequency (kHz)
-30
-40
-50
-60
-70
-80
-90
-100
0
1
2
3
4
5
6
7
8

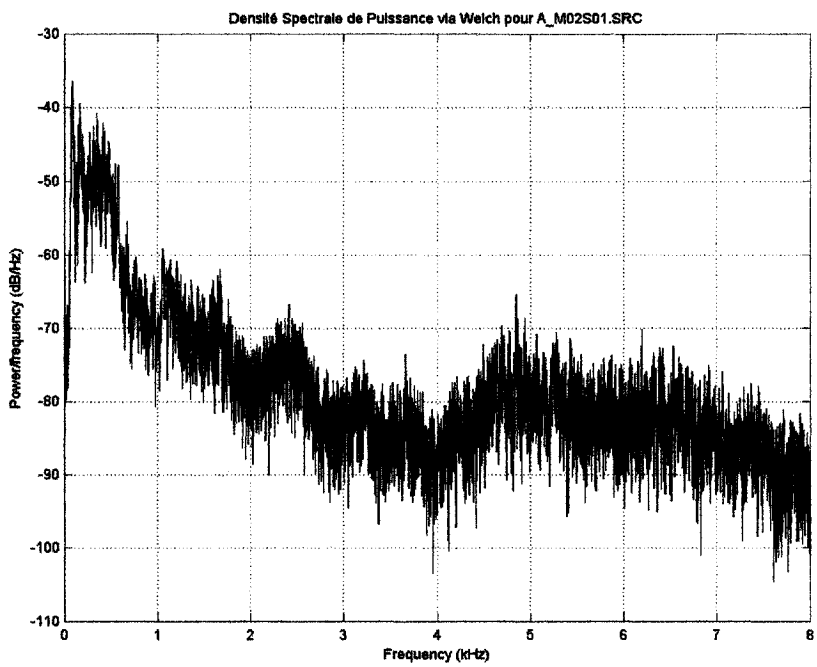

Densité Spectrale de Puissance via Welch pour A_F02S04.SRC
Power/frequency (dB/Hz)
Frequency (kHz)
-30
-40
-50
-60
-70
-80
-90
-100
-110
0
1
2
3
4
5
6
7
8

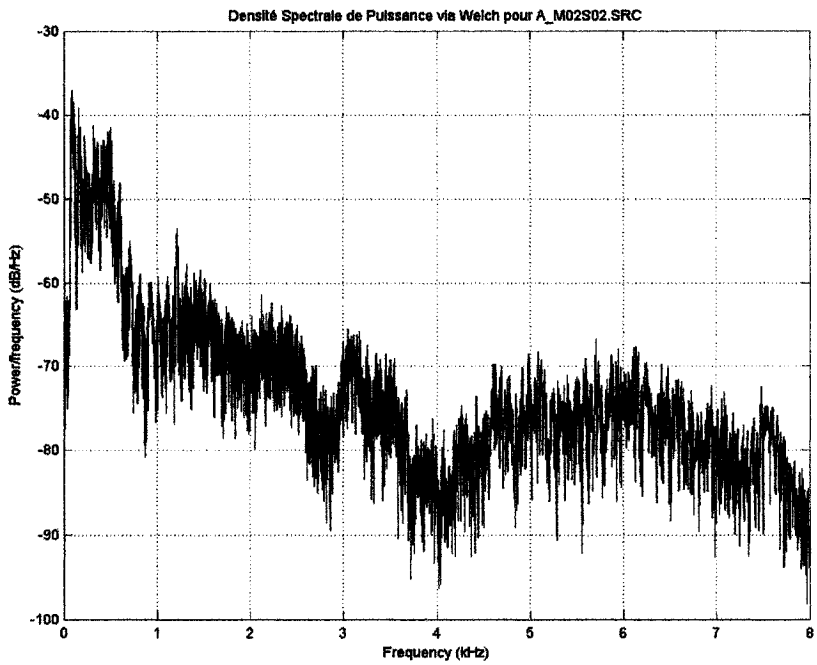

Densité Spectrale de Puissance via Welch pour A_M01S01.SRC
Power/frequency (dB/Hz)
Frequency (kHz)
-30
-40
-50
-60
-70
-80
-90
-100
-110
0
1
2
3
4
5
6
7
8

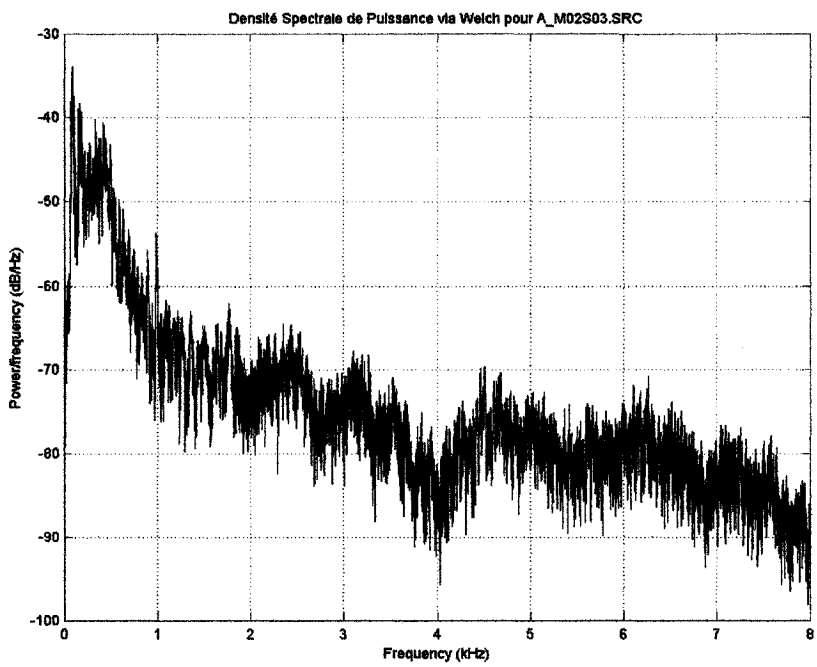

Densité Spectrale de Puissance via Welch pour A_M01S02.SRC
Power/frequency (dB/Hz)
Frequency (kHz)
-30
-40
-50
-60
-70
-80
-90
-100
-110
-120
0
1
2
3
4
5
6
7
8

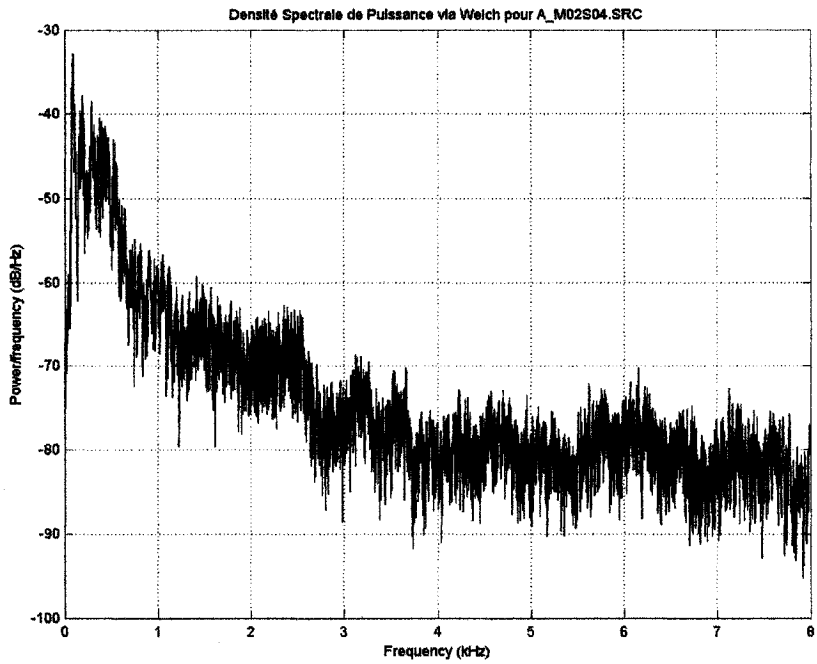

Densité Spectrale de Puissance via Welch pour A_M01S03.SRC
Power/frequency (dB/Hz)
Frequency (kHz)
-30
-40
-50
-60
-70
-80
-90
-100
0
1
2
3
4
5
6
7
8


Densité Spectrale de Puissance via Welch pour A_M01S04.SRC
Power/frequency (dB/Hz)
Frequency (kHz)
-30
-40
-50
-60
-70
-80
-90
-100
-110
0
1
2
3
4
5
6
7
8

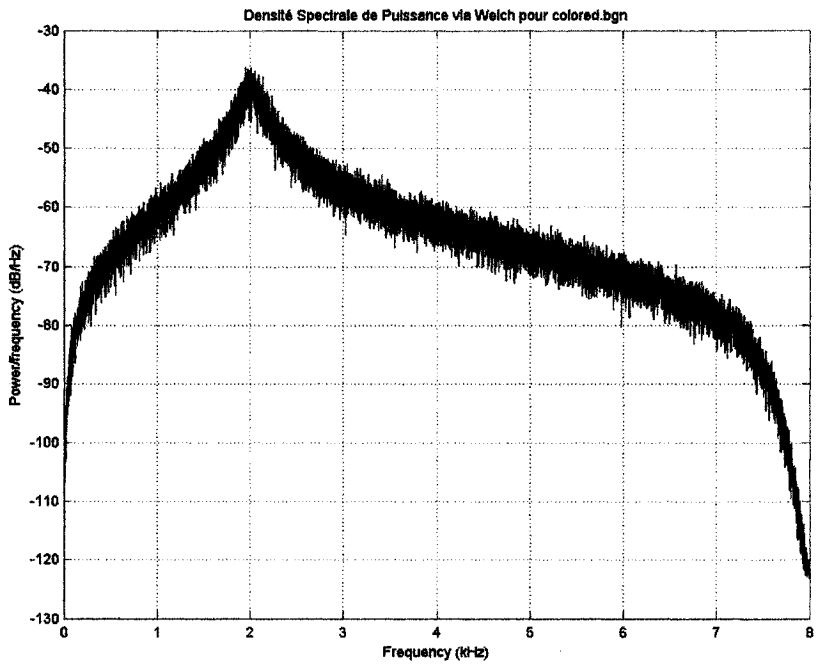

Densité Spectrale de Puissance via Welch pour A_M02S01.SRC
Power/frequency (dB/Hz)
Frequency (kHz)
-30
-40
-50
-60
-70
-80
-90
-100
-110
0
1
2
3
4
5
6
7
8

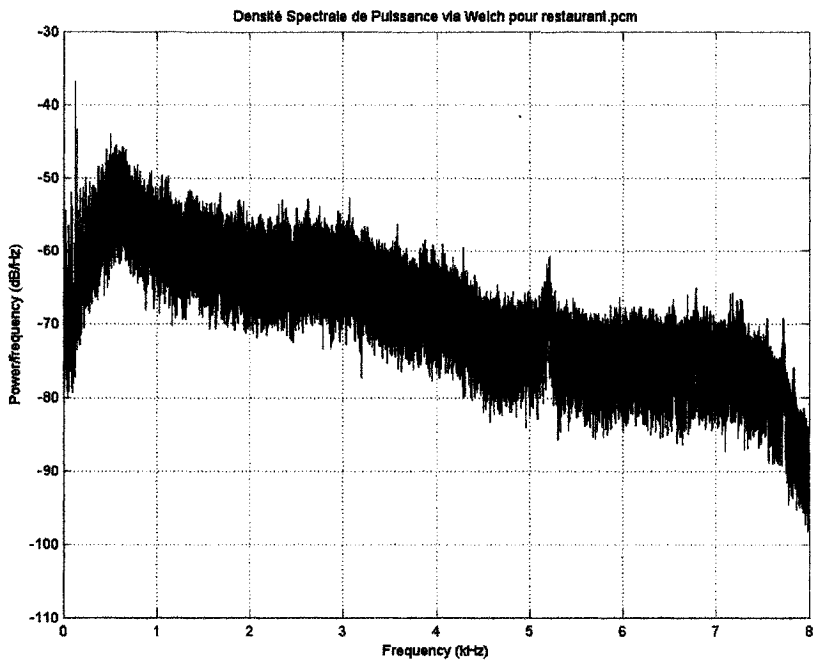

Densité Spectrale de Puissance via Welch pour A_M02S02.SRC
Power/frequency (dB/Hz)
Frequency (kHz)
-30
-40
-50
-60
-70
-80
-90
-100
0
1
2
3
4
5
6
7
8

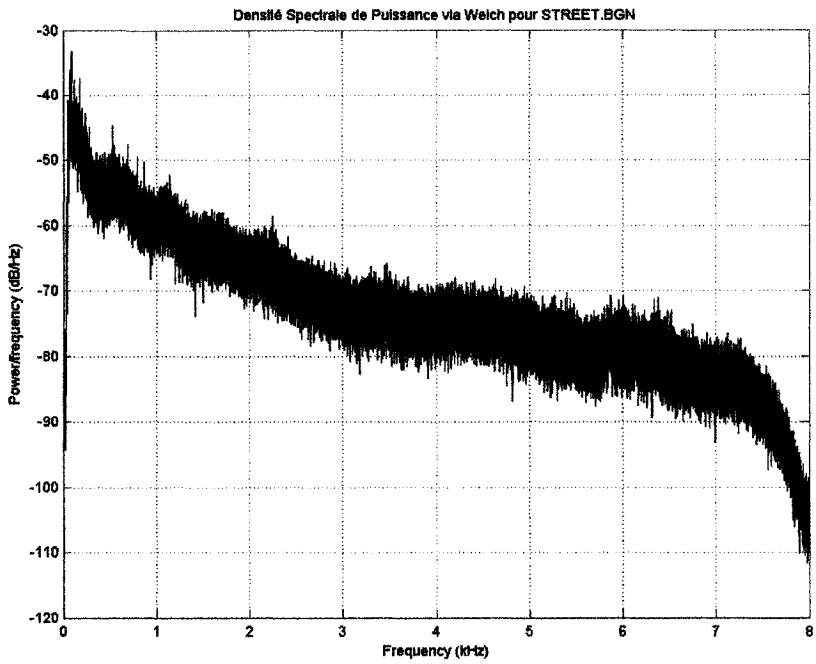

Densité Spectrale de Puissance via Welch pour A_M02S03.SRC
Power/frequency (dB/Hz)
Frequency (kHz)
-30
-40
-50
-60
-70
-80
-90
-100
0
1
2
3
4
5
6
7
8

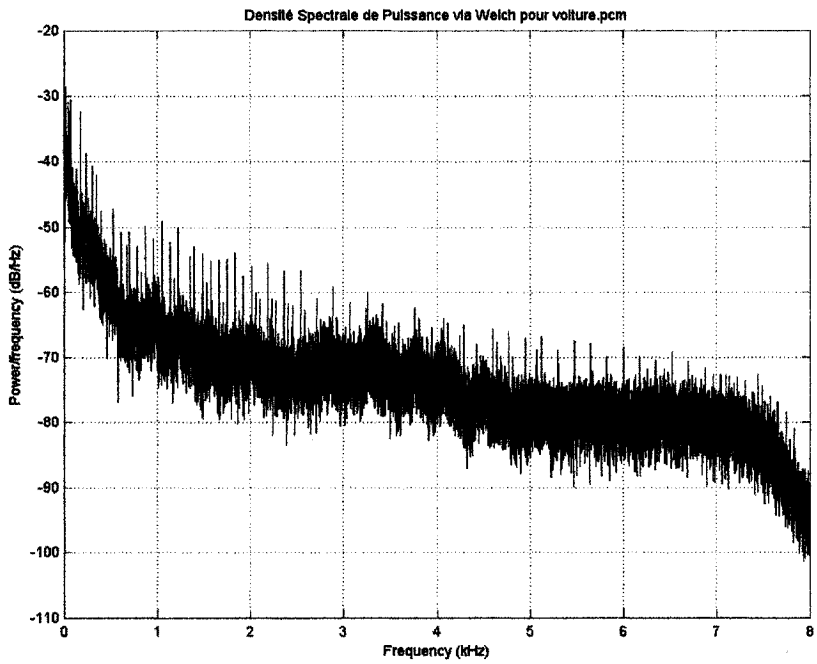

Densité Spectrale de Puissance via Welch pour A_M02S04.SRC
Power/frequency (dB/Hz)
-30
-40
-50
-60
-70
-80
-90
-100
0
1
2
3
4
5
6
7
8
Frequency (kHz)

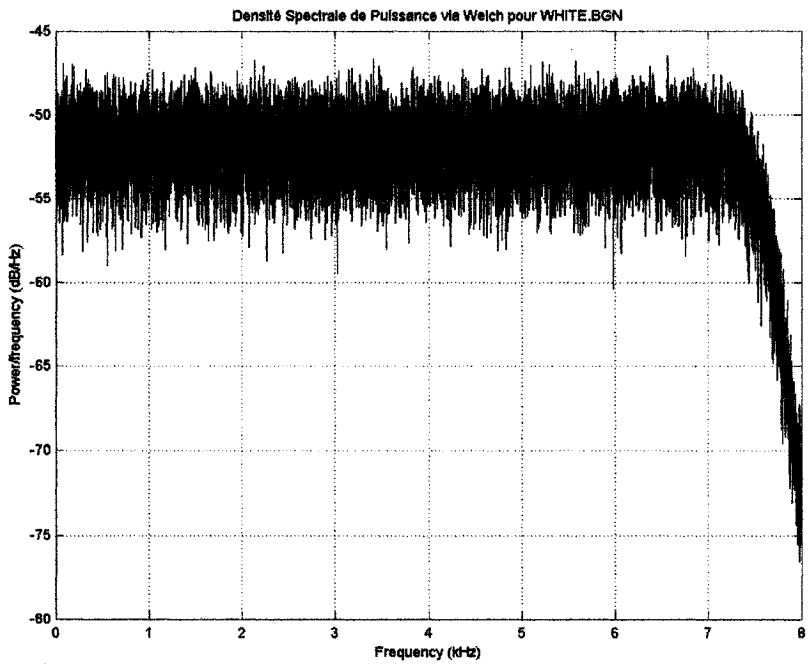
PSD des signaux de bruit






Densité Spectrale de Puissance via Welch pour STREET.BGN
Power/frequency (dB/Hz)
Frequency (kHz)
-30
-40
-50
-60
-70
-80
-90
-100
-110
-120
0
1
2
3
4
5
6
7
8


Densité Spectrale de Puissance via Welch pour voiture.pcm
Power/frequency (dB/Hz)
Frequency (kHz)
-20
-30
-40
-50
-60
-70
-80
-90
-100
-110
0
1
2
3
4
5
6
7
8


Densité Spectrale de Puissance via Welch pour WHITE.BGN
Power/frequency (dB/Hz)
-45
-50
-55
-60
-65
-70
-75
-80
0
1
2
3
4
5
6
7
8
Frequency (kHz)

ANNEXE C – FONCTION GETCLOCKCYCLES

```
/*
  Fonction: getClockCycles.c
  Auteur  : Simon Robidas,
            http://www.site.uottawa.ca/~srobidas
  But     : Cette fonction retourne le nombre de cycles
            d'horloge.
  Note    : La fonction GetCC a été gracieusement fournie par
            Mike Bélanger de PIKA Technologies.
            Pour compiler cette fonction il suffit d'invoquer
               > mex getClockCycles.c
            à partir de la ligne de commande MATLAB.

  Révision: 1.0
  Date    : 6 mai 2006
 */


#include "mex.h"
#include "matrix.h"

#include "windows.h"
#include "stdio.h"

#define NB_RANGEE 1
#define NB_COL    1

_inline void GetCC(__int64 *x)
{

 __int64 cycles;

 _asm {
       RDTSC
       mov DWORD PTR cycles, eax
       mov DWORD PTR cycles+4, edx
 }
 *x = cycles;
}

//* Interface pour communiquer avec Matlab
void mexFunction (int nlhs,       mxArray *plhs[],
    int nrhs, const mxArray *prhs[])
{
    unsigned double *ptrClock;
    const int dims[] = {1,1};

    if (nrhs > 0)
```

```
      mexErrMsgTxt ("Aucune entrée nécessaire");
    else if (nlhs > 1)
      mexErrMsgTxt ("Un seul argument de sortie est
fourni.\n");

    // Cette fonction retourne un élément de type uint64.
    plhs[0] = mxCreateNumericArray (1,dims,
mxUINT64_CLASS,mxREAL);
    ptrClock = mxGetPr (plhs[0]);

    GetCC (ptrClock);
```