ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À
L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

COMME EXIGENCE PARTIELLE
À L'OBTENTION DE LA
MAITRISE EN GÉNIE ÉLECTRIQUE
M.Ing.

PAR
ALEXANDRU CRACIUN

IMPLÉMENTATION D'UNE MÉTHODE ROBUSTE DE DÉTECTION D'ACTIVITÉ VOCALE SUR LE PROCESSEUR DE SIGNAL TMS320C6711

MONTRÉAL, LE 18 NOVEMBRE 2004

CE MÉMOIRE A ÉTÉ ÉVALUÉ
PAR UN JURY COMPOSÉ DE :

M. Marcel Gabréa, directeur de mémoire
Département de génie électrique à l'École de technologie supérieure

M. Christian Gargour, président du jury
Département de génie électrique à l'École de technologie supérieure

Mme Rita Noumeir, professeure
Département de génie électrique à l'École de technologie supérieure

IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC
LE 18 OCTOBRE 2004
À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

IMPLÉMENTATION D'UNE MÉTHODE ROBUSTE DE DÉTECTION D'ACTIVITÉ VOCALE SUR LE PROCESSEUR DE SIGNAL TMS320C6711

Alexandru Craciun

SOMMAIRE

Un algorithme de détection d'activité vocale (VAD) est un algorithme capable de discriminer entre les régions où la parole est présente et les régions où la parole est absente dans le signal vocal analysé.
Le VAD est un module important utilisé dans une large gamme d'applications dans le domaine du traitement de la parole comme la reconnaissance, la transmission ou le rehaussement de la parole.
La nature non-stationnaire ainsi que la grande variété de signaux vocaux et de bruits de fond dans les conditions où on n'a pas d'informations a priori sur la nature ou le niveau du bruit rendent ce problème difficile spécialement dans le cas d'un faible rapport signal bruit RSB.
Bien que le problème de détection d'activité vocale ait été étudié depuis plusieurs décennies, une solution optimale ne s'est pas encore imposée. De nombreux algorithmes qui utilisent une large gamme de paramètres, certains d'entre eux présentés dans cet ouvrage, ont été proposés pour répondre aux problèmes pratiques rencontrés.
Dans cet ouvrage on a implémenté sur le processeur numérique de signal TMS320C7611 un nouvel algorithme robuste de VAD qui utilise le concept d'analyse court-terme. La décision pour chaque trame de signal est fournie en temps-réel. Les distorsions spectrales, par rapport au spectre du bruit de fond, sont assimilées à des régions de parole et sont évaluées à l'aide de deux paramètres : le coefficient de corrélation spectrale et la moyenne de RSB de sous-bandes. Le filtrage médian et une approche statistique originale sont utilisés pour la détection robuste des régions de parole.
Pour évaluer les performances de l'algorithme proposé, on a utilisé un signal vocal de test complexe qui présente plusieurs régions de parole et de silence, corrompu avec plusieurs types de bruits réels dans le cas de trois RSB différentes. Les résultats de tests montrent le comportement robuste de l'algorithme proposé.

IMPLEMENTATION OF A NEW ROBUST VOICE ACTIVITY DETECTION ALGORITHM ON THE TMS320C6711 DSP

Alexandru Craciun

ABSTRACT

A voice activity detector (VAD) is an algorithm able to distinguish the speech regions from the background noise of the input signal and it is an important step in many speech-processing applications.
The problem of end point detection has been studied for several decades but despite this fact it remains an open field of research. Various types of VAD algorithms have been proposed and currently most of them use one or more parameters to met practical requirement.
In this work we propose a new VAD algorithm designed to improve the solution of word boundary detection problem for variable background noise level in a real time application. This algorithm is based on the short-time analysis and uses two parameters: the correlation coefficient between the instantaneous spectrum and an average of the background noise spectrum, and the average of the subband signal to noise ratio. The speech regions may be detected based on a median filtering and a statistical approach.
To evaluate the performance of the proposed method a clean speech dataset from the TIMIT database corrupted with different types of noise from NOISEX database for different SNR levels has been utilized
The algorithm was then implemented and tested on the TMS320C6711 floating-point DSP of Texas Instruments. The tools used to develop this application are the Code Composer Studio, which provide an integrated development environment and the DSP starter kit (DSK) with the TMS320C6711 processor on board and complete support for input and output.
The new proposed algorithm is proved to be robust and flexible. The structure of the algorithm allows adjustments to make it more efficient for some specified condition.

REMERCIEMENTS

Je tiens à remercier de façon particulière mon directeur de mémoire, M. Marcel Gabrea, pour le soutien matériel, académique et moral offert tout au long du projet.

Mes remerciements sont adressés également au collective de l'École de technologie supérieure pour avoir crée et m'avoir accepté dans cette merveilleuse école de génie.

Ce mémoire est dédié à ma cher épouse Creola.

TABLE DES MATIÈRES

LISTE DE TABLEAUX

LISTE DE FIGURES

LISTE DES ABRÉVIATIONS

AR	Auto régressif
CCS	Code Composer Studio
CS	Coefficient de corrélation spectrale
DSP	Digital Signal Processeur
IIR	Infinie Impulse Réponse
LSP	Paires de railles spectrales
MAP	Maximum a posteriori
P_d	Probabilité de détection
P_f	Probabilité de faux alarme
PG	Paramètre globale
RS	Moyenne de RSB de sous bande
RSB	Rapport signal bruit
S_{bruite}	Spectre du bruite
S_{inst}	Spectre instantanée
SLIT	Système linéaire invariant dans le temps
TF	Transformé de Fourier
TFD	Transformé de Fourier discrète
TFR	Transformé de Fourier rapide
VAD	Détection d'activité vocale

INTRODUCTION

Le traitement numérique de la parole est une des disciplines qui a profité pleinement du progrès technologique des dernières décennies. Des algorithmes autrefois utopiques à cause du volume de calcul sont maintenant utilisés dans des applications complexes telles que le codage, le rehaussement ou la reconnaissance de la parole. Plusieurs de ces applications utilisent un module de détection d'activité vocale pour augmenter les performances et réduire le coût du traitement numérique.

Un algorithme de détection d'activité vocale VAD a comme but de discriminer entre les régions où la parole est présente et les régions où la parole est absente dans le signal vocal analysé. Un algorithme de VAD fonctionne d'après une logique binaire. Il produit les valeurs logiques 1 ou 0 pour chaque segment ou trame de signal analysé, indiquant respectivement la présence ou l'absence de la parole.

Le VAD est un module important dans une large gamme d'applications concernant le traitement de la parole soit la reconnaissance, la transmission ou le rehaussement de la parole.

Dans le domaine de reconnaissance de la parole, le VAD est utilisé pour localiser le début et la fin des régions à reconnaître. La précision du VAD utilisé se matérialise dans une amélioration du taux de reconnaissance.

Pour les systèmes de transmission de la parole tels que la téléphonie cellulaire, le VAD est utilisé pour contrôler la transmission discontinue qui active la transmission uniquement pendant les périodes d'activité vocale. La transmission discontinue permet d'augmenter la capacité du système pour l'opérateur tandis que pour l'abonné prolonge l'autonomie du mobile [12].

Dans le cas du rehaussement de la parole les périodes de silence détectées par le VAD peuvent servir à actualiser le paramètre du bruit.

Une mesure objective des performances d'un algorithme de VAD est donnée par l'ensemble des paramètres : probabilité de détection P_d et probabilité de fausse alarme P_f rapportée à une décision idéale. P_d représente le ratio entre le nombre de trames contenant signal vocal correct classifié et le nombre réel de trames de parole. P_f est le ratio entre le nombre de trames de silence incorrect classifié par l'algorithme et le nombre réel de trames de silence [14]. La décision idéale de référence est obtenue par un marquage manuel des régions de silence et de parole pour le signal non bruité.
Autres aspects importants dont il faut tenir compte quand on apprécie un algorithme de VAD sont la précision, le délai introduit dans la réponse, la robustesse par rapport au bruit et le coût du traitement numérique.

Ainsi il existe des algorithmes de VAD qui sont conçus pour répondre aux exigences du travail en temps réel et utilisés spécialement dans les applications de transmission de la parole telles que la téléphonie. Un tel algorithme doit fournir la décision pour la trame courante avant qu'une nouvelle trame soit réceptionnée et donc disponible.
Dans d'autres applications telles que la reconnaissance de la parole, la condition de fonctionnement en temps réel n'est pas exigée; ce qu'on impose est plutôt une meilleure précision.

La tâche d'un algorithme de détection d'activité vocale est loin d'être facile sauf pour le cas d'un rapport entre le signal vocal et le bruit RSB très élevé, condition qui est loin d'être réalisable dans des applications réelles de traitement de la parole. La nature non stationnaire et la grande variété des bruits de fond et du signal vocal auquelles s'ajoute un RSB inconnu au concepteur et parfois variable rendent le problème de détection d'activité vocale difficile. Évidemment, ce qu'on cherche est un algorithme précis, robuste par rapport au bruit et qui demande un minimum de calculs.

Dû à son importance le problème de détection d'activité vocale préoccupe les chercheurs depuis plusieurs décennies et malgré les efforts dépensés pour le résoudre, une solution optimale ne s'est pas encore imposée. Un grand nombre d'algorithmes de VAD ont été proposés pour répondre aux conditions spécifiques rencontrées dans diverses applications. Pour chacune des situations concrètes on adopte la solution qui répond mieux en terme de précision, délai, robustesses et coût du traitement numérique.

L'objectif de ce travail est d'implémenter sur le processeur numérique de signal TMS320C7611 un nouvel algorithme robuste de VAD qui soit capable de travailler en temps réel et de performer dans les conditions où l'on n'a pas d'informations a priori sur la nature ou le niveau du bruit.

Pour atteindre cet objectif, dans un premier chapitre on a étudié le mécanisme de production et de réception, les caractéristiques physiques et la modélisation du signal vocal. Dans une deuxième étape on a étudié les principaux paramètres qui caractérisent le signal vocal numérique qui sont utilisés habituellement dans la détection d'activité vocale. On a introduit ainsi le concept d'analyse court-terme pour l'analyse spectrale du signal qui va jouer un rôle central dans le développement ultérieur de l'ouvrage.

Bien qu'une revue exhaustive de la littérature qui traite la VAD est impossible à ce niveau, le troisième chapitre présente une série d'algorithmes représentatifs afin de mettre en évidence la problématique reliée à ce sujet. Une attention particulière est portée aux plusieurs types de bruits rencontrés dans les applications réelles et leur effet sur la détection d'activité vocale.

Le quatrième chapitre décrit en détail le nouvel algorithme robuste de VAD proposé ainsi que les conditions et les résultats des tests expérimentaux. On retrouve ici l'apport de l'algorithme proposé qui repose principalement sur un nouveau paramètre robuste

utilisé dans un schème de décision originale. Les nouveautés introduites par l'algorithme de décision proposé sont le filtrage médian et la façon dont on a utilisé l'approche statistique basée sur le ratio de vraisemblance et un schème de Markov de premier ordre pour calculer le paramètre de décision.
L'ensemble des paramètres proposés et utilisés par l'algorithme ont été obtenus en observant le comportement de l'algorithme pour une base de données d'environ 5000 trames de parole et de silence.

Le dernier chapitre présente les principales caractéristiques d'un processeur dédié au traitement numérique du signal et les détails d'implémentation et de test de l'algorithme présenté au chapitre antérieur sur le processeur numérique de signal TMS320C6711.

CHAPITRE 1

LA PRODUCTION ET LA RÉCEPTION DE LA PAROLE

1.1 Préambule

Si l'on tente une définition, la parole est la capacité de l'être humain de communiquer la pensée par l'intermédiaire de sons articulés [1]. Dû à son importance, la parole a préoccupé depuis toujours les scientistes. Ainsi quelques-unes des sciences qui se préoccupent de l'étude de la parole ont déjà des centaines d'années. D'autres sont plus récentes, comme le traitement numérique de la parole, qui ne compte pas plus de quarante ans.

La production de la parole commence avec la formulation de la pensée à être communiquée. La personne qui parle, suite à des processus neurologiques et musculaires, produit les fluctuations de la pression de l'air qui constituent le signal vocal. Celui-ci se propage dans le milieu, qui d'habitude est l'air, jusqu'aux oreilles de l'écouteur où il est reçu et, après une certaine analyse, il est envoyé vers le cerveau qui l'interprète. Donc le signal vocal a une nature duale. Il peut être analysé de point de vue objectif comme étant une réalité physique ou de point de vue subjectif si on regarde la sensation psycho-acoustique produite au niveau du cerveau [2-6].

Comme on vient de voir, la production de la parole est un processus complexe, qui implique des phénomènes neurologiques, physiologiques et physiques. Dans un tel contexte l'étude de la parole est une science multidisciplinaire. Pour une meilleure compréhension, l'ingénieur qui travaille dans ce domaine devrait connaître les notions de base caractéristiques à chacune de ces disciplines qu'il rencontre dans son travail. Certaines des ces notions seront présentées dans les chapitres qui suivent.

1.2 L'appareil phonatoire humain

Techniquement parlant, la parole est une onde sonore produite par l'action volontaire et coordonnée des structures anatomiques qui forment l'appareil phonatoire humain. Ce processus est coordonné par le système nerveux central. Les sons produits sont analysés par rétroaction auditive pour assurer la qualité acoustique de la parole.

Les muscles abdominaux actionnent sur le diaphragme, en poussant l'air des poumons vers la trachée artère. Au bout supérieur de celle-ci se trouve le larynx qui module le courant d'air sous la forme d'impulsions périodiques appliquées au conduit vocal. Celui-ci est formé d'un ensemble de cavités : la cavité pharyngienne suivie de la cavité buccale et en dérivation la cavité nasale. La luette, qui prolonge le bord postérieur du voile du palais, contribue à la fermeture des fosses nasales. Comme résultat, pendant la production de la parole, la cavité nasale peut être couplée soit totalement, soit partiellement, où même découplée de la cavité buccale. Autres organes anatomiques importants qui participent à la production des sons sont : la langue et les dents dans la cavité buccale, les narines dans la cavité nasale et les lèvres [5], comme montré dans la figure 1.

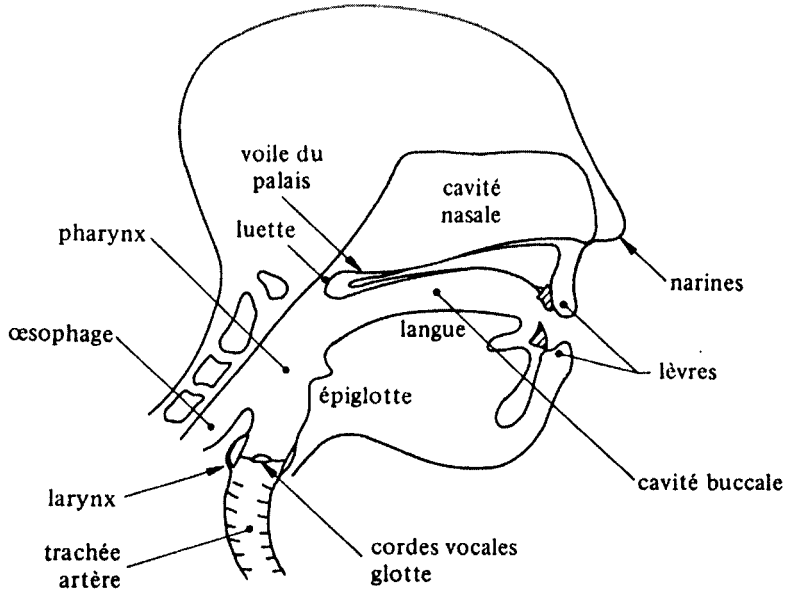


Figure 1 Appareil phonatoire [3]

Le larynx a un rôle extrêmement important dans la production de la parole. Il est formé d'un ensemble de muscles et cartilages mobiles entourant une cavité située à la partie supérieure de la trachée. Les cordes vocales, partie intégrale du larynx, peuvent le fermer ou peuvent former une ouverture variable appelée glotte. La fonction du larynx est de fournir une excitation périodique au reste du système sous la forme d'une suite d'impulsions périodiques de pression d'air pendant la phonation du son voisé. Au contraire, il laisse passer librement l'air pendant la voix chuchotée et la phonation des sons sourds ou non voisés. Ainsi le conduit vocal peut être vu comme une suite de tubes acoustiques dont la section peut varier avec le temps. Son diagramme est représenté dans la figure 2.

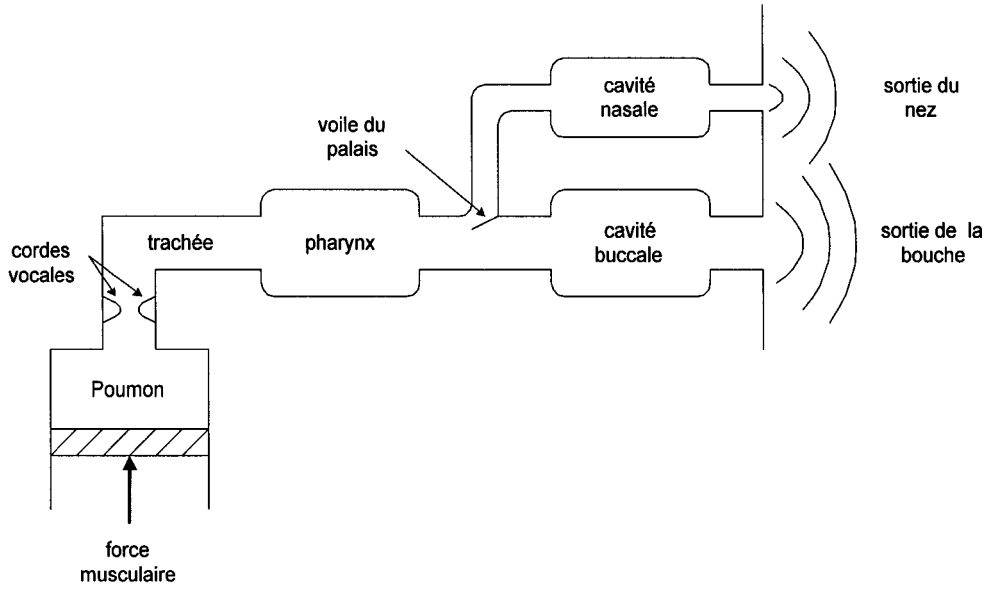


Figure 2 Modèle mécanique de production de la parole [5]

1.3 Eléments d'analyse acoustique du signal vocal

Les paramètres qui décrivent le signal vocal réel changent avec le temps car le système physique qui les produit change rapidement avec le temps. Ainsi le signal vocal peut être divisé en segments temporels de longueurs comprises entre 10 et 30 ms dont les propriétés acoustiques demeurent quasi stationnaires [5-6].

L'étude de la forme d'onde de la parole révèle des caractéristiques telles que l'intensité, le comportement périodique ou non, les limites et la durée de chaque son qui forme le signal vocal. La figure 3 représente l'évolution temporelle du signal vocal pour le mot *she* / ʃ i: /. Une des plus importantes caractéristiques de la parole qui découle de l'analyse temporelle est le fait que la parole n'est pas une succession de sons discrets et très bien définis. En réalité, le signal vocal est une suite de sons qui s'approchent plus ou moins d'un set de sons cible qui sont les sons idéaux de la parole. De plus le passage d'un son à l'autre, la coarticulation, a une durée finie. Ceci est nécessaire aux transitions

physiques de l'appareil phonatoire pour arriver à la forme qui lui permettrait de produire le son ciblé.

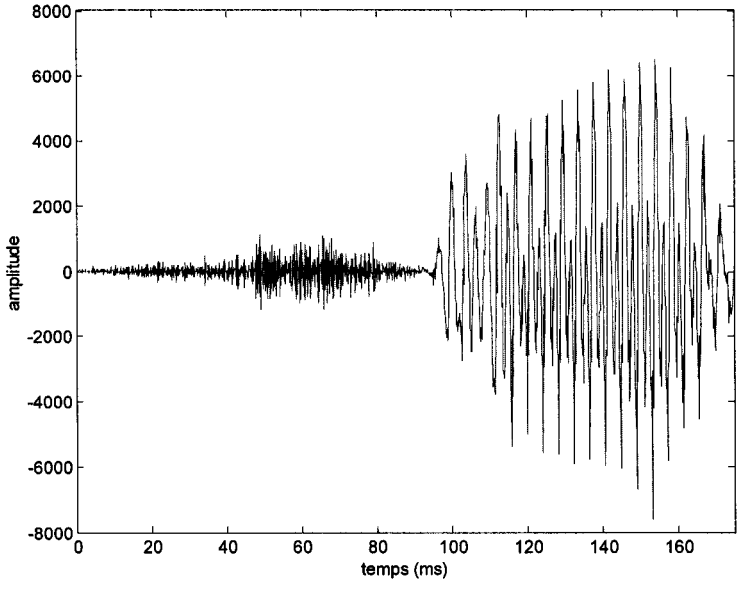


Figure 3 L'évolution temporelle du signal vocal pour le mot *she* / ʃ i: /

L'étude dans le domaine fréquentiel du signal vocal peut révéler aussi un comportement périodique ou non et éventuellement les fréquences les plus importantes qui décrivent le signal vocal. La figure 4 représente le spectre du son voisé / i / et la figure 5 le spectre du son non voisé / ʃ / dans le même mot *she*. On a utilisé une fréquence d'échantillonnage de 8000 Hz et la TFD en 256 points.

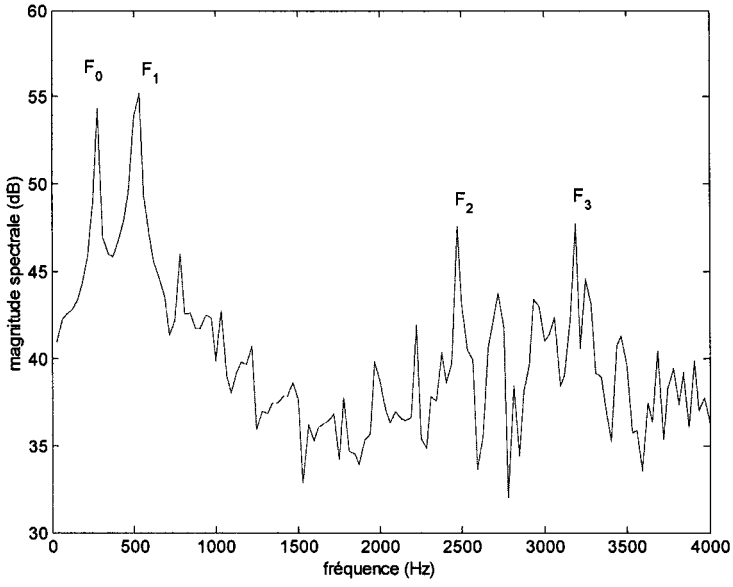


Figure 4 Le spectre du son voisé / i /

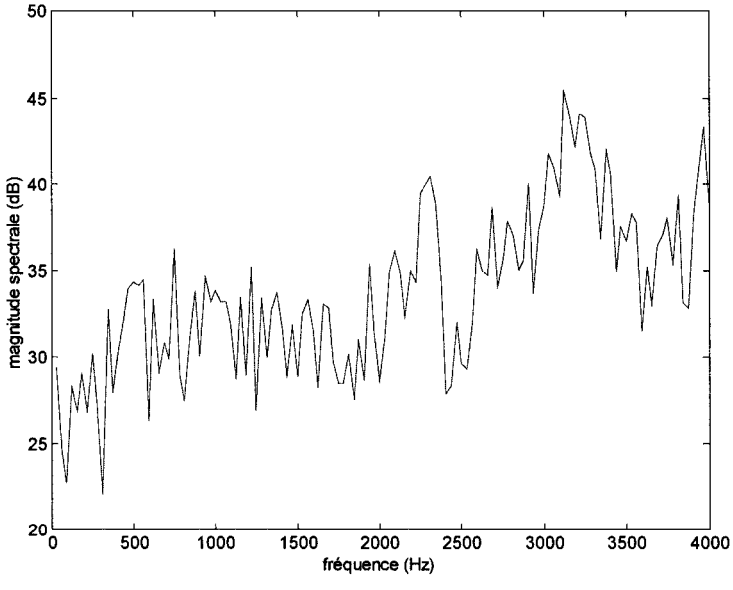


Figure 5 Le spectre du son non voisé / ∫ /

Comme on a vu, l'appareil phonatoire contient plusieurs cavités acoustiques qui sont reliées entre elles pendant la production de la parole. Cet ensemble acoustique est excité par les pulsions d'air modulé par le larynx pour produire les vibrations sonores. Dans le spectre du signal vocal, on trouve des fréquences amplifiées et des fréquences atténuées par le conduit vocal. La position de ces fréquences dans le domaine fréquentiel sont propres à chaque personne. Elles peuvent être vues comme des fréquences de résonance en rapport direct avec la forme et la dimension de chaque conduit vocal quand celui-ci produit un son voisé. En conséquence, chaque conduit vocal est caractérisé par un set de fréquences qui tentent de former tout le spectre du signal vocal appelé structure formantique. Théoriquement il existé un nombre infini de formants (F_1, F_2, F_3,...), mais dans la pratique les premiers trois ou quatre sont suffisants pour la caractérisation du signal vocal [2].

Parce que le signal vocal varie dans le temps, son spectre change aussi. Parmi les sons idéaux de la parole il y en a qui possèdent une telle propriété. Dans ces cas, une représentation tridimensionnelle du spectre en rapport avec le temps est plus suggestive et donc souvent utilisée [5].

1.4 Mécanisme de la phonation

L'une de plus importantes caractéristiques du signal vocal est la nature de l'excitation. Il existe deux types élémentaires d'excitation qui produisent les sons voisés et non voisés.

1.4.1 Phonation de sons voisés

Les sons voisés sont produits à partir d'une excitation qui actionne sur le conduit vocal et qui consiste en une suite d'impulsions périodiques d'air fournies par le larynx. Les cordes vocales au début sont fermées. Sous la pression continue de l'air qui vient des poumons elles s'ouvrent graduellement délivrant cette énergie potentielle. Pendant cette

ouverture la vitesse de l'air et l'énergie cinétique augmentent jusqu'à ce que la tension élastique des cordes vocales égale la force de séparation du courant d'air. A ce point l'ouverture de la glotte est maximale. L'énergie cinétique qui a été accumulée comme tension élastique dans les cordes vocales commence à rétrécir cette ouverture et de plus la force de Bernoulli accélère encore la fermeture abrupte de la glotte [5]. Ce processus périodique est caractérisé par une fréquence propre à chaque personne, connue sous le nom de fréquence du fondamental (F_0) où pitch et il donne la hauteur normale de la voix. La fréquence fondamentale peut varier de 80 à 200 Hz pour une voix masculine, de 150 à 450 Hz pour une voix féminine et de 200 à 600 Hz pour une voix d'enfant [2]. Cette fréquence fondamentale peut varier suite à des facteurs liés au stress, intonation et émotions. Le timbre de la voix est déterminé par les amplitudes relatives des harmoniques du fondamental.

L'intensité du son émis est liée à la pression de l'air en amont de larynx. Tous ces aspects pour un son voisé peuvent être observés dans la figure 4.

1.4.2 Phonation de sons non voisés

Les sons non voisés sont générés par le passage de l'air dans une constriction étroite situé en un point du conduit vocal. Ils sont générés sans l'apport du larynx et ne présentent pas de structure périodique [5]. Ces caractéristiques d'un son non voisé peuvent être observés dans la figure 5.

1.5 Caractéristique phonétique

Les phonèmes sont les sons d'une langue définis par les propriétés distinctives qui les opposent aux autres sons de cette langue [1]. La totalité des phonèmes d'une langue assurent toutes les nécessités d'expression de cette langue. La majorité des langues naturelles sont composées d'un ensemble relatif restreint de phonèmes. Par exemple on

considère que la langue française comprend 36 phonèmes [3]. À cause de plusieurs facteurs, par exemple accents, genre, l'effet de la coarticulation, un certain phonème peut avoir une variété de manifestations acoustiques pendant le discours. Du point de vue acoustique, un phonème représente plutôt une classe de sons qui ont la même signification [5].

Afin de faciliter et standardiser la transcription phonétique a été adopté un alphabet phonétique international. Il est assez complet pour couvrir les nécessités de transcription phonétique pour toutes les langues du monde.

1.6 Classification des phonèmes

Il y a plusieurs façons de classifier les phonèmes. Un phonème est stationnaire ou continuant si la configuration du conduit vocal ne change pas pendant la production du son. Un phonème est non continuant si pendant sa production il y a des changements dans la configuration du conduit vocal [5].

On peut grouper les 36 phonèmes de la langue française en classes et sous classes d'après le mode d'articulation de l'appareil de phonation (tableau I).

Tableau I

Phonèmes de la langue française [3]

Phonèmes								
Voyelles		Semi-consonnes	Consonnes					
Orales	Nasales		Liquides	Nasales	Fricatives		Occlusives	
					Voisées	Non-voisées	Voisées	Non-voisées
i (I)	ɛ̃ (IN)	j (Y)	l (L)	m (M)	v (V)	f (F)	b (B)	p (P)
e (E)	œ̃ (UN)	w (W)	R (R)	n (N)	z (Z)	s (S)	d (D)	t (T)
ɛ (AI)	ɑ̃ (AN)	ɥ (UI)		ɲ (GN)	ʒ (J)	ʃ (CH)	g (G)	k (K)
a (A)	ɔ̃ (ON)							
ɔ (O)								
u (OU)								
y (U)								
ø (EU)								
œ (OE)								
ə (E)								
o (AU)								

1.6.1 Les voyelles

Les voyelles sont des sons voisés, continus, normalement avec la plus grande amplitude parmi tous les phonèmes et elles peuvent varier beaucoup en durée, entre 40 et 400 ms. Les voyelles orales sont produites sans l'intervention de la cavité nasale pendant que pour les voyelles nasales, le conduit nasal est couplé à la cavité buccale et la production de son se fait par la bouche et par les narines en même temps. Les voyelles sont différentiées en trois groupes d'après la position de la courbure de la langue et le dégrée de la constriction induit dans le conduit vocal.

différentiées en trois groupes d'après la position de la courbure de la langue et le dégrée de constriction induit dans le conduit vocal.

L'analyse dans le domaine temporel et fréquentiel révèle plusieurs caractéristiques acoustiques qui aident à la classification de chaque son. L'analyse dans le domaine temporel montre que les voyelles sont de sons quasi périodiques dus à l'excitation. Les voyelles peuvent être identifiées par les locations de leurs formants dans le domaine fréquentiel. La position des deux premiers formants est suffisante pour caractériser la majorité des voyelles, le troisième formant est nécessaire juste pour quelques-uns. La position de formants de fréquence plus élevée reste presque inchangé et n'apporte pas d'information utile pour l'identification.

1.6.2 Les diphtongues

Les diphtongues impliquent un mouvement d'une voyelle initiale vers une autre voyelle finale. Donc les diphtongues sont essentiellement des sons non continus. La différence entre une diphtongue et les deux voyelles individuelles composantes est que la durée de la transition est plus grande que la duré de chaque voyelle. De plus la voyelle initiale est plus longe que la voyelle finale. Dans la parole les deux voyelles composant une diphtongue peuvent ne pas être réalisées entièrement ce qui accentue l'idée de non-stationnarité qui caractérise les diphtongues.

1.6.3 Les semi-consonnes

Les semi-consonnes sont des sons non continus et voisés qui possèdent des caractéristiques spectrales semblable aux voyelles. On peut voir les semi-consonnes comme des sons transitoires qui s'approchent, atteint et après s'éloignent d'une positionne cible. La durée des transitions est comparable à la durée passée en position cible.

1.6.4 Les consonnes

Les consonnes sont de sons pour lesquels le conduit vocal est plus étroit pendant la production, par rapport aux voyelles. Les consonnes impliquent les deux formes d'excitation pour le conduit vocal et elles peuvent être continuantes ou non.

1.6.4.1 Les consonnes fricatives

Les fricatives non voisées résultent d'une turbulence créée par le passage de l'air dans une constriction du conduit vocal qui peut se trouver près des lèvres pour les labiales, au milieu du conduit vocal pour les dentales et au fond du conduit vocal pour le palatales. Dans ce cas la constriction cause une source de bruit et aussi divise le conduit vocal en deux cavités. La première cavité actionne comme une enceinte antirésonante qui atténue les basses fréquences d'où la concentration de l'énergie vers les hautes fréquences dans le domaine spectral.
Pour les fricatives voisées l'excitation est mixte et à la source de bruit s'ajoutent les impulses périodiques crées par la vibration de cordes vocales.

1.6.4.2 Les consonnes occlusives

Les consonnes occlusives sont des sons non continuants qui sont des combinaisons de sons voisés, non voisés et de courtes périodes de silence. Une forte pression d'air s'accumule avant une occlusion totale dans un point du conduit vocal qui après est relâché brusquement. Cette période d'occlusion s'appelle la phase de tenue.
Pour les occlusives non voisés la phase de tenue est un silence et la période de friction qui suit est plus longue que pour les occlusives voisés. Pour les occlusives voisés, pendant la phase de tenue, un son de basse fréquence est émis par vibration des cordes vocales.

1.6.4.3 Les consonnes nasales

Les consonnes nasales sont des sons continus et voisés. Les vibrations produites par les cordes vocales excitent le conduit vocal que cette fois est formé de la cavité nasale ouverte et la cavité buccale fermée. Même fermée, la cavité buccale est couplée à la cavité nasale et influence la production de sons comme une enceinte antirésonante qui atténue certaines fréquences, en fonction du point où elle est fermée. Les formes d'onde des consonnes nasales ressemblent aux celles des voyelles mais sont normalement plus faibles en énergie due à la capacité réduite de la cavité nasale de radier des sons par rapport à la cavité buccale.

1.6.4.4 Les consonnes liquides

Les consonnes liquides sont des sons non continus et voisés qui possèdent des caractéristiques spectrales similaires aux voyelles. Elles sont plus faibles en énergie due au fait que le conduit vocal et plus étroit pendant leur production.

1.7 Modélisation mathématique de la production de la parole

1.7.1 La propagation du son

L'onde sonore est produite par vibrations et sa propagation se réalise par l'oscillation des particules qui composent le milieu de propagation. En conséquence certaines lois physiques sont utilisées pour décrire la production et la propagation des sons dans l'appareil phonatoire. La loi de conservation de la masse, conservation du moment et conservation de l'énergie, et les lois de la thermodynamique et de la mécanique des fluides sont appliquées à l'air qui est le milieu de propagation est qui est un fluide compressible et peu visqueux. En utilisant ces principes physiques on obtient un set

d'équations différentielles à dérivés partielles qui caractérise le mouvement de l'air dans l'appareil phonatoire [5-6]. Un modèle détaillé de la production de la parole doit tenir compte des facteurs suivants:

a. la modification temporelle de la forme de l'appareil phonatoire pendant la production de sons ;
b. les pertes dues a la friction et au transfert de chaleur entre l'air et la surface de l'appareil phonatoire ;
c. la consistance de la surface de l'appareil phonatoire ;
d. le couplage entre la cavité nasale et la cavité buccale ;
e. le type de l'excitation.

Un modèle théorique complet, incluant tous ces facteurs, n'a pas encore été développé. Des modèles mécaniques simplifiés qui font abstraction de certaines réalités physiques se sont imposés et ont été utilisés pour développer des modèles mathématiques qui utilisent le formalisme spécifique au traitement numérique du signal.

1.7.2 Le modèle numérique de la production de la parole

Comme on a vu, une modélisation exhaustive pour la production de la parole est très difficile et pour des raisons pratiques inefficace. L'idée de base dans la modélisation numérique est d'arriver á un modèle linéaire qui produit en sortie un signal équivalent au signal vocal. Le modèle est correct dans la mesure ou sa sortie s'approche du signal vocal sans modéliser les phénomènes physiques intrinsèques á la production du signal vocal [6]. La figure 6 présente un tel modèle général qui est utilisé dans le traitement numérique de la parole.

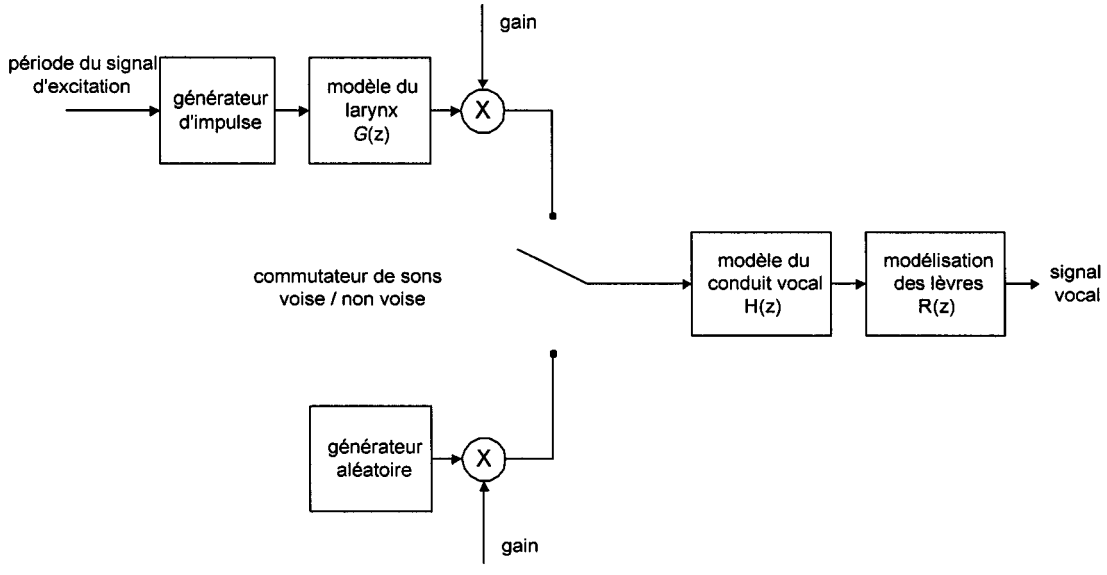


Figure 6 Le modèle numérique de la production de la parole [6]

Dans ce modèle général on utilise deux sources d'excitation. Pour les sons non voisés la source d'excitation est un bruit blanc. Pour la production des sons voisés la source d'excitation est un train périodique d'impulsions qui traverse un filtre passe bas d'ordre deux *G(z)* [3].

$$G(z) = \frac{A}{(1+a z^{-1})(1+b z^{-1})} \tag{1.1}$$

Ce filtre, qui modélise le fonctionnement du larynx, a une fréquence de coupure d'environ 100 Hz. Le résultat est un train périodique d'ondes de forme particulière, montée rapidement suivie d'une chute graduelle comme dans la figure 7.

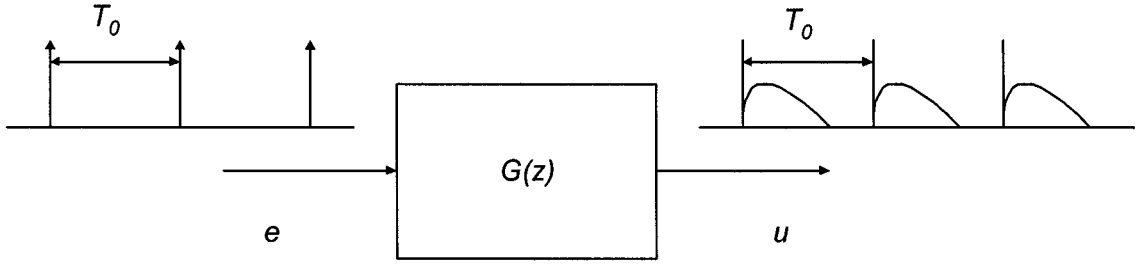


Figure 7 Le modèle de la source d'excitation pour les sons voisés [3]

Un modèle mécanique simplifié du conduit vocal le représente sous la forme d'une succession de tubes acoustiques élémentaires. Chaque tube où résonateur mécanique est assimilé à un filtre numérique d'ordre deux. La transmittance globale du modèle est de la forme [3] :

$$H(z) = \frac{B}{\prod_{k=1}^{K}(1+b_{1k}\,z^{-1}+b_{2k}\,z^{-2})} \tag{1.2}$$

La fréquence centrale de chaque résonateur correspond à un formant et est donnée par [3] :

$$F_k = \frac{1}{2\pi} f_s \cos^{-1}\left(\frac{-b_{1k}/2}{\sqrt{b_{2k}}}\right) \tag{1.3}$$

Au bout du conduit vocal le son passe à travers l'ouverture des lèvres. Celles-ci sont vues comme une composante qui transforme le débit volumique dans une onde de pression à une certaine distance. Dans le domaine spectral le rayonnement des lèvres a l'effet d'un filtrage passe haut. Le plus simple filtre numérique qui a cette propriété est [3] :

$$R(z) = C(1 - z^{-1}) \tag{1.4}$$

Pour des raisons de stabilité numérique et encore certaines détails physiques le zéro introduit par *R(z)* est déplacé à l'intérieur du cercle unité [5] :

$$R(z) = C(1 - z_0 z^{-1}), \quad z_0 \approx 1, \; z_0 < 1 \tag{1.5}$$

La présence d'un numérateur différent d'une constante rend difficile l'estimation des paramètres du système. Pour éliminer cet inconvénient on spécule sur l'identité :

$$1 - z_0 z^{-1} = \frac{1}{\sum_{k=0}^{K} z_0^k \; z^{-k}} \tag{1.6}$$

où K est théoriquement infinie mais pratiquement finie car $z_0 < 1$.

En conclusion la fonction de transfert globale est de la forme [5] :

$$T(z) = \begin{cases} H(z) R(z) & \text{pour les sons non voisés} \\ \\ G(z) H(z) R(z) & \text{pour les sons voisés} \end{cases} \tag{1.7}$$

Cette fonction de transfert de type tous-pôles qui est à la base de la modélisation par prédiction linéaire a été justifiée théoriquement et vérifiée pratiquement. Cependant elle présente une série de limitations.

Tout d'abord, la source d'excitation est soit un bruit blanc, soit un train périodique d'impulsions. Ce modèle ne peut pas donc produire les sons fricatifs voisés qui impliquent les deux sources d'excitation simultanément.

Par la suite, les sons nasalisés impliquent le couplage en parallèle des deux cavités, nasale et buccale. Dans ce cas la fonction de transfert résultant est de la forme [3] :

$$H(z)=\frac{d_1}{A_1(z)}+\frac{d_2}{A_2(z)}=\frac{d_1 A_2(z)+d_2 A_1(z)}{A_1(z)A_2(z)} \tag{1.8}$$

et elle présente des zéros, ce qui n'est pas permis par le modèle adopté.

L'analyse du signal vocal montre que les paramètres qui le caractérisent varient dans le temps. Toutefois ces variations sont assez lentes pour pouvoir les négliger pendant des intervalles de l'ordre de 30 à 40 ms [3,5-6]. En conséquence la source d'excitation et les coefficients de *T(z)* sont actualisés pour tenir le pas avec la nature non stationnaire du signal vocal chaque 30 ms au moins.

1.8 Notions d'acoustique

L'onde sonore est produite par la vibration d'une source. Le transfert de l'énergie vers le récepteur se réalise par une variation de pression dans un milieu élastique, l'air dans le cas du signal vocal.

La pression acoustique p_a dans un point de l'espace est la différence entre la pression en présence et la pression en absence de l'onde sonore.

L'intensité du son I est la quantité de l'énergie engendrée par le son qui traverse l'unité de surface normale à la direction de propagation de son dans une unité de temps [2].

$$I=\frac{{p_a}^2}{\delta c} \tag{1.9}$$

où c est la vitesse de propagation du son dans le milieu considéré et δ est la densité du milieu de propagation.

Pour comparer l'intensité de deux sons on utilise la notion de niveau d'intensité sonore *NI* [4]:

$$NI = 10\log\frac{I_1}{I_2} = 20\log\frac{p_1}{p_2} \tag{1.10}$$

En acoustique les niveaux d'intensité sonore s'expriment en fonction d'un niveau de référence $I_{ref} = 10^{-12}$ W/m^2, qui est le seuil d'audibilité et qui correspond à une pression sonore de $2\text{x}10^{-5}$ N/m^2 [7]. Le Tableau II fournit un exemple de niveaux de pression sonore pour diverses conditions.

Tableau II

Niveaux de pressions sonores pour diverse condition [7]

Condition	Pression sonore dB
moteur d'une fusée	200
moteur d'un avion à réaction	150
seuil de la douleur	140
tonnerre	110
les chutes du Niagara	100
métro	90
usine	80
rue agitée	70
bureau	50
salle d'audience	40
maison tranquille	30
studio d'enregistrement	20
seuil d'audibilité	0

1.9 Propriétés acoustiques de l'appareil auditif

L'appareil auditif est un système physiologique complexe qui joue un rôle important dans la réception et l'analyse de sons et donc du signal vocal dont certaines caractéristiques acoustiques seront présentées par la suite.

Les ondes sonores sont recueillies par l'appareil auditif et analysées par le cerveau. La relation complexe entre le son physique qui constitue l'excitation et la sensation auditive résultante dépend d'une part de la nature de l'excitation et d'autre part des particularités de l'appareil auditif humain. Pour pouvoir être perçues comme des sons, les vibrations acoustiques doivent satisfaire certaines conditions concernant la fréquence l'intensité et la durée.

Les mesures ont montré que l'oreille humaine perçoit comme des sons les vibrations acoustiques dont les fréquences se trouvent dans le domaine de 16 à 16000 Hz. Les variations de fréquence perceptibles dépendent de la fréquence et du niveau d'intensité sonore. Pour un niveau d'intensité sonore de 40 dB et pour des fréquences supérieures à 100 Hz on peut percevoir des variation de fréquence de jusqu à 0.3% [4].

En ce qui concerne l'intensité des vibrations acoustiques, pour pouvoir être perçu comme des sons elles doivent dépasser une certaine valeur limite inférieure. Cette valeur est variable en fonction de la fréquence ; pour la fréquence de 1000 Hz elle est égale à 10^{-16} W/cm^2 et appelée seuil d'audibilité. Si l'intensité des vibrations acoustiques augmente, on atteint une valeur limite supérieure d'environ 10^{-4} W/cm^2 au-delà duquel apparaissent la sensation de douleur [7]. Les variations d'intensité sonore perceptibles dépendent de la fréquence et du niveau d'intensité sonore. Pour des niveaux d'intensité sonore de 40-50 dB et pour des fréquences supérieures à 100 Hz on peut percevoir des variations d'intensité sonore inférieure à 2 dB [4].

Compte tenu de ces limites on peut définir une surface d'audibilité comme dans la figure 8 :

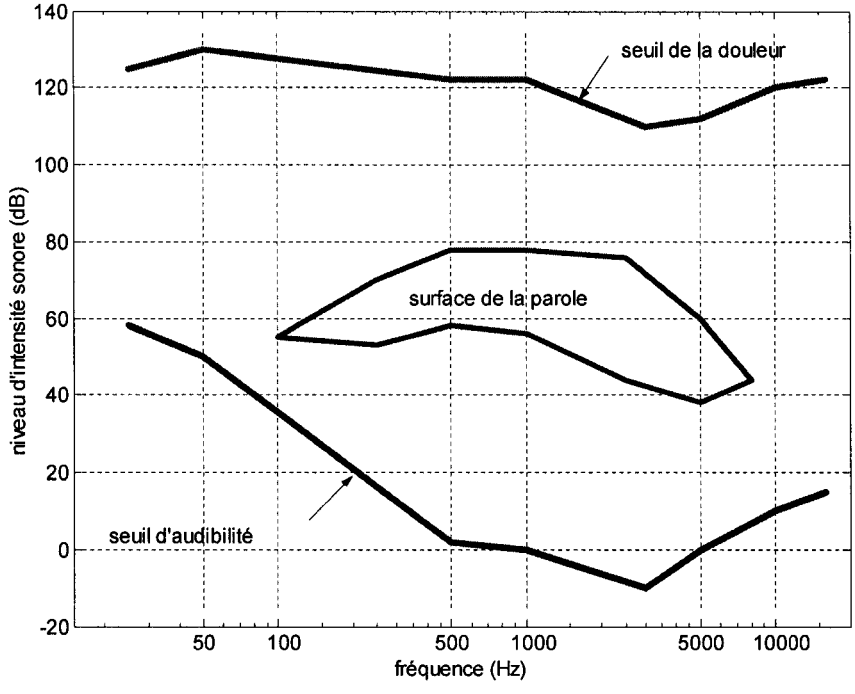


Figure 8 Surface d'audibilité de l'oreille [4]

Pour qu'une vibration acoustique puisse être perçue comme son, sa durée minime doit être supérieure à 12 ms pour une fréquence de 100 Hz et 4 ms pour les fréquences de 2000 à 10000 Hz. L'oreille, comme les yeux, présente une certaine inertie par rapport à la disparition de l'excitation d'environ 50-60 ms [4].

CHAPITRE 2

CARACTÉRISTIQUES DU SIGNAL VOCAL NUMÉRIQUE

2.1 Introduction

La représentation numérique d'un signal analogique implique l'échantillonnage, la quantification du chaque échantillon et le codage. La fréquence d'échantillonnage doit respecter le théorème de Shannon [8-9]. Le pas de quantification est en rapport avec la précision désiré [3,8]. Le codage est relié au problème technique particulier en vue de sa transmission ou de son enregistrement. Ce domaine est très vaste, il s'étend depuis le codage MIC (Modulation par Impulsions Codées) utilisé en téléphonie numérique commerciale jusqu'aux algorithmes très complexes qui permettent d'éliminer la redondance du signal vocal [4].

Le spectre du signal vocal s'étend jusqu'à 12 kHz et en conséquence, si on veut garder toute l'information, une fréquence d'échantillonnage d'au moins 24 kHz s'impose [4]. En faisant un compromis sur la qualité en faveur du coût du traitement numérique la fréquence d'échantillonnage peut varier dans des limites assez larges et devenir efficace pour le problème concret à résoudre. Ainsi on utilise une fréquence d'échantillonnage de 8 kHz pour la téléphonie et de 6 à 16 kHz pour l'analyse ou la synthèse du signal vocal [5-6].
Dans ce chapitre par signal vocal on sous-entend la version discrétisée de celui-ci.

Le signal vocal duquel on dispose dans une application pratique est formé d'une succession de régions de parole et de silence ou bruit intercalé. L'information utile de ce signal est contenue dans les régions de parole et généralement on veut détecter les limites des ces régions pour pouvoir les analyser séparément ce qui conduit vers un traitement plus efficace de l'ensemble du signal. Un algorithme de détection d'activité

vocale VAD est donc utilisé pour détecter les limites des régions de parole dans le signal vocal analysé. Ainsi un algorithme de VAD devrait fournir une décision pour chaque échantillon de signal numérique. Dans la pratique cette approche est inefficace, d'une part à cause du coût du traitement numérique et d'autre part à cause de la structure du signal vocal, les régions de parole et de silence étant des régions compactes qui incluent plusieurs centaines d'échantillons.

L'approche couramment utilisée dans la détection d'activité vocale est basée sur l'analyse à court terme du signal vocal. Le signal vocal est divisé en segments très courts qui deviennent ainsi l'unité de base d'analyse et de décision.

La justification de cette approche sera exposée dans le prochain chapitre. Par la suite on va présenter une série de paramètres qui utilisent le concept d'analyse court-terme pour caractériser le signal vocal numérique. Ces paramètres sont utilisés dans les divers algorithmes de VAD présentés dans le chapitre suivant.

2.2 Traitement court-terme du signal vocal

Une simple inspection visuelle de la forme d'onde du signal vocal (figure 3) met en évidence la nature non stationnaire du celui-ci. On peut facilement observer les variations en amplitude ou dans la fréquence. Etant donné la nature non stationnaire du signal vocal, une analyse globale ou à long terme est dans la majorité des cas inefficace. D'autre part, on possède des moyens très puissants pour l'étude des systèmes linéaires et invariants dans le temps. Dans ces conditions l'hypothèse le plus utilisée dans le traitement de la parole est le fait que les propriétés du signal vocal changent lentement dans le temps [6]. Cette hypothèse conduit vers un traitement à court terme. Les segments du signal vocal sont isolés et traités comme s'ils étaient des fragments composant des sons soutenus et invariants. Pour ces segments, qui d'habitude se

chevauchent, on peut faire usage des mêmes outils que dans le cas d'un système linéaires et invariant dans le temps SLIT [5-7].

Du point de vue statistique, le signal vocal pour des segments courts de temps est considéré la réalisation d'un processus aléatoire stationnaire et ergodique. Ces deux propriétés permettent respectivement d'être indépendant d'un décalage temporel et d'identifier les moyennes d'ensembles avec les moyennes temporelles [5].

Le résultat du traitement d'un tel segment peut être un seul numéro ou un set de numéros qui devient une nouvelle représentation du signal vocal. La représentation mathématique de ce processus est décrite par la relation [6] :

$$Q_n = \sum_{m=-\infty}^{\infty} T(x(m))\, w(n-m) \tag{2.1}$$

Le signal vocal est transformé par l'opérateur *T( )* et le résultat est multiplié par une fenêtre alignée à l'échantillon *n*. D'habitude cette fenêtre contient un nombre fini d'échantillons mais ce n'est pas toujours le cas. La relation (2.1) est le produit de convolution de la fenêtre *w(n)* et de la séquence *T(x(n))*. Donc Q_n peut être vu comme étant la sortie d'un SLIT qui pourrait être un filtre caractérisé par la réponse impulsionnelle *h(n) = w(n)*. Cette interprétation est représentée graphiquement dans la figure 9.

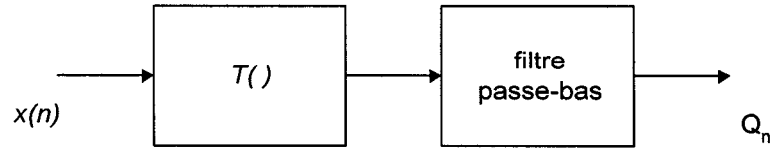


Figure 9 Représentation graphique du principe d'analyse court-terme [6]

Pour les choix de la fenêtre on regarde deux aspects : la dimension en nombre d'échantillons et la forme, chaque aspect ayant des répercussions différentes sur les analyses ultérieures. L'effet du fenêtrage peut être mis en évidence par l'étude des propriétés de deux fenêtres représentatives : la fenêtre rectangulaire et la fenêtre de Hamming.

La fenêtre rectangulaire est donnée par [8] :

$$h(n)=\begin{cases}1 & 0\le n\le N-1\\ 0 & ailleurs\end{cases} \tag{2.2}$$

Le module de la réponse fréquentielle de la fenêtre rectangulaire est [8] :

$$|H(e^{j\omega})|=\frac{|\sin(\omega N/2)|}{|\sin(\omega/2)|} \tag{2.3}$$

Le premier zéro dans la relation (2.3) se réalise pour $F=F_s/N$ où F_s est la fréquence d'échantillonnage. Cette fréquence F délimite la largeur du premier lobe qui caractérise le module de la réponse fréquentielle de la fenêtre rectangulaire, (figure 10) et qui diminue quand N augmente.

La fenêtre de Hamming est décrite par [8] :

$$h(n)=\begin{cases}0.54-0.46\cos(2\pi\ n(N-1)) & 0\le n\le N-1\\ 0 & ailleurs\end{cases} \tag{2.4}$$

La largeur du premier lobe du module de la réponse fréquentielle de la fenêtre de Hamming est le double du premier lobe correspondant à la fenêtre rectangulaire de même longueur. D'autre part la fenêtre de Hamming a une plus grande atténuation à l'extérieur du premier lobe par rapport à la fenêtre rectangulaire.

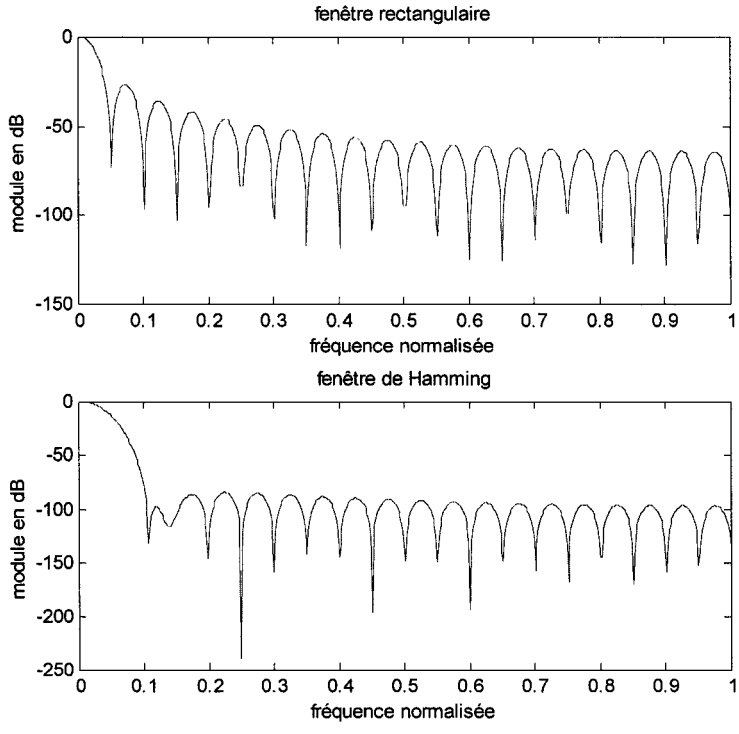


Figure 10 Comportement fréquentiel pour la fenêtre rectangulaire et la fenêtre de Hamming $N = 40$

Pour les deux fenêtres l'augmentation de N conduit vers une diminution de la largeur du lobe principal. Donc une meilleure résolution spectrale est obtenue avec une fenêtre plus longue.

Parfois dans la pratique on est obligé d'utiliser une fenêtre assez longue pour surprendre certaines aspects qui caractérisent le signal vocal, par exemples la fréquence fondamentale de la voix d'un homme qui peut nécessiter jusqu'à 200 échantillons à une fréquence d'échantillonnage F_s = 8 kHz. D'autre part une valeur trop grande pour N se concrétise par une fenêtre trop longue pour laquelle l'hypothèse d'invariance temporaire est violée.

Compte tenu de ces aspects, le choix du N se fait en fonction du problème concret à résoudre. Il est généralement dans la plage de 80 à 160 échantillons pour une fréquence d'échantillonnage F_s = 8 kHz.

2.3 Énergie court-terme

On a déjà vu que l'amplitude du signal vocal varie d'une façon importante dans le temps. L'amplitude des régions voisées du signal vocal est généralement plus grande que celle des régions non voisées. L'énergie court-terme est un paramètre qui reflète ces variations d'amplitude dans le signal vocal et elle a été un de premiers paramètres utilisés dans la détection d'activité vocale. La définition de ce paramètre est [6] :

$$E_n = \sum_{m=-\infty}^{m=\infty} [x(m)w(n-m)]^2 \tag{2.5}$$

ou encore

$$E_n = \sum_{m=-\infty}^{m=\infty} x^2(m)h(n-m) \tag{2.6}$$

où

$$h(n) = w^2(n) \tag{2.7}$$

L'équation (2.6) voit l'énergie court-terme comme étant la sortie d'un filtre défini par la réponse impulsionnelle $h(n)$, relation (2.7), excité par le signal d'entrée $x^2(n)$. La discussion portée sur les choix de la fenêtre dans le chapitre précédent peut être maintenant vérifiée par un exemple concret. La figure 11 présente l'énergie court-terme pour la même phrase utilisant deux fenêtres rectangulaires de longueurs différentes. La longueur de la deuxième fenêtre N = 160 échantillons est le double de la première pour laquelle N = 80 échantillons.

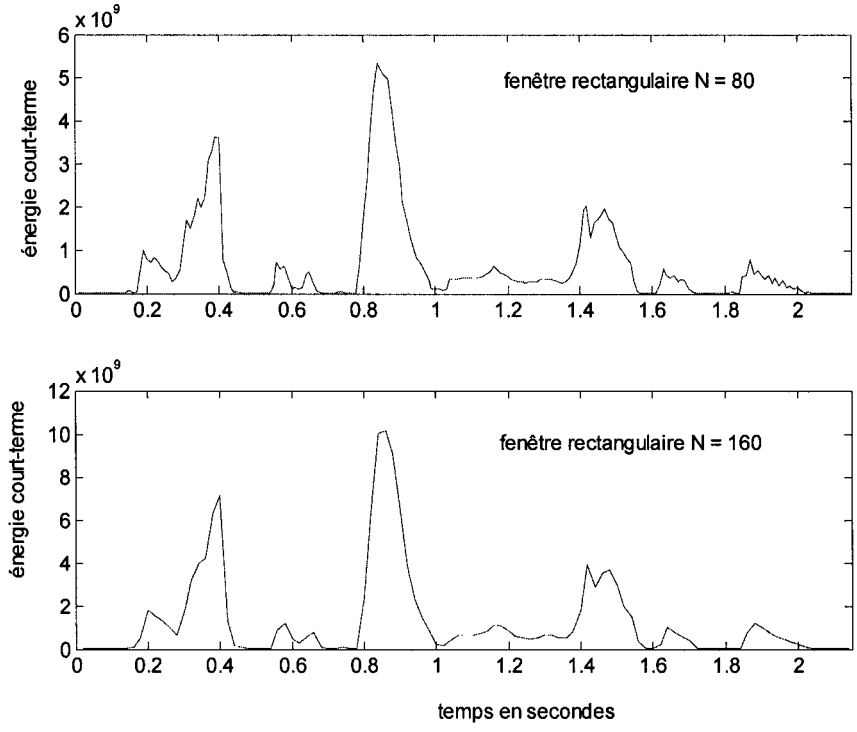


Figure 11 Énergie court-terme pour deux fenêtres rectangulaires de dimensions différentes

On peut voir clairement l'effet de lissage induit par la deuxième fenêtre par rapport à la première.

L'énergie court-terme définie par la relation (2.5) est très sensible aux niveaux des signaux car elle fait intervenir l'amplitude au carré. Parfois dans la pratique on préfère utiliser une forme simplifiée pour estimer l'énergie court-terme comme suit [10] :

$$M_n = \sum_{m=-\infty}^{\infty} |x(n)| w(n-m) \tag{2.8}$$

L'effet est de diminuer les différences entre les valeurs du paramètre pour les régions voisées et non voisées comme on peut voir dans la figure 12.

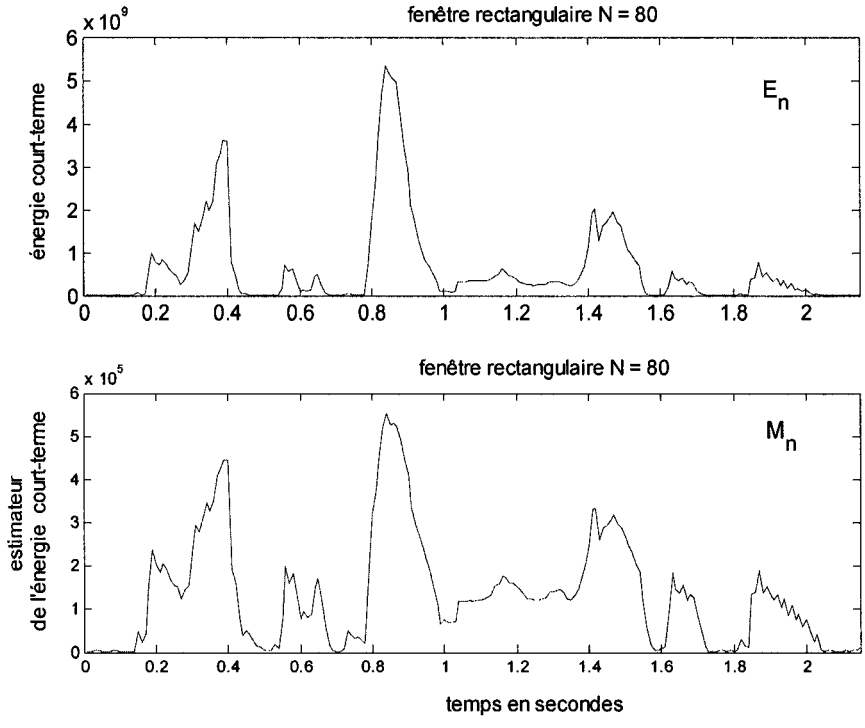


Figure 12 Énergie court-terme et l'estimateur de l'énergie court-terme M_n

2.4 Taux de passage par zéro

Le taux de passage par zéro est une estime grossière du contenu fréquentiel du signal analysé. Il est aussi un des premiers paramètres utilisés dans le VAD car la structure spectrale du bruit est différente de celle de la parole.

Pour un signal discret il y un passage par zéro quand deux échantillons successifs ont le signe différent. Ce paramètre est estimé par l'équation :

$$Z_n = \sum_{m=-\infty}^{m=\infty} |\,\mathrm{sgn}[(x(m)] - \mathrm{sgn}[x(m-1)]\,|\,w(n-m) \qquad (2.9)$$

où

$$\text{sgn}[x(n)] = \begin{cases} 1 & s(n) \geq 0 \\ -1 & s(n) < 0 \end{cases} \tag{2.10}$$

et

$$w(n) = \begin{cases} \dfrac{1}{2N} & 0 \leq n \leq N-1 \\ 0 & \textit{ailleurs} \end{cases} \tag{2.11}$$

On sait que les sons voisés sont caractérisé par une forte composante de basse fréquence, due à l'excitation, et que pour les sons non voisés plus d'énergie est concentrée dans la région de haute fréquence [5]. On s'attend donc d'avoir un taux de passage par zéro moins élevé pour les régions voisées du signal vocal que pour les régions non voisées. En observant des segments de 10 ms, on trouve des distributions gaussiennes avec une moyenne de 14 pour les sons voisés et une moyenne de 49 pour les sons non voisés, ces deux répartitions se recouvrent partiellement [4].

Un exemple pour ce paramètre est représenté dans la figure 13 où l'on a utilisé une fenêtre de 10 ms.

Dans le cas d'une utilisation pratique de ce paramètre il faut diminuer au maximum certains types de bruit qui ont un impact très important sur le résultat. Par exemple si le convertisseur analogue numérique introduit un décalage par rapport à la valeur zéro, comparable avec l'amplitude du signal, le taux de passage par zéro devient nul ou très petit. Cet effet est plus important pour les régions de silence ou non voisées du signal vocal. Il est préférable d'utiliser un filtre passe-bande capable d'éliminer les bruits de base fréquence au lieu du filtre passe-bas anti-recouvrement.

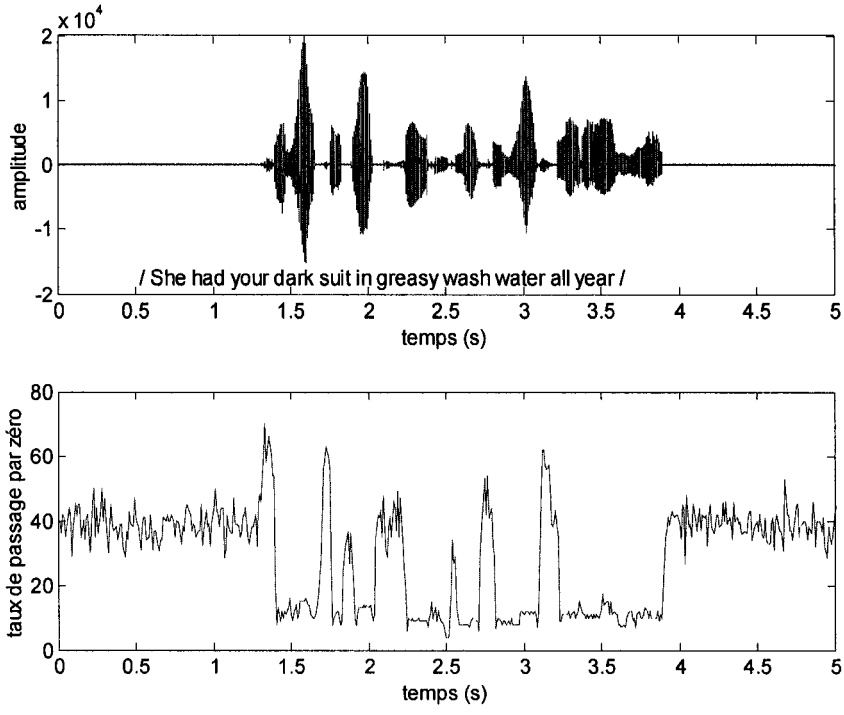


Figure 13 Taux de passage par zéro en utilisant une fenêtre de 10 ms

2.5 La fonction d'autocorrélation

La fonction d'autocorrélation d'un signal ergodique et stationnaire est donnée par l'équation [6] :

$$\phi(k) = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{m=-N}^{N} x(m)\,x(m+k) \tag{2.12}$$

La fonction d'autocorrélation est une fonction paire en k, elle atteint son maximum pour $k = 0$. Dans le cas de signaux périodiques de période P, $\phi(0)$ est la puissance moyenne et la relation suivante est vérifiée [6] :

$$\phi(k) = \phi(k+P) \tag{2.13}$$

Comme suite à ces propriétés, on peut déduire que la fonction d'autocorrélation d'un signal périodique présente maximums pour les échantillons $\pm P$, $\pm 2P$, $\pm 3P\ldots$. Cela fait d'elle un bon outil pour estimer la périodicité dans une large gamme de signaux incluant le signal vocal.

Dans le contexte du traitement court-terme du signal vocal donc sur un nombre constant et fini de N échantillons, la fonction d'autocorrélation peut être calculée par [6] :

$$R_n(k) = \sum_{m=0}^{N-1-k} [x(n+m)w(m)][x(n+m+k)w(k+m)] \tag{2.14}$$

À cause de la valeur finie du N on utilise moins d'échantillons dans le calcul de R_n quand k augmentent. Cela se concrétise dans une réduction de l'amplitude de la fonction d'autocorrélation court-terme quand k augmente et donc une diminution de l'amplitude des maximums correspondants aux signaux corrélés. Pour diminuer ce phénomène on peut choisir une fenêtre plus longue mais ceci a l'inconvénient de multiplier le nombre d'opérations arithmétiques nécessaires. De plus on sait que l'hypothèse de stationnarité ne tient que pour des fenêtres de temps inférieures à 30 ms. Une autre approche est de modifier la définition de la fonction d'autocorrélation comme suit [6] :

$$\widehat{R}_n(k) = \sum_{m=0}^{N-1} x(n+m)x(n+m+k) \quad 0 \le k \le K \tag{2.15}$$

où K est le plus grand pas pour lequel on calcule $\widehat{R}_n$. Dans cette approche on utilise K échantillons à l'extérieur de la fenêtre utilisée dans le calcul de R_n. On se réfère à la relation (2.15) comme étant la fonction d'autocorrélation court-terme modifiée.

2.6 La fonction de différence moyenne d'amplitude

Une autre fonction qui est capable de mettre en évidence le caractère périodique d'un signal est basée sur l'idée que pour un signal x périodique de période P le paramètre défini avec l'équation :

$$d(n) = x(n) - x(n-k) \tag{2.16}$$

est zéro pour $k = \pm P, \pm 2P, \pm 3P\ldots$. Par la suite on s'attend que pour des intervalles courts de temps dans le cas du signal voisé la fonction décrite par l'équation [5] :

$$\gamma(k) = \sum_{m=-\infty}^{m=\infty} | x(n+m)w(m) - x(n+m-k)w(m-k) | \tag{2.17}$$

présente minimums quand k approche un multiple de la période.
Cette nouvelle fonction n'implique pas des multiplications. Pour cette raison elle est moins coûteuse en termes de temps de calcul et donc parfois préférée dans des applications temps-réel.

2.7 Lisage médian et filtrage linéaire

Une technique largement utilisée pour éliminer le bruit dans un signal est le filtrage linéaire. Dépendant du type de donnée utilisée, cette technique ne donne pas toujours les meilleurs résultats. Un exemple est le cas de signaux caractérisés par des points de discontinuité qu'il faut préserver mais aussi des points largement erronés qu'il faut éliminer. Bien qu'un algorithme idéal pour résoudre ce problème n'existe pas, certaines techniques de lisage qui utilisent une combinaison des moyennes temporaires et filtrage linéaire donne de bons résultats.

On utilise cette technique pour réduire la variance des paramètres employés dans l'analyse court-terme pour décrire le signal vocal. Dans le cas spécifique de la détection d'activité vocale ; cette approche est justifiée car les régions de parole et de silence sont des régions compactes formées des plusieurs trames.

Dans le cas de filtrage linéaire le signal est vu comme une somme pondérée de sinusoïdes et l'effet du filtrage est de modifier les amplitudes des ces sinusoïdes. Dans le cas du lisage non linéaire il est plus utile de voir le signal comme étant formé d'une composante lisse et d'une autre bruyante. Une description analytique pour ce signal est de la forme [6] :

$$x(n) = S[x(n)] + R[x(n)] \tag{2.18}$$

où $S[x(n)]$ est la partie lisse est $R[x(n)]$ est la partie bruyante du signal. Une technique non linéaire qui est capable de séparer le deux composantes S et R est le lisage médian. Ceci est la moyenne des valeurs comprises dans la fenêtre alignée au point n. Le lisage médian préserve les discontinuités qui se manifestent dans le signal sur une durée plus grande que la longueur de la fenêtre L et suit la tendance générale du signal mais n'est pas assez efficace en terme d'élimination de la composante bruyante du signal.

Un bon compromis est assuré par une combinaison de lisage médian et filtrage linéaire comme dans la figure 14(a). Le rôle du filtrage linéaire est d'éliminer la partie bruyante qui reste après le lisage médian. On utilise d'habitude un filtre de type FIR a coefficients symétriques.

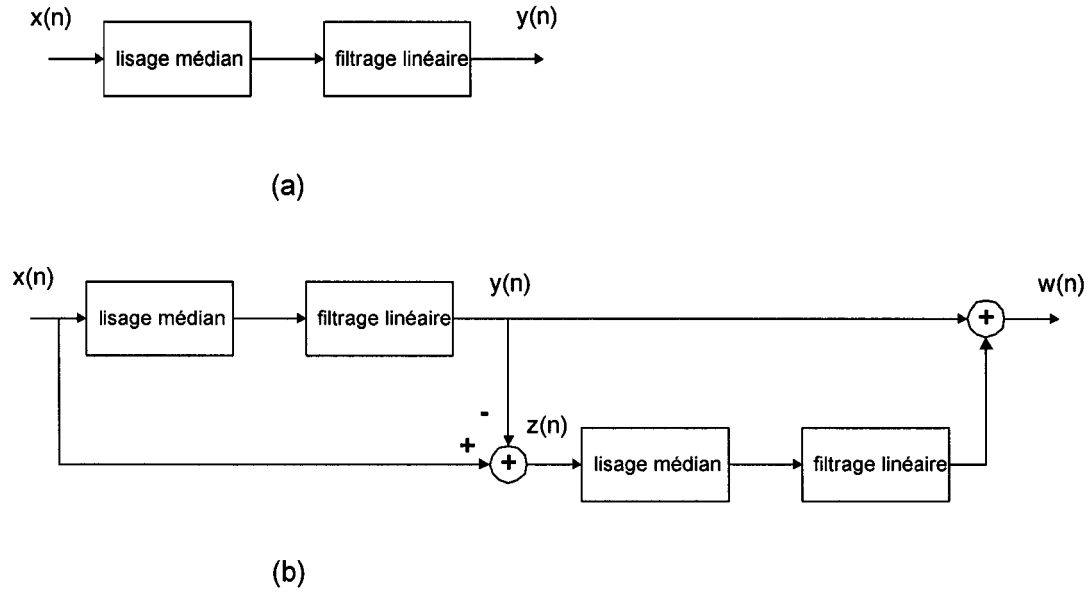


Figure 14 Schéma bloc d'un système de lisage non linéaire [6]

Dans la figure 14(a) le signal $y(n)$ est une approximation de $S[x(n)]$. Une réalisation pratique est suggérée dans la figure 14(b) ou :

$$y(n) = S[x(n)] \tag{2.19}$$

et alors

$$z(n) = x(n) - y(n) = R[x(n)] \tag{2.20}$$

Dans la figure 14(b) le signal $z(n)$ est à son tour lissé et additionné au $y(n)$ pour obtenir une meilleure approximation de $S[x(n)]$. Le signal $w(n)$ satisfait la relation :

$$w(n) = S[x(n)] + S[R[x(n)]] \tag{2.21}$$

Dans le cas d'un lisage idéal la quantité $z(n)$ serait juste la partie bruyante du signal pour laquelle la valeur $S[R[x(n)]]$ est nulle et donc la correction inutile.

Il faut souligner que le lisage linéaire introduit un retard de $(L\text{-}1)/2$ échantillons ou L est la longueur de la fenêtre utilisée.

2.8 Transformée de Fourier court-terme

La représentation des signaux par une somme d'exponentiels complexes ou des sinusoïdes est connue sous le nom de transformée de Fourier. Du à ses propriétés, cette transformée est un outil largement utilisé dans le traitement de signal [2-9].

Puisque le signal vocal peut être considéré stationnaire pour des intervalles courts de temps, il est utile d'introduire la notion de transformée de Fourier court-terme. Cette extension logique de la transformée de Fourier classique d'un signal discret introduit pour surprendre les variations temporelles du spectre du signal vocal nouvelle notion est une et elle est définie par [3,5-7] :

$$X_n(e^{j\omega}) = \sum_{m=-\infty}^{\infty} w(n-m)\,x(m)\,e^{-j\omega m} \tag{2.22}$$

Cette représentation du signal vocal est une fonction de deux variables : l'indice du temps n qui prend des valeurs discrètes et la pulsation ω qui est une variable continue.

Cette équation peut être interprétée de deux façons différentes. Premièrement pour n fixe $X_n(e^{j\omega})$ est la transformée de Fourier classique de la séquence $w(n\text{-}m)x(m)$ et donc en possède toutes les propriétés. L'existence de cette transformée est assurée car la quantité $w(n\text{-}m)x(m)$ est toujours absolument sommable dans le cas d'une fenêtre w finie en temps. La fenêtre w est utilisée pour délimiter les segments du signal vocal. De plus on

peut retrouver la valeur $x(n)$ par l'intermédiaire de la transformée de Fourier inverse [8-9] :

$$x(n) = \frac{1}{2\pi\, w(0)} \int_{\pi}^{\pi} X_n(e^{j\omega}) e^{j\omega n} d\omega \tag{2.23}$$

à condition que $w(0)$ ne soit pas nul.

Deuxièmement pour ω fixe $X_n(e^{j\omega})$ est écrit sous la forme d'une convolution et donc on peut interpréter la relation (2.22) en terme de filtrage linéaire.

Un facteur important dans le calcul de la transformée de Fourier court-terme est la fenêtre utilisée. On suppose que les transformées de Fourier suivantes existent :

$$X(e^{j\omega}) = \sum_{m=-\infty}^{\infty} x(m) e^{-j\omega m} \tag{2.24}$$

et

$$W(e^{j\omega}) = \sum_{m=-\infty}^{\infty} w(m) e^{-j\omega m} \tag{2.25}$$

alors la transformé de Fourier de $w(n-m)x(m)$ pour n fixe est la convolution entre les transformées de $w(n-m)$ et $x(m)$. La transformée de $w(n-m)$ est $W(e^{-j\omega}) e^{-j\omega n}$ et donc [6] :

$$X_n(e^{j\omega}) = \frac{1}{2\pi} \int_{-\pi}^{\pi} W(e^{j\theta}) e^{j\theta n} X(e^{j(\omega+\theta)}) d\theta \tag{2.26}$$

L'équation précédente est correcte dans le cas où on supposerait que $X(e^{j\omega})$ est la transformée de Fourier d'un signal qui a le même spectre ou qui est nul à l'extérieur de la fenêtre étudiée. Dans ces conditions la transformée de Fourier court-terme peut être vue comme une version lisse du spectre idéal du signal. Le lissage induit dans le spectre du signal est la conséquence de l'utilisation du filtre médian résultant de la transformée de Fourier de la fenêtre temporelle *w*. L'effet de la fenêtre devient clair maintenant, la meilleure fenêtre temporaire du point de vue de la résolution est celle qui s'approche le plus possible d'une impulsion dans le domaine fréquentiel.

Comme on l'a déjà vu au début de ce chapitre, la largeur du lobe central de $W(e^{-j\omega})$ est en rapport inverse avec la largeur de la fenêtre lorsque l'amplitude des lobes latéraux est indépendante de la dimension de la fenêtre. On peut diminuer l'amplitude des lobes latéraux au détriment de l'épaisseur du lobe central par un choix judicieux de la forme de la fenêtre. De ce point de vue la figure 10 montre un meilleur comportement de la fenêtre de Hamming par rapport à la fenêtre rectangulaire.

2.8.1 Transformée de Fourier discrète

La transformée de Fourier d'un signal discret est évaluée pour un nombre infini de pulsations ce qui pose un problème pour l'évaluation numérique. La solution est de remplacer la variable continue ω par une variable discrète k où k vaut de zéro à N-1 et divise l'intervalle de $-\pi$ à π en N sous-intervalles égaux.
La définition de la transformée de Fourier discrète à temps discret TFD est [8-9] :

$$X(k) = \sum_{n=0}^{N-1} x(n) V^{nk} \qquad k = 0,1,...,N-1 \tag{2.27}$$

avec

$$V = e^{-j\frac{2\pi}{N}} \tag{2.28}$$

Le calcul direct de la TFD nécessite un grand nombre d'opérations arithmétiques : (N^2-N) additions complexes et N^2 multiplications complexes. Plusieurs méthodes ont été développées pour réduire le nombre d'opérations nécessaire au calcul de la TFD . Une telle méthode pour le calcul de la transformée de Fourier rapide TFR est présenté dans l'annexe 1.

2.9 Analyse spectrale du signal vocal

Le signal vocal ne possède pas une description temporelle analytique mais on peut l'assimiler à un signal aléatoire gouverné par des lois statistiques. De plus, pour des intervalles courts de temps, inférieurs à 35 ms, on peut le considérer comme la réalisation d'un processus aléatoire stationnaire et ergodique. La fonction d'autocorrélation est l'outil principal que l'on utilise pour caractériser les processus aléatoires dans le domaine temporel. La transformée de Fourier de la fonction d'autocorrélation, qui produit la densité spectrale de puissance, est une description fréquentielle des processus aléatoires.

Le problème principal qui se pose est d'estimer la densité spectrale de puissance du signal vocal à partir d'un nombre N fini d'échantillons correspondants à la fenêtre temporelle utilisée.

Comme il s'agit principalement d'un problème d'estimation, il est utile de définir deux paramètres qui caractérisent les estimateurs : le biais et la variance. Le biais d'un estimateur est la différence entre son espérance mathématique et sa vraie valeur. La variance d'un estimateur est la mesure de l'étendue de la densité de probabilité de l'estimation. La variance et le biais d'un bon estimateur doivent tendre vers zéro quand N tend vers l'infinie et ceci est la définition d'un estimateur consistant. La somme des

carrés de ces deux erreurs, variance et biais, donne une erreur globale connue sous le nom d'erreur quadratique moyenne [3-9].

2.9.1 Analyse spectrale non paramétrique

L'hypothèse d'ergodicité permet d'utiliser l'équation (2.11) pour calculer la fonction d'autocorrélation. Cette équation requiert un nombre infini d'échantillons d'où la nécessité d'une estimation pour $\phi(k)$ quand N est finie. Un premier estimateur non biaisé est celui dont la variance est inversement proportionnelle à N, donc consistant. Il est donné par la relation [8]:

$$r_x'(k) = \frac{1}{N-|k|} \sum_{n=0}^{N-|k|-1} x(n)x(n-k) \tag{2.29}$$

Pour des valeurs k proches de N la variance de cet estimateur est grande parce qu'il est calculé avec peu de termes. On utilise alors un autre estimateur, cette fois biaisé mais qui possède une erreur quadratique moyenne plus faible. Il est donné par l'équation [8] :

$$r_x(k) = \frac{1}{N} \sum_{n=0}^{N-|k|-1} x(n)x(n-k) \tag{2.30}$$

Sa variance est inversement proportionnelle à N et son biais tend vers zéro quand N tend vers infinie. Il faut remarquer que dans le cas d'une fenêtre de dimensions constantes le terme $1/N$ devient une constante et on retrouve la relation (2.14).
Par définition on a la densité spectrale de puissance [8] :

$$S_x(\omega) = \sum_{k=-(N+1)}^{N-1} r_x(k) e^{-j\omega k} \tag{2.31}$$

Si on remplace r_x avec (2.29) on a [8] :

$$S_x(\omega) = \frac{1}{N} | \sum_{n=0}^{N-1} x(n) e^{-j\omega k} |^2 = \frac{1}{N} | X(\omega) |^2 \qquad (2.32)$$

Cet estimateur simple de la densité spectrale de puissance est connu sous le nom de periodogram. Le biais de l'estimateur simple n'est pas nul et sa variance ne dépend pas de la durée d'observation N. L'estimateur simple n'étant pas consistant, d'autres estimateurs plus performants ont été proposés.

Pour diminuer la variance de l'estimateur simple, le signal observé de dimension N est segmenté en L sections de durée M chacune. On a [8] :

$$x_l(n) = x(n + (l-1)M) \qquad avec \quad n = 0,...,M-1, N = ML, l = 1,...L \qquad (2.33)$$

On calcule les L estimateurs simples [8] :

$$S_{xl}(\omega) = \frac{1}{M} | \sum_{n=0}^{M-1} x_l(n) e^{-j\omega n} |^2 \quad l = 1,...,L \qquad (2.34)$$

L'estimateur spectral moyenné proposé par Barttlet est [8] :

$$\overline{S}_x(\omega) = \frac{1}{L} \sum_{l=1}^{L} S_{lx}(\omega) \qquad (2.35)$$

Le biais de l'estimateur moyenné est plus grand que celui de l'estimateur simple mais sa variance est L fois plus petite. Comme $L=M/N$ pour N fixe, on a un compromis entre le biais qui diminue avec l'augmentation de M et la variance qui diminue avec l'augmentation de L.

Une autre façon de réduire la variance de l'estimateur simple est de le considérer comme un signal bruité et de le filtrer. Il s'agit d'effectuer un filtrage des variations rapides en fonction de la fréquence en utilisant une fenêtre spectrale $W(\omega)$. Ainsi, pour obtenir un estimateur adouci, on a [3,8] :

$$\tilde{S}(\omega) = \sum_{k=-(M-1)}^{M-1} w(k) r(k) e^{-j\omega k} \tag{2.36}$$

où $w(k)$ est la fenêtre temporelle de longueur M.

Un dernier estimateur, dit modifié, met en valeur les avantages de deux derniers estimateurs. Le signal de longueur N est divisé en L sections de durée M qui se chevauchent. Chaque section est d'abord pondérée par la fonction fenêtre $w(k)$ avant de calculer l'estimateur simple pour chaque section qui est [3,8] :

$$S_{xl}(\omega) = \frac{1}{MP} \left| \sum_{k=0}^{M-1} x(k) w(k) e^{-j\omega k} \right|^2$$

$$avec \quad l = 1,\ldots,L \qquad et \qquad P = \frac{1}{M} \sum_{k=0}^{M-1} w^2(k) \tag{2.37}$$

L'estimateur modifié est donné par [3,8] :

$$\hat{S}(\omega) = \frac{1}{L} \sum_{l=1}^{L} S_{xl}(\omega) \tag{2.38}$$

Les biais et la variance de cet estimateur sont inversement proportionnels à M et à L respectivement.

Pour le signal vocal on peut appliquer cette méthode d'estimation de la densité spectrale de puissance d'une façon répétitive pour des intervalles de temps inférieurs à 35 ms en utilisant une fenêtre de pondération. On déplace cette fenêtre à chaque 10 ms, donc pour une fenêtre de 20 ms on a un chevauchement de 50% entre deux trames consécutives. Pour chaque intervalle on estime la densité spectrale de puissance à l'aide de la relation (2.53). La moyenne de ces trois densités spectrales de puissance représente le résultat cherché. On remarque que ces paramètres assurent une nouvelle estimée à chaque 10 ms avec un minimum de calcul.

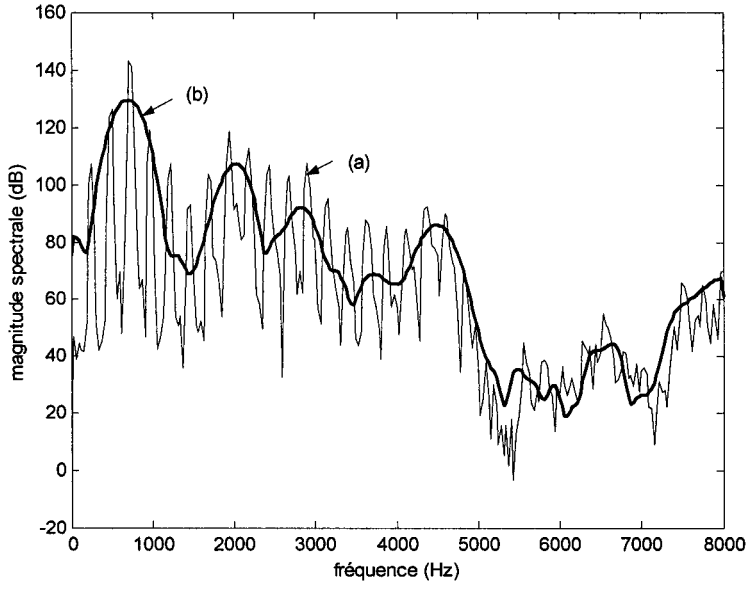


Figure 15 Le comportement de deux estimateurs de densité spectrale de puissance estimateur simple (a) et estimateur modifié (b) [5]

Le comportement des deux estimateurs est présenté dans la figure 15 pour un exemple réel. L'analyse se fait sur un segment de signal contenant la voyelle / ə / d'une durée de 32 ms échantillonné à 16000 Hz. Dans un premier cas le signal est pondéré par une fenêtre de Hamming et la densité spectrale de puissance est estimée à l'aide de

l'estimateur simple en 512 points. Dans un deuxième cas on a utilisé la même fenêtre pour l'estimateur modifié avec $L = 3$ et $M = 256$. On peut remarquer clairement la réduction de la variance associée à l'estimateur modifié.

2.10 Le modèle autorégressif pour la production de la parole

L'une des plus puissantes techniques utilisées dans l'analyse du signal vocal est la prédiction linéaire. Elle est largement utilisée dans l'estimation des principaux paramètres du signal vocal et pour réduire le débit binaire pour la transmission ou le stockage.

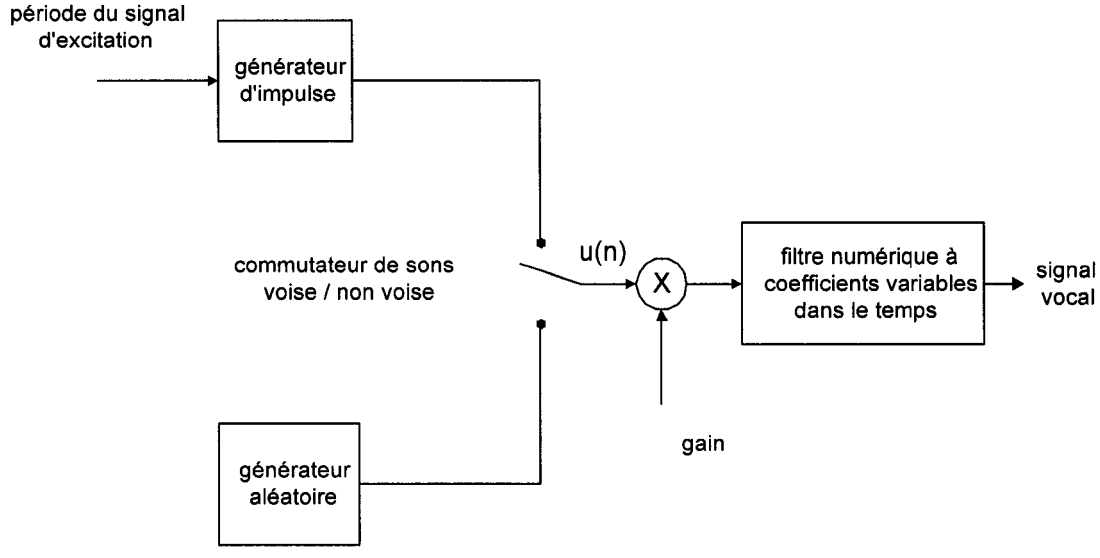


Figure 16 Modèle autorégressif de production de la parole [6]

On représente un échantillon quelconque du signal vocal comme une combinaison linéaire d'échantillons qui le précèdent. Un ensemble unique de coefficients de prédiction est obtenu en minimisant la somme des carrés des différences entre les

échantillons du signal et les échantillons prédits pour un intervalle fini pour lequel le signal est stationnaire.

Dans le chapitre § 1.8 on a formulé un modèle numérique général pour la production de la parole. Un modèle équivalent qui est particulièrement adapté pour la modélisation autorégressive est présenté dans la figure 16.

Dans ce modèle l'effet du larynx, du conduit vocal et des lèvres sous le spectre de sortie est modélisé par un filtre numérique à coefficients variables dans le temps. La forme de la fonction de transfert de ce filtre est [5-6] :

$$H(z) = \frac{X(z)}{U(z)} = \frac{G}{1 - \sum_{k=1}^{p} a_k z^{-k}} \tag{2.39}$$

Le système est excité par un train périodique d'impulsions pour les sons voisés et par un bruit blanc pour les sons non voisés. Les paramètres du modèle sont : la période du fondamental pour l'excitation dans le cas des sons voisés, le gain G, et les coefficients du filtre. Ces paramètres varient lentement dans le temps.

Pour ce système l'échantillon $x(n)$ est relié à l'excitation $u(n)$ par une récurrence linéaire [5-6] :

$$x(n) = \sum_{k=1}^{p} a_k x(n-k) + Gu(n) \tag{2.40}$$

En réalité on ne connaît pas l'excitation, donc l'estimation des paramètres du modèle sera basée exclusivement sur l'observation du signal. Dans ce cas l'estimé de chaque échantillon $x(n)$, à partir de p échantillons qui le précèdent, est [5-6] :

$$\tilde{x}(n) = \sum_{k=1}^{p} \alpha_k x(n-k) \tag{2.41}$$

L'erreur de prédiction linéaire est [5-6] :

$$e(n) = x(n) - \tilde{x}(n) = x(n) - \sum_{k=1}^{p} \alpha_k x(n-k) \tag{2.42}$$

Donc l'erreur de prédiction linéaire est la sortie d'un système dont la fonction de transfert est [5-6] :

$$A(z) = 1 - \sum_{k=1}^{p} \alpha_k z^{-k} \tag{2.43}$$

Si le signal vocal obéit à l'équation (2.40) et que l'on pose $a_k = \alpha_k$ dans (2.42), alors l'excitation coïncide avec l'erreur de prédiction linéaire à un facteur prés [5-6] :

$$e(n) = Gu(n) \tag{2.44}$$

L'énergie résiduelle de prédiction est définie par [5-6]:

$$E_n = \sum_m e_n^2(m) = \sum_m \left(x_n(m) - \sum_{k=1}^{p} \alpha_k x_n(m-k) \right)^2 \tag{2.45}$$

où $x(m)$ est un segment du signal vocal autour de l'échantillon n. Les limites de somations sont finies et choisies de telle sorte que le signal vocal reste stationnaire à l'intérieur d'eux.

On trouve les valeurs $\hat{\alpha}_k$ qui minimisent E_n posant $\partial E_n / \partial \alpha_i = 0$ pour $i=1,2, \ldots,p$ ce qui conduit vers les système [5-6] :

$$\sum_m x_n(m-i)x_n(m) = \sum_{k=1}^{p} \hat{\alpha}_k \sum_m x_n(m-i)x_n(m-k) \qquad i = 1,\ldots,p \tag{2.46}$$

Les valeurs $\hat{\alpha}_k$ sont uniques ; on va le nommer α_k et avec la notation :

$$\phi_n(i,k) = \sum_m x_n(m-i)x_n(m-k) \tag{2.47}$$

on a [6] :

$$\sum_{k=1}^{p} \alpha_k \phi_n(i,k) = \phi_n(i,0) \quad i = 1,\ldots p \tag{2.48}$$

L'énergie résiduelle de prédiction peut être exprimée sous la forme [6] :

$$\begin{aligned} E_n &= \sum_m x_n^2(m) - \sum_{k=1}^{p} \alpha_k \sum_m x_n(m)x_n(m-k) \\ &= \phi_n(0,0) - \sum_{k=1}^{p} \alpha_k \phi_n(0,k) \end{aligned} \tag{2.49}$$

Pour trouver les coefficients de prédiction linéaire optimaux il faut tout d'abord calculer les valeurs de $\phi_n(i,k)$. Les limites de la sommation peuvent être choisies des façons différentes ce qui va donner naissance à des systèmes linéaires semblables mais pas identiques.

2.10.1 La méthode d'autocorrélation

Dans la méthode d'autocorrélation on suppose que le segment analysé x_n est nul à l'extérieur de l'intervalle de 0 à N. Comme en réalité cette supposition n'est pas vraie, on essaye de la corriger en utilisant une fenêtre adoucie aux extrémités, donc :

$$x_n(m) = x(m+n)w(m) \tag{2.50}$$

où $w(m)$ est une fenêtre de longueur finie N et $\phi_n(i,k)$ peut être exprimé [5-6] :

$$\phi_n(i,k) = \sum_{m=0}^{N-1-(i-k)} x_n(m)x_n(m+i-k) \qquad i = 1,..,p; \qquad k = 0,...,p \tag{2.51}$$

On peut montrer que dans ce cas $\phi_n(i,k)$ est identique à la fonction d'autocorrélation court terme (2.14) évaluée en *i-k* et comme la fonction d'autocorrélation est une fonction paire on a [6] :

$$\phi_n(i,k) = R_n(|i-k|) \quad i = 1,..,p; \qquad k = 0,...,p \tag{2.52}$$

Le set d'équations (2.48) devient [6] :

$$\sum_{k=1}^{p} \alpha_k R_n(|i-k|) = R_n(i) \qquad i = 1,...p \tag{2.53}$$

Ce sont les équations normales, dites de Yule-Walker pour lesquelles la matrice des coefficients est une matrice Teoplitz. L'énergie résiduelle de prédiction (2.49) devient [5-6] :

$$E_n = R_n(0) - \sum_{k=1}^{p} \alpha_k R_n(k) \tag{2.54}$$

2.10.1.1 Algorithme de résolution pour la méthode d'autocorrélation

Pour résoudre le système linéaire d'ordre p (2.53), qui requiert l'inversion d'une matrice d'ordre p, les méthodes classiques nécessitent un nombre d'opérations arithmétiques de l'ordre p^3. La structure particulière de la matrice des coefficients permet aux algorithmes de résolution de réduire le nombre d'opérations nécessaire à p^2.

L'un des algorithmes les plus connus utilisé pour résoudre ce système est la méthode de Levinson-Durbin [3,5-8] :

$$E^{(0)} = R(0) \tag{2.55}$$

$$k_i = \frac{\left(R(i) - \sum_{j=1}^{i-1} \alpha_j^{(i-1)} R(i-j)\right)}{E^{(i-1)}} \quad i = 1,...p \tag{2.56}$$

$$\alpha_i^{(i)} = k_i \tag{2.57}$$

$$\alpha_j^{(i)} = \alpha_j^{(i-1)} - k_i \alpha_{i-j}^{(i-1)} \quad j = 1,...i-1 \tag{2.58}$$

$$E^{(i)} = (1 - k_i^2) E^{(i-1)} \tag{2.59}$$

Le ensemble d'équations (2.70) à (2.74) est résolu pour i=1,2 ,…p et la solution finale est :

$$\alpha_j = \alpha_j^{(p)} \qquad j = 1,..., p \tag{2.60}$$

L'algorithme trouve ainsi tous les coefficients de prédiction linéaire pour les systèmes inférieurs à l'ordre p.

2.10.2 La méthode de covariance

Une deuxième approche pour choisir les limites de sommation pour la fonction d'autocorrélation est la méthode dite de covariance [3,5-6]. Dans ce cas on utilise dans le calcul de $\phi_n(i,k)$ p échantillons à l'extérieur de l'intervalle de 0 à N, donc la fonction d'autocorrélation court terme modifiée (2.15). Dans ce cas on n'a pas besoin d'une fenêtre adoucie alors que tous les échantillons à l'extérieur de l'intervalle de 0 à N sont maintenant disponibles.

La résolution efficace du système résultant implique la décomposition Cholesky pour la matrice des coefficients qui est une matrice symétrique positive définie.

2.10.3 Le gain du modèle

Le gain G du modèle est estimé en imposant la condition d'égalité entre l'énergie de l'excitation et l'énergie résiduelle de prédiction [6] :

$$G^2 \sum_{m=0}^{N-1} u^2(m) = \sum_{m=0}^{N-1} e^2(m) = E_n \tag{2.61}$$

Dans le cas où l'ordre p du filtre serait assez élevé pour modéliser l'effet du larynx, du conduit vocal et des lèvres, et si on suppose l'excitation comme étant un bruit blanc pour les sons non voisés et un train d'impulsions périodiques pour les sons voisés, on a [6] :

$$G^2 = R_n(0) - \sum_{k=1}^{p} \alpha_k R_n(k) \tag{2.62}$$

2.10.4 Discussion des méthodes d'analyse

Dans le cas d'une fréquence d'échantillonnage fixe de 8000 Hz il nous reste encore une série de paramètres à choisir pour établir les conditions optimales d'analyse.
L'ordre de prédiction p est indépendant de la méthode d'analyse mais en relation avec la fréquence d'échantillonnage. L'excitation glottique et l'effet de radiation des lèvres nécessitent 4 pôles et si on assure une paire de pôles par chaque kHz de bande passante on a donc $p = 12$. De plus l'examen de l'évolution de l'énergie résiduelle en fonction de l'ordre p (figure 17) montre une diminution négligeable au-delà de $p = 14$.

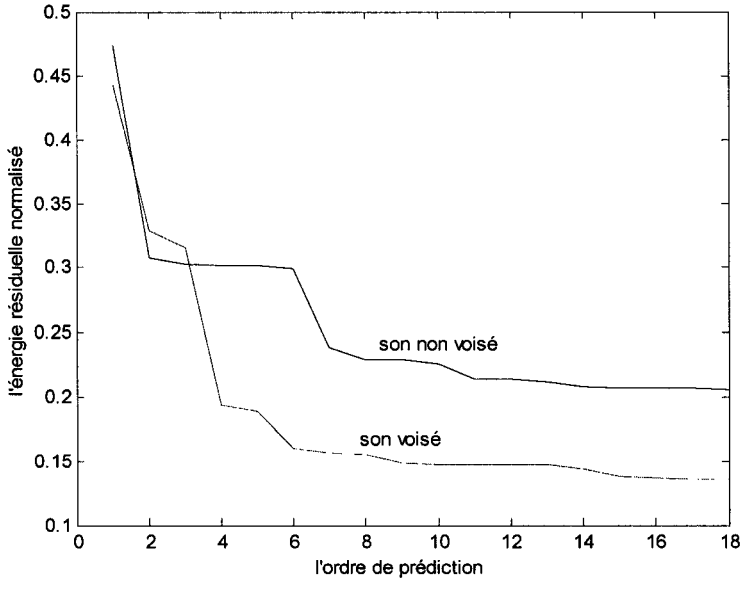


Figure 17 L'énergie résiduelle en fonction de l'ordre de la prediction [5-6]

La durée des segments d'analyse dépend de la méthode et des conditions d'analyse. Pour la méthode d'autocorrélation la fenêtre de pondération doit contenir plusieurs périodes

du fondamental pour les sons voisés. Dans la pratique on utilise couramment des fenêtres de 30 ms décalées de 10 ms donc N = 240 et L = 80 [5-6].

Pour la méthode de covariance on peut effectuer une analyse synchrone sur des segments alignés aux impulsions du fondamental pour les sons voisés. Cela nous permet une analyse sur des intervalles plus courts de temps de 2 à 5 ms. On peut ainsi suivre de plus près l'évolution des caractéristiques du signal. L'analyse synchrone exige une localisation précise du début du chaque période du pitch, chose qui peut s'avérer ardue et pour cela en pratique l'analyse est principalement asynchrone [3].

La pré-accentuation du signal se réalise par un filtrage qui accentue la partie haute fréquence du spectre. Le filtre utilisé est de transmitance $1 - \mu z$ avec $\mu = R_n(0)/R_n(1)$ pour chaque segment mais lorsque la valeur de μ n'est pas critique en pratique on utilise $\mu \approx 0.95$. Le but de ce pré-traitement est d'assurer un bon conditionnement des algorithmes de résolution [3,5].

Le choix de la méthode d'analyse a des répercussions directes sur le coût du traitement numérique exprimé en terme d'opérations arithmétiques élémentaires. Un exemple est donné dans le Tableau III.

Tableau III

Nombre d'opérations par analyse [3]

Méthode	Autocorrélation	valeur numériques pour $N = 240$ et $p = 12$
Pondération	N	240
Coefficients	$(N\text{-}p/2)(p+1)$	3042
Résolution	$p(p+1)$	156
Méthode	Covariance	
Pondération	0	0
Coefficients	$N(p+1)$	3120
Résolution	$1/3p^3+3/2p^2+6p$	864

2.10.5 Analyse spectrale basée sur le modèle autorégressif

L'énergie résiduelle de prédiction peut être exprimée dans le domaine temporaire comme suit [6] :

$$E_n = \sum_{m=0}^{N+p-1} e_n^2(m) \tag{2.63}$$

Une fois les coefficients de prédiction linéaire α_k obtenue pour un ordre p donné, si on considère $X_n(e^{j\omega})$ la transformée de Fourier du segment de signal vocal x_n et [6] :

$$A(e^{j\omega}) = 1 - \sum_{k=1}^{p} \alpha_k e^{-j\omega k} \tag{2.64}$$

on peut utiliser le théorème de Parseval pour exprimer l'énergie résiduelle de prédiction dans le domaine fréquentiel [6] :

$$E_n = \frac{1}{2\pi} \int_{-\pi}^{\pi} |X_n(e^{j\omega})|^2 |A(e^{j\omega})|^2 d\omega \tag{2.65}$$

dans le modèle de production de la parole adopté on a [6]:

$$H(e^{j\omega}) = \frac{G}{A(e^{j\omega})} \tag{2.66}$$

et donc :

$$E_n = \frac{G^2}{2\pi} \int_{-\pi}^{\pi} \frac{|X_n(e^{j\omega})|^2}{|H(e^{j\omega})|^2} d\omega \tag{2.67}$$

Cette relation nous montre que lorsqu'on minimise E_n on minimise le rapporte entre l'énergie du spectre du signal original et l'énergie du spectre du signal synthétique produit par le modèle adopté. De plus, quand le gain du modèle respecte la relation (2.62), le signal original et le signal synthétique ont la même fonction d'autocorrélation pour les premières p+1 valeurs. Donc le spectre du signal synthétique est une meilleure approximation du spectre du signal original quand l'ordre de prédiction est élevé. Il faut remarquer toutefois que, lorsque la fonction de transfert qui caractérise le modèle est à déphasage minimal, l'égalité entre les spectres du signal synthétique et original n'implique pas l'égalité entre la réponse fréquentielle du modèle et la transformée de Fourier du signal original.

On appelle le spectre du modèle le carré du module de sa transmitance [3] :

$$S_{\tilde{x}}(\omega) = |H(e^{j\omega})|^2 = \left|\frac{G}{A(e^{j\omega})}\right|^2 \tag{2.68}$$

Si l'excitation du modèle est le bruit blanc de variance unité, le spectre du modèle coïncide avec la densité d'énergie du signal synthétique. Le spectre du modèle peut être calculé par l'application de la TFR au vecteur des coefficients de prédiction prolongé par des zéros jusqu'à la longueur nécessaire pour la résolution spectrale souhaitée.

Dans l'équation (2.67) on peut voir que les régions où $|X_n(e^{j\omega})| > |H_n(e^{j\omega})|$ ont une influence plus importante dans le calcul de E_n par rapport aux régions où $|X_n(e^{j\omega})| < |H_n(e^{j\omega})|$. Le spectre du modèle AR approche mieux les pics, donc les formants dans le cas des sons voisés, que les vallées du spectre S_x [5-6]. On remarque cet effet dans la figure 18 où le spectre de la voyelle / u / a été obtenu par la TFD et en utilisant la modélisation AR.
La TFR a été calculé en 512 points pour une fenêtre de Hanning de 32 ms. Pour le spectre du modèle AR on a utilisé la méthode d'autocorrélation et un ordre $p = 24$.

L'ordre de prédiction du modèle AR contrôle le degré de lisage du spectre du modèle. Chaque paire de pôles permet l'existence d'un pic dans le spectre du modèle et donc plus on augmente l'ordre de prédiction plus le modèle est capable de suivre de plus près les détails du spectre du signal originel.

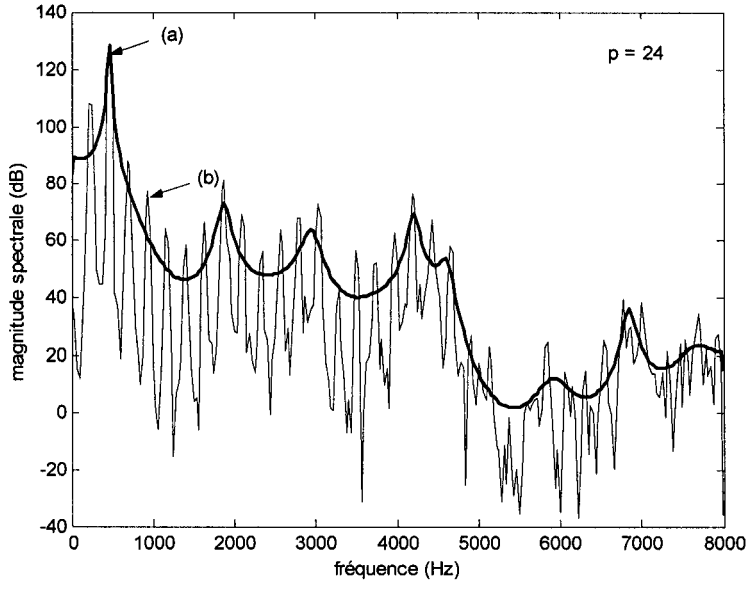


Figure 18 Le spectre du modèle AR versus le module de la TFD [6]

2.11 Propriétés statistiques du signal vocal

Dans l'étude du signal vocal il est parfois utile de le considérer comme une réalisation particulière d'un processus aléatoire non stationnaire. Lorsque le signal est échantillonné à une fréquence compatible avec le théorème de Shannon, l'estimation de sa statistique peut être faite sur les échantillons. A cause de sa non-stationnarité, les propriétés statistiques doivent être estimées sur des intervalles longs de temps, plusieurs secondes, et moyennées pour plusieurs locuteurs pour obtenir ce qui on appelle statistique longue-terme.

La densité de probabilité du signal $p(x)$ est estimée à l'aide de l'histogramme des amplitudes. Les expériences ont montré que la distribution Gamma est très voisine de la densité de probabilité du signal vocal $p(x)$ [3,6]. Une approximation acceptable est la distribution de Laplace dont l'expression est plus simple à utiliser [3,6].

La figure 19 compare le résultat d'une estimation faite sur un segment de parole d'environ 7.5 s avec quelques lois de répartition usuelles dont les expressions sont données par la suite. Dans la figure les lois de répartition sont normalisées pour avoir une moyenne nulle et une variance unitaire.

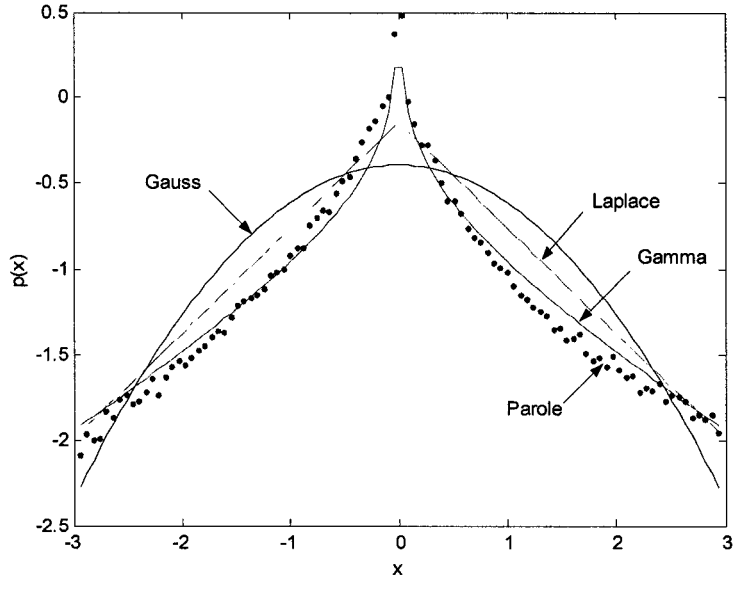


Figure 19 Lois de répartition usuelles est densité de probabilité long terme du signal vocal

Tableau IV

Lois de répartition usuelles [3]

Lois de répartition	Expression analytique pour la moyenne nulle
Gaussienne	$\frac{1}{\sqrt{2\pi\delta_x^2}}\exp\left[\frac{-x^2}{2\delta_x^2}\right]$
Laplace	$\frac{1}{\sqrt{2}\delta_x}\exp\left[\frac{-\sqrt{2}\,\lvert x\rvert}{\delta_x}\right]$
Gamma	$\left[\frac{\sqrt{3}}{8\pi\delta_x\lvert x\rvert}\right]^{1/2}\exp\left[\frac{-\sqrt{3}\,\lvert x\rvert}{2\delta_x}\right]$

On définit également une statistique court-terme estimée sur des segments temporels inférieurs à 30 ms pendant lesquels le signal est quasi stationnaire. Les expériences utilisant diverses testes statistiques montrent que pour des segments de temps de 5 ms à 0.5 s la distribution de Laplace est la meilleure approximation pour la densité de probabilité du signal. Pour des segments de temps plus longues les expériences favorisent la distribution Gamma, comme dans le cas de statistique à longue terme [11].

Le Tableau V présente le résultat moyen du test statistique χ^2 (annexe 2), effectué sur un fragment de parole de 20 s en utilisant plusieurs fonctions de densité de probabilité et plusieurs longueurs de trame [11].

Tableau V

Résultat du test statistique χ^2 pour plusieurs longueurs de trame [11]

Longueur de trame	Gamma	Laplace	Gaussienne
2.5 ms	113	78	68
5 ms	140	96	124
10 ms	192	127	245
20 ms	284	181	562
50 ms	496	350	4502
100 ms	725	717	26477
200 ms	1013	926	31477
0.5 s	1739	1726	72665
1 s	2910	2971	50886
2 s	5525	5773	46000
5 s	14750	15692	80330

Les mêmes expériences prouvent que pour les intervalles de silence la distribution Gaussienne est la plus près de la densité de probabilité du bruit.

Dans un deuxième pas les segments du signal vocal sont transformés en composantes non corrélées et analysées du point de vue statistique dans ces nouveaux domaines. Les transformées couramment utilisées sont TFD, TCD et la transformée de Karhunen Loève TKL. Les résultats des tests statistiques sur ces composantes assimilés à des variables aléatoires convergent vers la même conclusion que dans le cas du domaine temporaire : la distribution de Laplace est la meilleure approximation pour la plus grande partie d'entre eux [11].

CHAPITRE 3

LA DÉTECTION D'ACTIVITÉ VOCALE

3.1 Préambule

Un problème très souvent rencontré dans le traitement du signal vocal est la détection d'activité vocale VAD. Autrement dit, il faut discriminer entre les régions où la parole est présente et les régions où la parole est absente dans le signal analysé. Un algorithme de détection d'activité vocale fonctionne d'après une logique binaire. Il produit les valeurs logiques 1 ou 0 pour chaque segment de signal analysé, indiquant respectivement la présence ou l'absence de la parole.
Le VAD est un module important dans une large gamme d'applications concernant le traitement de la parole soit la reconnaissance, la transmission ou le rehaussement de la parole.

Dans le domaine de reconnaissance de la parole le VAD est utilisé pour localiser le début et la fin des régions à reconnaître. La précision du VAD utilisé se matérialise dans une amélioration du taux de reconnaissance.

Pour les systèmes de transmission de la parole telle que la téléphonie cellulaire, le taux d'activité vocale est en moyenne de 40%, donc 60% du temps le système serait inutilisé [12]. Le VAD est utilisé pour contrôler la transmission discontinue qui active la transmission uniquement pendant les périodes d'activité vocale. La transmission discontinue permet d'augmenter la capacité du système pour l'opérateur et pour l'abonné prolonge l'autonomie du mobile [12].

Dans le cas du rehaussement de la parole les périodes de silence détectés par le VAD peuvent servir à actualiser le paramètre du bruit.

La tâche d'un algorithme de détection d'activité vocale est loin d'être facile sauf pour le cas d'un rapport entre le signal vocal et le bruit RSB très élevé quand l'énergie du plus faible son excède l'énergie du bruit du fond et juste une mesure de l'énergie suffit pour la décision. Toutefois cette condition est loin d'être réalisable dans des applications réelles de traitement de la parole où cette approche simple est inefficace. De plus le concepteur du système n'a pas des connaissances a priori sur la nature du bruit ou sur le RSB qui peut changer, d'une façon aléatoire, d'une application à l'autre et avec le temps dans le cadre d'une même application. On peut conclure que la nature non stationnaire et la grande variété des bruits de fond et du signal vocal auquel s'ajoute un RSB inconnue et parfois variable rendent le problème de détection d'activité vocale difficile. Alors l'ingénieur doit faire preuve d'inventivité pour résoudre ce problème dans les conditions spécifiques pour chaque application.

3.2 L'effet du bruit dans un VAD

Comme on vient de le voir, la plus grande difficulté pour un algorithme de détection d'activité vocale repose sur le bruit qui s'ajoute au signal vocal. Dans les applications réelles il y a plusieurs types de bruit qui interviennent parmi lesquels on compte certains bruits électriques, le bruit de quantification, le bruit ambiant.

Le rapport signal bruit est défini par le logarithme du rapport entre la variance du signal et celle du bruit [3] :

$$RSB = 10\log\frac{\delta_x^2}{\delta_b^2} \qquad [dB] \tag{3.1}$$

3.2.1 Bruits électriques

Pour éliminer l'effet dû au bruit de 60 Hz du réseau d'alimentation et à tout autre niveau de courant continu, le signal est passé par un filtre passe-haut dont la fréquence de coupure est de 100 Hz. Ce filtrage ne devrait pas affecter le signal vocal car, comme on a déjà vu , généralement il y a peu d'information utile au-dessous de cette fréquence. De plus la pré-accentuation du signal qui consiste en un filtrage passe haut est une technique répandue dans le traitement du signal vocal pour accentuer les hautes fréquences [3,5]

Le bruit du microphone est négligeable lorsqu'on utilise un microphone de qualité caractérisé par un RSB supérieur à 60 dB [4].

3.2.2 Le bruit de quantification

La représentation numérique du signal implique la quantification de chaque échantillon selon un nombre fini de valeurs discrètes. Dans ce qui suit, on traitera la quantification uniforme instantanée.

Une loi de quantification sans mémoire ou instantanée est définie par L+1 niveaux de décision. Pour toutes les amplitudes comprises dans l'intervalle [$x(i$-1) $x(i)$] on fait coresponsable une valeur quantifiée $y(i)$ située au milieu de l'intervalle. Les niveaux de saturation $\pm x_s$ sont les niveaux extrêmes quantifiables à l'extérieur desquels on a la situation de dépassement.

La différence entre deux niveaux de décision successifs est appelée pas de quantification et notée ε. Lorsque le nombre de valeurs quantifiables est une puissance de 2 soit $L = 2^b$, chaque valeur quantifiée est représentée par un mot de b bits quand l'erreur de quantification vaut [3] :

$$\varepsilon = \frac{2x_s}{L} \tag{3.2}$$

Comme le signal vocal est un signal aléatoire, l'erreur de quantification est aussi un signal aléatoire. On parle donc de bruit de quantification ou de granulation dont les propriétés seront établies par des méthodes statiques.

Le facteur de charge Γ du quantificateur est défini par le rapport [3] :

$$\Gamma = \frac{x_s}{\delta_x} \tag{3.3}$$

où δ_x est l'écart type du signal. Pour un signal gaussien la probabilité de dépassement est inférieure à 0.0027 pour $\Gamma = 3$ [3].

Pour un signal aléatoire gaussien et un rapport δ_x/ε supérieur à 4 la corrélation entre l'erreur de quantification et le signal est négligeable. L'erreur de quantification est un bruit blanc de répartition uniforme, de moyenne nulle et dont la variance est égale à $\varepsilon/12$ ou si on utilise (3.2).

$$\delta_e = \frac{x_s}{3} 2^{-2b} \tag{3.4}$$

Le RSB en l'absence de dépassement vaut par conséquent [3,8] :

$$RSB = 6.02b + 4.77 - 20\log\Gamma \qquad [dB] \tag{3.5}$$

Pour un facteur de charge constant le RSB augmente avec 6 dB pour chaque bit de plus disponible. Dans le cas d'une quantification sur 16 bits et $\Gamma = 5$ on a RSB = 87 dB [3].

En cas de dépassement le RSB est dégradé car [3] :

$$RSB = 10\log\frac{\delta_x^2}{\delta_b^2 + \delta_D^2} \tag{3.6}$$

ou δ_D est la variance du bruit de dépassement. Cette dégradation dépend principalement de la loi de distribution du signal x. Dans le cas d'un signal aléatoire qui respecte une distribution de Laplace, telle que le signal vocal, une quantification sur 16 bits permet un RSB d'environ 77 dB quand $\Gamma = 17$ [3].

Des meilleurs RSB sont obtenus lorsque la loi de quantification est adaptée à la densité de probabilité du signal. En effet l'erreur de granulation est réduite pour les amplitudes du signal les plus probables [3].

3.2.3 Le bruit ambiant

Lorsque l'acquisition signal vocal se fait dans des applications réelles il y aurait toujours un bruit de fond dû aux autres sources de signal sonore existantes dans le milieu d'enregistrement. Ce bruit ambiant additif contamine le signal de parole et rend difficile l'extraction des caractéristiques propres au signal utile. Si on regarde le tableau II et la figure 8 on s'aperçoit vite que même pour des conditions d'utilisation normales, bureau ou rue, le RSB entre la parole et le bruit ambiant varie à la limite de 30 dB à –30 dB. On peut essayer d'augmenter le RSB par diverses techniques. Par exemple on peut exploiter les caractéristiques de directivité du microphone et réduire la distance entre le microphone et la source sonore d'intérêt avec le risque d'introduire certains artefacts

comme l'effet de la respiration etc. Généralement ces techniques dépendent d'utilisateur et leur effet est limité et difficile à quantifier.

De plus le RSB est une mesure globale qui ne tient pas compte de la nature forte non stationnaire du signal vocal qui présente lui-même une gamme dynamique d'environ 40 dB. Si on regarde de plus près ce qui se passe quand on ajoute un bruit stationnaire avec un RSB constant, à un signal vocal formé d'une phrase entière, on peut observer que le RSB pour chaque trame de signal peut varier beaucoup par rapport au RSB global. La figure 20 présente l'évolution du RSB pour chaque trame de signal dans le cas d'un signal vocal formé d'une phrase et des régions de silence contaminé par un bruit rose avec un RSB global de 5 dB. De plus on a marqué différemment les trames du signal vocal contenant voyelles et consonnes.

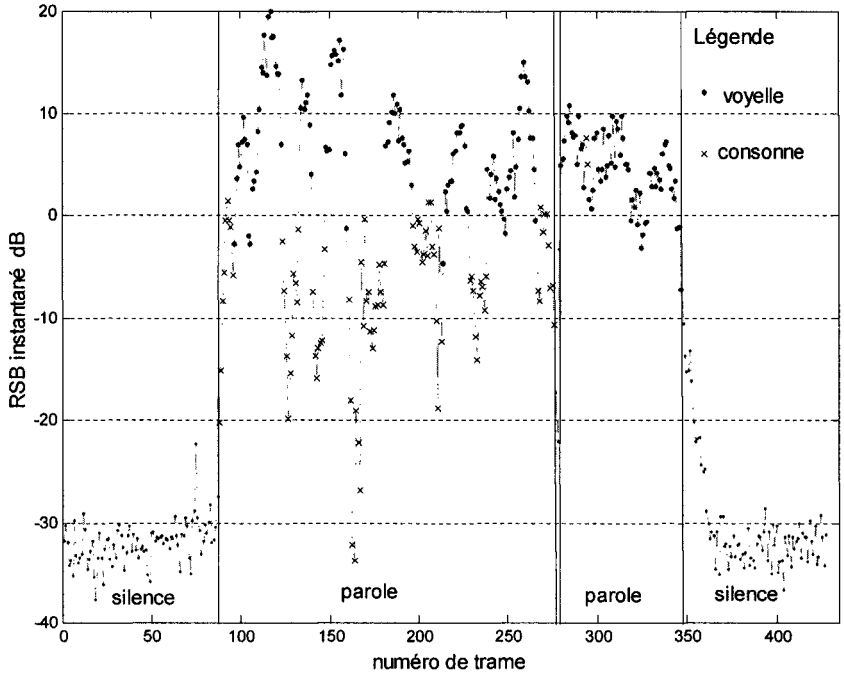


Figure 20 L'évolution du RSB instantané pour un signal vocal bruité

On observe que même si le RBS global est de 5 dB, il y a des trames pour lesquelles les RSB est beaucoup plus bas. Grâce à leur énergie les voyelles se différentient nettement des consonnes et elles imposent le RSB global. Dans la figure 20 on peut remarquer que, lorsque pour les voyelles le RSB est situé autour de la moyenne de 5 dB, pour les trames contenant consonnes le RSB est inférieure à 0 dB allant jusqu'à -20 dB et même -30 dB. Il est clair que la détection des ces dernières serait beaucoup plus difficile pour un algorithme de VAD.

3.3 Revue des algorithmes utilisés dans la détection d'activité vocale

Au fils des années plusieurs concepts on été utilisés et un grand nombre d'algorithmes ont été proposé et même aujourd'hui, après plus de trente ans de recherches, ce problème reste d'actualité [10-29].

La discrimination entre le bruit de fond et le signal vocal peut être vue comme un problème de reconnaissance des formes [10,14]. Une approche possible pour résoudre ce problème est d'analyser une série de paramètres suffisamment invariables spécifiques au bruit et d'essayer de découvrir certains changements dans leur évolution qui pourraient indiquer l'activité vocale.

Généralement un algorithme de VAD utilise un intervalle d'analyse d'une durée de 10 à 30 ms et dans une première étape sont obtenus les paramètres de la chaque trame. Dans une deuxième étape les paramètres associés à la trame courante sont comparés avec les paramètres qui caractérisent le bruit en utilisant une logique spécifique à chaque algorithme. Les paramètres du bruit soit sont supposées connus a priori soit sont estimés à partir des premières trames du signal. Suite à cette comparaison une première décision est prise. En fonction de chaque algorithme la décision initiale est lissée pour tenir compte de la forte corrélation qui existe au niveau de l'occurrence de la parole ou du

silence entre deux trames consécutives de signal. Une dernière étape permet à l'algorithme d'actualiser les paramètres du bruit en utilisant les trames de silence antérieures.

Évidemment ce qu'on cherche est un algorithme précis, robuste par rapport au bruit et qui demande un minimum de calcul. Une mesure des performances d'un algorithme de VAD est donnée par l'ensemble de paramètres : probabilité de détections P_d et probabilité de fausse alarme P_f rapporté à une décision idéale. P_d représente le ratio entre le nombre de trames contenant signal vocal correct classifié et le nombre réel de trames de parole. P_f est le ratio entre le nombre de trames de silence incorrect classifié par l'algorithme et le nombre réel de trames de silence [14]. La décision idéale de référence est obtenue par un marquage manuel des régions de silence et de parole pour le signal non bruité.

Dans ce qui suit quelques algorithmes représentatifs seront exposés afin de mettre en évidence la problématique reliée à ce sujet.

3.3.1 VAD basé sur l'énergie court terme et le taux de passage par zéro [10]

L'idée de base est d'utiliser l'estimé de l'énergie court terme M_n comme un paramètre robuste pour découvrir les régions voisées. La décision est raffinée par la suit pour inclure les régions non voisées à l'aide d'un deuxième paramètre, le taux de passage par zéro Z_n.

Pour cela l'analyse est basée sur une trame de 10 ms. Les 100 premières ms sont considérées juste du bruit et sont utilisées pour calculer la moyenne $\overline{IZC}$ et l'écart type δ_{IZC} du taux de passage par zéro, et la moyenne de l'estimateur de l'énergie court-terme *IMN*. Le seuil minime de passages par zéro pour les régions non voisées est choisi d'après la relation :

$$\begin{aligned} IZTC &= MIN(IF, \overline{IZC} + 2\delta_{IZC}) \\ IF &= 0.25 \cdot longueur\ de\ w \end{aligned} \tag{3.7}$$

IF est une constante, dépendante de la longueur de la fenêtre d'analyse en nombre d'échantillons. On définit encore deux seuils d'énergie ITL et ITU :

$$\begin{aligned} I1 &= 0.03(IMX - IMN) + IMN \\ I2 &= 4IMN \\ ITL &= MIN(I1, I2) \\ ITU &= 5ITL \end{aligned} \tag{3.8}$$

ou *IMX* est le maximum de l'énergie court terme pour le signal entier.

L'algorithme cherche tout d'abord les régions pour lesquelles l'énergie dépasse la valeur *ITL* et après *ITU* sans retomber au-dessous de *ITL*. Cela donne un premier indice N_1 du debout de l'activité vocale. Un premier indice de la fin de l'activité vocale N_2 est quand l'énergie court terme devient inférieur à *ITL* après avoir dépassé *ITU*.

A partir de N_1 on se déplace 250 ms vers le début du signal et on regarde cette fois le taux de passage par zéro. Si Z_n dépasse trois fois ou plus le seuil *IZTC* le point N_d où *IZTC* a été dépassé pour la dernière fois est le début de l'activité vocale si non la décision initiale reste inchangée. La même procédure s'exécute à partir de N_2 vers la fin du signal pour trouver l'indice final N_f de la fin de l'activité vocale.

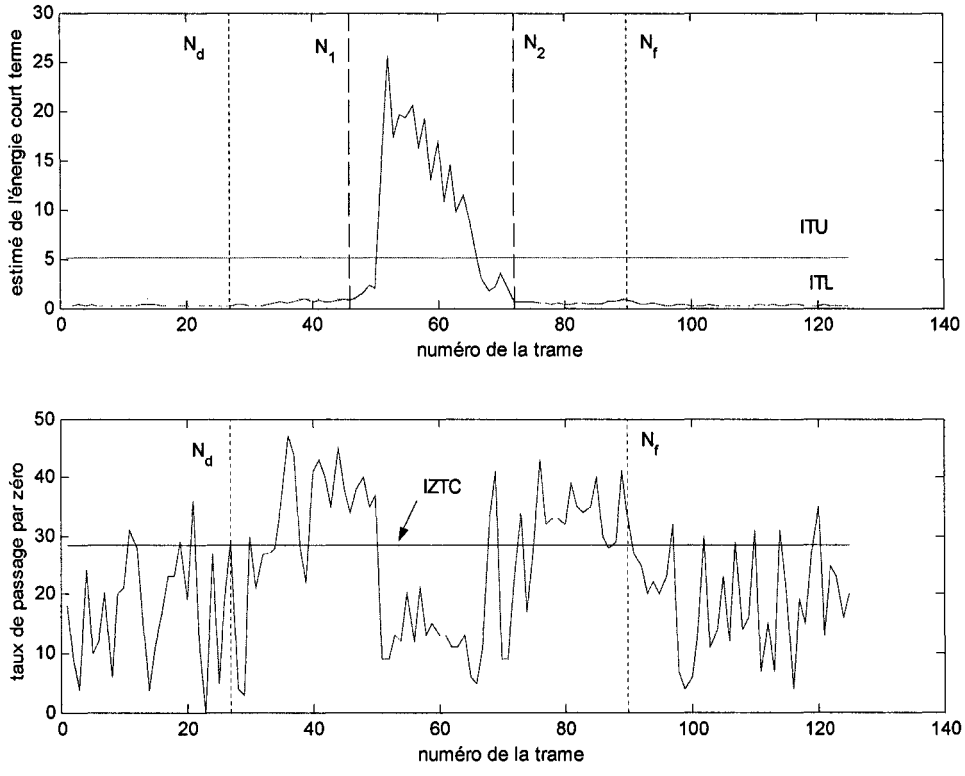


Figure 21 VAD basé sur l'énergie court terme et le taux de passage par zéro [10]

La figure 21 exemplifie le résultat obtenu avec cet algorithme pour le mot *six* qui est un exemple de mot qui commence et se termine avec un son non voisé. L'intervalle $[N_1, N_2]$ correspondant à la voyelle / i / est découvert par l'algorithme grâce au paramètre M_n. La décision finale $[N_d, N_f]$ qui inclut les deux consonnes / s / au début et à la fin du mot est prise en se basant sur le deuxième paramètre Z_n.

Cet algorithme se comporte bien pour un RBS supérieur à 30 dB quand le paramètre Z_n est fiable, pour des RSB plus petits, la valeur de Z_n est vite corrompue par le bruit et conduit vers des conclusions erronées. Des plus, l'algorithme ne s'adapte pas à

l'évolution du bruit, les valeurs des paramètres qui caractérisent le bruit déterminé au début de la période restent inchangées pour tout l'intervalle d'analyse.

3.3.2 VAD basé sur un filtrage optimale de l'énergie court-terme [18]

3.3.2.1 Conception du filtre optimal

Cet algorithme est inspiré par une de techniques utilisées pour la détection du contour dans le traitement des images [31]. La procédure est similaire au lissage médian sauf que cette fois les caractéristiques du filtre $f(y)$ utilisé sont optimisées pour :

- éliminer les effets du bruit
- être capable de détecter les débuts et les fins des régions d'intérêt
- avoir une longueur finie
- avoir le niveau de la réponse finie
- avoir une réponse maximale et précise dans le cas d'un changement dans l'évolution du paramètre utilisé
- minimiser la probabilité de fausse alarme

Suite à ces conditions on obtient un filtre antisymétrique de longueur finie qui décroît vers zéro aux extrémités. On utilise la méthode de multiplicateurs de Lagrange pour trouver les paramètres du filtre optimal. La solution est donnée par la relation [30] :

$$f(y) = e^{Ay}[K_1 \sin(Ay) + K_2 \cos(Ay)] + e^{-Ay}[K_3 \sin(Ay) + K_4 \cos(Ay)] + K_5 + K_6 e^{sy} \quad (3.9)$$

où $A = 0.41s$. Pour une longueur du filtre $W = 7$ et $s = 1$ on trouve K = [1.583, 1.468, -0.078, -0.036, -0.872, -0.56] [30]. Les coefficients du filtre sont obtenus en utilisant l'éqution [30] :

$$h(i) = \{-f(-W \le i \le 0), f(1 \le i \le W)\} \quad (3.10)$$

Pour la détection d'activité vocale une valeur W = 13 donne de bons résultats, le filtre optimal utilisé est présenté dans la figure 22.

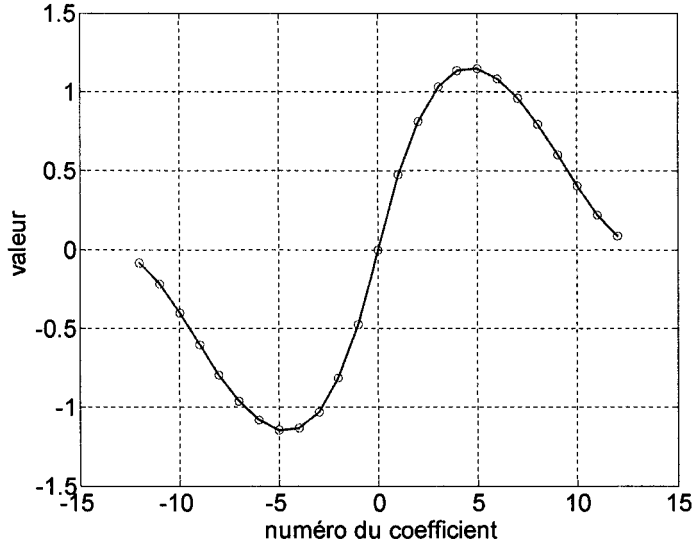


Figure 22 Filtre optimal W= 13 [30]

Le paramètre utilisé est le logarithme de l'énergie court terme et le détecteur d'activité vocale fonctionne comme dans le cas du filtrage médian, pour chaque trame on a :

$$F(n) = \sum_{i=0}^{2W} h(i-W) \log_{10} E(n+i) \tag{3.11}$$

3.3.2.2 Algorithme de décision

La valeur $F(n)$ doit être comparée à des seuils pré-déterminés et évalués par un diagramme à trois états pour obtenir la décision finale. Le diagramme de décision emploie trois états *silence*, *parole* et *quitter parole* et il est représenté dans la figure 23.

Au début on se trouve dans l'état *silence* pour $F(1)$. Les entrées dans le diagramme sont les valeurs de $F(n)$ et les sorties sont les points de début et de fin de l'activité vocale. Le compteur est un compteur de trames, T_U et T_L sont deux seuils $T_U > T_L$ et l'écart est un nombre entier qui indique le nombre de trames nécessaires pour passer dans l'état *silence* après la détection d'un point de fin de l'activité vocale. Les conditions de transition sont marquées sur le diagramme à côté des flèches indiquant la transition et les actions sont entre parenthèses. Le fonctionnement du diagramme est expliqué à l'aide d'un exemple. La figure 24 partie (A) présente l'énergie court terme du mot six et la partie (B) la sortie $F(n)$ du filtre optimal.

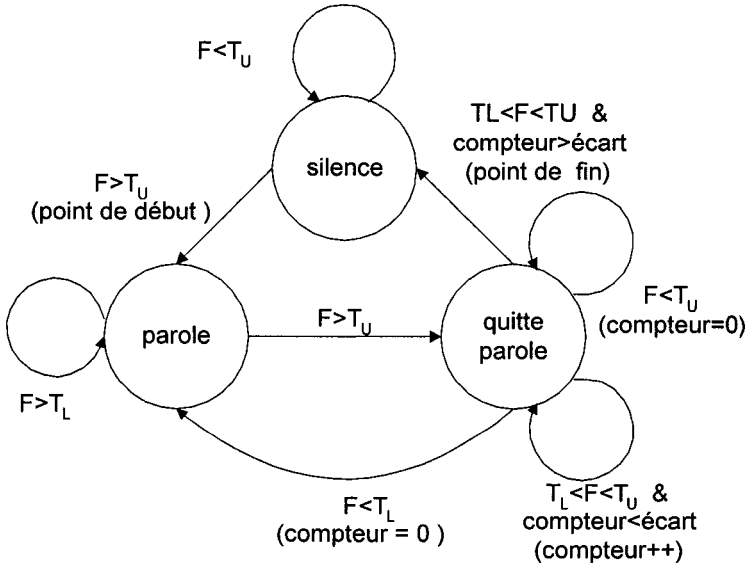


Figure 23 Diagramme de décision à trois états [10]

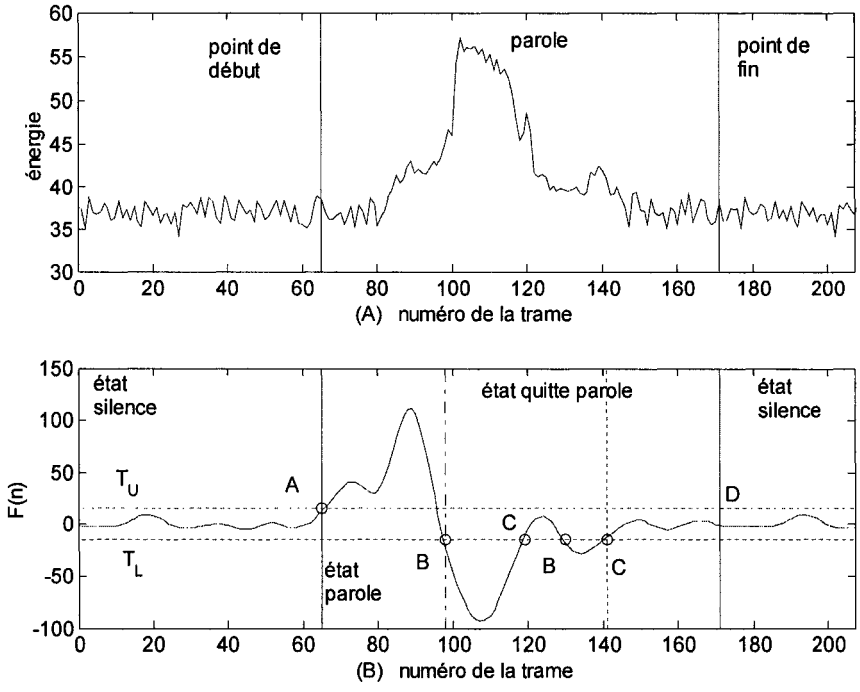


Figure 24 Exemple (A) énergie du mot six et (B) la sortie du filtre $F(n)$ [10]

Le diagramme des états reste dans l'état *silence* jusqu'au point A où $F(n)>T_U$ et un point de début d'activité vocale est détecté, l'état courant devient *parole*. Quand $F(n)<T_L$ au point B, l'état courant devient *quitte parole* et le compteur est tenu à zéro tant que $F(n)$ ne dépasse pas T_L. Au point C, $F(n)>T_L$ et le compteur est incrémenté tant que $T_U > F(n) > T_L$. Si l'écart est dépassé par la valeur du compteur, un point de fin d'activité vocale est détecté est le diagramme des états revient à l'état de départ. La valeur de l'écart est choisie égale à 30 et correspond à la période de descendant de l'énergie avant d'arriver à un point de fin d'activité vocale. Les valeurs de T_L et T_U sont choisies empiriquement, à partir de quelques exemples, l'algorithme est assez stable par rapport aux valeurs de T_L et T_U.

3.3.2.3 Observations

L'effet du bruit fait diminuer la gamme dynamique pour les valeurs de l'énergie et en conséquence les variations de $F(n)$. La sortie du filtre $F(n)$ reste plus longtemps entre les seuils T_L et T_U et donc les régions de silence seront élargies au détriment des régions de parole comme on peut voir dans la figure 25.

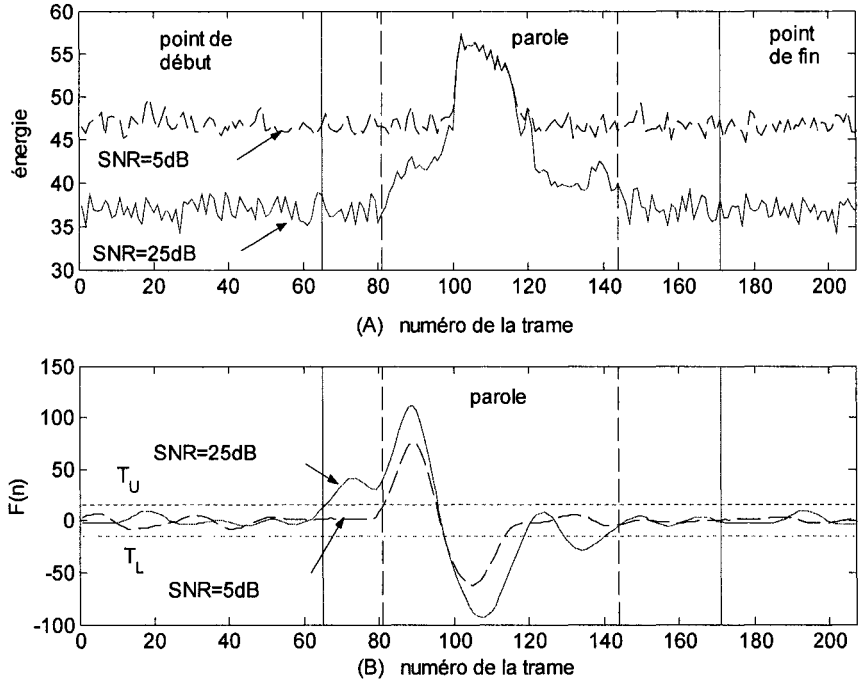


Figure 25 (A) énergie du mot six (B) la sortie du filtre $F(n)$

Malgré le fait qu'elle n'est pas très précise, cette méthode reste intéressante car elle demande un minimum de calculs et se comporte bien même dans le cas des bruits non stationnaires.

3.3.3 VAD basé sur l'analyse de l'énergie court-terme en sous bandes de fréquence [20]

3.3.3.1 Problématique

A partir de l'observation que les deux paramètres déjà utilisés pour la VAD, l'énergie court-terme et le taux de passage par zéro, ne sont pas suffisants pour une détection robuste même si l'on utilise des algorithmes élaborés pour la décision [17], un nouveau paramètre qui présente de meilleures caractéristiques a été proposé. Ce paramètre est la somme du logarithme de l'énergie court-terme lissé et normalisé et l'énergie du signal dans la bande de fréquence de 250 à 3500 Hz à son tour lissée et normalisée. Basé sur ce paramètre et certains seuils d'énergie, l'algorithme proposé [19] trouve tout d'abord les zones de signal vocal de haute énergie correspondant aux sons voisés. La décision finale est le résultat d'une procédure de raffinement qui utilise le taux de passage par zéro et des seuils de durée. Si les taux de passage par zéro moyenné pour une durée de 100 ms avant et 150 ms après les frontières déjà trouvés dépasse les taux de passage par zéro moyenné des 100 premières ms, ces régions sont classifiées comme étant des régions de parole. Sinon la décision initiale reste inchangée.

3.3.3.2 Définition des paramètres

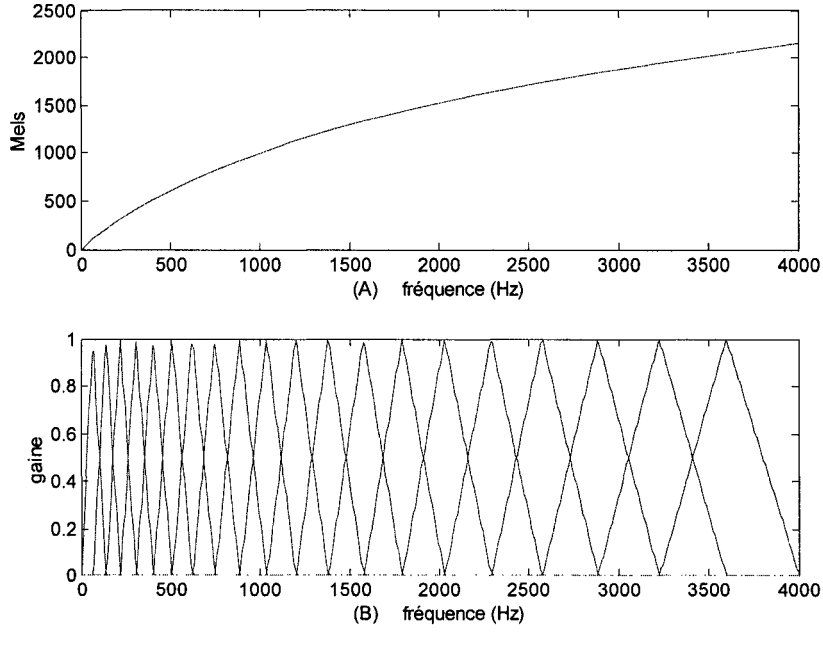
L'idée d'un paramètre composé est aussi à la base de l'algorithme proposé en [20] qui essaie de résoudre le problème de VAD dans le cas d'un RSB variable. Pour cela deux paramètres sont proposés.

Dans le cas d'un RSB variable l'énergie du bruit change en temps et les seuils basés sur l'analyse en début du signal doivent être adaptés pour tenir le pas avec l'évolution du bruit.

Pour modéliser le fait que l'oreille humaine perçoit les sons non-linéaires en rapport avec leur fréquence réelle, a été introduite l'échelle de mel qui est une mesure subjective pour la hauteur d'un son. La relation entre le mel et la fréquence est donnée par la relation [12] :

$$mel = 2595\log(1 + f/7000) \tag{3.11}$$

où f est la fréquence. Une façon d'obtenir ce spectre subjectif est d'utiliser une banque de filtres passe-bande de gain constant et qui ont une largeur de bande en rapport avec l'échelle de mel. Dans la figure 26 on pressente une banque de 20 filtres triangulaires conçue pour le domaine de fréquence de 0 à 4000 Hz. Chaque filtre est multiplié par le module de la TFR de la trame courante pour générer le spectre subjectif qui est donc calculé en 20 points pour chaque trame.



Figue 26 (A) la relation entre le mel et la fréquence (B) le banque de 20 filtres triangulaires utilisés pour obtenir le spectre subjectif de l'oreille [20]

On réalise un lisage temporel pour chacune de 20 bandes du spectre subjectif résultant utilisant un filtre médian en trois points. Le spectre ainsi obtenu est normalisé. Les valeurs moyennes pour chacune de 20 bandes de fréquence des 5 premières trames, considérées juste du bruit, sont soustraites de la bande correspondante pour le reste du signal.

$$X(n,i) = X_{lise}(n,i) - Noise = X_{lise}(n,i) - \frac{\sum_{j=0}^{4} X_{lise}(j,i)}{5} \tag{3.12}$$

où $X(n,i)$ est l'énergie lissée et normalisée de la i-ième bande de la n-ième trame. Etant donné que le spectre du bruit à été soustrait, on peut maintenant définir l'énergie du signal propre $E(i)$ pour chaque bande i, comme suit :

$$E(i) = \sum_{n=0}^{N-1} X(n,i) \tag{3.13}$$

où N est le nombre total des trames du signal.

Un exemple du spectre subjectif pour le mot six est représenté dans la figure 27. On a utilisé 128 points pour la TFR et une fenêtre temporelle de 15 ms.

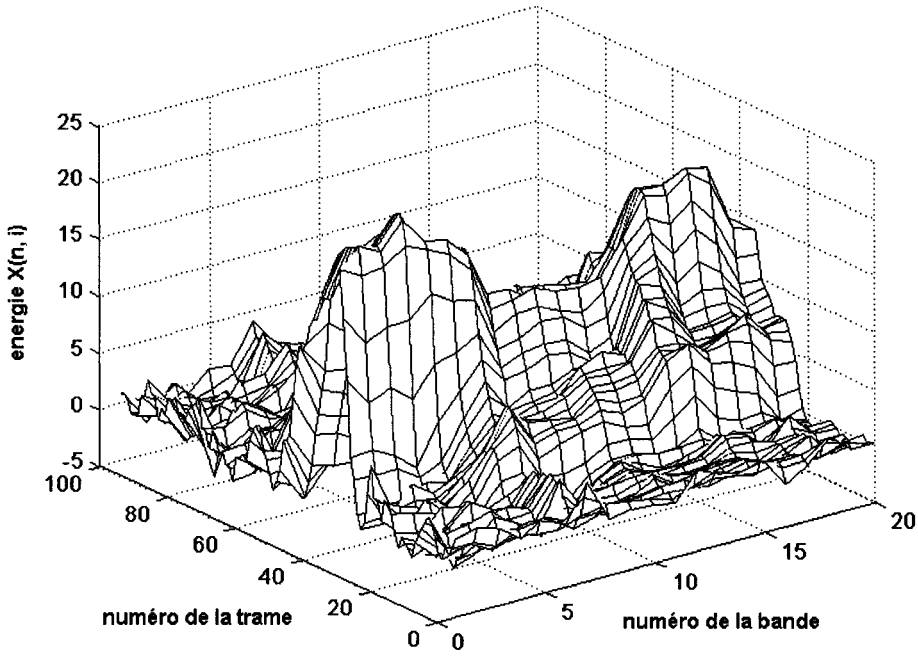


Figure 27 Exemple de spectre subjectif pour le mot six

Ensuite on effectue un tri de ces vingt valeurs de $E(i)$. On obtient un nouvel ensemble d'indices $I(i)$ pour lesquels $E(I(1))$ est la bande qui contient le plus d'énergie et $E(I(20))$ est la bande qui contient le moins d'énergie. Lorsque $E_{min} = E(I(20))$ contiennent le moins d'énergie du signal vocal, son évolution temporelle est un bon indicateur pour l'évolution temporelle du bruit. Le paramètre appelé *VAR* est une mesure de la variation du bruit pour tout le signal :

$$VAR = \frac{\sum_{n=0}^{N-1} | X(n, I(20)) |}{N} \tag{3.14}$$

Les essais ont montré que la plus grande partie de l'énergie du signal vocal est concentrée dans les six premières bandes du spectre subjectif trié. L'estimation de l'énergie du signal se fait donc à l'aide d'un nouveau paramètre, l'énergie temps-fréquence, *ETF* qui est la somme lissée de l'énergie du domaine temps T et des six premières bandes du spectre subjectif :

$$ETF(n) = lisse(T(n) + cF(n)) \qquad avec \quad F(n) = \sum_{i=1}^{6} X(n, I(i)) \tag{3.15}$$

où $c \cong 1.1$ est une constante de proportionnalité.

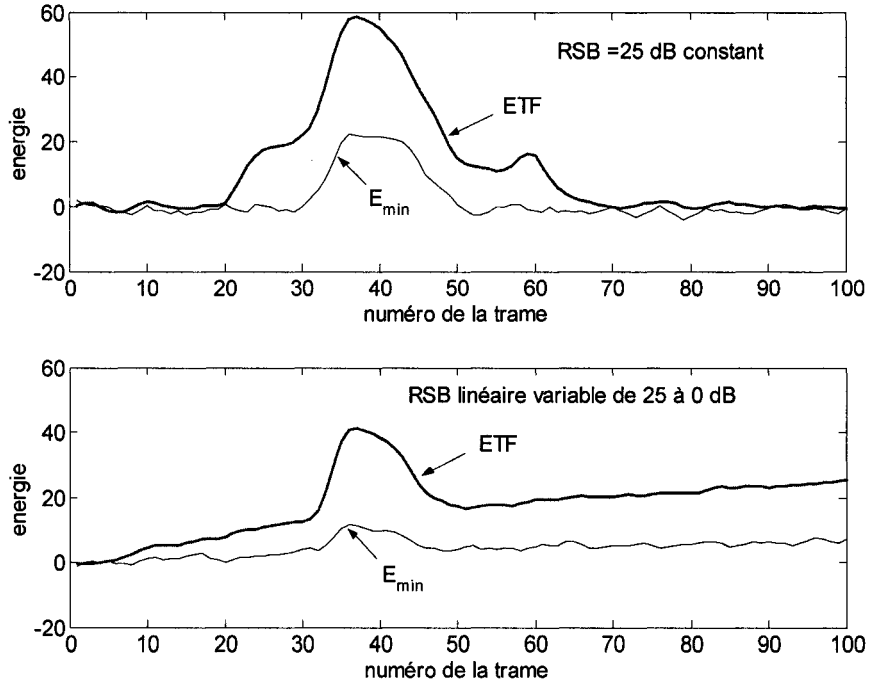


Figure 28 Les paramètres *ETF* et E_{min} pour le mot six

L'énergie du domaine temps $T(n)$ est le logarithme de l'énergie court-terme pour chaque trame, lissé et normalisé, exactement comme dans le cas de chaque bande du spectre subjectif.

Dans la figure 28, les deux paramètres *ETF* et E_{min} sont représentés pour un RSB constant et pour un RSB linéaire variable dans le cas du mot six. On a utilisé le bruit rose et une longueur de trame qui correspond à 15 ms.

3.3.3.3 Algorithme de décision

Avec ces paramètres on peut passer à l'étape de décision résumée dans la figure 29. Quand dépasse un certain seuil *th1*, le paramètre *VAR* indique une variation importante dans le niveau de bruit et les seuils de décision *th2* et *th3* sont modifiés en conséquence. Dans la partie A du diagramme de décision, le paramètre *ETF* et le seuil conservateur *th1* sont utilisés pour trouver une première estimation des frontières pour les régions de parole qui possèdent plus d'énergie et dépassent un certain seuil de durée *th4* = 90 ms. Dans la partie B, le début de la région de parole est poussé vers le début du signal tant que *ETF* est plus grande que le deuxième seuil moins conservateur *th3* ou le taux de passage par zéro dépasse le seuil *th5* et le seuil de durée n'est pas dépassé. Le seuil de passage par zéro est fixé à partir du taux de passage par zéro moyen des 5 premiers trames de signal. La partie C déplace la fin de la région de parole vers la fin du signal dans les mêmes conditions.

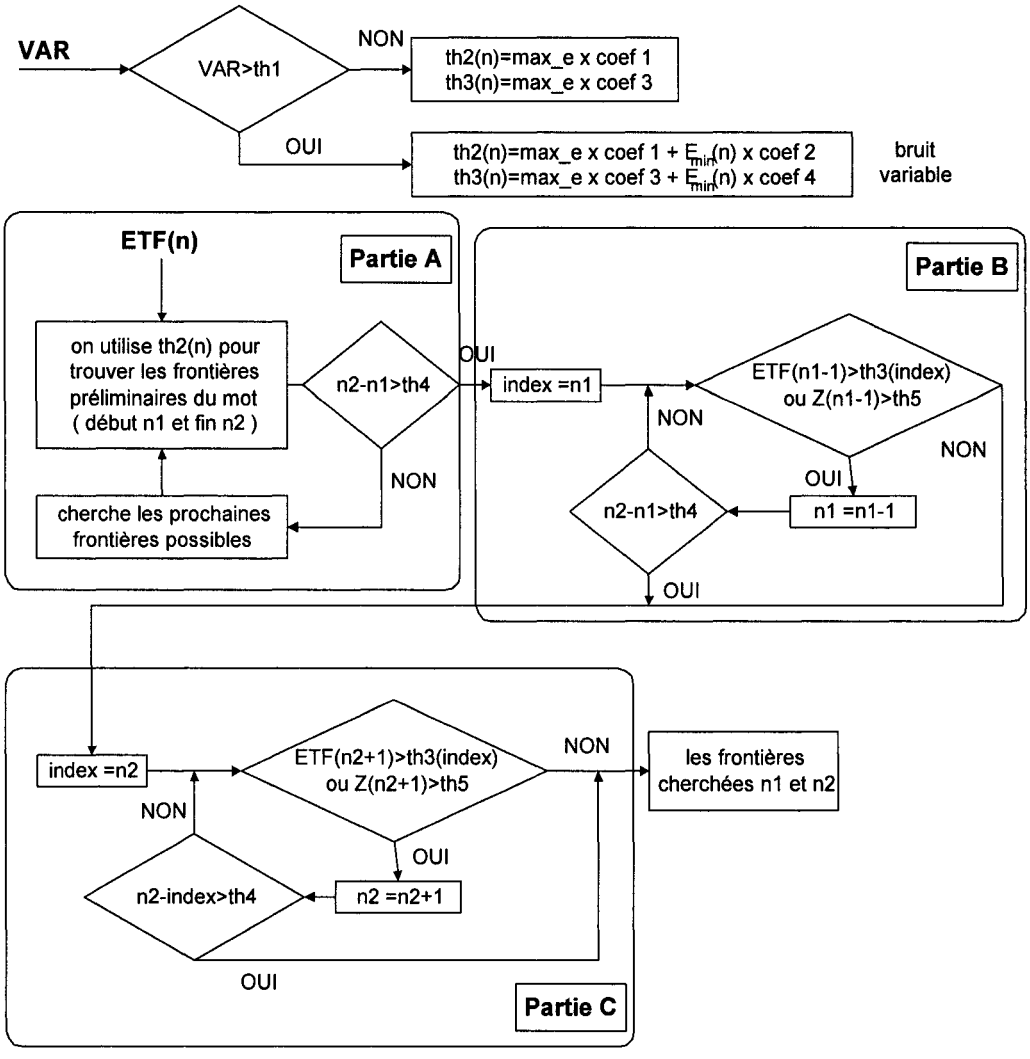


Figure 29 Diagramme de décision [20]

Le paramètre *max_e* est le maximum de $T(n)$ pour tout le signal. Les valeurs des paramètres *th1*, *coefi*, i=1 ,...,4 sont choisies pour optimiser un set d'exemples donnée pour lesquels les frontières on été marquées manuellement.

3.3.3.4 Observations

L'algorithme présenté est conçu pour résoudre le problème de VAD dans le cas d'un RSB constant ou variable. Sa faiblesse provient du fait que pour des signaux vocaux longs les paramètres proposés sont moins robustes. De plus il n'est pas approprié pour un traitement séquentiel car il a besoin de tout le signal pour l'étape de décision.

3.3.4 L'algorithme de VAD de l'annexe G.729 B de l'ITU [12]

Cet algorithme de VAD, utilise une trame de 10 ms donc 80 échantillons à une fréquence d'échantillonnage de 8000 Hz. Il effectue une décision pour chaque trame sans introduire aucun retard entre la décision pour la trame courante et l'arrivée d'une nouvelle trame. Si la décision du module de VAD indique la présence de la parole, le codeur de parole code la trame qui sera après transmise ; si l'absence de la parole est indiquée, les algorithmes de transmission discontinue et de génération de bruit de confort génèrent les trames d'inactivité vocale.

L'algorithme adopte une approche de classification des formes. On utilise un ensemble de quatre paramètres qui décrit chaque trame de signal et un ensemble de frontières de décision dans l'espace 4D des paramètres. La décision initiale est lissée tenant compte d'un certain nombre de trames passées. Dans une dernière étape on procède à l'actualisation des moyennes courantes du bruit du fond si les seuils énergétiques imposés au bruit de fond sont dépassés.

Le fonctionnement de l'algorithme est résumé dans le diagramme de la figure 30 :

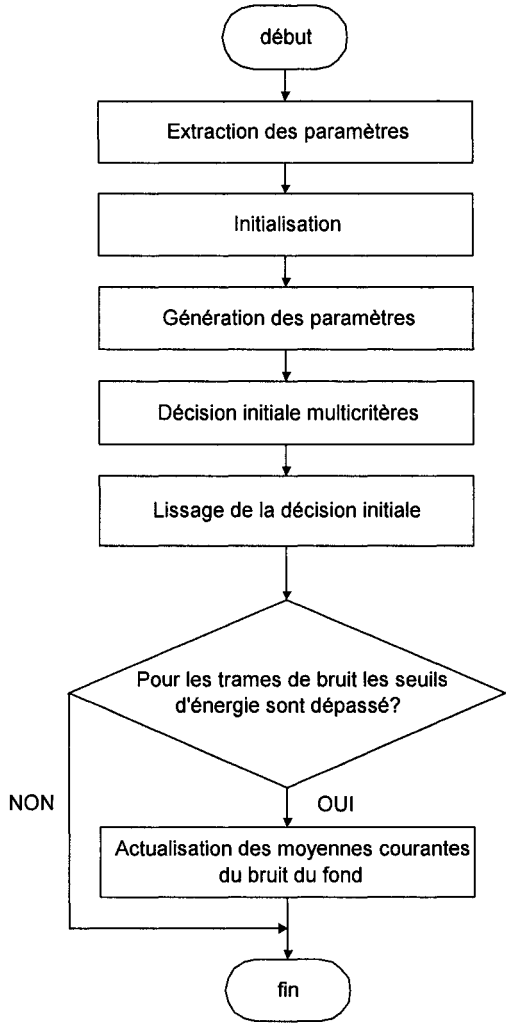


Figure 30 Diagramme du fonctionnement de l'algorithme de VAD de l'annexe G.729 B de l'ITU [12]

3.3.4.1 Extraction des paramètres

Dans une première étape on calcule les douze premiers coefficients d'autocorrélation court-terme R_n, n = 1,...12. A partir des ces coefficients on dérive les coefficients de prédiction linéaire et le deuxième coefficient de réflexion. A partir des coefficients de prédiction linéaire on obtient une série de 10 paramètres qui représentent le spectre du

signal en terme paires de raies spectrales, notés LSP_i, i = 1,...10, dont une méthode efficace de calcul est présenté en [33]. Ces paramètres représentent des angles ayant les valeurs dans l'intervalle de 0 à π.

On calcule l'énergie pleine bande E_f qui correspond au logarithme du R_1 normalisé par la taille de la fenêtre L :

$$E_f = 10\log_{10}\left(\frac{R_1}{L}\right) \tag{3.16}$$

L'énergie basse fréquence E_b est obtenue par une multiplication de la matrice de Toeplitz, engendrée par les coefficients d'autocorrélation, avec les 13 premiers coefficients de la réponse impulsionnelle d'un filtre RIF ayant la fréquence de coupure 1000 Hz. Le dernier paramètre de la trame courante est le taux de passage par zéro Z.

On définit le minimum de l'énergie long-terme E_{min} comme étant le minimum de l'énergie pleine bande sur un intervalle d'une seconde.

3.3.4.2 Initiation

Les moyennes de paramètres déjà vus pour un nombre de N_i premières trames constituent l'initiation des paramètres du bruit. Pendant ce temps la décision de l'algorithme est basée sur un seuil d'énergie seulement.

3.3.4.3 Génération des paramètres

La différence entre les paramètres qui caractérisent la trame courante et les paramètres qui caractérisent le bruit de fond au même instant génère les quatre paramètres utilisés pour la décision initiale. Ces paramètres sont :

- la distorsion spectrale

$$\Delta S = \sum_{i=1}^{10} (LSP_i - \overline{LSP})^2 \tag{3.17}$$

- la différence d'énergie pleine bande

$$\Delta E_f = \overline{E_f} - E_f \tag{3.18}$$

- la différence d'énergie basse fréquence

$$\Delta E_b = \overline{E_b} - E_b \tag{3.19}$$

- la différence de taux de passage par zéro

$$\Delta Z = \overline{Z} - Z \tag{3.20}$$

3.3.4.4 Décision initiale multicritères

Les quatre paramètres de décision se trouvent dans une région de l'espace Euclidienne quadri-dimensionnelle. Les paramètres indiquant l'activité vocale se groupent dans un certain hyper-volume de l'espace lorsque les paramètres indiquant le manque d'activité vocale se groupent dans un autre hyper-volume. Ces hyper-volumes ont été identifiés et séparés à l'aide de quatorze hyper plans définis dans des espaces tri-dimensionnels qui constituent donc les frontières pour la décision initiale. Pour chaque trame la décision initiale correspond à la région où le point, dont les coordonnés sont les quatre paramètres de décision, se retrouve.

3.3.4.5 Lissage de la décision initiale

Normalement les régions de parole ou de silence ont une longueur de quelques dizaines de trames, la décision initiale est lissée pour refléter cette caractéristique de stationnarité. Le lisage s'effectue en quatre étapes dérivées d'une observation approfondie d'une large base de données.

Dans la première étape la décision d'activité vocale est prolongée à la trame courante si l'énergie de la trame courante dépasse un certain seuil.

Dans un deuxième pas la décision d'activité vocale est prolongée à la trame courante si les deux trames précédentes sont des trames de parole et la différence entre l'énergie de la trame courante et l'énergie des deux trames précédentes est au-dessous d'un certain seuil.

Dans un troisième pas la décision d'inactivité vocale est prolongée à la trame courante si les dix trames précédentes sont des trames de silence et la différence entre l'énergie de la trame courante et l'énergie des dix trames précédentes est au-dessous d'un certain seuil. Dans une dernière étape la décision d'activité vocale et corrigée si l'énergie de la trame courante est inférieure à l'énergie du bruit avec un certain écart, le deuxième coefficient de réflexion est plus petit que 0.6 et aucun des deux premiers pas n'ont pas été exécutés.

Les tests ont montré qu'on obtient des meilleurs résultats lorsqu'une logique floue remplace la partie de décision de l'algorithme [21] et que même la langue utilisée pour tester l'algorithme a des influences sur le résultat de classification [14].

3.3.4.6 Actualisation des paramètres du bruit de fond

La dernière étape de l'algorithme est l'actualisation des moyennes courantes du bruit de fond. Pour cela on utilise une version simplifiée de l'algorithme de VAD qui utilise seulement l'énergie pleine bande, le coefficient de distorsion spectrale et le deuxième coefficient de réflexion. Si la décision est silence les paramètres du bruit de fond sont actualisés à l'aide d'un filtre autorégressif d'ordre un. Différents coefficients de régression sont utilisés pour chaque paramètre et une adaptation plus rapide est exécutée en début du signal et après chaque réinitialisation. Cet algorithme de VAD simplifié est plus conservateur en ce qui concerne la détection de bruit pour éviter l'adaptation des paramètres du bruit avec des valeurs qui en réalité proviennent de trames de parole. Il se comporte bien pour des bruits qui changent lentement dans le temps. Dans le cas des bruits moins stationnaires il perd plus de trames de bruit qui pourraient être utilisées pour l'adaptation des paramètres du bruit. Dans ces conditions le bruit estimé est loin du bruit réel ce qui augmente la probabilité de fausse alarme.

S'il y a une augmentation brusque dans le niveau d'énergie du bruit, l'algorithme peut se bloquer dans l'état d'activité vocale ; pour éviter cette possibilité on utilise un mécanisme de réinitialisation qui initialise le niveau du bruit de fond avec la valeur du E_{min}.

3.3.5 VAD basé sur un modèle statistique [22-23]

Cet algorithme de VAD emploie une modélisation statistique des paramètres qui caractérisent le bruit de fond est le signal vocal. Cette approche permet d'utiliser la règle de décision de Bayese (annexe 3) pour la prise de décision. On considère que les paramètres du bruit de fond sont connus a priori grâce à une méthode d'estimation de la statistique du bruit qui sera exposée ensuite.

3.3.5.1 Le calcul du ratio de vraisemblance

On suppose que le bruit et le signal vocal sont des processus aléatoires non-corrélés gouvernés par des lois de distribution de probabilité Gaussiennes. Les coefficients de la TFD pour chaque processus sont aussi des variables aléatoires indépendantes qui respectent une loi de distribution de probabilité Gaussiennes. Les vecteurs de coefficients L dimensionnels du signal vocal, du bruit et du signal vocal bruité sont notés S, N et X respectivement. L'indice k indique le k-ième élément de ces vecteurs.
Les variances de $N(k)$ et $S(k)$ sont :

$$\begin{aligned} v_N(k) &= S_N(2\pi k / L) \\ v_S(k) &= S_S(2\pi k / L) \end{aligned} \tag{3.21}$$

où $S_N(\omega)$ et $S_S(\omega)$ est la densité spectrale de puissance du bruit et du signal vocal propre.
La variance de $X(k)$ est :

$$v_X(k) = v_N(k) + v_S(k) \tag{3.22}$$

Les deux hypothèses sont : H_0 quand la parole est absente et $X = N$, et H_1 quand la parole est présente et $X = N + S$. Si les paramètres du bruit de fond sont connus a priori, les sets de paramètres $V_S = \{v_S(k),\ k = 0,\ldots,L\text{-}1\}$ qui sont propres à la parole sont inconnus et sont estimés par la méthode de soustraction spectrale :

$$\hat{V}_S(k) = | X(k)^2 | - v_N(k) \tag{3.23}$$

Les fonctions de densité de probabilité conjointe conditionnée par H_0, H_1 et V_S sont :

$$p(X \mid H_0) = \prod_{k=0}^{L-1} \frac{1}{\pi v_N(k)} \exp\left\{ -\frac{|X(k)|^2}{v_N(k)} \right\} \tag{3.24}$$

$$p(X \mid V_S, H_1) = \prod_{k=0}^{L-1} \frac{1}{\pi [v_N(k) + v_S(k)]} \exp\left\{ -\frac{|X(k)|^2}{v_N(k) + v_S(k)} \right\} \tag{3.25}$$

En remplaçant (3.23) dans (3.25) le ratio de vraisemblance généralisé est :

$$\begin{aligned} \Lambda_g &= \frac{1}{L} \log \frac{p(X \mid \hat{V}_S, H_1)}{p(X \mid H_0)} \\ &= \frac{1}{L} \sum_{k=0}^{L-1} \left\{ \frac{|X(k)|^2}{v_N(k)} - \log \frac{|X(k)|^2}{v_N(k)} - 1 \right\} \end{aligned} \tag{3.26}$$

La règle de décision en utilisant le ratio de vraisemblance généralisé peut être maintenant énoncé. Si Λ_g est plus grand qu'un seuil la décision est H_1 donc l'existence de la parole, dans le cas contraire la décision est H_0 et la trame analysée est classifiée comme bruit.

Cette relation (3.26) est une approximation de la mesure de distorsion d'Itakura-Saito DIS. Dans la figure 31 on représente Λ_g pour le même signal et deux RSB différents. On utilise le bruit rose et une fenêtre rectangulaire de 30 ms qui se déplace à chaque 10 ms.

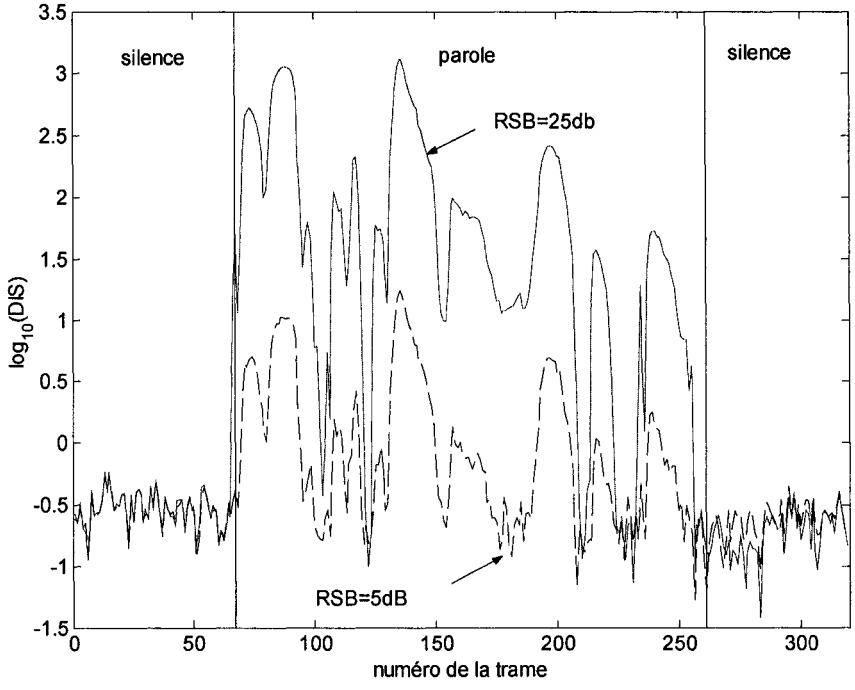


Figure 31 Λ_g pour le même signal et deux RSB différents

Les paramètres qui caractérisent le bruit de fond ont été obtenus en faisant la moyenne des paramètres correspondant aux 10 premières trames du signal. On observe que l'effet du bruit est de diminuer la valeur de Λ_g correspondant aux intervalles de parole.

3.3.5.2 L'estimation de la statistique du bruit de fond

Lorsque le bruit de fond change en fonction du temps, les paramètres qui le caractérisent doivent être actualisés pour refléter cette variation. L'estimation optimale de la variance de $v_N(k)$ dans le sens de l'erreur quadratique moyenne minimale est donnée par :

$$\begin{aligned}\hat{v}_N(k) &= E(v_N(k) \mid X(k)) \\ &= E(v_N(k) \mid H_0)P(H_0 \mid X(k)) + E(v_N(k) \mid H_1)P(H_1 \mid X(k))\end{aligned} \tag{3.26}$$

En utilisant la règle de Bayes pour $P(H_0|X(k))$ et $P(H_1|X(k))$ on a :

$$E(v_N(k)\,|\,X(k)) = \frac{1}{1+\rho\,\Lambda(k)} E(v_N(k)\,|\,H_0) + \frac{\rho\,\Lambda(k)}{1+\rho\,\Lambda(k)} E(v_N(k)\,|\,H_1) \quad (3.27)$$

où $\rho = P(H_0)/P(H_1)$ et $\Lambda(k) = p(X(k)\,|\,H_1)/p(X(k)\,|\,H_0)$.

Quand la parole est absente $E(v_N(k)|H_0)$ est remplacée par $|X(k)|$. Quand la parole est présente on ne peut pas utiliser la trame courante pour actualiser les paramètres du bruit et on utilise la trame précédente. On obtient une relation récursive pour l'estimation des paramètres du bruit au moment n, $v_N(k,n)$:

$$\hat{v}_N(k,n) = \frac{1}{1+\rho\,\Lambda(k,n)} |\,X(k)\,|^2 + \frac{\rho\,\Lambda(k,n)}{1+\rho\,\Lambda(k,n)} \hat{v}_N(k,n-1) \quad (3.28)$$

Parce qu'on ne possède pas une estimation de l'ensemble de paramètres V_S et parce que la décision n'est pas prise pour chaque bande de fréquence k mais par rapport à la moyenne des ratios de vraisemblances de toutes les bandes de fréquence, on utilise Λ_g au lieu de Λ :

$$\hat{v}_N(k,n) = \frac{1}{1+\rho\,\Lambda_g(n)} |\,X(k)\,|^2 + \frac{\rho\,\Lambda_g(n)}{1+\rho\,\Lambda_g(n)} \hat{v}_N(k,n-1) \quad (3.29)$$

La relation (3.29) montre que la vitesse d'adaptation des paramètres du bruit est plus grande quand le spectre de la trame courante est plus près de l'estimation du spectre du bruit. Le paramètre ρ peut être vu comme une constante qui détermine la vitesse de convergence plutôt qu'une probabilité a priori de l'existence de la parole. Avec cette

approche le spectre du bruit est actualisé en permanence, même si la parole est détectée dans la trame courante.

3.3.5.3 L'algorithme de décision

Au lieu de comparer la valeur de Λ_g avec un certain seuil, l'algorithme de décision est modifié pour éviter les possibles fausses décisions d'inactivité vocale causées par des régions de faible énergie appartenant au signal vocal.

Dans le cas particulier de la détection d'activité vocale il est utile de modéliser le processus de décision avec une chaîne de Markov (annexe 4) de premier ordre à deux états : parole H_1 est silence H_0. Ce modèle est justifié si on prend en considération la forte corrélation qui existe au niveau de l'occurrence de la parole ou du silence entre deux trames consécutives de signal.

La matrice de probabilités de transitions Π qui décrit le modèle de Markov binaire adopté est une matrice stochastique :

$$\Pi = \begin{bmatrix} a_{00} & a_{10} \\ a_{01} & a_{11} \end{bmatrix} \tag{3.30}$$

Où a_{00} représente la probabilité de silence quand a été silence à l'instant précédent, a_{10} représente la probabilité de silence quand a été parole à l'instant précédent. Pour cette matrice on a $\sum_j a_{ij} = 1$ et donc il suffit juste une des deux lignes pour décrire le processus de transition. Les valeurs de a_{ij} sont déterminées par la relation :

$$a_{ij} = \frac{T_{ij}}{\sum_q T_{iq}} \tag{3.31}$$

où T_{iq} est un nombre mesuré de transition de l'état i à l'état q pour un signal vocal normal.

Une représentation graphique du modèle est donnée dans la figure :

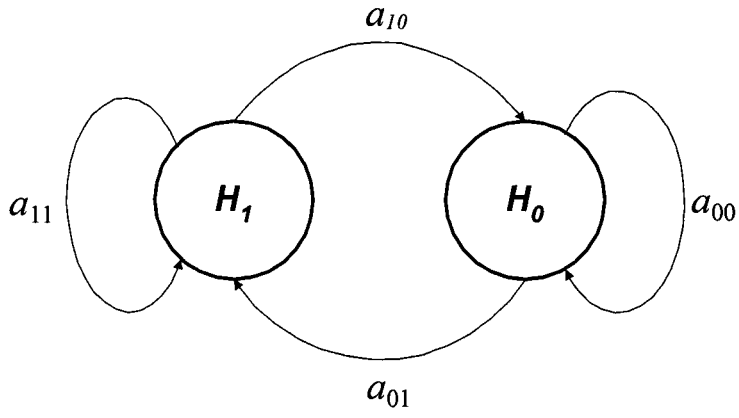


Figure 32 Modèle de Markov binaire [23]

On adopte la notation $P_{n-1}(H_1)$ pour représenter la probabilité a priori que la parole soit présente à l'instant n étant donné les observations jusqu'à l'instant n-1. Si on suppose $P_{n-1}(H_1)$ connue, l'objectif est d'estimer la probabilité a posteriori $P_n(H_1)$ que la parole soit présente à l'instant n étant donné les observations jusqu'à l'instant n. La règle de décision de Bayes et le théorème de la probabilité totale conduit vers la probabilité de classe a posteriori :

$$P_n(H_1) = P(H_1 \mid X(k,n)) = \frac{\Lambda_g P_{n-1}(H_1)}{\Lambda_g P_{n-1}(H_1) + (1 - P_{n-1}(H_1))} \tag{3.32}$$

La règle de décision est donc :

$$H(n)=\begin{cases}H_0 & si\ P_n(H_1)\geq S_{H1}\\ H_1 & si\ P_n(H_1)<S_{H1}\end{cases} \tag{3.33}$$

Le seuil S_{HI} qui contrôle le compromis entre P_d et P_f est choisi expérimentalement.
La probabilité a priori pour l'instant suivant n+1 est obtenue à l'aide du modèle de Markov adopté :

$$\begin{pmatrix}1-P_{n+1}(H_1)\\ P_{n+1}(H_1)\end{pmatrix}=\begin{pmatrix}a_{00} & a_{10}\\ a_{01} & a_{11}\end{pmatrix}\begin{pmatrix}1-P_n(H_1)\\ P_n(H_1)\end{pmatrix} \tag{3.34}$$

et donc :

$$P_{n+1}(H_1)=a_{01}(1-P_n(H_1))+a_{11}P_n(H_1) \tag{3.35}$$

3.3.5.4 Discussion

Cette méthode, qui est une approche statistique de la VAD, a été améliorée en utilisant d'autres paramètres et lois de densité de probabilité pour représenter le bruit et le signal de la parole [24-25]. D'autres algorithmes de VAD qui utilisent une approche statistique on trouve dans [26] et [27].

Dans [24] on utilise la transformé de Karhune-Loève TKL ou la transformé de cosinus discret TCD pour représenter le signal. On adopte une loi de densité de probabilité Gaussienne pour le bruit et une distribution de Laplace pour le signal vocal. Les paramètres du bruit et du signal vocal sont actualisés en utilisant deux filtres IIR d'ordre un. Les coefficients de chaque filtre permettent une vitesse d'adaptation appropriée du paramètre du bruit et du signal vocal. Avec ces modifications les résultats obtenus sont supérieurs par rapport à la méthode de départ qui emploie une distribution gaussienne pour la parole.

CHAPITRE 4

ALGORITHME DE DÉTECTION D'ACTIVITÉ VOCALE BASÉ SUR L'ANALYSE SPECTRALE

4.1 Justification du concept utilisé

Comme on a vu dans le chapitre précédent le plus grand problème pour un algorithme de détection d'activité vocale est constitué par le bruit de fond. Plus le rapport signal bruit est faible plus les paramètres propres au signal vocal sont affectés par le bruit et la détection de la parole est difficile en spécial pour les régions de faible énergie du signal vocal. Le RSB est une mesure objective mais n'est pas le seul paramètre d'intérêt dans la caractérisation du bruit. Chaque type de bruit a un impact différent sur les paramètres qui décrivent le signal vocal et lorsqu'on conçoit un algorithme de VAD il faut tenir compte de cet aspect. Par exemple on peut avoir des algorithmes de VAD qui se comportent bien dans le cas du bruit blanc, qui est un bruit plutôt stationnaire, mais qui échouent dans le cas des bruits légèrement non stationnaires. La majorité des bruits de fond réels présentent un certain degré de non stationnarité qui se matérialise dans une variation temporelle des paramètres de bruit. Toutefois la majorité des algorithmes de VAD est basée sur la mesure des variations de ces paramètres dans l'hypothèse générale que le bruit est plus stationnaire que la parole ce qui rend la discrimination possible. Le concepteur d'un tel algorithme est donc toujours à la recherche d'un set de paramètres qui soit robuste par rapport à la variabilité du bruit mais sensible aux plus faibles variations engendrées par la présence du signal vocal. Généralement on réalise une actualisation des paramètres du bruit pendant les périodes de silence pour tenir le pas avec l'évolution du bruit. Pendant les périodes de parole les paramètres du bruit sont difficiles à évaluer et le processus d'actualisation risquant.

Dans les efforts de trouver de tels paramètres robustes nous nous somme arrêté sur deux paramètres, le coefficient de corrélation spectrale *CS*, qui est un contribution originale, et le moyenne de RSB de sous-bandes *RS*, un paramètre déjà utilisé dans la détection d'activité vocale [22].

4.1.1 Le coefficient de corrélation spectrale

Parce que les propriétés spectrales du signal présentent un intérêt majeur pour la perception auditive [2] on a choisi le spectre comme paramètre qui décrit le signal dans l'algorithme de VAD proposé.

En principe, le concept de densité spectrale ne s'applique qu'à un signal stationnaire. Le signal vocal est essentiellement non stationnaire, c'est pourquoi on a introduit l'analyse à court terme. Ce concept permet l'analyse spectrale continue du signal vocal. Le spectre court terme $S_n(k)$ est défini à partir de la transformée de Fourier discrète court-terme. Pour chaque trame n de signal on a :

$$S_n(k) = \mathrm{Re}(X_n(k))^2 + \mathrm{Im}(X_n(k))^2 \tag{4.1}$$

où $X_n(k)$ est la transformée de Fourier discrète de la trame n isolée du reste du signal à l'aide d'une fenêtre de pondération.

On définit le coefficient de corrélation *CS* entre deux spectres court-terme du signal $S_n(k)$ et $S_m(k)$ avec la relation [35] :

$$CS(S_n, S_m) = 1 - \frac{\mathrm{cov}(\sqrt{S_n}, \sqrt{S_m})}{\sqrt{\mathrm{cov}(\sqrt{S_n}, \sqrt{S_n})\,\mathrm{cov}(\sqrt{S_m}, \sqrt{S_m})}} \tag{4.2}$$

où

$$\mathrm{cov}(S_n, S_m) = E[(\sqrt{S_n} - E(\sqrt{S_n}))(\sqrt{S_m} - E(\sqrt{S_m}))] \tag{4.3}$$

L'opérateur E symbolise l'espérance mathématique. Ce coefficient de corrélation est une mesure normalisée de la dépendance linéaire entre deux sets de données ; spectres dans ces cas. Les valeurs possibles pour ce paramètre sont dans l'intervalle de 0 à 2.

L'algorithme proposé utilise la relation (4.2) pour calculer la corrélation entre le spectre du bruit de fond S_{bruit} et le moyenne de trois dernières trames appelé spectre instantanée S_{inst}.

Une valeur près de zéro indique une forte corrélation entre les deux spectres. Dans ces cas les deux trames du signal possèdent à peu près la même structure spectrale et on peut penser qu'il s'agit du même son même si l'amplitude, donc l'énergie diffère significativement. En fait ce paramètre est totalement indépendant par rapport à l'amplitude des deux spectres lorsque leur forme est pareille.

Plus la valeur du coefficient de corrélation s'éloigne de zéro plus la corrélation des deux spectres est faible, une valeur près de l'unité indique que les deux spectres sont complètement non corrélés. Une valeur près des deux indique que les deux spectres sont complémentaires.

On initialise le spectre du bruit à partir de quelques trames en debout du signal, trames qui sont suppose contenir juste du bruit. Quand la valeur du paramètre *CS*, dépasse un certain seuil on peut décider l'existence de l'activité vocale. Lorsque la valeur du paramètre *CS* reste au-dessous du seuil choisi, la décision est d'inactivité vocale et on peut actualiser le spectre du bruit avec le spectre de la trame courante.

La propriété la plus intéressante de ce paramètre est son insensibilité par rapport aux amplitudes des deux spectres, ce qui le rend totalement insensible par rapport aux variations en amplitude du bruit de fond. Cette propriété est intéressante pour l'analyse de signaux caractérisés par une RSB variable quand le bruit de fond garde la même structure spectrale mais présente des variations d'énergie importantes dans le temps.

4.1.2 La moyenne des RSB des sous-bandes

Ce coefficient défini avec la relation (4.4) a été déjà utilisé avec succès dans d'autre algorithmes de VAD [22,23]. Il est un indicateur sensible aux variations de l'énergie dans les sous-bandes du signal :

$$RS(S_{inst}(k), S_{bruit}(k)) = \frac{1}{L}\sum_{k=1}^{L}\frac{S_{inst}(k)}{S_{bruit}(k)} \tag{4.4}$$

Théoriquement la valeur de ce paramètre est près de l'unité quand les deux spectres présentent la même énergie pour toutes les sous-bandes et une valeur plus grande quand il y a une augmentation de l'énergie pour une ou plusieurs sous-bandes du signal. Lorsque dans une application réelle le signal vocal s'ajoute au bruit, une variation de l'énergie, mesurée à laide de *RS*, pourrait indiquer le début de l'activité vocale.
La somme des logarithmes de ces deux paramètres pondérés par une constante de proportionnalité est le paramètre global *PG* utilisé dans l'algorithme de VAD proposé.

4.1.3 Le choix de la méthode de calcul du spectre du signal

A partir des observations théoriques du chapitre précédent on va tester le *CS* tout d'abord dans le cas du bruit et après pour le signal vocal. On a choisi comme signal de test le bruit rose car la majorité des bruits de fond réels présentent une concentration de l'énergie dans la partie de basse fréquence.

La fréquence d'échantillonnage du signal est 8000 Hz. On utilise une fenêtre d'analyse rectangulaire de 160 échantillons que l'on divise en deux sous fenêtres de 128 échantillons alignés au début et a la fin de la fenêtre principale. Les deux sous fenêtres ainsi obtenues sont compatibles avec la TFR en 128 points. Le spectre correspondant à la fenêtre principale est obtenu en moyennant les spectres des deux sous fenêtres.

Le vecteur qui contient le spectre du signal est augmenté avec un élément qui est la somme de toutes ses valeurs donc proportionnelle avec l'énergie totale de la trame analysée. L'effet de ce dernier élément est d'augmenter la valeur de CS spécialement dans le cas du bruit blanc.

La fenêtre principale d'analyse se déplace avec 80 échantillons donc on obtient un nouveau spectre de signal à chaque 10 ms. On améliore la variance de l'estimation du spectre instantané S_{inst} en faisant la moyenne des trois derniers spectres par rapport à l'instant n. Comme dans le cas de l'estimateur spectral modifié, S_{inst} est obtenue par la moyenne de 6 spectres qui se chevauchent. S_{inst} correspond à une fenêtre de longueur totale de 320 échantillons donc une durée de 40 ms pour laquelle on peut considérer le signal vocal comme étant stationnaire.

Le spectre du bruit S_{bruit} est initialisé avec la moyenne des 8 premiers spectres et actualisé en utilisant un filtre IIR d'ordre un [24] :

$$S_{bruit}(n,k) = aS_{bruit}(n-1,k) + (1-a)S_{inst}(n,k) \tag{4.5}$$

La valeur de la constante a influence la vitesse d'adaptation, quand a diminue la vitesse d'adaptation augmente. On a utilisé $a = 0.98$.

4.1.4 Comportement du *CS* dans le cas du bruit

Dans la figure 33 on présente l'évolution du paramètre *CS* pour un signal de 6 s et un exemple de S_{bruit} et S_{inst}.

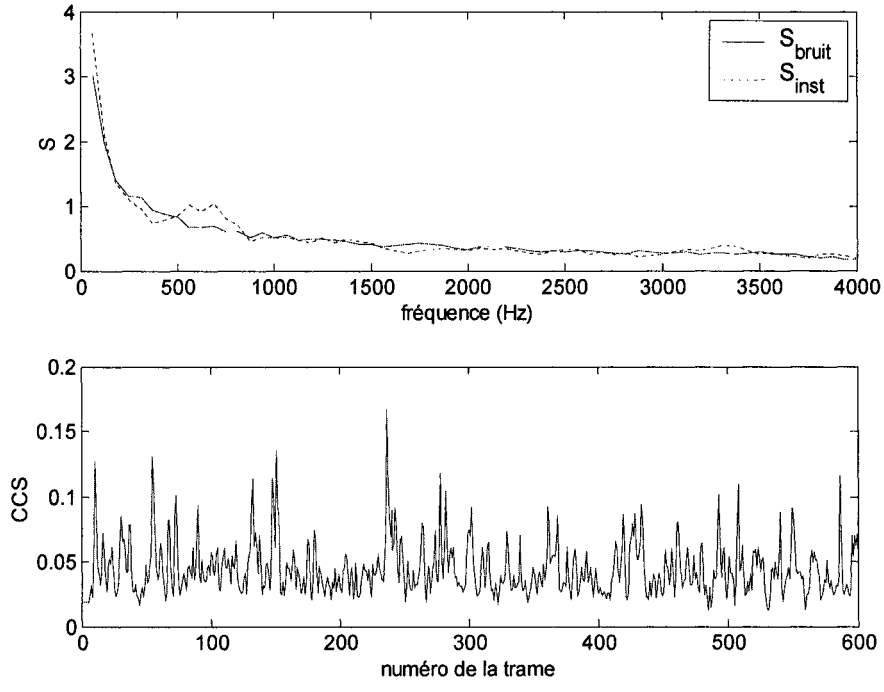


Figure 33 Les spectres S_{bruit} et S_{inst} et le paramètre *CS*

Comme on s'attend, on observe que la valeur de *CS* est inférieure à 0.15, ce qui indique une forte corrélation des deux spectres.

Dans la figure 34 on représente graphiquement la probabilité empirique versus les valeurs de ln(*CS*). Le but d'une telle représentation et de confirmer si le ensemble de

données étudiées provient d'une distribution normale. Lorsque les probabilités empiriques suivent une droite l'hypothèse de normalité est raisonnable.

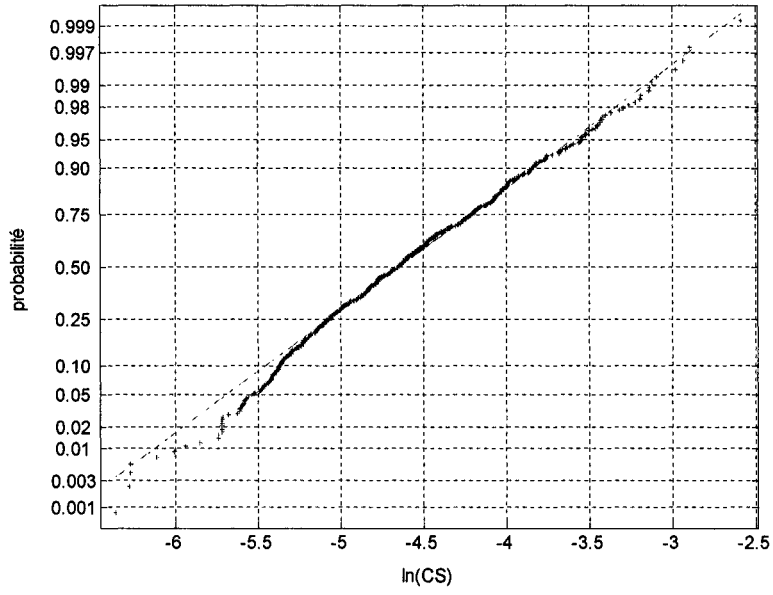


Figure 34 Probabilité empirique versus les valeurs de ln(*SC*)

En regardant le graphique on peut conclure que le logarithme de *CS* présente une distribution normale. Les tests réalisés avec d'autres types de bruits confirment l'hypothèse de normalité et conduisent vers les résultats exposés dans le Tableau VI où on a noté la moyenne de ln(*CS*) avec μ_b et l'écart type de ln(*CS*) avec σ_b.

Tableau VI

Moyenne et écart type du paramètre $ln(CS)$

Type bruit	μ_b	σ_b
rose	-4.63	0.61
véhicule	-5.52	0.86
cockpit	-4.25	0.57
cabine de commande	-4.7	0.61

4.1.5 Comportement du *CS* dans le cas du signal vocal

Parce que le spectre des trames de parole est différent du spectre du bruit, on s'attend à ce que pour ces régions la valeur de *CS* soit supérieure aux celles qui caractérisent le bruit. Lorsque *CS* dépasse un certain seuil on arrête le processus d'actualisation du spectre de bruit et on peut décider la présence de la parole. Dans la figure 35 on présente l'évolution du paramètre *CS* pour deux RSB, 30 et 10 dB et le signal originel sans bruit.

On observe que dans le cas d'un RSB élevé le coefficient *CS* fait très facilement la différence entre les régions de parole et de silence, sa valeur ne descend jamais au-dessus de 0.15. On remarque que pour les régions de parole qui présentent une faible énergie, associées généralement aux consonnes et qui posent normalement le plus de difficultés pour la détection, due aux différences spectrales, la valeur de *CS* est parfois plus élevée que pour les régions de haute énergie associées généralement aux voyelles.

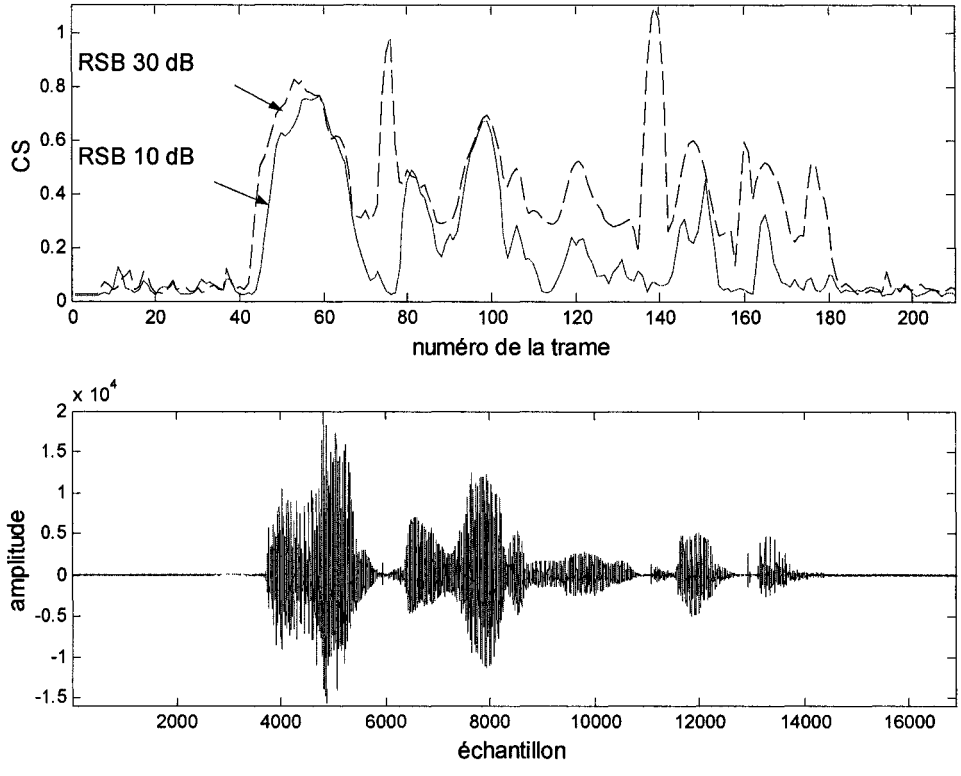


Figure 35 Évolution du *CS* pour le signal vocal et deux RSB

Malheureusement les choses changent radicalement quand le RSB augmente. Le spectre des régions de basse énergie du signal vocal est englouti par le spectre du bruit et la valeur du *CS* tombe vers des valeurs assez porches de celles qui caractérisent le bruit, et la détection de ces régions devient difficile. Le choix de la valeur du seuil devient moins évident et une petite variation autour d'une valeur optimale cause des variations importantes pour les probabilités de détection et de fausse alarme P_f et P_d. Pour améliorer cette situation on propose un algorithme plus complexe pour la partie de décision au lieu de comparer la valeur de *CS* avec un seuil fixe.

4.2 Algorithme de décision

L'algorithme de décision proposé est axé sur trois directions pour atteindre l'objectif d'un VAD robuste. On réalise pour cela un filtrage médian suivi d'une décision statistique corrigée par une méthode de lisage qui implique l'utilisation d'un modèle de Markov binaire pour le processus de décision.

4.2.1 Filtrage médian

L'effet bénéfique du filtrage médian se manifeste pour des RSB faibles. Il réduit la variance du paramètre *CS* pour les régions de bruit et de parole ce qui nous permet de descendre le seuil de décision sans augmenter la probabilité de fausse alarme P_f.

L'effet négatif est que, pour les RSB élevés, le filtre médian amplifie la valeur du *CS* qui caractérise les trames de silence situées au voisinage des trames de signal vocal et ainsi il peut augmenter la valeur de P_f dans le cas d'une valeur petite du seuil de décision.

Dans la figure 36 on représente la dépendance de P_d en fonction de P_f pour différents seuils de décision quand on utilise le paramètre proposé avec et sans filtrage médian. On utilise une fenêtre de lisage rectangulaire symétrique de longueur $2U+1$. Au signal vocal on a ajouté un bruit rose à un RSB de 5dB et 25 dB respectivement. On peut observer que le filtrage médian permet d'augmenter le P_d pour une même P_f.

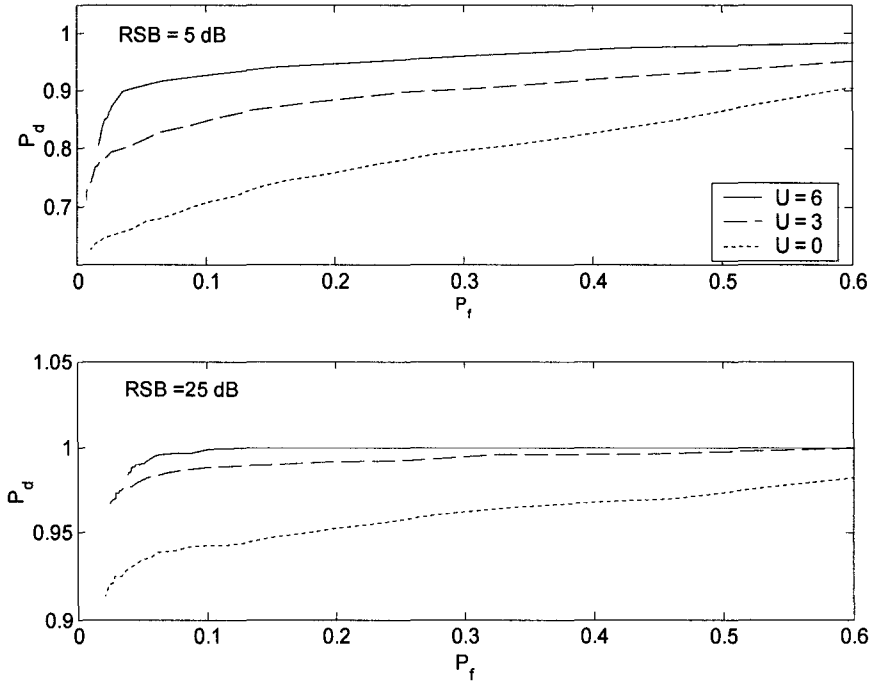


Figure 36 L'effet du lisage sur les valeurs de P_f et P_d

Le seuil optimal peut être déduit si on cherche le maximum pour la différence entre P_d et P_f figure 37. On observe que, pour le RSB de 25 dB, une petite variation du seuil autour de la valeur optimale n'est pas critique comme dans le cas du RSB de 5 dB. Lorsqu'on veut avoir un algorithme de VAD robuste, on va choisir la valeur du seuil qui donne les meilleurs résultats pour une large gamme de RSB même si cette valeur n'est pas optimale pour aucune de RSB étudiés.

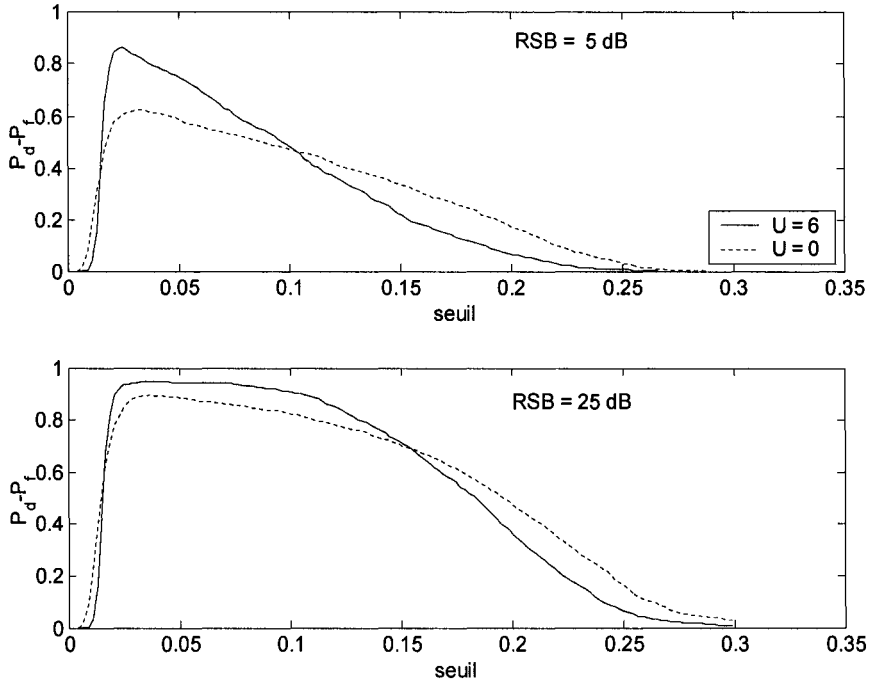


Figure 37 Le choix du seuil optimal pour différents RSB

Les tests effectués sur plusieurs types de bruits conduisent vers la conclusion que dans le contexte de l'algorithme de VAD proposé un filtre médian de longueur 13 est souhaitable pour des RSB allant de 25 dB à 5 dB.

On applique le filtrage médian sur le *ln*(*CS*) et le *ln*(*RS*) pour obtenir une version lissée de *CS* et *RS* à l'instant *n* soit *CSL*(*n*) et *RSL*(*n*) :

$$\begin{aligned} CSL(n) &= \frac{1}{2U}\sum_{i=-U}^{U}\log(CS(n+i)) \qquad avec \quad U=6 \\ RSL(n) &= \frac{1}{2U}\sum_{i=-U}^{U}\log(RS(n+i)) \end{aligned} \tag{4.6}$$

4.2.2 Décision statistique

Le paramètre global $PG(n)$ utilisé dans l'algorithme proposé est donné par la relation :

$$PG(n) = CSL(n) + b\,RSL(n) \tag{4.7}$$

où b est un constante de proportionnalité.

Pour les régions de bruit, $PG(n)$ présente une distribution normale avec la moyenne μ_b et l'écart type σ_b. Si on fait la supposition que ce même paramètre respecte une distribution normale pour les régions de parole avec la moyenne μ_v et l'écart type σ_v, on peut utiliser le ratio de vraisemblance Λ pour la décision. Pour cela on calcule les probabilités associées à l'hypothèse de parole active $p(H_1|PG)$ et de silence $p(H_0|PG)$ pour la valeur de PG courante.

Dans le calcul du ratio de vraisemblance on ajoute le paramètre d qui permet de contrôler la vitesse de variation et les valeurs extrêmes de Λ car lorsque $p(H_0|PG)$ tend vers zéro Λ tend vers infini et pourrait causer des problèmes pour la représentation numérique. Avec ce paramètre on a :

$$\Lambda(n) = \frac{p(H_1 \mid PG(n)) + d}{p(H_0 \mid PG(n)) + d} \tag{4.8}$$

Quand Λ dépasse l'unité on peut décider l'existence de la parole et actualiser les valeurs de μ_v et σ_v en utilisant un filtre IIR d'ordre un avec la valeur de PG courante. Dans le cas contraire la parole est absente et on actualise les valeurs de μ_b et σ_b.

4.2.3 L'utilisation du modèle de Markov binaire pour la décision

On utilise le modèle de Markov présenté au sous chapitre § 3.3.5.3 pour modifier la décision basée sur le ratio de vraisemblance. Dans ce cas la probabilité a posteriori $P_n(H_1)$ que la parole soit présente à l'instant n étant donné les observations jusqu'à l'instant n et la probabilité a priori $P_{n-1}(H_1)$ est donné par la relation :

$$P_n(H_1) = P(H_1 \mid P(n)) = \frac{\Lambda(n) P_{n-1}(H_1)}{\Lambda(n) P_{n-1}(H_1) + (1 - P_{n-1}(H_1))} \tag{4.9}$$

La règle de décision est donc :

$$H(n) = \begin{cases} H_0 & si\ P_n(H_1) \geq S_{H1} \\ H_1 & si\ P_n(H_1) < S_{H1} \end{cases} \tag{4.10}$$

Le seuil S_{H1} qui contrôle le compromis entre P_d et P_f est choisi expérimentalement. On choisit une valeur de 0.5.

La probabilité a priori pour l'instant suivant n+1 est obtenue à l'aide du modèle de Markov adopté :

$$P_{n+1}(H_1) = a_{01}(1 - P_n(H_1)) + a_{11} P_n(H_1) \tag{4.11}$$

En fait ce modèle de Markov réalise un lissage au niveau de la décision statistique initiale qui se concrétise par une homogénéisation des régions de bruit et de parole. Plusieurs algorithmes essaient de réaliser la même chose imposant une condition de durée minimale, choisie empiriquement, pour les régions de parole et de silence détectées initialement.

Le paramètre d introduit dans la relation (4.8) qui contrôle les extrêmes de Λ peut être utilisé pour contrôler, par l'intermédiaire de Λ, la valeur de $P_n(H_1)$. Un d petit permet au Λ d'atteindre une valeur grande et d'avoir une influence dominante dans la relation (4.9). Ainsi $P_n(H_1)$ dépend plus des valeurs $p(H_0|PG(n))$ et $p(H_1|PG(n))$ et donc de l'instant n et moins de l'instant n-1. Dans ces conditions l'algorithme est plus sensible aux variations du paramètre $PG(n)$.

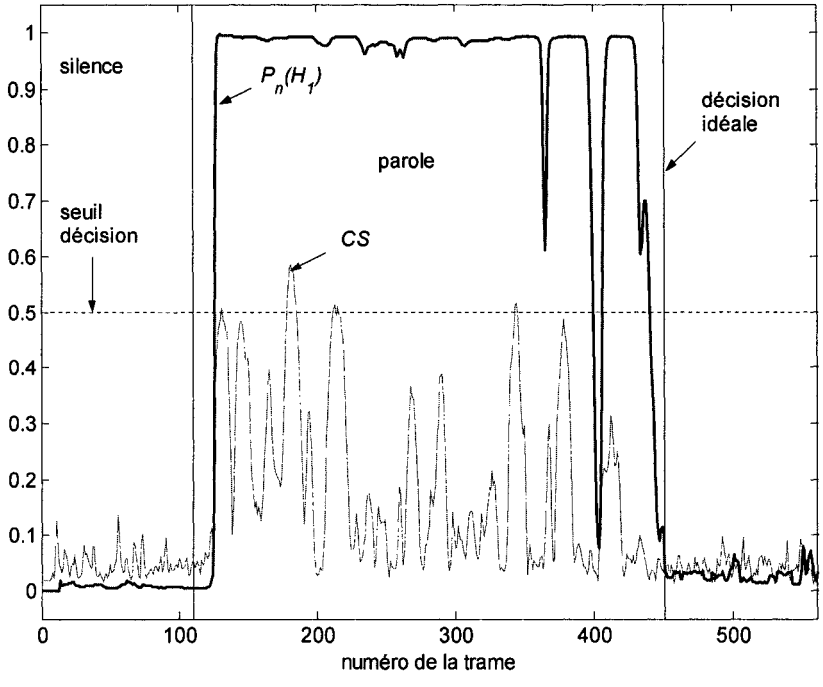


Figure 38 $P_n(H_1)$ versus CS (RSB = 5 dB)

Quand on augmente la valeur de d la valeur maximale de Λ diminue et l'influence de $P_{n-1}(H_1)$ est plus importante. Dans ces conditions l'algorithme est plus conservateur par rapport aux variations du paramètre $PG(n)$ et tend à rester dans l'état antérieur. On

utilise donc une valeur petite du d quand l'algorithme se trouve dans l'état H_0 pour être capable de détecter vite le changement de $PG(n)$. La valeur du d est augmentée progressivement jusqu'à une valeur limite d_{max} quand l'algorithme se trouve dans l'état H_1 pour éviter les passages accidentels à l'état H_0 dus aux faibles valeurs de l'énergie. Dans la figure 38 on exemplifie l'efficacité de l'algorithme de décision proposé pour un signal corrompu avec un bruit rose à un RSB de 5dB.

On peut observer dans la figure 38 que la décision a partir de $P_n(H_1)$ est plus évidente que dans le cas du paramètre de départ CS. La valeur du seuil de décision n'est pas critique, une valeur autour de 0.5 change peu la décision.

4.2.4 Initiation et actualisation des paramètres

Dans l'algorithme proposé il y a plusieurs paramètres dont les valeurs doivent être initialisées puis actualisées pendant le déroulement de l'algorithme. Pour tous les paramètres qui subissent une actualisation on utilise une régression linéaire d'ordre 1. La relation (4.12) permet de choisir la valeur du coefficient d'actualisation a en rapport avec le nombre N_T de valeurs antérieures souhaité dans le calcul de la valeur courante du chaque paramètre [8].

$$y(n) = a\,y(n-1) + (1-a)x(n) \quad avec \quad a = \frac{N_T}{N_T - 1} \tag{4.12}$$

Le spectre du bruit S_{bruit} est initialisé avec la moyenne des 8 premiers spectres et actualisé pendant les régions détectées de silence. Parce qu'on utilise un filtre médian de longueur $2U + 1 = 13$, à l'instant n la décision est pour la trame n-(U+1) où $U = 6$ et donc on utilise pour actualiser S_{bruit} le spectre du bruit à l'instant n-(U+1).

Les premiers $2U+1$ trames de signal sont utilisées pour initialiser la valeur de μ_b et pour cet intervalle de temps l'algorithme reste dans l'état de silence. Les deux paramètres qui caractérisent la statistique du bruit sont actualisés pendant les régions détectées de silence en utilisant la valeur de $PG(n-(U+1))$.

Les valeurs initiales pour μ_v et σ_v sont choisies expérimentalement et sont actualisées pendant les régions détectées de parole en utilisant la valeur de $P(n-(U+1))$. Quand l'algorithme détecte l'état silence pour un intervalle de temps supérieur à une certaine constante comparable avec la durée d'une consonne, les valeurs de μ_v et σ_v sont réactualisées pour atteindre leurs valeurs initiales. On fait cet ajustement pour diminuer le P_f quand le RSB est faible. Les valeurs de a_{01} et a_{11} restent inchangées après initiation.

L'algorithme peut être résumé comme suit :

Les constantes utilisées sont

$a_{01} = 0.3$ *et* $a_{11} = 0.98$

$a_{mv} = 0.999$ *pour actualiser* μ_v

$a_{vv} = 0.96$ *pour actualiser* σ_v

$a_{mv} = 0.98$ *pour actualiser* μ_b

$a_{vv} = 0.96$ *pour actualiser* σ_b

$a_s = 0.96$ *pour actualiser* S_{bruit}

$\mu_v = -1.5$ *et* $\sigma_v = 0.9$

constante de retard = 10 ;

Initiation des paramètres

pour n=1, ...$2U+1$

$a_s = 0.98$

a1 = 0.995 et a2 = 0.96;

$S_{bruit}(S_n, n = 1,...,8)$ et $S_{inst}(S_n, n = n-2,...,n)$

μ_b $(n=8,...,13)$, $\sigma_b = 0.3$

$d = 0$, $r = 0$ et $P_0(H_1) = 0$

Corps de l'algorithme

pour n = $2U+2,...,fin$

calcule de

$Sinst(S_n, n = n-2,...,n)$

$CS(S_{bruit}, S_{inst})$ *relation (4.2)*

$RS(S_{bruit}, S_{inst})$ *relation (4.4)*

CSL(CS, U) et RSL(RS, U) *relation (4.6)*

PG(n) *relation (4.7)*

$\Lambda(p(H_0|PG), p(H_1|PG), d)$ *relation (4.8)*

$P_n(\Lambda, P_{n-1}(H_1))$ *relation (4.9)*

si $P_n(H_1) > S_{H1}$

décision d'activité vocale

actualisation de μ_v *et* σ_v

$r = 0$ *et augment d*

autrement

décision d'inactivité vocale

$d = 0$ *et augmente r*

si $r >$ *constante de retard*

actualisation de μ_v *et* σ_{vv}

actualisation de μ_b , σ_b *et* S_{bruit}

calcul de

$P_{n+1}(H_1)(a_{01}, a_{11}, P_n(H_1))$

4.3 Évaluation des performances

Pour évaluer les performances de l'algorithme de VAD proposé on a utilisé un set de signaux vocaux appartenant à la base de données TIMIT car cette base de données fournit les étiquettes pour chaque échantillon. A partir des ces étiquettes on associe une valeur 0 aux trames de silence et une valeur 1 aux trames de parole, qui représentent la décision idéale de référence. Pour les trames qui contiennent des échantillons de signal vocal et de silence on attribue la valeur 1 si la majorité des échantillons sont de parole et la valeur 0 autrement.

Le signal de test présente 37% de trames de parole et 63% de trames de silence. On facilite ainsi l'interprétation des résultats et on est en concordance avec la proportion réelle entre les régions de parole et de silence rencontré dans le discours spontané.

Les signaux de parole sont normalisés et quantifiés pour être représentés par des valeurs entières ayant une gamme dynamique de –20000 à 20000. La raison de la quantification est la représentation numérique point fixe sur 16 bits du signal d'entrée dans le processeur numérique utilisé. La gamme dynamique a été limitée pour éviter l'erreur de dépassement lorsqu'on ajoute le bruit. La normalisation est nécessaire pour avoir un RSB consistant quand on analyse un signal continue formé de plusieurs signaux vocaux concaténés.

Les signaux de bruit utilisés proviennent de la base de données NOISEX. Dans le calcul du RSB on a utilisé l'énergie de signal originel qui est considéré sans bruit. A partir de cette énergie on calcule la constante avec laquelle on multiplie chaque échantillon de bruit qu'on ajoute pour obtenir le RSB souhaité.

Pour réaliser les tests on a choisi 12 phrases prononcées par un nombre de trois femmes et trois hommes concaténé dans un signal continu de 50 s pour un total de 1900 trames de parole et 3100 trames de silence. On a ainsi un signal de test complexe qui présente plusieurs régions de parole est de silence de longueur variable.

Pour tester la robustesse de l'algorithme proposé on a utilisé plusieurs types de bruits réels dans le cas de trois RSB différents. Les résultats obtenus sont résumés dans le tableau VII. Comme référence on a utilisé les résultats obtenus en utilisant l'algorithme proposé en [23].

Tableau VII

Résultats de simulation

Type bruite	RSB dB	Algorithme proposé		Algorithme en [23]	
		P_d	P_f	P_d	P_f
blanc	25	99.48	5.66	89.03	0.83
	15	98.59	4.16	71.91	0.32
	5	93.68	2.38	52.79	0.10
rose	25	99.63	*5.56*	92.74	1.02
	15	98.38	4.13	77.18	0.35
	5	91.28	2.57	56.50	0.10
véhicule militaire	25	100.00	7.31	98.22	3.72
	15	99.53	6.70	96.97	2.92
	5	97.08	4.73	92.69	2.00
voiture	25	100.00	9.44	99.37	3.78
	15	99.84	8.36	97.81	2.57
	5	98.96	7.02	96.81	2.03
cockpit	25	99.22	12.33	95.09	1.30

	15	97.65	10.77	85.01	0.48
	5	92.43	9.41	61.72	0.10
cabine de commande	25	99.27	17.00	96.55	14.90
	15	97.49	15.60	89.66	13.63
	5	92.85	15.00	71.96	13.00
rumeur	25	99.95	14.27	98.54	39.56
	15	99.16	17.64	96.03	38.70
	5	87.36	14.68	85.38	38.07

La P_d résultant de l'algorithme proposé est meilleure surtout pour des faibles RSB aux prix d'un P_f plus grande pour des RSB élevé et des bruits plus stationnaires.

Les simulations effectuées ont montré que les régions de silence et de parole détectées par l'algorithme proposé sont plus compactes, ce qui, d'un point de vue subjectif, est un avantage important.

La méthode utilisée pour obtenir le spectre du signal dans l'algorithme présenté, basée sur la TF, est une méthode non paramétrique. Les simulations effectuées ont prouvé qu'on obtient les mêmes résultats lorsqu'on utilise la méthode paramétrique basée sur la fonction d'autocorrélation, présentée au chapitre 3, pour obtenir le spectre du signal

CHAPITRE 5

IMPLÉMENTATION DE L'ALGORITHME SUR LE PROCESSEUR NUMÉRIQUE DE SIGNAL TMS320C6711

5.1 Problématique

La partie pratique de ce projet consiste à implémenter en temps-réel sur le processeur numérique de signal point flottant TMS320C6711 l'algorithme de VAD présenté ceci en utilisant la trousse d'outils de démarrage fournie par TI. Cette trousse inclut le Code Composer Studio CCS qui est un logiciel de développement des applications, la carte DSK qui contient le processeur et le support matérielle pour réaliser les entrées-soties, le câble parallèle qui connecte le DSK avec le PC et la source d'alimentation.

5.2 Considérations générales sur un processeur dédié au traitement numérique du signal

Dans ce qui suit on présente quelques caractéristiques générales des DSP avec des références concrètes au processeur utilisé.

5.2.1 Traitement analogue versus traitement numérique

Pour le traitement du signal, il existe deux techniques, l'approche analogique et la méthode numérique. Chacune a ses avantages et ses désavantages.
Pour la technologie analogique on peut noter les avantages suivants :

- large bande de fréquence et grande résolution
- les signaux n'ont pas à être convertis
- ajustement facile et rapide
- méthode d'analyse et de conception très bien connue

Il existe néanmoins une série de désavantages liés à cette technologie :

- sensibilité au bruit
- dérive en température
- vieillissement des composantes
- on peut implémenter seulement des conceptions simples
- aucune capacité de stockage

La technologie numérique s'est imposée dernièrement grâce à ses avantages :

- très peu sensibles à l'environnement (température, vieillissement, amplitude du signal, bruit électrique)
- solution programmable
- les algorithmes peuvent être complexes
- grande capacité d'adaptation
- capacité de communication et de stockage

Certaines des limitations auxquelles doit faire face un système de traitement numérique sont données par :

- il nécessite convertisseur (CAN/CNA)
- les délais de calcul limitent la bande passante du système et peuvent affecter la stabilité du système
- les erreurs numériques peuvent affecter la précision des résultats
- très exigeant en puissance du CPU

5.2.2 Numérisation du signal

Les processeurs numériques du signal DSP sont des processeurs spécialisés qui essaient de surpasser les limitations présentées. Tout d'abord, les DSP sont travaillés avec des numéros. Pour pouvoir utiliser le DSP dans le traitement de signal il faut numériser le

signal. Les deux opérations nécessaires sont l'échantillonnage et la quantification du signal.

L'échantillonnage se réalise en respectant le théorème d'échantillonnage. En général, les échantillons sont prélevés périodiquement avec une période T_e appelée période d'échantillonnage. Quelle que soit la valeur choisie pour T_e, le signal obtenu après échantillonnage sera toujours un signal à temps discret (ou échantillonné) et donc une approximation du signal analogique $x_a(t)$.

La quantification consiste dans une discrétisation dans l'amplitude. Si pour l'échantillonnage les choses sont assez claires, la quantification du signal supporte plusieurs approches et on choisit la loi de quantification optimale en fonction de l'application.

5.3 Entrée sortie dans un système de traitement numérique du signal

Les applications typiques qui utilisent le DSP demandent la présence d'un convertisseur A/D qui a le rôle de transformer le signal analogique d'entrée dans un signal numérique qui serait traité par le DSP. Pour cette raison le signal d'entrée passe par un filtre passe-bas qui élimine les fréquences supérieures à la fréquence de Nyquist.

Le convertisseur A/D réalise une approximation numérique du signal analogique réel. Par exemple on considère un ADC sur 8 bits et un domaine pour le signal d'entrée de ± 1.5V. La plus petite variation saisissable de l'ADC est le domaine divisé par 2^8, donc 3/256 = 11.72 mV. Cette quantification produit des erreurs de jusqu'à ± 5.86mV. Pour les valeurs d'entrée qui ne sont pas multiples entiers de 11.72 mV, on a une approximation à la sortie de l'ADC. La quantification d'un signal analogue implique une perte d'information résultant de l'ambiguïté introduite par quantification. On peut facilement déduire que plus le nombre de bits de l'ADC est grand, plus l'erreur de

quantification est petite. La qualité de sortie d'un ADC est mesurée par le RSBQ. Cette notion est présentée plus en détails dans le chapitre § 3.2 2.

Le convertisseur D/A a le rôle de transformer le signal numérique de sortie dans un signal analogique correspondant. Cette opération consiste dans une interpolation d'ordre 1 ou plus élevée. En pratique cela se fait en général à l'aide d'un filtre analogique nommé filtre de lissage.

Le DSK utilisé contient le convertisseur AD533 [35] qui utilise une technologie sigma-delta et réalise une conversion sur 16 bits à une fréquence d'échantillonnage fixe de 8 kHz. La gamme dynamique du signal d'entrée est 3V p-p. Les connecteurs qui fournissent l'entrée et la sortie sont notés IN(J7) et OUT(J6), l'accès se fait avec un câble audio de 3.5 mm. La communication avec le processeur se réalise via le port MCBSP0. Le DSK dispose de 16 MB de mémoire SDRAM et 128 kB de mémoire ROM.

5.4 Architecture du système

L'architecture du système est la façon dont les éléments d'un système à microprocesseur sont interconnectés et changent l'information. L'architecture a une grande influence sur la manière dont le CPU accède aux périphériques. Les architectures typiques sont : Von Neumann et Harvard.

L'architecture Von Neumann est apparue en premier et est la plus répandue aujourd'hui. Les différents éléments du système (CPU, mémoire, périphériques) sont interconnectés par un système unique de trois buses : bus de données, bus d'adresses et bus de contrôle et il ne dispose que d'un seul espace mémoire. Suite à cette structure un processeur dans une architecture Von Neumann peut faire une seule lecture ou écriture donc un seul accès mémoire dans un cycle d'horloge.

Dans une architecture Harvard existent deux systèmes de bus séparés. Le bus programme est réservé aux transferts des instructions de la mémoire vers le CPU. Le bus donnés est réservé aux échanges de données entre le CPU et les périphériques. Suite à cette structure, l'architecture Harvard permet au CPU d'acheminer le code et échanger des données avec les périphériques en même temps. L'architecture Harvard permet une exécution plus rapide des instructions. Elle est aussi plus sure car il n'y a aucun risque que le CPU aille écrire des données dans la mémoire programme et corrompe le programme. Le problème d'auto corruption du programme donne lieu à des comportements logiciels difficiles à cerner.

5.4.1 L'architecture des DSP

En adoptant l'architecture Harvard, les DSP présentent au moins deux espaces mémoire, données et programme, qui peuvent être appelées dans un seul cycle d'horloge. De plus, différents DSP ont des techniques ad-hoc pour réduire la bande passante : répéter une instruction (jusqu'à 256 fois), désactiver les interruptions. Les DSP les plus récents ont de la mémoire cache.

Pour l'adressage de la mémoire, les DSP détiennent des unités spécialisées qui permettent :

- adressage direct, DMA (direct memory access)
- incrémentation automatique
- adressage immédiat
- adressage circulaire (utile pour la convolution)
- adressage bit-reverse (pour FFT).

Le processeur TMS320C6711 [37] inclut deux niveaux de mémoire cache, le premier niveau et séparé en deux segments de 4 kB pour programme et données respectivement.

Le système interne de buses et formé d'un bus d'adresse programme de 32 bits et un bus de données programme de 256 bits qui permet d'acheminer 8 instructions de 32 bits à la fois, deux buses d'adresse de données de 32 bits et quatre buses de données de 64 bits deux pour amener les données au processeur et deux pour stoker les données dans la mémoire.

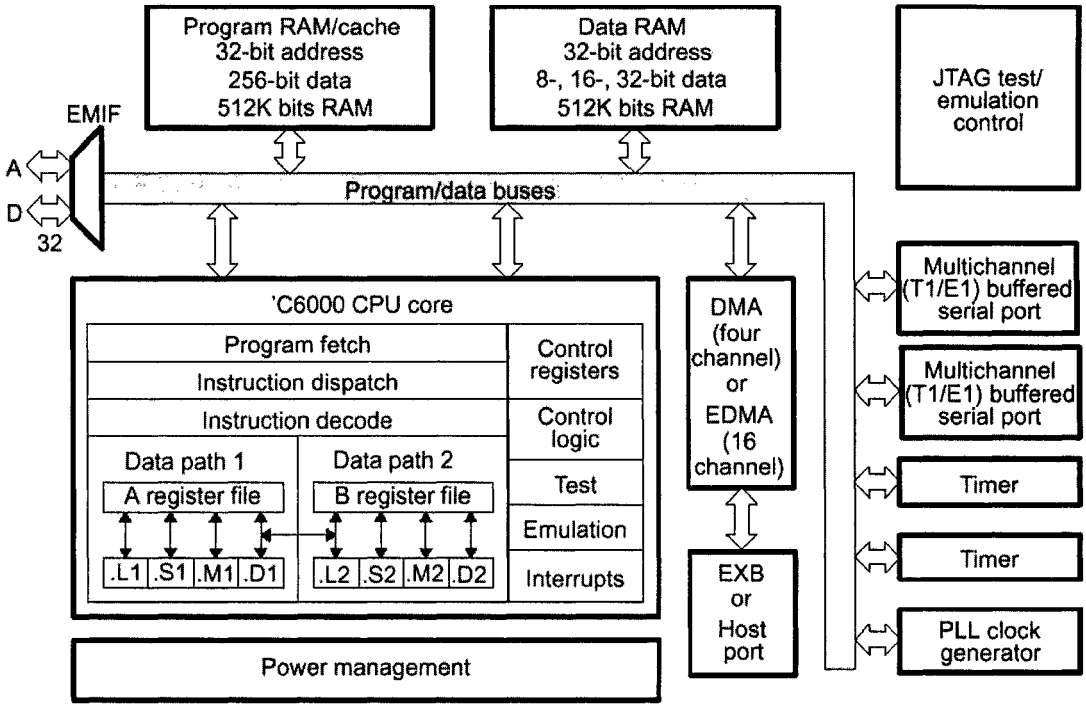


Figure 39 TMS320C6x et périphériques – diagramme bloc [36]

Les périphériques [37-38] sont accessibles par deux ports serials dotés de mémoire tampon McBSP, un port hôte de 16 bits HPI et une interface pour la mémoire externe de 32 bits qui permet d'accéder jusqu'à 4GB de données organisées en 4 espaces de mémoire externe. Les blocs des mémoires séparés permettent deux accès mémoire en parallèle dans un seul cycle.

L'adressage direct permet les transferts de données entre la mémoire interne et un dispositif externe sans intervention du CPU. On dispose de quatre canaux séparés configurables pour le transfert de données et un canal réservé pour réaliser le DMA avec le HPI.

5.4.2 Les unités fonctionnelles et les registres

Le CPU possède 8 unités fonctionnelles indépendantes groupées en deux blocs 1 et 2. Chaque bloc consiste en 4 unités spécialisées pour effectuer certaines opérations :

- l'unité M. pour les opérations de multiplication point fixe/flottant
- l'unité L. pour les opérations logiques et arithmétiques point fixe/flottant
- l'unité S. pour les opérations de branchement, de manipulation au niveau de bit et arithmétiques point fixe/flottant
- l'unité D. pour les opérations d'échange de données et arithmétiques point fixe seulement

Chaque bloc contient un set de 16 registres de 32 bits d'usage général avec certaines restrictions. Les registres de A0 à A16 appartiennent à l'unité 1 et les registres de B0 à B16 appartiennent à l'unité 2. Les registres A0, A1, B0, B1 et B2 sont utilisés pour réaliser les instructions conditionnalités. Les registres de A4 à A7 et de B4 à B7 sont utilisés pour réaliser l'adressage circulaire. Les registres de A0 à A9 et de B0 à B9 excepté B3 sont des registres temporaires lorsque chacun des registres de A10 à A15 utilisé est sauvé et reconstitué pendant l'appel d'une procédure.

Une valeur de 40 bits peut être représentée en utilisant une paire de registres, les 32 moins significatifs bits sont contenus dans le registre pair et les 8 bits qui restent dans les 8 bits moins significatifs du registre impair. Une technique similaire est utilisée pour représenter une valeur double précision sur 64 bits.

Chaque bloc fonctionnel peut accéder les registres propres mais aussi les registres qui appartiennent au bloc opposé.

5.4.3 L'adressage

Le processeur TMS320C6711 supporte l'adressage linéaire et l'adressage circulaire. Le mode d'adressage le plus couramment utilisé est l'adressage indirect. On utilise l'un des registres d'usage général comme un pointer vers l'adresse mémoire pour déposer ou trouver la donné d'intérêt. On peut effectuer l'adressage indirect avec ou sans déplacement.

L'adressage circulaire est implémenté matériel. Il est utilisé en conjonction avec deux buffers circulaires, disponibles via BK0 et BK1. Les dimensions et les registres contenant les adresses des deux buffers BK0 et BK1 sont indiqués dans le registre de mode d'adressage AMR.

On utilise l'adressage circulaire pour implémenter d'une façon plus efficace certains algorithmes très répandus dans le traitement numérique du signal tels que le filtrage numérique ou le calcul de la fonction d'autocorrélation.

5.5 Format de représentation des nombres

Basés sur les architectures Harvard modifiées, les DSP disposent du matériel spécialisé pour effectuer toutes les opérations arithmétiques dans un seul cycle et assurent une bonne représentation numérique. Il y a le support matériel permettant les déplacements, l'existence de bits de retenue ou saturation. Les DSP utilisent des nombres réels, entiers ou fractionnaires pour réaliser les calculs. D'après le format de représentation des nombres on a des DSP point fixe et point flottant.

5.5.1 Erreurs dues à la représentation

Une contrainte importante pour un système de traitement consiste à commettre des erreurs de calcul qui soient négligeables vis-à-vis de l'incertitude du signal lui-même.
A cause de la précision finie pour la représentation des nombres dans le DSP, il peut intervenir plusieurs erreurs durant les calculs.

L'erreur de dépassement (overflow) arrive dès que le résultat d'une opération dépasse la capacité de représentation pour la notation choisie. Selon cette notation le dépassement peut devenir un retournement, quand lors d'une addition le résultat traverse la frontière de signe. Un dépassement a en général un effet catastrophique sur une opération arithmétique. Un résultat qui aurait dû prendre une valeur positive très élevée va prendre à la place une valeur négative. Pour éviter un dépassement, le DSP peut utiliser la saturation à la plus grande valeur du signe adéquat, dans le cas d'une résultat affecté d'une telle erreur. Cette fonctionnalité est réalisée de façon matérielle et ne demande pas de puissance de calcul supplémentaire.

En fonction de la résolution de la représentation une valeur réelle positive peut être tronquée au nombre représentable juste inférieur ou pour une valeur négative au nombre représentable juste supérieur ; on a ainsi une erreur de troncature.

Un arrondi suppose l'attribution de la valeur représentable la plus proche de la valeur réelle. Les erreurs d'arrondi s'accumulent lentement au cours du calcul d'une somme et influencent éventuellement le résultat. Il faut donc toujours être prudent et analyser tous ces aspects quand on choisit un algorithme.

Les annulations (underflow), en point flottant arrive quand on additionne des valeurs très éloignées, une ou plusieurs chiffres du résultat disparaissent à cause de la largeur limitée du mot de l'ordinateur.

5.5.2 Processeur point fixe versus point flottant

Pour les DSP point fixe la taille du mot affecte la précision. Il y a des DSP point fixe de 16 bits, 20 bits ou 24 bits. Pour le DSP point fixe le programme doit maîtriser l'échelle à l'intérieur du code, les phénomènes de débordements (overflow), annulation (underflow) et saturation. Pour cela le programmeur a quelques options offertes par le processeur : avoir un ou plusieurs bits de retenue (utilisant le registre accumulateur, ex. Motorola sur 24 bits dispose d'accumulateurs de 56 bits), déplacement à gauche ou à droite avant les opérations mathématiques d'addition ou multiplication.

La réalisation de DSP point-fixe est plus simple mais elle exige un conditionnement adéquat des algorithmes de traitement. Ces processeurs demandent moins de puissance électrique et coûtent moins cher que les processeurs point flottant.

Pour le processeur point flottant, le développement du logiciel est assez simple parce qu'il offre une plage de représentation de nombres beaucoup plus élevée mais cela au détriment de la vitesse de calcul, de la consommation et du prix.
De ce point de vue le TMS320C6711 est un processeur point flottant [39].

5.6 Les interruptions

Une interruption peut être générée de façon interne ou externe. Une interruption arrête le processus courant du CPU pour qu'il puisse réaliser une tâche demandée par l'interruption. Le flux du programme est redirectionné vers une routine de service d'interruption (ISR). La source d'interruption peut être une ADC, un temporisateur, etc. Pendant une interruption, l'état du processus doit être sauvegardé afin d'être continué à la fin de l'interruption.

Pour les processeurs TMS320C6x, il existe 16 sources d'interruptions [37], parmi lesquelles 2 temporisatrices, 4 interruptions externes, 4 interruptions McBSP et 4 interruptions DMA.

5.7 Vitesse du processeur

L'une de plus restrictives contraintes imposées au DSP est le travail en temps réel. Les algorithmes qui supposent le temps réel exigent qu'un cycle de calcul soit fait dans un intervalle maxime de temps qui sépare deux événements successifs. Dans le cas du traitement court-terme un cycle de calcul est l'intervalle de temps nécessaire à l'acquisition d'une trame complète de signal. Les délais de calcul limitent la bande passante du système et peuvent affecter la stabilité du système.

Le temps nécessaire au processeur pour effectuer un certain algorithme est une constante qui dépend de la fréquence de travail du DSP et de la complexité d'algorithme (nombre d'opérations mathématiques requis). Cet intervalle de temps doit être plus petit que l'intervalle de temps équivalent à l'acquisition d'une trame de signal.

Pour augmenter la puissance du processeur il y a deux techniques. Une première technique est d'augmenter la fréquence du travail du processeur ce qui lui permet d'exécuter plusieurs cycles dans le même intervalle de temps. Une autre est de faire plusieurs opérations dans un même cycle du processeur. C'est à dire d'essayer l'exécution de plusieurs opérations en parallèle. Les DSP modernes tels que le C6711 [36] disposent de support matériel et logiciel qui offre la possibilité de traitement parallèle mais elle n'est pas automatique. Le programmeur doit adapter l'algorithme de telle façon que cette possibilité soit mise en valeur.

5.7.1 Le parallélisme dans le processeur TMS320C6711

Le processeur TMS320C6711 utilise la notion de paquet d'exécution EP qui consiste dans un groupe d'instruction qui sont exécutées en parallèle dans le même cycle [40]. Comme on a vu le bus de données programme de 256 bits permet d'acheminer 8 instructions à la fois vers le processeur appelé paquet reçu (fetch packet) FP. Le nombre de EP dans un FP peut varier de 1, avec 8 instructions en parallèle, à 8 quand il n'a pas de parallélisme.

Idéalement, on peut utiliser toutes les 8 unités fonctionnelles du processeur dans le même cycle. Toutefois, pour arriver à cela un effort substantiel est demandé de la part du programmeur car il doit adapter l'algorithme pour supporter le traitement parallèle et réaliser la synchronisation dans l'exécution. Dans ces conditions idéales, avec une fréquence de 150 MHz, on peut arriver à 1200 millions instructions par seconde (MIPS) avec un temps moyen d'instruction de 6,67 ns.

5.8 Les instructions

Les DSP ont des instructions caractéristiques spécialisées et complexes qui permettent d'effectuer plusieurs opérations dans une seule instruction. Les instructions sont aussi capables de manipuler plusieurs dimensions de données comme on peut voir dans les exemples suivants.

Pour les processeurs TMS320C6x, la forme générale d'une instruction est [41-43] :

```
Etiquette || [] Instruction Unité Opérants ; commentaires
```

L'étiquette représente une adresse de mémoire qui contient instructions ou données. Les barres parallèles existent si l'instruction s'exécute en parallèle avec l'instruction précédente. Le champ suivant est optionnel et il existe si l'instruction s'exécute d'une façon conditionnelle. Cinq registres A1, A2, B0, B1, B2, peuvent être utilisés comme des registres conditionnels. Par exemple, [A2] signifie que l'instruction associée s'exécute si la valeur d'A2 est différente de 0. Autrement, [!A2] signifie que l'instruction associée s'exécute si la valeur d'A2 est 0. On indique l'unité qu'on veut exécuter, l'instruction et les opérants associés dans les derniers champs de l'instruction.

Les types des instructions suivants sont disponibles :

- addition, soustraction, multiplication

Ex. : `ADD .L1 A3, A7, A7 ;` addition A3+A7→A7

Additionne les valeurs des registres A3 et A7 et place le résultat dans le registre A7. L'unité .L1 est optionnelle. Si la destination était B7, l'unité aurait été .L2.

- appel et stockage dans la mémoire

Ex. : `STW .D2 A1, * +A4[20] ;` stockage A1→ (A4) augmenté avec 20

Stocke le mot de 32 bits A1 dans l'adresse de mémoire spécifiée par la valeur du registre A4 qui a été augmentée avec 20 mots (des 32 bits). L'adresse contenue dans le registre A4 est pré-incrémentée mais elle n'est pas modifiée (le signe ++ est utilisé si on veut modifier le contenu du registre).

- branchement, déplacement de données

Ex. : `Loop MVK .S1 x, A4 ;` déplace 16 LSBs de x à l'adresse A4

```
                .
                .
                .
[A1]   B    .S2 Loop ;
```

va à l'étiquette *Loop* si A1 est différent de 0.

La première instruction déplace la moitié inférieure (16 MSB) du mot de l'adresse `x` dans le registre A4.

5.9 Le Code Compose Studio

Le CCS [44-46] fournit le support logiciel nécessaire pour développer les applications d'une façon très efficace. C'est un environnement de développement intégré qui réunit le compilateur C, l'assembleur, l'éditeur de liaisons. Il facilite la correction des programmes, a des capacités de représentation graphique d'analyse et de contrôle en temps réel et encore d'autres options qui facilitent la tâche du programmeur et accroît le rendement de l'implémentation.

Le compilateur C/C++ compile le programme source avec l'extension `.c` et produit un fichier en langage assembleur avec l'extension `.asm` spécifique au DSP. L'assembleur assemble le fichier `.asm` et produit un objet en langage machine avec l'extension `.obj`. L'éditeur de liaisons combine les fichiers objet et les objets qui appartiennent à la bibliothèque pour produire un fichier exécutable avec l'extension `.out` qui est chargé et exécuté directement par le DSP.

L'outil DSP/BIOS [47-48] permet, parmi d'autres facilités, l'analyse et l'échange de données en temps réel, l'organisation de la mémoire et la gestion des interruptions dans un environnement visuel composé de plusieurs modules d'interface programmée. Les objets utilisés par cet outil sont définis dans un fichier de configuration avec l'extension `.cdb`.

Par exemple l'échange de données en temps réel (RTDX) entre le PC hôte et la carte DSK se réalise via l'interface JTAG (Joint Test Action Team) pendant que le processeur travaille. Ainsi on peut visualiser le graphe d'exécution qui montre quand sont exécutés

les différentes tâches par le processeur et s'il manque de temps pour le traitement temps réel.

Les outils de correction permettent d'imposer des points d'arrêt, de visualiser les variables, les registres et régions de mémoire. Le compilateur C génère le code en assembleur de telle façon que le programmeur puisse voir les instructions générées pour chaque ligne de code source en langage C. On peut visualiser le nombre de cycles machine associés à une instruction ou une fonction et représenter graphiquement les données.

5.10 Réalisation pratique

5.10.1 Test de fonctions

L'implémentation de l'algorithme est faite en deux étapes. Tout d'abord les fonctions programmées en langage C qui réalisent l'algorithme sont testées en utilisant des fichiers pour obtenir les mêmes résultats comme dans le cas des simulations qui utilisent le logiciel Matlab.

On organise la fonction principale de telle façon qu'on puise après l'adapter facilement aux exigences du travail en temps réel. Le fichier audio originel qui contient les échantillons du signal vocal est converti dans un fichier texte à l'aide du logiciel Matlab. Ce fichier constitue l'entrée dans le programme de test. A ce niveau on est intéressé par le fonctionnement correct de chaque fonction programmée. Pour cela on génère plusieurs fichiers texte contenant les résultats de chaque fonction programmée. Après les simulations ces fichiers sont chargés en Matlab et comparés avec les résultats obtenus en utilisant les fonctions programmées en Matlab. Étant donné qu'on utilise les mêmes données d'entrée et la même précision dans les calculs il faut obtenir les mêmes résultats dans les deux simulations.

La fonction utilisée pour le calcul du spectre de chaque séquence de 160 échantillons appelle deux fois la fonction qui réalise la TF de premiers et respectivement de 128 derniers échantillons. Lorsque le signal vocal est une série de nombres naturels on a utilisé l'algorithme présenté dans l'annexe 2 pour le calcul efficace de la TFR d'une séquence réelle de longueur $2N$, $N = 64$ est le nombre de points de la TF.
Une fois cette étape de test réalisé avec succès, on peut passer à l'implémentation de l'algorithme sur le DSP.

5.10.2 L'implémentation sur DSP

Dans une deuxième étape on adapte la fonction principale pour travailler avec l'entrée et la sortie du DSK en utilisant les interruptions dans le but de tester de façon concrète le fonctionnement temps-réel du programme réalisé.

Pour implémenter une application en CCS on définit tout d'abord un projet qui est une collection de répertoires contenant plusieurs types de fichiers différents nécessaires à la réalisation de l'algorithme désiré. A ce niveau on peut combiner les fichiers source programmés en langage C et assembleur avec les fichiers de bibliothèque pour parvenir le plus efficacement possible à la réalisation du fichier exécutable.

Pour la réalisation de l'algorithme exposé au chapitre précédent on utilise une série de fichiers support [34] qui sont englobés dans le projet et dont la nécessite sera exposé ensuite.

On réalise la configuration de la mémoire avec le fichier de commande `C6xdsk.cmd`. Son rôle est d'allouer dans les sections définies dans la mémoire physique les blocs de programme et de données. Ces sections peuvent être initialisées ou non et en fonction de

leur fréquence d'accès elles sont disposées dans la mémoire interne, qui est plus rapide, ou dans la mémoire externe.

Le fichier d'en tête `C6xdsk.h` qui définit les adresses de la mémoire externe, ports serials etc, est inclus dans CCS. Le fichier `C6xinterrupts.h` qui contient les fonctions qui initialisent les interruptions est un fichier support fourni par TI. Le fichier `C6xdsk.h` contient les prototypes des fonctions utilisées. Tous ces fichiers sont regroupés dans le répertoire Include du projet. Dans le même répertoire se trouve la fonction `compw.c` qui est appelée une seule fois en début du programme pour initialiser le vecteur qui contient les valeurs de V^k, k=0,N-1 nécessaires pour le calcul de la TFR.

Dans le répertoire Librarie on a le fichier `rts6701.lib` qui est un fichier de librairie fourni par TI et qui supporte les architectures C67x/C62x.

Le répertoire Source contient les fonctions source du projet `.asm`, et `.c`. On place ici le fichier de communication `c6xdskinit.c`. qui et formé des fonctions qui initialisent le DSK, le convertisseur AD535 et les deux port serial. Une fois le DSK initialisé l'interruption INT11 qui permet la communication avec le McBSP est configurée et activée par la fonction `comm_intr.c`. Deux autres fonctions sont utilisées pour réaliser l'entrée et la sortie d'un échantillon dans le programme. La fonction `input_samples` lis la valeur du registre McBSP0_DRR (Data Reciver Register) du McBSP et la fonction `ouput _sample` écrit une valeur dans le registre McBSP0_DTR (Data Transmiter Register) du McBSP.
Le fichier `vectors_11.asm` fournit l'adresse de la routine de service d'interruption (ISR), `c_int11` qui se trouve dans la fonction principale du programme. Pour cela un branchement à l'adresse de ISR est indiqué dans `vectors_11.asm` quand l'interruption INT11 se produit.

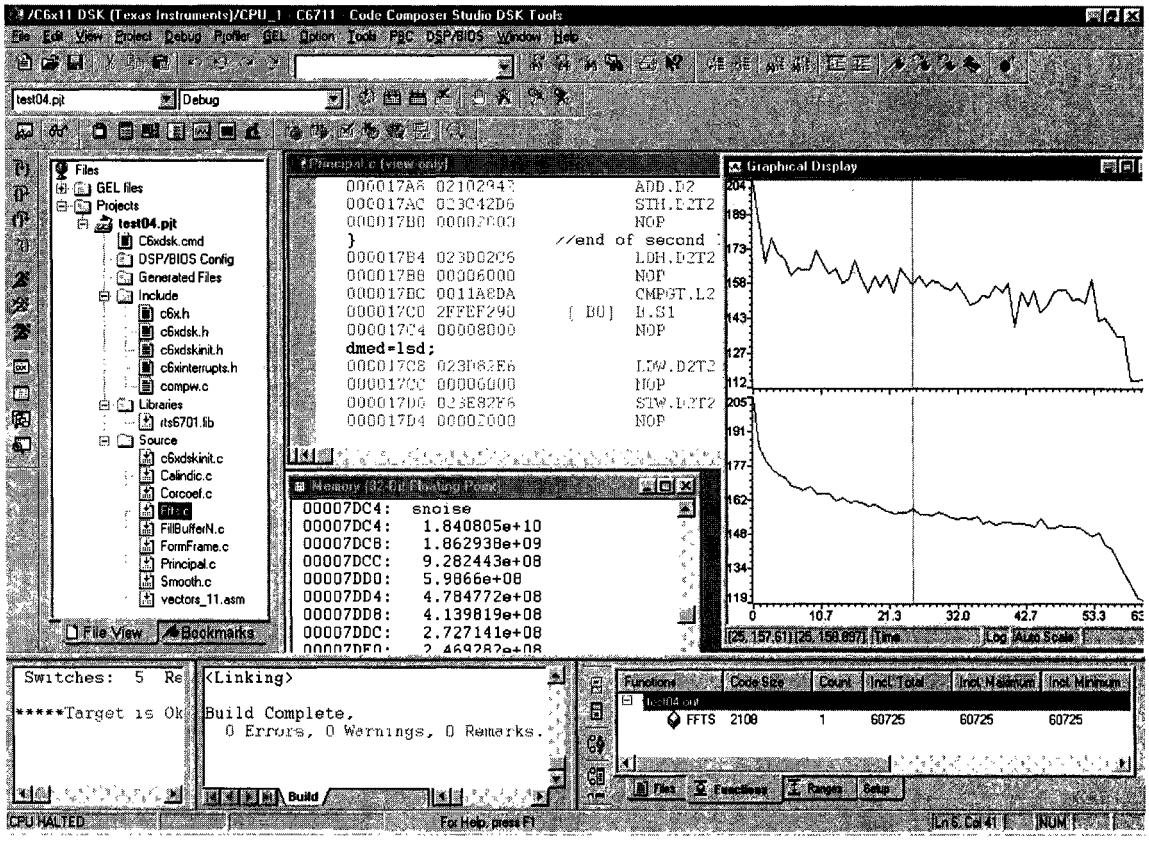


Figure 40 L'interface du CCS

Le CCS met à la disposition du programmeur toute une panoplie d'options pour la compilation et l'édition de liaisons. Les options de compilation permettent entre autres de garder les fichiers `.asm` et d'intercaler les instructions en langage c et assableur ce qui est utile pour l'étape de correction du programme. On peut aussi indiquer une implémentation point fixe ou point flottant et plusieurs niveaux d'optimisation.

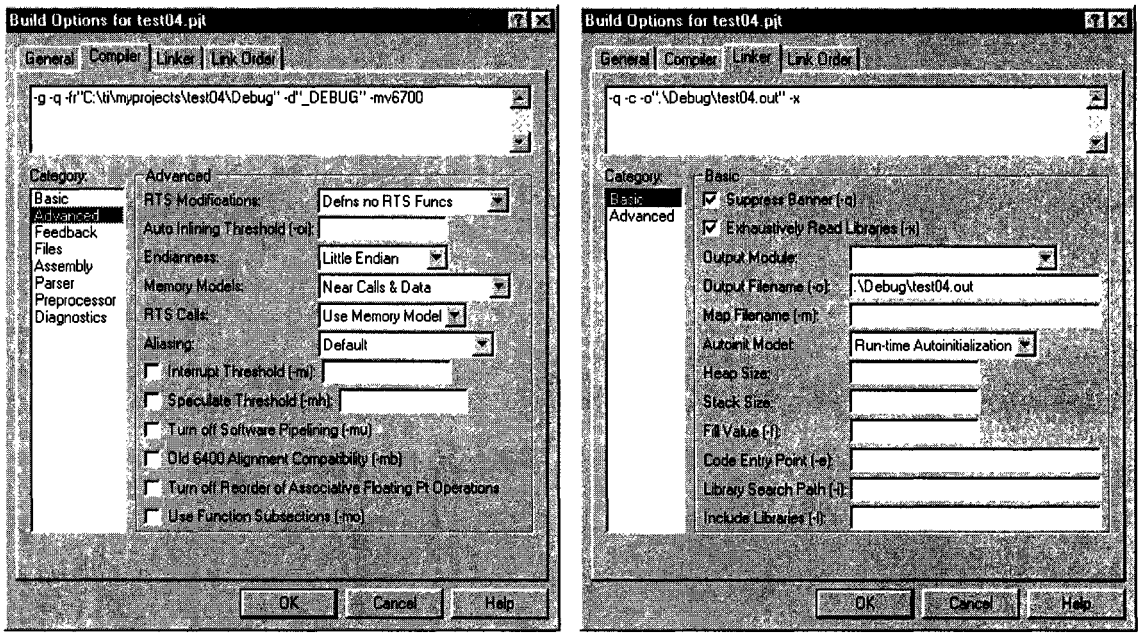


Figure 41 Fenêtres d'options pour la compilation et l'édition de liaisons

5.10.3 Explication du programme

L'organigramme du programme est représenté dans la figure 42. Pour les tests temps-réel on utilise la carte de son du PC pour envoyer le signal vocal vers le DSK via le connecteur IN(J7), le signal de sortie est envoyé vers un haut parleur connecté à la borne OUT(J6).

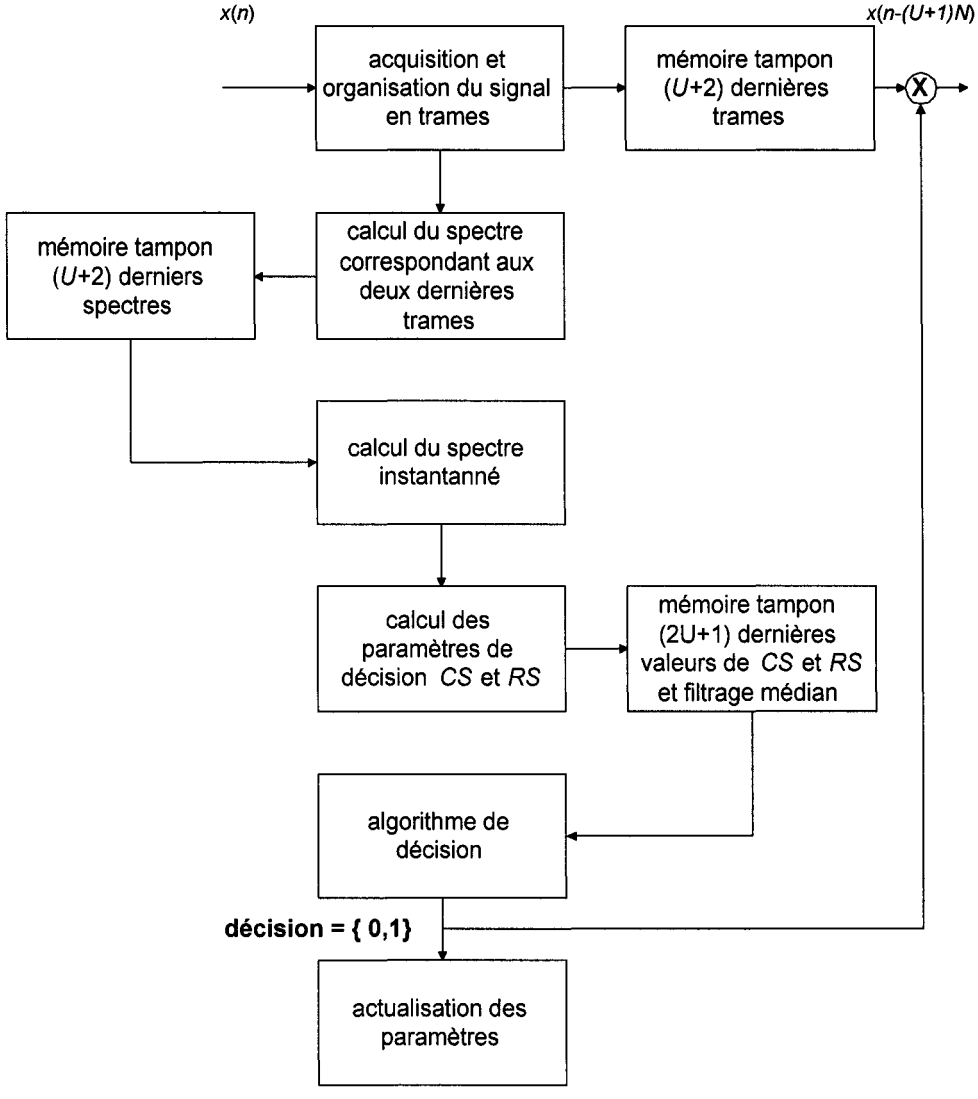


Figure 42 Organigramme de l'algorithme de VAD proposé

Après l'étape d'initiation, dans la fonction principale une instruction `while(1)` réalise une boucle infinie. Dans le corps de cette boucle est placée tout le code. Une interruption arrive à chaque 0.125 ms et le flux de programme est direction vers ISR, la fonction `c_int11`. On utilise deux vecteurs de dimension 80 éléments de type `char` pour mémoriser les 160 derniers échantillons du signal. La mémoire tampon qui contient les 8

dernières trames du signal est organisée sous la forme d'un bouffer circulaire et est utilisé lorsqu'on veut écouter la partie du signal détecté comme parole.

Après chaque 80 échantillons reçus on calcule le spectre correspondant aux 160 derniers échantillons, tel que présenté dans le chapitre précédent. Ce spectre est gardé dans une mémoire tampon avec les 7 derniers spectres qui le précèdent. On utilise les trois derniers spectres pour obtenir le spectre instantané. Le spectre du bruit est actualisé en fonction de la sortie du bloc de décision avec le spectre qui se trouve 8 trames en arrière et d'ici la nécessite de garder les 8 derniers spectres.

Les deux spectres, instantané et du bruit, sont utilisés pour calculer les deux paramètres *CS* et *RS*. Le logarithme naturel de 13 dernières valeurs de ces deux paramètres sont gardés dans deux vecteurs au but de réaliser le filtrage médian. La valeur du paramètre global *PG* est passée au bloc de décision.

L'étape de décision, telle qu'exposée au chapitre précédent, utilise la valeur de *PG* et une série de paramètres dont les valeurs sont initialisées en début de l'algorithme et qui sont actualisées en fonction de la décision antérieure. A cause de la longueur du filtre médian, la décision à l'instant n est pour la trame $n-7$, pour cela on doit mémoriser les 8 dernières trames du signal dans une application ou on veut transmettre ou reproduire juste la partie de parole du signal total.

5.10.4 Méthodologie du test de l'implémentation sur DSP

Le test du fonctionnement temps réel de l'implémentation de l'algorithme exposé sur DSP pose quelques problèmes spécifiques lorsqu'on veut comparer les valeurs obtenues en utilisant le DSP et celles fournies par les simulations en Matlab. En principe on peut charger le même signal vocal utilisé pour les simulations Matlab dans la mémoire qui se trouve sur la carte DSK en utilisant les *points-probes* du CCS. Une fois le signal

disponible on peut utiliser une version modifiée du programme, qui exclut les interruptions, pour obtenir certaines valeurs d'intérêt qui seront transférées en Matlab pour réaliser la comparaison. Cette approche, même si elle teste le fonctionnement correct de l'algorithme sans introduire une surcharge de calcul, présente plusieurs désavantages. Tout d'abord elle ne teste pas vraiment le comportement temporel de l'algorithme et deuxièmement elle n'utilise pas le convertisseur et donc omet toute erreur d'acquisition.

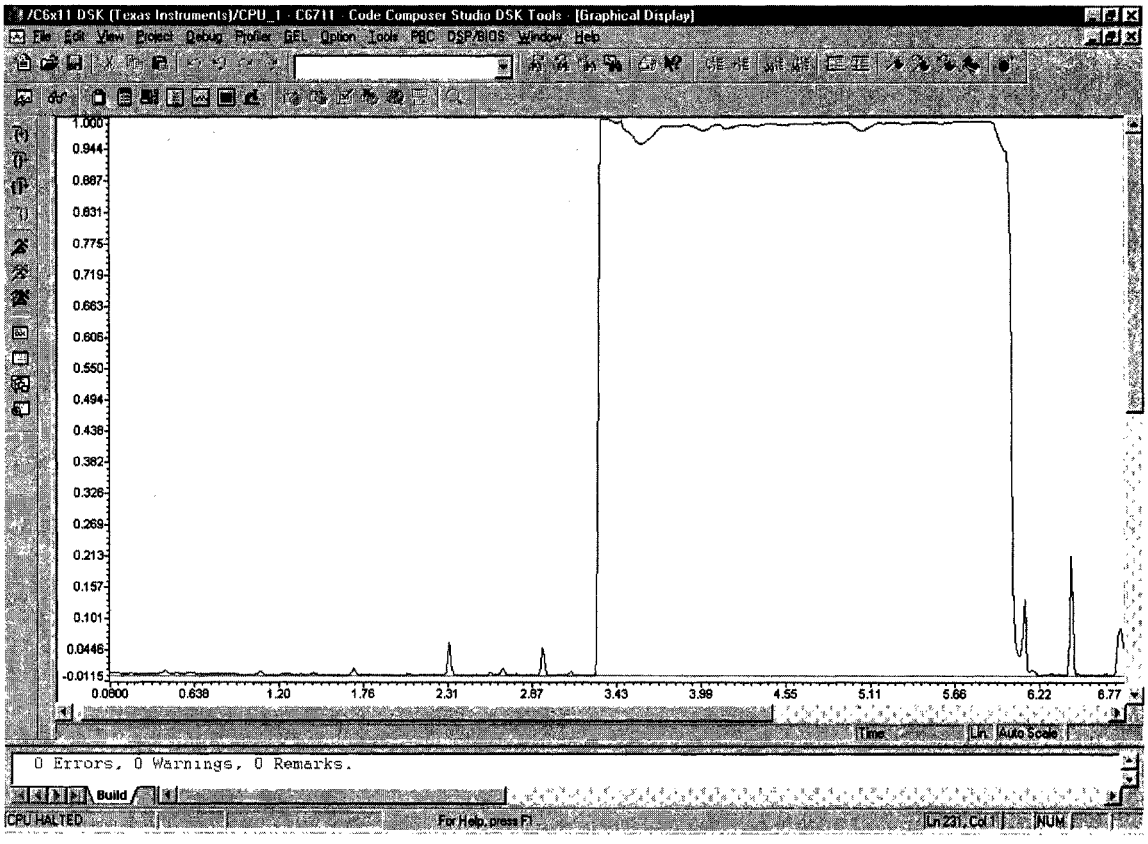


Figure 43 Résultats de simulation temps-réel sur DSP (paramètre P_n)

Pour éliminer ces inconvénients on a élaboré une méthodologie différente de test. On définit un vecteur de 80 000 éléments dans la mémoire du DSK qui est accessible via un pointer `far` et qui est utilisée pour garder tous les échantillons du signal. Ce vecteur est

suffisant pour mémoriser 10 s du signal vocal et donc pour le test on utilise des segments de signal vocal plus court que 10 s. La carte de son du PC est utilisée pour envoyer le signal vers le convertisseur de la carte DSK. On utilise un autre vecteur de 1000 éléments pour garder les valeurs d'intérêt telle que le paramètre global *PG* ou la décision pour chaque trame du signal. Une fois le signal vocal envoyé vers les DSK en totalité on arrête l'algorithme. On transfère les deux vecteurs qui contiennent le signal vocal qui a été réceptionné par le DSK et le résultat de la simulation dans deux fichiers sur le HD du PC.

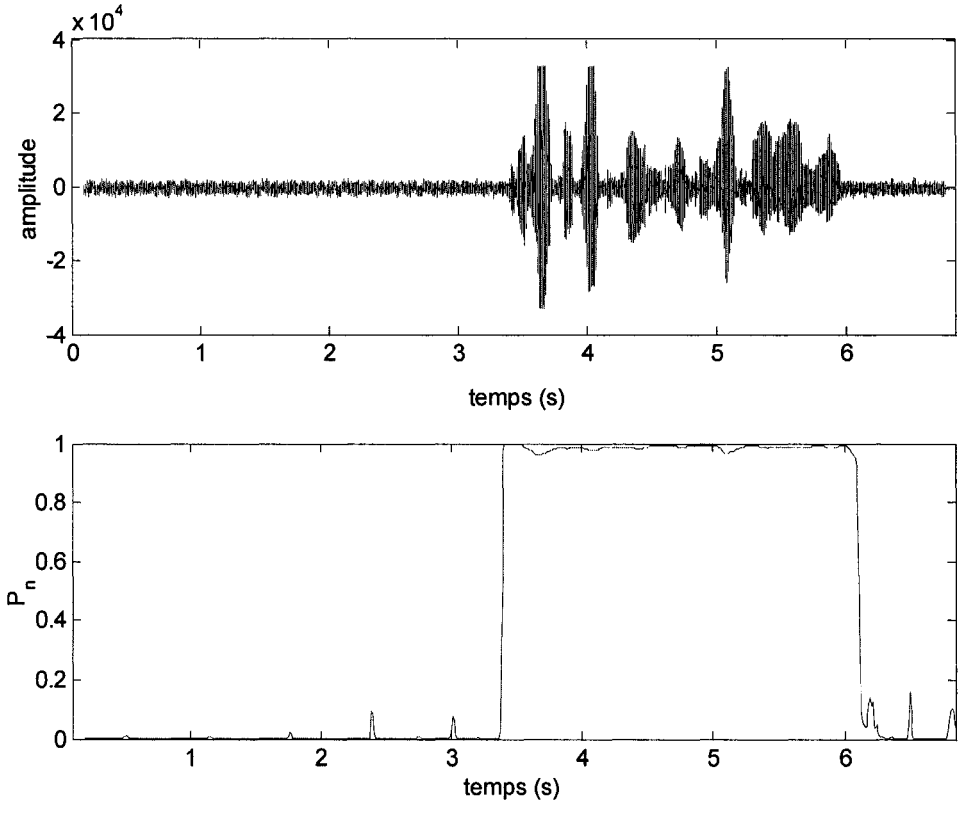


Figure 44 Résultats de simulation Matlab (paramètre P_n)

On utilise le signal vocal ainsi obtenu comme l'entrée de la fonction Matlab qui réalise l'algorithme proposé. Les résultats des deux simulations peuvent être maintenaient comparés. Dans le cas d'un fonctionnement correct de l'implémentation sur DSP il ne doit pas exister de différences entre les résultats des deux simulations.

5.11 Recommandations

Le facteur d'échelle utilisé dans la sommation des deux paramètres *CS* et *RS* peut être utilisé aussi pour accorder plus de poids au paramètre qui présente un meilleur comportement dans une application plus spécifique. Par exemple, dans une application où le niveau de bruit est fortement variable le coefficient de corrélation spectrale est plus robuste car il est totalement indépendant du niveau d'énergie du signal. Ainsi on peut utiliser le facteur d'échelle pour diminuer l'apport du paramètre *RS* qui, dans cette condition spécifique, ne peut pas aider à la décision.

La méthode, paramétrique ou non-paramétrique, utilisée dans le calcul du spectre du signal n'est pas essentielle, ainsi l'algorithme proposé peut partager le même ensemble de paramètres utilisés dans l'application dont il fait partie si c'est le cas.

Pour des applications où le RSB et assez élevé la dimension du filtre de lisage peut être diminuée et ainsi réduire le P_f sans diminuer le P_d.

Si on veut utiliser cet algorithme dans une application de transmission de données temps-réel il faut tenir compte du fait que le lissage médian introduit un retard de L trames dans la décision, pour un filtre de longueur $2L+1$.

On peut utiliser une version modifiée de cet algorithme de VAD dans une application plus grande qui implique la modélisation AR du signal vocal. Le coefficient de corrélation spectrale peut être remplacé par la distance euclidienne entre les coefficients

de réflexion et la moyenne de RSB de sous-bandes par le rapport des énergies. Ainsi l'algorithme introduirait un minium de calcul pour la VAD.

CONCLUSION

Un algorithme de détection vocale simple, précis et robuste est souhaitable dans plusieurs systèmes de traitement de la parole. La reconnaissance, la transmission ou le rehaussement de la parole emploient souvent un module de VAD pour améliorer les performances et minimiser le coût du traitement numérique. De plus dans certaines conditions on impose le fonctionnement temps réel de l'algorithme de VAD.

Plusieurs types d'algorithmes de VAD ont été proposés au cours des trois dernières décennies. La plus grande partie d'entre eux utilisent une ou plusieurs des paramètres suivants : l'énergie court-terme, le taux de passage par zéros, période de pitch, durée et diverses distances entres les coefficients de prédiction linéaire ou plus récemment les paramètres cepsatrales, les paires de railles spectrales ou les transformées en ondellettes. Dans la majorité des cas l'étape de décision se réalise par la comparaison des valeurs des paramètres extraits du signal d'entrée avec certains seuils prédéfinis. Les seuils de décision sont déduits d'une analyse préalable du comportement des paramètres utilisés pour le signal vocal pour une large gamme de types de bruits et de RSB.

Dans cet ouvrage on a proposé, testé et implémenté sur un processeur numérique de signal un nouvel algorithme robuste de VAD. L'étude de l'évolution temporelle du signal révèle la nature non stationnaire de celui-ci pour des intervalles longs de temps. Ainsi pour surpasser cette difficulté et être en mesure d'utiliser les techniques d'analyse spécifiques aux SLIT on a utilisé le concept d'analyse court-terme. Cette approche classique dans l'analyse du signal vocal utilise une fenêtre de pondération pour diviser le signal vocal en segments courts à l'intérieur desquels le signal vocal peut être considéré quasi stationnaire.

L'algorithme proposé utilise une méthode non paramétrique pour obtenir le spectre instantané du signal et mesure les différences de forme et d'amplitude entre le spectre

instantanée et le spectre du bruit de fond pour trouver certaines distorsions spectrales qui pourraient indiquer la présence de la parole. Pour cela on utilise deux paramètres : le coefficient de corrélation spectrale et la moyenne de RSB de sous-bandes.
Le premier paramètre, qui est un apport original, présente deux propriétés utiles dans le VAD. Ce paramètre est indépendant de la variation d'amplitude du signal vocal parce qu'il est une mesure normalisée de la dépendance linéaire entre les deux spectres et en même temps il est sensible aux changements de la forme des deux spectres.
Le deuxième paramètre a déjà été utilisé avec succès dans d'autres algorithmes de VAD [22,23]. Il est un indicateur sensible aux variations de l'énergie dans les sous-bandes du signal.

Dans des applications réelles il y aura toujours un bruit de fond dû aux autres sources de signal sonore existantes dans le milieu d'enregistrement. Ce bruit ambiant additif contamine le signal de parole et rend difficile l'extraction des caractéristiques propres au signal utile. Pour répondre aux exigences de robustesse nécessaires dans les applications réelles, on a réalisé un filtrage médian de deux paramètres en utilisant un filtre symétrique de longueur 13. Cette approche, utilisée d'une façon originale, nous a permis d'augmenter la probabilité de détection pour une même probabilité de fausse alarme.

La détection des régions de parole se réalise en utilisant un algorithme original basé sur une approche statistique qui utilisé le ratio de vraisemblance et une chaîne de Markov de premier ordre à deux états qui modélisent le processus de décision. Ce schème permet au seuil de décision de s'adapter au type de bruit et au niveau de RSB, ainsi on augmente la P_d et on obtient des régions de parole et silence plus compactes.

On a testé la robustesse de l'algorithme proposé à l'aide d'un signal vocal complexe qui présente plusieurs régions de parole est de silence de longueur variable corrompu avec plusieurs types de bruits réels dans le cas de trois RSB différents. Les résultats obtenus

montrent l'efficacité de l'algorithme proposé par rapport à un autre algorithme qui utilise un seuil de décision fixe.

Dans une dernière étape l'algorithme proposé à été implémenté sur le processeur numérique de signal TMS320C6711. L'objectif principal de cette étape a été de vérifier le fonctionnement temps réel de l'algorithme. Cette contrainte temporaire oblige l'algorithme de fournir la décision pour la trame courante avant qu'une nouvelle trame de signal soit disponible.
Pour cela on a utilisé la trousse des outils de développement fournie par TI qui inclue le logiciel CCS et la carte DSK. Le signal vocal analogique de test a été numérisé via le convertisseur AD535 qui se trouve sur la carte DSK en utilisant l'interruption INT11. La méthodologie de test utilisé nous a permis de comparer les résultats de simulation temps-réel sur le DSP avec celles obtenues en utilisant le logiciel Matlab.

L'ensemble des paramètres utilisés par l'algorithme a été choisi en observant le comportement de l'algorithme sur une base de données. Dans un travail futur on envisage la possibilité d'utiliser une méthode mathématique d'optimisation pour le calcul de ces paramètres.

On peut aussi tester les performances de cet algorithme de VAD par rapport aux autres algorithmes disponibles en l'introduisant dans une application de reconnaissance de la parole. Un meilleur taux de reconnaissance dans les mêmes conditions indique un meilleur algorithme de VAD.

ANNEXE 1

Transformée de Fourier rapide

ANNEXE 1

Transformée de Fourier rapide [8]

A1.1 Méthodes de calcul pour la transformée de Fourier rapide

Les méthodes de calcul de la TFD par entrelacement temporel et par entrelacement fréquentiel [8-9] exploitent les propriétés de périodicité et de symétrie de V :

$$V^{k+N} = V^N \tag{A1.1}$$

et

$$V^{k+N/2} = -V^k \tag{A1.2}$$

Ces méthodes pour une base 2 décomposent une TFD de N points en deux TFD de $N/2$ points et répètent le processus pour chaque nouvelle TFD jusqu'à ce qu'on arrive à $N/2$ transformée de deux points. Ces méthodes permettent de réduire le nombre d'opérations arithmétiques d'une manière très importante, on passe de N^2 à $N \log_2 N$ multiplication. On peut augmenter la base de l'algorithme à 4, 8, 16 pour améliorer les performances mais il faut mentionner que le nombre N doit être une puissance entière de la base utilisée. Il existe encore d'autres algorithmes plus complexes et efficaces, par exemple ceux qui combinent deux algorithmes d'entrelacement de bases différentes dans le but de réduire le nombre d'opérations.

A1.2 Calcul de la transformée de Fourier rapide par la méthode d'entrelacement fréquentiel

La séquence temporelle *x(n)* est séparée en deux [8] : *x(0)*, *x(1)*,... *x(N/2-1)* et *x(N/2)*, *x(N/2+1)*,..., *x(N-1)*. La TFD appliquée a chaque séquence donne :

$$X(k) = \sum_{n=0}^{N/2-1} x(n)V^{nk} + \sum_{n=N/2}^{N-1} x(n)V^{nk} \qquad \text{(A1.3)}$$

Si on pose pour la deuxième sommation $n = n + N/2$ on a :

$$X(k) = \sum_{n=0}^{N/2-1} x(n)V^{nk} + V^{kN/2} \sum_{n=0}^{N/2-1} x(n+N/2)V^{nk} \qquad \text{(A1.4)}$$

On utilise

$$V^{kN/2} = e^{-jk\pi} = (\cos\pi - \sin\pi)^k = (-1)^k \qquad \text{(A1.5)}$$

dans (A1.4) et *X(k)* devient :

$$X(k) = \sum_{n=0}^{N/2-1} [x(n) + (-1)^k x(n+N/2)]V^{nk} \qquad \text{(A1.6)}$$

Mais $(-1)^k$ =1 pour *k* paire et –1 pour k impaire et donc *X(k)* peut être séparé d'après *k* :

$$X(2k) = \sum_{n=0}^{N/2-1} [x(n) + x(n+N/2)]V^{2nk} \qquad k = 0,1,...,N/2-1 \qquad \text{(A1.7)}$$

$$X(2k+1) = \sum_{n=0}^{N/2-1} [x(n) - x(n+N/2)] V^n V^{2nk} \quad k = 0,1,...,N/2-1 \qquad \text{(A1.8)}$$

V est de longueur N, on le note V_N alors $(V_N)^2$ peut être écrite $V_{N/2}$. Avec les notations :

$$a(n) = x(n) + x(n+N/2) \qquad \text{(A1.9)}$$

$$b(n) = [x(n) - x(n+N/2)] V_N^n \qquad \text{(A1.10)}$$

ces deux équations sont exemplifiées graphiquement par le papillon tel qu'illustré dans la figure 44

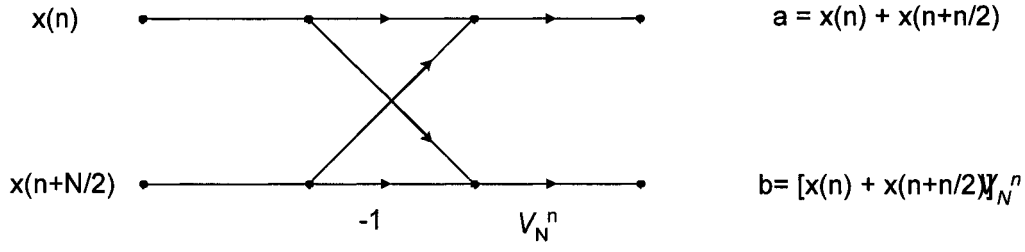


Figure 45 La structure papillon utilisée pour le calcul de la TFR par la méthode d'entrelacement fréquentielle

Les équations (A1.7) et (A1.8) peuvent être écrites plus clairement comme deux TFD de $N/2$ points :

$$X(2k) = \sum_{n=0}^{N/2-1} a(n) V_{N/2}^{nk} \qquad k = 0,1,...N/2-1 \qquad \text{(A1.11)}$$

$$X(2k+1) = \sum_{n=0}^{N/2-1} b(n) V_{N/2}^{nk} \qquad k = 0,1,...N/2-1 \qquad \text{(A1.12)}$$

On note que lorsque l'arrangement de la séquence d'entrée est en ordre naturel, l'arrangement de la séquence de sortie de cet algorithme est en ordre des bits renversés et il faut la réarranger pour l'obtenir en ordre naturel.

A1.3 Calcul efficace de la TFR pour deux séquences réelles

Jusqu'à maintenant on a supposé le cas général quand la séquence d'entrée *x(n)* est formée de valeurs complexes. Dans le cas de signal vocal la séquence d'entrée *x(n)* est formée de valeurs réelles. Comme l'algorithme de calcul de la TFD est conçu pour manipuler des valeurs complexes on peut exploiter cette caractéristique pour obtenir d'une manière plus efficace la TFD de deux séquences réelles [8].
On suppose que $x_1(n)$ et $x_2(n)$ sont deux séquences réelles de longueur N, alors la séquence $x(n)$ est complexe :

$$x(n) = x_1(n) + j\,x_2(n) \quad n = 0,1,\ldots,N-1 \tag{A1.13}$$

et donc

$$\begin{aligned} x_1(n) &= \frac{x(n) + x^*(n)}{2} \\ x_2(n) &= \frac{x(n) - x^*(n)}{2j} \end{aligned} \tag{A1.14}$$

La TFD est une transformée linéaire et donc :

$$X(k) = X_1(k) + j\,X_2(k) \tag{A1.15}$$

Comme la TFD de $x^*(n)$ est $X^*(N-k)$, les TFD de $x_1(n)$ et $x_2(n)$ donnent :

$$X_1(k) = \frac{1}{2}[X(k) + X^*(N-k)]$$
$$X_1(k) = \frac{1}{2j}[X(k) - X^*(N-k)] \qquad \text{(A1.16)}$$

A1.4 Calcul efficace de la TFR pour une séquence réelle de longueur 2*N*

On suppose que *g(n)* est une séquence réelle de longueur $2N$ et on pose [8] :

$$x_1(n) = g(2n)$$
$$x_2(n) = g(2n+1) \qquad \text{(A1.17)}$$

Pour ces deux séquences on applique le résultat obtenu précédemment.
La TFD de $2N$ points s'exprime en termes de deux TFD de N points:

$$G(k) = \sum_{n=0}^{N-1} g(2n) V_{2N}^{2nk} + \sum_{n=0}^{N-1} g(2n+1) V_{2N}^{2(n+1)k}$$
$$= \sum_{n=0}^{N-1} x_1(n) V_N^{nk} + V_{2N}^{nk} \sum_{n=0}^{N-1} x_2(n) V_N^{nk} \qquad \text{(A1.18)}$$

et en conséquence :

$$G(k) = X_1(k) + V_{2N}^k X_2(k) \qquad k = 0,1,\ldots,N-1$$
$$G(k+N) = X_1(k) - V_{2N}^k X_2(k) \qquad k = 0,1,\ldots,N-1 \qquad \text{(A1.19)}$$

Donc on a obtenu la TFD d'une séquence réelle de longueur $2N$ à partir d'une TFD en N points et encore quelques opérations tel qu'indiqué par (A1.16) et (A1.19).

ANNEXE 2

Description du test statistique χ^2

ANNEXE 2

Description du test statistique χ^2

Le test statistique χ^2 [11] compare la distribution des données expérimentales avec certaines fonctions de densité de probabilité données. Il est une mesure des distorsions existant entre les données est les fonctions de densité de probabilité théoriques. La relation que donne la valeur du test est :

$$\chi^2 = \frac{\sum_{i=1}^{K}(n_i - N\,p_i)^2}{N\,p_i} \qquad \text{(A2.1)}$$

L'espace de valeurs possible pour les données est divisé en K intervalles ; n_i est le numéro d'échantillons qui se retrouvent dans l'intervalle i ; p_i est la probabilité théorique qu'un échantillon se retrouve dans le même intervalle i et N est le numéro total d'échantillons.

Plus la valeur résultant du test est petite, plus la distribution expérimentale s'approche de la distribution théorique

ANNEXE 3

Théorie bayésienne de la décision

ANNEXE 3

Théorie bayésienne de la décision [35]

A3.1 La théorie classique de décision de Bayes

La théorie Bayésienne de la décision est une approche statistique pour les problèmes de classification Cette approche est basée sur l'ajustement des probabilités connues a priori pour un paramètre inconnu, à des probabilités a posteriori plus certaines.

En statiques classiques, on associe à une variable aléatoire discrète C, une distribution de probabilités notée $P(C)$ qui rallie la notion de fréquence d'occurrence. Pour une variable aléatoire continue x la distribution est représentée par une fonction de densité de probabilité (fdp) notée $p(x)$.

On suppose X comme étant un vecteur aléatoire continu de caractéristiques qui décrit des objets appartenant à M classes discrètes nommées c_i, $i=1,\ldots,M$. On utilise la probabilité conditionnelle $P(X \mid c_i)$ pour représenter la fdp du vecteur des observations X étant donné la classe c_i. De même, on utilise la probabilité conditionnelle $P(c_i \mid X)$ pour indiquer la classe c_i étant donné l'observation X.

La probabilité conjointe $P(X, c_i)$ est la probabilité que X et c_i se réalisent simultanément. Si les événements X et c_i sont des événements indépendants alors :

$$P(X,c_i) = P(X)p(c_i) \tag{A3.1}$$

Si les événements X et c_i sont des événements dépendants alors :

$$P(X,c_i) = P(X \mid c_i)p(c_i) = P(c_i \mid X)p(X) \tag{A3.2}$$

A partir de cette relation on peut écrire la formule de Bayes :

$$P(c_i \mid X) = \frac{P(X \mid c_i)P(c_i)}{P(X)} \tag{A3.3}$$

où

$$P(X) = \sum_{i=1}^{M} P(X \mid c_i)P(c_i) \tag{A3.4}$$

On suppose que $P(X, c_i)$ et $P(X)$ sont connues au concepteur de classificateur. En d'autres termes, le concepteur a la pleine connaissance de la nature aléatoire de la source. L'objectif d'un classificateur est de classifier correctement chaque X dans la classe c_i à laquelle il appartient.

Pour mesurer la performance du classificateur, nous définissons pour chaque paire de classe (i,j) une fonction de perte ou de coût e_{ij}, qui signifie le coût de classifier un objet appartenant à la classe i dans la classe j. Cette fonction est positive définie avec $e_{ii} = 0$ pour une classification correcte.

Etant donné une observation arbitraire X, une fonction de perte moyenne ou de risque conditionnel pour classifier X dans la classe i peut être définie comme suit :

$$R(c_i \mid X) = R_i(X) = \sum_{j=1}^{M} e_{ij} P(c_j \mid X) \tag{A3.5}$$

Une fonction de perte simple très utilisée est la fonction sigmoïde un-zéro :

$$e_{ij} = \begin{cases} 0, & i = j \\ 1, & i \neq j, \quad avec \quad i, j = 1...M \end{cases} \tag{A3.6}$$

qui accorde une perte zéro pour une bonne classification et une perte unitaire pour une classification erroné. Avec cette fonction de perte la fonction de perte conditionnée est de la forme :

$$R_i(X) = \sum_{i \neq j} P(c_j \mid X) = 1 - P(c_i \mid X) \tag{A3.7}$$

Dans ce cas, il est évident que la minimisation de $R_i(X)$ revient à maximiser $P(c_i \mid X)$. Donc, pour minimiser la rate d'erreur de classification, le classifieur utilise la règle de décision (A3.7) appelé maximum a posteriori (MAP).

La connaissance nécessaire sur le système pour une décision optimale est une probabilité a posteriori. Cette probabilité n'est pas connue dans des problèmes réels et elle doit être estimée à partir d'un set de données d'apprentissage. La théorie Bayésienne de la décision transforme le problème de classification dans un problème d'estimation de paramètres de lois de probabilités. Le taux d'erreur minimal réalisé par la règle de décision de MAP s'appelle le risque de Bayes. Puisque $P(X)$ n'est pas une fonction de l'index de classe et n'a, ainsi, aucun effet dans la décision de MAP, la connaissance probabiliste nécessaire peut être représentée par la probabilité a priori de classe $P(c_i)$ et la probabilité conditionnelle $P(X| c_i)$.

Il y a plusieurs issues liées à cette approche classique. D'abord, les distributions doivent habituellement être paramétrisées afin qu'elles puissent être pratiquement utilisées pour l'exécution de la règle de MAP. Le concepteur de classificateur doit donc déterminer la bonne forme paramétrique des distributions. Pour la plus grande part des vrais problèmes du monde, ceci est une tâche difficile. Notre choix de la forme de distribution est souvent limité par la tractabilité mathématique des fonctions de distribution

particulières et peut être une faible approximation pour la distribution réelle. Ceci signifie que la véritable règle de décision de MAP peut rarement être mise en application, et le risque minimal de Bayes demeure généralement une limite inférieure intouchable.

Deuxièmement, une fois la loi de distribution choisie, les paramètres inconnus définissant la distribution doivent être estimés à partir des données de formation. Une bonne méthode d'évaluation des paramètres est donc nécessaire.

Troisièmement, l'approche exige un set de données de formation connus. Afin d'avoir une évaluation fiable des paramètres, l'ensemble de formation doit être de taille suffisante. Habituellement, plus la taille du set de données de formation fourni est grande, plus l'évaluation des paramètres est meilleure. La difficulté est le fait que la collecte de données est un travail coûteux, en particulier pour des applications dans le domaine de la parole et des fois on ne dispose pas que d'une quantité limitée de données d'apprentissage. Quand la quantité de données de formation est limitée, la qualité d'évaluation des paramètres pour les lois de distribution ne peut pas être garantie.

A3.2 Décision Bayésienne à deux classes

Dans le cas de la détection d'activité vocale, le nombre de classes est limité à deux, représenté, par l'absence H_0 ou la présence H_1 de la parole dans le segment de signal à classifier. On classifie les observations en minimisant la perte moyenne. Les pertes moyennes associés sont :

$$\begin{aligned} R_0(X) &= e_{00}P(H_0 \mid X) + e_{01}P(H_1 \mid X) \\ R_2(X) &= e_{10}P(H_0 \mid X) + e_{11}P(H_1 \mid X) \end{aligned} \tag{A3.8}$$

La règle de décision est d'affecter l'observation X à la classe qui présente la perte moyenne minimale. En utilisant la règle de Bayes, on peut exprimer la règle de décision en terme des probabilités a priori de classes :

$$\frac{P(X \mid c_0)}{P(X \mid c_1)} = \frac{e_{01} - e_{11}}{e_{10} - e_{00}} \frac{P(H_1)}{P(H_0)} \tag{A3.9}$$

Le gauche de l'égalité (A3.9) s'appelle ratio de vraisemblance et le terme droit est un seuil indépendant du vecteur de caractéristiques X. La règle de décision est de choisir la clase H_1 si le ratio de vraisemblance est supérieur au seuil, sinon le choix est H_0.

Quand on regarde une seule trame d'un signal d'entrée dans un algorithme de VAD, il est raisonnable d'attribuer des probabilités a priori égaux aux hypothèses qu'elle contienne ou non de la parole. Si de plus on considère la fonction de perte décrite par la relation (A3.6) qui attribue des coûts égaux aux décisions erronés. Le seuil dans la relation (A3.9) est 1 et la règle de décision de Bayes est équivalente à la règle de décision du maximum de vraisemblance.

ANNEXE 4

Modèles de Markov

ANNEXE 4

Modèles de Markov

Pour les problèmes qui ont une temporalité inhérente comme dans le cas de la production de la parole, on regarde la possibilité de faire plutôt une séquence de décision qu'une seule décision. On peut donc avoir des états à un moment donné t qui sont influencés directement par les r états antérieurs. En particulier, une chaîne de Markov d'ordre 1 est une séquence pour laquelle la probabilité d'occurrence de chaque état ne dépend que de la nature de l'état qui la précède.

Pour produire une séquence des états $w = \{c(1), c(2), \ldots, c(k)\}$, ce système utilise la notion de probabilité de transition entre les états du système :

$$P(c_i(t) \mid c_j(t-1)) = a_{ij} \tag{A4.1}$$

qui est la probabilité d'avoir l'état c_i au temps t quand on a l'état c_j au temps t-1.

On suppose avoir un modèle particulier O caractérisé par le set de paramètres a_{ij}, la probabilité que le modèle génère la séquence w est :

$$P(w \mid O) = \prod_{i=2}^{k} a_{i-1i} \tag{A4.2}$$

BIBLIOGRAPHIE

[1] *Le petite Larrouse* (1995)

[2] Calliope (1989). *La parole et son traitement automatique*. Paris : Masson.

[3] Boite, R., Kunt, M. (1987). *Traitement de la Parole*, (1e éd.). Lausanne : Presses Polytechniques Romandes.

[4] Alexa F. (1999) *Introducere in tehnica sunetului*. Timisoara : Editura de vest.

[5] Deller, J.R., Hansen, J. H. L., Proakis J. G. (1999). *Discrete Time Processing of Speech Signals*, (3e éd). New Jersey : IEEE Press.

[6] Rabiner, L. R., Schafer, R. W. (1978). *Digital Processing of Speech Signals*, (1e éd.). New Jersey : Prentice-Hall.

[7] Childers D.G. (2000). *Speech processing and synthesis toolboxes*, (1e éd). New York : John Wiley & Sons, Inc

[8] Proakis, J. G., Manolakis, D. G. (1996). *Digital Signal Processing: Principles, Algorithms, and Applications*,(3e éd.). Upper Saddle River, New Jersey : Prentice-Hall.

[9] C.-S. Gargour, (2001). *Traitement numérique de signaux*,(1e éd). École de technologie supérieure.

[10] Rabiner, L. R., Sambur, M. R. (1975). An algorithm for determining the endpoints of isolated utterances. *Bell Syst. Tech. J.*, 54(2), 297–315.

[11] Gazor S., Zhang W. (2003). Speech Probability Distribution. *IEEE Signal Processing Letters*, 10(7), 204-207.

[12] Benyassine A., Sholomot E., Su H.-Y., Massaloux D., Lambline C., Petit J.-P. (1997). ITU-T Recommendation G.729 Annex B: A silence compression schema for use with G.729 optimized for V.70 digital simultaneous voice and data application. *IEEE Communication Magazine*, 35(9), 64–73.

[13] Atal B. S., Rabiner, L. R. (1976) A Pattern Recognition Approach to Voiced-Unvoiced-Silence Classification with Application to Speech Recognition. *IEEE Transaction on Acoustics, Speech, and Signal Processing*, ASSP-24(3), 201-212.

[14] Beritelli L., Casale S, G. Ruggeri , S. Serrano (2002). Performance Evaluation and Comparison of G.729/AMR/Fuzzy Voice Activity Detectors. *IEEE Signal Processing Letters*, 9(3), 85-88.

[15] Rabiner, L. R., Schmidt C. E., Atal B. S. (1977). Evaluation of a Statistical Approach to Voiced-Unvoiced-Silence Analyses for Telephone-Quality Speech. *The Bell System Technical Journal*, March, 455-483.

[16] Rabiner, L. R., Sambur, M. R. (1977). Application of an LPC Distance Measure to the Voiced-Unvoiced-Silence Detection Problem. *IEEE Transaction on Acoustics, Speech, and Signal Processing*, ASSP-25(4), 338-343.

[17] Lamel L. F., Rabiner, L. R., Rossenberg A. E. Wilpon J. G. (1981). An improved end point detector for isolated Word Recognition. *IEEE Transaction on Acoustics, Speech and Signal Processing*, ASSP-29(4), 777-785.

[18] Li Q., Zheng J., Tsai A., Zhou Q. (2002). Robust endpoint detection end energy normalization for real time speech and speaker recognition. *IEEE Tran. on Speech and Audio Processing*, 10(3), 146–157.

[19] Junqua J.-C., Mak B., Reaves B. (1994). A robust algorithm for word boundary detection in presence of noise. *IEEE Tran. on Speech and Audio Processing*, 2(3), 406-412.

[20] Lin, C.-T. Lin, J.-Y. Wu G.-D. (2002). A robust word boundary detection algorithm for variable noise-level environment in cars. *IEEE Tran. on Intelligent Transportation Systems*, 3, 89 -101.

[21] Cavallaro A., Beritelli L., Casale S (1998). A robust robust voice activity detector for wireless communication using soft computing. *IEEE J. Select. Areas. Commun.*, 16(12), 1818-1829

[22] Shon, J., Sung, W. (1998). A Voice Activity Detection Employing Soft Decision Based Noise Spectrum Adaptation. *Proc. Int. Conf. Acoustics, Speech and Signal Processing*, 365-368.

[23] Shon, J., Kim, N. S., Sung W. (1999). A Statistical Model Based Voice Activity Detection, *IEEE Signal Processing Letters*, 6(1), 1-3.

[24] Gazor S., Zhang W. (2003). A Soft Voice Activity Detector Based on a Laplacian-Gausian Model. *IEEE Trans. on Speech and Audio Processing*, 2(5), 498–505.

[25] N. G. Philipe, T. Fukada, Y. Komori (2004) A differential spectral voice activity detector. *ICASSP*, 1,597 – 600

[26] S. G. Tanyer, H. Özer (2000) Voice activity detection in non-stationary noise. *IEEE Trans. on Speech and Audio Processing*, 9(3), 217-231.

[27] B. Ahmed, W. H. Holmes (2004). A voice activity detector using chi–square test. *ICASP*, 1, 625 – 628.

[28] Chen S. H., Wang J.-F. (2002). A wavelet-based voice activity detection in noisy environments. *International Conference on Electronics, Circuits and Systems*, 3, 995–998.

[29] Craciun A., M. Gabrea (2004) Correlation coefficient- based voice activity algorithm. *CCECE*, Niagara Falls.

[30] Petrou M., Kittler J (1991). Optimal edge detectors for ramp edges. *IEEE Trans. Pattern Anal. Machine Intell.*, 13(5), 483-491.

[31] Canny J. (1986). A computational approach to edge detection. *IEEE Trans. Pattern Anal. Machine Intell.*, PAMI-8, Nov, 679-698.

[32] Allen J. B. (1985). Cochlear modeling. *IEEE Acoust., Speech, Signal, Processing*, 2, 3-29.

[33] Xiao-dan Mei, Sheng-he Sun (2000). An efficient Method to compute LSFS from LPC Coefficients. *Processing of ICSP*, 655-658.

[34] Ephraim Y., D. Malah (1984). Speech enhancement using a minimum mean-square error short-time spectral amplitude estimator. *IEEE Transaction on Acoustic, Speech and Signal Processing*, ASSP-32(6), 1109-1121.

[35] Cheriet M., (1997). *Reconnaissance des formes et inspection*, notes de cours, vol. 1. Ecole de technologie supérieure.

[34] Chassaing, R. (2002). DSP Applications Using C and the TMS320C6x DSK, (1e éd.). New York : John Wiley & Sons, Inc.

[35] *TLC320AD535C/I Data Manual Dual Channel Voice/Data Codec*, SLAS202A, Texas Instruments, Dallas, TX, 1999.

[36] *TMS320C600 Technical Brief*, SPRU197D, Texas Instruments, Dallas, TX, 1999.

[37] *TMS320C600 Peripherals Reference Guide*, SPRU190D, Texas Instruments, Dallas, TX, 2001.

[38] *TMS320C6x Peripheral Support Library Programmer's Reference* , SPRU273B, Texas Instruments, Dallas, TX, 1998.

[39] *TMS320C6211 Fixed-Point Digital Signal Processor – TMS320C6711 Floating Point Digital Signal Processor*, SPRS073C, Texas Instruments, Dallas, TX, 2000.

[40] *TMS320C600 Optimizing Compiler Guide User's Guide*, SPRU187G, Texas Instruments, Dallas, TX, 2000.

[41] *TMS320C600 Programmer's Guide*, SPRU198D, Texas Instruments, Dallas, TX, 2000.

[42] *TMS320C600 CPU and Instruction Set Reference Guide*, SPRU189F, Texas Instruments, Dallas, TX, 2000.

[43] *TMS320C600 Assembly Language Tools User's Guide*, SPRU186G, Texas Instruments, Dallas, TX, 2000.

[44] *Code Composer Studio User's Guide*, SPRU328B, Texas Instruments, Dallas, TX, 2000.

[45] *Code Composer Studio Getting Started Guide*, SPRU509, Texas Instruments, Dallas, TX, 2001.

[46] *Code Composer Studio Tutorial*, SPRU301C, Texas Instruments, Dallas, TX, 2000.

[47] *TMS320C600 DSP/BIOS User's Guide*, SPRU303B, Texas Instruments, Dallas, TX, 2000.

[48] *TMS320C600 DSP/BIOS Application Programming Interface (API)*, SPRU403A, Texas Instruments, Dallas, TX, 2000.