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
KHALED ZAABI

IMPLÉMENTATION D'UNE MÉTHODE DE RECONNAISSANCE DE LA PAROLE SUR LE PROCESSEUR DE TRAITEMENT NUMÉRIQUE DU SIGNAL TMS320C6711

MONTRÉAL, LE 10 JUIN 2004

CE MÉMOIRE A ÉTÉ ÉVALUÉ
PAR UN JURY COMPOSÉ DE :

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

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

M. Maarouf Saad, professeur
Département de génie électrique à l'École de technologie supérieure

IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC
LE 27 MAI 2004
À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

IMPLÉMENTATION D'UNE MÉTHODE DE RECONNAISSANCE DE LA PAROLE SUR LE PROCESSEUR DE TRAITEMENT NUMÉRIQUE DU SIGNAL TMS320C6711

Khaled Zaabi

RÉSUMÉ

Les systèmes actuels de reconnaissance automatique de la parole RAP permettent d'atteindre des performances relativement élevées. Ces systèmes nécessitent souvent de grandes capacités de mémoire et de calcul du fait de la complexité des algorithmes auxquels ils font appel. De nos jours on assiste de plus en plus à l'intégration des systèmes RAP dans divers domaines, ce qui a pour conséquence la nécessité de limiter fortement la consommation de ressources matérielles, et ceci dans le but de réduire le coût du produit ou du service dans des marchés de plus en plus compétitifs.
L'objectif de ce mémoire vise la réalisation d'une méthode de reconnaissance vocale à l'aide d'algorithmes efficaces et rapides. Une méthode est proposée, fondée sur la segmentation en syllabes pour décomposer les chiffres connectés, et le seuillage de l'énergie afin de cibler les segments les plus représentatifs d'un mot. Cette méthode est ensuite implémentée sur un processeur de traitement numérique du signal, le DSP TMS320C6711 de Texas Instrument, pour cela nous utiliserons l'environnement du développement Code Composer Studio.

IMPLÉMENTATION D'UNE MÉTHODE DE RECONNAISSANCE DE LA PAROLE SUR LE PROCESSEUR DE TRAITEMENT NUMÉRIQUE DU SIGNAL LE TMS320C6711

Khaled Zaabi

SOMMAIRE

Les systèmes actuels de reconnaissance automatique de la parole RAP permettent d'atteindre des performances relativement élevées. Ces systèmes nécessitent souvent de grandes capacités de mémoire et de calcul du fait de la complexité des algorithmes auxquels ils font appel. De nos jours on assiste de plus en plus à l'intégration des systèmes RAP dans divers domaines, ce qui a pour conséquence la nécessité de limiter fortement la consommation de ressources matérielles, et ceci dans le but de réduire le coût du produit ou du service dans des marchés de plus en plus compétitifs.
L'objectif de ce mémoire vise la réalisation d'une méthode de reconnaissance vocale à l'aide d'algorithmes efficaces et rapides. Une méthode est proposée, fondée sur la segmentation en syllabes pour décomposer les chiffres connectés, et le seuillage de l'énergie afin de cibler les segments les plus représentatifs d'un mot. Nous avons testé les performances de reconnaissance en effectuant des expériences sur un corpus de 65 locuteurs extraits de la base de données TIDIGITS de Texas Instrument. Cette méthode est ensuite implémentée sur un processeur de traitement numérique du signal, le DSP TMS320C6711 de Texas Instrument, pour cela nous utiliserons l'environnement du développement Code Composer Studio dont les différents outils permettent de tirer le meilleur parti des ressources du processeur, les algorithmes utilisés sont entièrement en langage C.

IMPLEMENTATION OF A SPEECH RECOGNITION METHOD ON DIGITAL SIGNAL PROCESSOR THE TMS320C6711

Khaled Zaabi

ABSTRACT

The current systems of automatic speech recognition ASR make it possible to reach performances relatively high. These systems often require great calculation and storage capacities because of the complexity of the algorithms to which they appeal. Nowadays one attends more and more the integration of the ASR systems in various fields, which has as a consequence the need for strongly limiting the consumption of resources materials, and this with an aim of reducing the cost of the product or the services in increasingly competitive markets. The objective of this memory aims at the realization of a method of voice recognition using effective and fast algorithms. A method is proposed, based on the segmentation in syllables to break up the connected digits, and the thresholding of energy at end to target the segments most representative of a word. We tested the performances of recognition by carrying out experiments on a corpus of 65 speakers extracted from the data base TIDIGITS of Texas Instrument. This method is then implemented on a digital signal processor, DSP the TMS320C6711 of Texas Instrument, for that we will use the environment of development the Code compose Studio from which the various tools make it possible to draw the best party from the resources of the processor, the algorithms used are entirely in C language.

REMERCIEMENTS

Je voudrais remercier sincèrement mon directeur de recherche, monsieur Marcel Gabréa, professeur à l'école de technologie supérieure, pour les nombreux conseils judicieux qu'il m'a prodigués et pour sa disponibilité tout au long de cette recherche, qu'il retrouve ici le témoignage de ma profonde gratitude.

Mes profonds remerciements vont aux membres de jury pour avoir accepter de juger ce travail.

Je remercie également mes chers parents, mes frères, mes sœurs, ma famille et tous mes amis pour leur soutien et encouragement.

TABLE DES MATIÈRES

Page

LISTE DES TABLEAUX

Page

LISTE DES FIGURES

Page

LISTE DES ABRÉVIATIONS

AR	Auto Régressif
ARMA	Auto Régressif à Moyenne Ajustée
CCS	Code Composer Studio
DAP	Décodage Acoustique Phonétique
DCT	Discret Cosine Transform
DSP	Digital Signal Processor
DTW	Dynamic Time Warping
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
HMM	Hidden Markov Models
Kpp	K plus proche voisins
LPC	Linear Prediction Coding
LPCC	Linear Predictive Cepstral Coefficients
LSF	Line Spectral Frequencies
MFCC	Mel Frequency Cepstral Coefficients
parcor	Partial Correlation
RAP	Reconnaissance Automatique de la Parole
SLIT	Système Linéaire Invariant dans le Temps
SUV	Silence Unvoiced Voiced
VAD	Voice Activity Detection
VQ	Vector Quantization

INTRODUCTION

La reconnaissance automatique de la parole est le processus qui consiste à décoder un signal acoustique de parole en la suite de mots effectivement prononcés, dans tout système de reconnaissance de la parole, le but ultime est d'arriver à un stade où on peut se servir de la voix comme un nouvel outil dans la communication homme-machine et ceci à l'aide d'une interface totalement en langage naturel, d'ailleurs l'enthousiasme pour cette idée est telle qu'on parle même de la "troisième main" [1] en référence à l'usage de la voix pour l'exécution des tâches au lieu des mains.

Bien que le début des travaux dans ce domaine remonte a plus d'un demi siècle [2], on peut considérer que c'est le début des années 70 qui a inauguré un virage marquant dans l'histoire de cette recherche avec entre autres l'apparition de la première réalisation commerciale en reconnaissance vocale: "Voice Command System" [1], un appareil autonome capable de reconnaître de manière fiable 24 mots isolés après cinq cycles d'apprentissage par le même locuteur, et ensuite l'émergence d'idées novatrices tel que les modèles de Markov cachés [3] (Hidden Markov Models, HMM), qui étaient sans le moindre doute d'un apport considérable dans l'amélioration des performances de reconnaissance des systèmes RAP.

Malgré les résultats remarquables des dernières années, la progression des recherches n'a pas été sans embûches, ainsi à leur début les chercheurs ont dû faire face à plusieurs problèmes, comme par exemple le problème de la parole continue et la difficulté d'isoler le début ou la fin des mots (la segmentation) des syllabes ou des phonèmes dans un flot continu de parole ou le problème de l'alignement temporel, en effet à cause de la variabilité des formes vocales même pour un même locuteur, il n'est guère possible d'envisager la comparaison de deux formes vocales prononcées à des débits différents, sans au préalable d'une manière ou d'une autre régler le problème de la normalisation

temporelle, parmi les solutions élaborées pour contrer ce problème on cite l'utilisation de la programmation dynamique (DTW) développé par des chercheurs de l'ex U.R.S.S [1], qui a fait ses preuves comme moyen puissant pour aligner et comparer deux sons prononcés à des débits différents.

La reconnaissance de la parole a de très nombreux débouchés, et l'utilisation de la voix comme un outil de commande devient une réalité dans la vie quotidienne, d'ailleurs on peut facilement retrouver des exemples concrets de ce changement avec l'apparition de plus en plus de systèmes à commandes vocales pour la gestion des tâches que jadis nécessitaient la mobilisation d'autres moyens, on peut citer ici comme exemple le système de reconnaissance vocale "Émilie " mis en service récemment par Bell Canada qui permet de reconnaître la raison de l'appel (« je déménage » par exemple) et de l'acheminer en conséquence.

Un autre domaine de la reconnaissance de la parole où nous assistons de plus en plus à des percées significatives, est la reconnaissance de la parole appliquée à la télécommunication dont l'une de ses formes est la reconnaissance via téléphone (speech recognition over the telephone) qui a permis d'élargir l'horizon de la reconnaissance à d'autres applications encore plus intéressantes tel que la validation des numéros de cartes ou de comptes bancaires ou la composition de numéros de téléphone par la voix pour les combinés téléphoniques classiques ou pour les cellulaires qui font partie de ce qu'on appelle de nos jours, communication "mains libres et œil-libre" [4-6].

Dans le même ordre d'idée, Le but de ce projet est la réalisation d'un système RAP, capable de reconnaître un vocabulaire limité (des chiffres isolés et connectés) comme un premier pas vers un système multi-locuteurs capable de reconnaître la parole continue. Des travaux de cette nature ont déjà existé, et spécialement pour reconnaître les chiffres connectés, un axe de recherche qui a connu des progrès remarquables à partir des années 80. Ainsi Rabiner [7] propose un système indépendant du locuteur pour la

reconnaissance des chiffres connectés en utilisant les modèles de Markov Cachés (HMM), puis Wilpon [8] propose une autre méthode qui permet d'améliorer celle de Rabiner [7] en incorporant d'autres paramètres tel que le log de l'énergie et des coefficients cepstrales d'ordre supérieur, on cite aussi Myers [9] qui a proposé une méthode de reconnaissance des chiffres connectés basée sur la programmation dynamique (DTW), enfin citons que d'autres méthodes existent, elles utilisent une autre approche qu'on appelle approche neuronale avec l'utilisation des réseaux de neurones pour la reconnaissance de la parole [10]

Parmi tous les travaux antérieurs, rares sont ceux qui ont été faits dans le cas de faibles ressources de calcul et de mémoire, d'où l'idée d'explorer la possibilité de réaliser un système RAP tout en évitant les algorithmes complexes qui sont coûteux en temps et en mémoire. Pour ce faire nous avons proposé une nouvelle méthode d'alignement basée sur le seuillage d'énergie, et l'utilisation de fenêtres de durée fixe [11] ensuite cette méthode a été améliorée en utilisant des fenêtres de durée variable, une durée qu'on ne connaît pas à priori et qu'on déterminera après seuillage de l'énergie.

Ce travail présente la conception et la réalisation d'un système de reconnaissance vocale à l'aide d'un processeur dédié au traitement numérique des signaux DSP le TMS320C6711 de Texas Instrument, ce système est composé principalement de trois modules celui de la phase de paramétrisation, celui de l'apprentissage et en dernier celui de la phase de reconnaissance ou de décision.

Dans la phase de paramétrisation, après un "seuillage" de l'énergie pour localiser la partie la plus énergétique du signal et par conséquent la plus riche en informations, chaque segment de parole est converti en un ensemble de paramètres LPCCs. Ce type de paramètres permet avec une pondération adéquate, une parfaite représentation des traits acoustiques du signal vocal, ce qui est d'une très grande utilité pour la phase de reconnaissance. La seconde phase celle de l'apprentissage a pour rôle de fournir le

dictionnaire des mots de référence à partir des chiffres isolés. Elle est basée sur l'algorithme K-means très utilisé dans la quantification vectorielle. Enfin la phase de reconnaissance est basée sur la règle de décision du plus proche voisin ou K_{pp}.

Ce programme a été réalisé en deux étapes, dans un premier temps, les algorithmes ont été réalisés et tester avec Matlab, et en deuxième temps avec le langage C avec quelques ajustements pour l'adapter à l'environnement logiciel que nous avons utilisé le "Code Composer Studio".

Pour tester les performances de ce programme, nous avons utilisé des chiffres isolés, ainsi que des chiffres connectés, dans ce dernier cas la chaîne de chiffres est séparée en chiffres isolés à l'aide d'un algorithme de segmentation.

Ce mémoire est organisée de la manière suivante, dans le premier chapitre nous avons exposé les notions théoriques de la production et de la perception de la parole, le deuxième chapitre traite les outils de base pour le traitement et la paramétrisation du signal vocal, le troisième chapitre est un aperçu sur les principales méthodes de la reconnaissance de la parole, et quelques algorithmes de classifications y sont aussi présentés, dans le quatrième chapitre nous décrirons notre système ainsi que les algorithmes utilisés pour l'élaborer, et enfin nous conclurons par les résultats expérimentaux obtenus et des recommandations pour les développements ultérieurs possibles.

CHAPITRE 1

PRODUCTION ET PERCEPTION DE LA PAROLE

1.1 Introduction

La parole, ce fascinant moyen de communication entre les humains, est selon la définition du Robert [12] "la faculté de communiquer la pensée par un système de sons articulés émis par les organes de phonation", sa particularité tient sans doute à la complexité des fonctions que le cerveau humain met en œuvre pour la produire ou la comprendre, et ceci d'une manière pratiquement instantanée. C'est ces fonctions, que pourtant le cerveau exécutent inconsciemment, que des décennies de recherches et d'efforts continus n'ont pas encore permis d'égaler les performances ou d'entièrement cernés.

Par le fait de la position de la parole, au croisement de plusieurs disciplines on lui distingue plusieurs niveaux de description entre autres [13]:

- Le niveau phonétique : en phonétique, on essaie d'explorer la façon dont le signal est produit par le système articulatoire, l'analyse est effectuée sur trois plans complémentaires, perceptif, articulatoire, et acoustique.

- Le niveau phonologique : la phonologie a comme objectif d'étudier les variantes phonétiques contextuelles, elle introduit la notion d'unité abstraite du discours le "phonème", qui sera exposé dans la suite de ce chapitre. En reconnaissance de la parole, la phonologie regroupe l'ensemble des modules de traitement des altérations possibles d'un phonème (allophones ou variantes) ou d'un mot dans un contexte donné.

- Le niveau acoustique : les acousticiens s'intéressent aux traits acoustiques de la parole : sa fréquence fondamentale, son énergie, et son spectre. L'analyse est effectuée sur un signal électrique, un transducteur (le microphone) est utilisé pour réaliser le passage de l'acoustique à l'électrique.

- Le niveau morphologique : la morphologie est la branche de la linguistique qui étudie comment les formes lexicales sont obtenues à partir d'un ensemble réduit d'unités porteuses de sens, appelées morphèmes.

- Le niveau syntaxique : la syntaxe est l'ensemble des règles qui définissent l'exactitude des phrases, en effet une suite de mots du lexique ne forme pas forcément une phrase correcte.

- Le niveau sémantique : la sémantique est l'étude des significations des mots, et la façon dont ils sont liés les uns aux autres. Une phrase peut être correcte du point de vue syntaxique, sans l'être du point de vue sémantique.

- Le niveau pragmatique : la pragmatique est l'étude des aspects du langage qui font référence aux relations entre locuteur et interlocuteur, d'une part et entre interlocuteurs et situation concrète, d'autre part, le sens pragmatique est défini comme dépendant du contexte.

Dans ce chapitre seront exposés le mécanisme de phonation, les aspects acoustiques et phonétiques de la parole, ainsi que le mécanisme d'audition et les propriétés perceptuelles qui s'y rattachent.

1.2 Description de l'appareil phonatoire

L'appareil phonatoire est composé principalement de trois éléments qui contribuent ensemble à la production de la parole. Ces éléments dont le contrôle et la coordination sont assurés par le système nerveux central, sont :

- les poumons : ils fournissent l'énergie (l'air) nécessaire à la production du son.

- le larynx (voir Figure 1): son rôle est la production des sons. C'est un ensemble de cartilages articulés comprenant les deux "cordes vocales". Ces dernières sont des organes vibratoires constituées de tissu musculaire et de tissu conjonctif résistant.

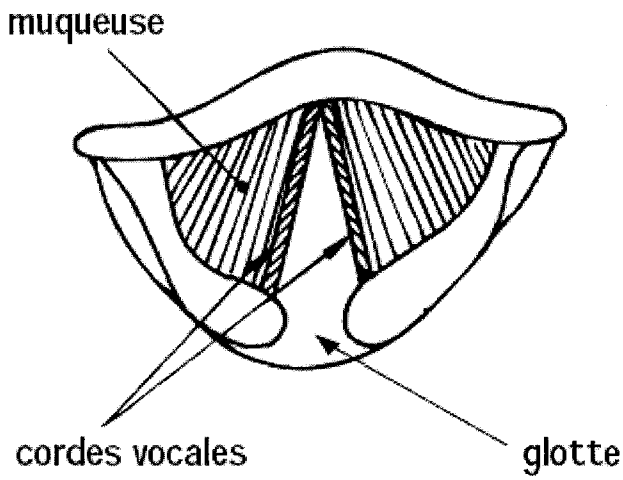

Figure 1 Le larynx [13]

- le conduit vocal (voir Figure 2): c'est le conduit entre le larynx et les lèvres, il est composé de plusieurs cavités reliées entre elles. On retrouve la cavité pharyngale

(le pharynx), la cavité nasale (les fosses nasales), la cavité buccale (la bouche) et la cavité labiale (les lèvres).

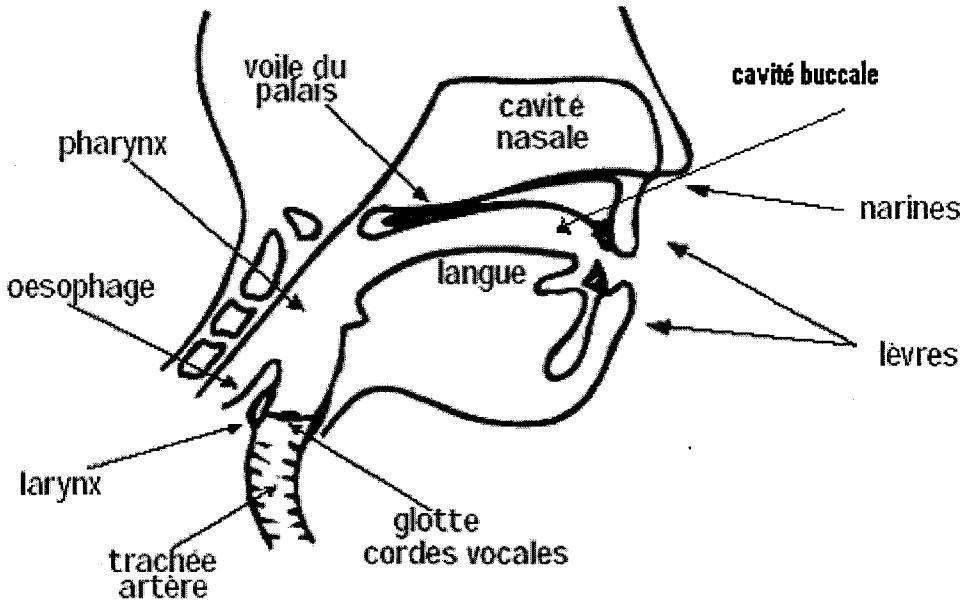

Figure 2 L'appareil phonatoire [13]

1.3 La production de la parole

Lors de la production de la parole, un flot d'air émanant des poumons est contraint à passer dans le larynx, qui par le biais des cordes vocales va générer une vibration, les sons étant des ondes (donc des vibrations). Lorsque la pression d'air s'accumule sous les cordes vocales, elles sont forcées de s'ouvrir partiellement, leur tension naturelle les amène ensuite à se refermer, ce sont le débit du flot d'air et le degré d'ouverture des

cordes vocales qui conditionnent l'intensité de l'onde ainsi produite. L'espace entre les cordes vocales s'appelle la glotte.

Les sons de parole sont produits soit par les vibrations des cordes vocales, dans ce cas on parle de sons voisés, soit par l'écoulement turbulent de l'air dans le conduit vocal, soit lors de relâchement d'une occlusion de ce conduit, alors on parle de sons non-voisés.

1.3.1 La fréquence du fondamental

La vitesse à laquelle les cordes vocales s'ouvrent et se referment lors du processus de phonation, produit une vibration d'une hauteur variable appelée fréquence du fondamental dont la valeur est étroitement liée à la taille de l'appareil phonatoire de la personne, cette fréquence est quasi stationnaire pour un signal de type voisé, elle varie de [14]:

- de 80 à 200 Hz pour une voix masculine,
- de 250 à 450 Hz pour une voix féminine,
- de 200 à 600 Hz pour une voix d'enfant.

Deux sons de même intensité et de même hauteur se distinguent par le timbre, qui est déterminé par les harmoniques du fondamental [14]. Un intérêt majeur pour la fréquence du fondamental se trouve dans les applications de la synthèse de parole.

1.3.2 Les formants

Le spectre du signal vocal résultant de l'action des sources de sons sur le conduit vocal présente des maximums et des minimums qui correspondent aux résonances et aux anti-résonances du conduit vocal, appelés formants et anti-formants. Du point de vue perceptif, seul les trois premiers formants jouent un rôle essentiel pour caractériser le

spectre vocal [14]. On peut caractériser toute voyelle en n'utilisant que ses trois premiers formants. En général la fréquence du premier formant varie de 200 à 900 Hz, celle du second de 500 à 2500 Hz et le troisième se situe entre 1500 et 3500 Hz. Des formants d'ordre supérieur existent même si leur rôle sur le plan perceptif est limité, ils contribuent à caractériser la voix.

1.3.3 Les sons en parole

Dans le processus de communication parlée, pour une langue donnée, les sons permettent de distinguer les différentes unités de signification du langage [2]. Pour réaliser cette distinction, les phonéticiens ont défini le phonème comme unité sonore minimale.

1.3.3.1 Le phonème

Le phonème [14] est la plus petite unité présente dans la parole et susceptible par sa présence de changer la signification d'un mot. Le nombre de phonèmes est toujours très limité, en générale il est inférieur à 50.

La notion de phonème ne tient compte que des caractéristiques acoustiques qui permettent une distinction entre des mots [2], elle ne tient pas compte des phénomènes physiques de la production du son.

1.3.3.2 La classification des phonèmes

Les phonèmes peuvent êtres rangés en catégories [2] selon des traits distinctifs qui indiquent une similitude au niveau articulatoire, acoustique ou perceptif. On retrouve les voyelles et les consonnes.

Les voyelles sont caractérisées par la vibration des cordes vocales, le lieu de l'articulation et la stabilité des articulations produisant des sons tenus pendant un certain laps de temps, les voyelles peuvent être rangées selon :

- la nasalité.
- l'ouverture du conduit vocal.
- la position de la constriction du conduit vocal.
- l'arrondissement des lèvres.

Les consonnes se prononcent avec un rétrécissement du passage de l'air et sont classées selon :

- le voisement (selon que les cordes vocales vibrent ou non à leur passage).
- le mode d'articulation (occlusif, nasal, fricatif)
- le lieu d'articulation (labiale, dentale, palatale)

1.3.4 La représentation de la parole dans les domaines temps et fréquence

Une représentation de l'évolution temporelle du signal vocal ou audiogramme est donnée dans la Figure 3, cependant pour avoir plus d'informations sur la fréquence du fondamental et les formants, généralement on utilise une représentation 3-D (amplitude/fréquence/temps) appelée spectrogramme.

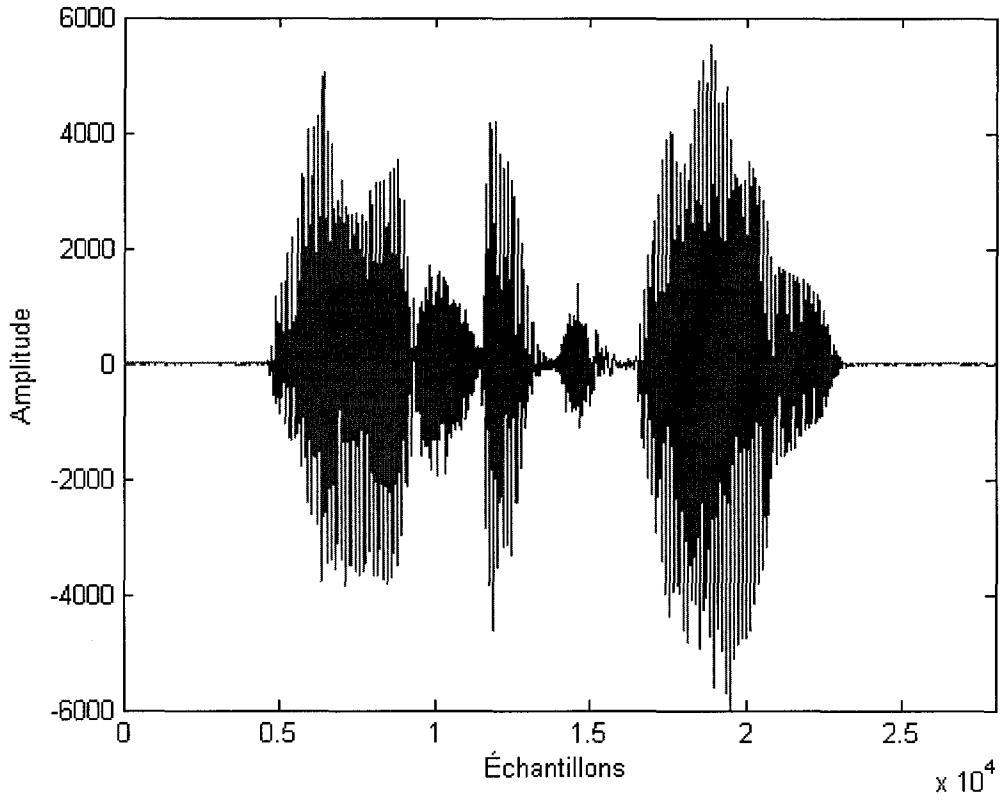

Figure 3 Représentation numérisée d'un signal vocal

1.3.4.1 Le spectrogramme

Le spectrogramme est une représentation tridimensionnelle, où le temps est représenté sur l'axe X, la fréquence sur l'axe Y et le niveau de chaque fréquence sur l'axe Z est symbolisé par le niveau du gris. Pour l'obtenir, on effectue sur le signal une FFT (Fast Fourier Transform) à fenêtre glissante.

On distingue deux types de spectrogrammes [13], les spectrogrammes à bandes larges (voir Figure 4) et les spectrogrammes à bandes étroites (voir Figure 5). Les premiers

sont obtenus avec des fenêtres de faible durée. Ils mettent en évidence l'enveloppe spectrale (les formants) du signal, les périodes voisées y apparaissent sous la forme de bandes verticales plus sombres. Les spectrogrammes à bande étroite sont obtenus avec des fenêtres de l'ordre de 30 à 40 ms, ils offrent une bonne résolution au niveau fréquentiel, les harmoniques du signal dans les zones voisées y apparaissent sous la forme de bandes horizontales.

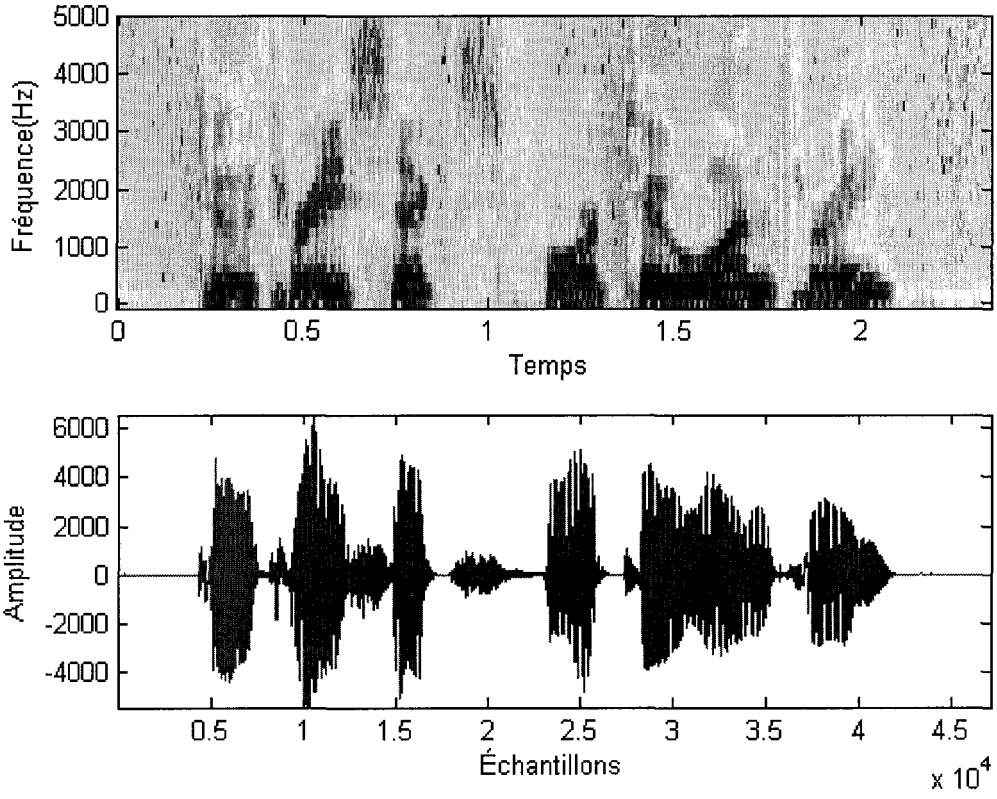

Figure 4 Spectrogramme à bande large (128 échantillons)

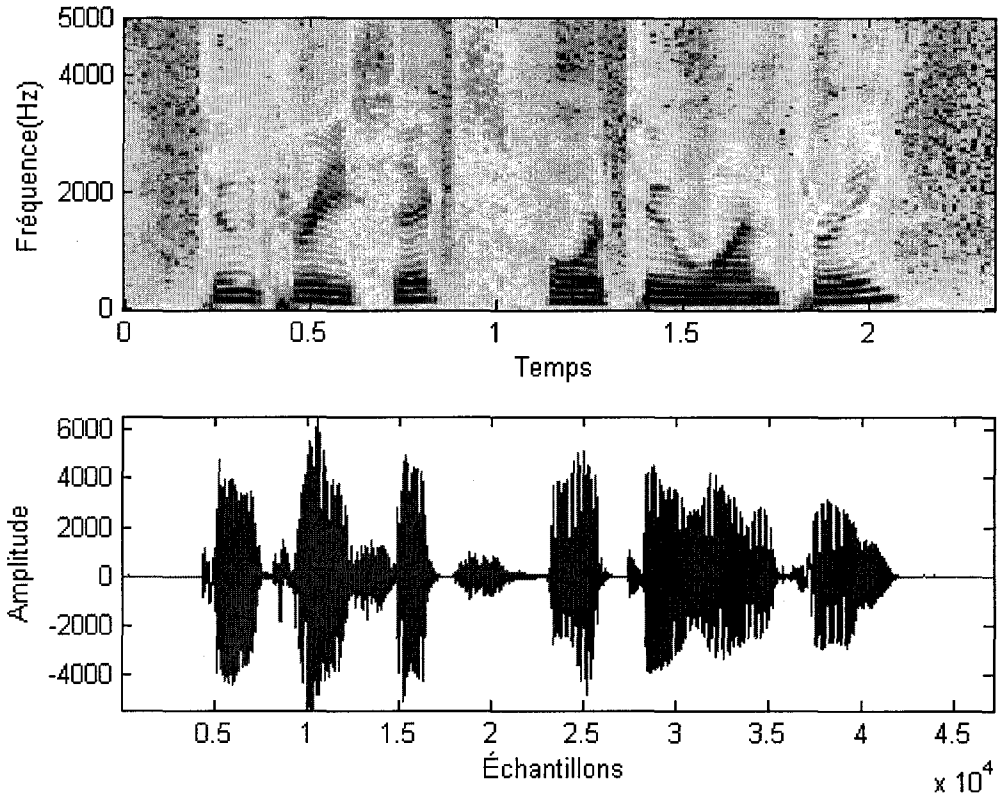

Figure 5 Spectrogramme à bande étroite (512 échantillons)

1.3.5 Modèle de production de la parole

À défaut de pouvoir travailler facilement sur le signal vocal dans les applications de reconnaissance, de codage ou de synthèse, généralement lors de l'analyse du signal on définit un modèle qui lui correspond. Une analyse est nécessaire pour déterminer les valeurs optimales des paramètres de ce modèle afin de réduire au minimum l'erreur entre le modèle et le signal modélisé.

Il existe de nombreux modèles de production de la parole on distingue :

1.3.5.1 Les modèles articulatoires

Ces modèles réalisent une simulation numérique du mécanisme de phonation. Dans ce cas on exploite les connaissances acquises sur la géométrie tridimensionnelle du conduit vocal, ainsi que celles sur les paramètres articulatoires (position de la langue, ouverture des lèvres,...etc.). En théorie ces modèles permettent la génération d'une multitude de voix au moyen d'une simple modification d'un ensemble de paramètres articulatoires.

1.3.5.2 Les modèle électriques

Ce modèle proposé par Fant en 1960 [14], utilise l'équivalent électrique du mécanisme de la production de la parole, on y décrit la parole comme le signal produit par un système constitué de générateurs et de filtres numériques, les paramètres de ces modèles sont ceux des générateurs et filtres qui les constituent. Dans cette catégorie on retrouve les modèles AR et les modèles ARMA qui seront détaillés dans la suite de ce travail.

1.4 Audition et perception de la parole

1.4.1 Structure du système auditif

L'oreille est composée de trois parties reliées l'une à l'autre, dans l'ordre on retrouve : l'oreille externe, l'oreille moyenne et l'oreille interne [14]. Un schéma de l'oreille est donné à la Figure 6.

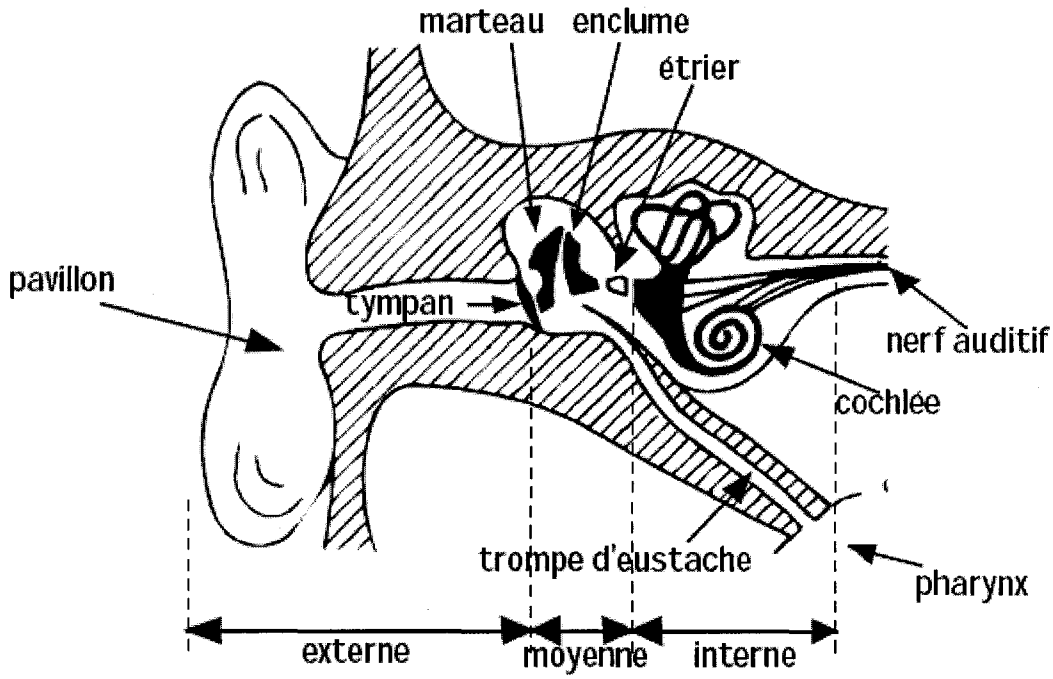

Figure 6 Structure du système auditif [13]

1.4.1.1 L'oreille externe

Chargée de recevoir le signal sonore, l'oreille externe est constituée du pavillon, qui dirige le flux sonore vers le conduit auditif externe, un tube acoustique de section uniforme, terminé par le tympan. Comme tout tube, le conduit auditif a des fréquences de résonance, la première étant proche de 3000 Hz, ce qui accroît la sensibilité du système auditif dans cette gamme de fréquences.

1.4.1.2 L'oreille moyenne

C'est une cavité d'air, sa fonction principale est la réalisation de l'adaptation d'impédance entre l'air et le milieu liquide de l'oreille interne pour améliorer la transmission du son. Elle a aussi pour mission de protéger l'oreille interne des sons très

élevés qui risquent de l'endommager. Du point de vue fréquentiel on peut considérer l'oreille moyenne comme un filtre passe-bas. Elle est composée des éléments suivants :

- le tympan : une membrane sensible aux variations de pression acoustique, l'équilibrage de la pression des deux côtés du tympan est assuré par la trompe d'eustache, reliée à l'appareil respiratoire
- les osselets : le marteau, l'enclume et l'étrier, qui jouent le rôle de levier, amplifient les vibrations reçues par le tympan.

1.4.1.3 L'oreille interne

Son rôle est de convertir les vibrations mécaniques issues de l'oreille moyenne est de les acheminer ensuite vers le cerveau. L'oreille interne est un milieu liquide constitué des éléments suivants :

- les trois canaux semi-circulaires : qui sont le départ du nerf vestibulaire.
- la cochlée : qui est un canal membraneux enroulé, elle contient la membrane basilaire qui transforme les vibrations mécaniques en impulsions nerveuses, aussi elle supporte par la membrane basilaire, l'organe de corti qui contient environ 25000 cellules ciliées. Ces cellules sont caractérisées par des fréquences de résonances (voir Figure 7) qui dépendent de la position de chaque cellule sur la membrane.

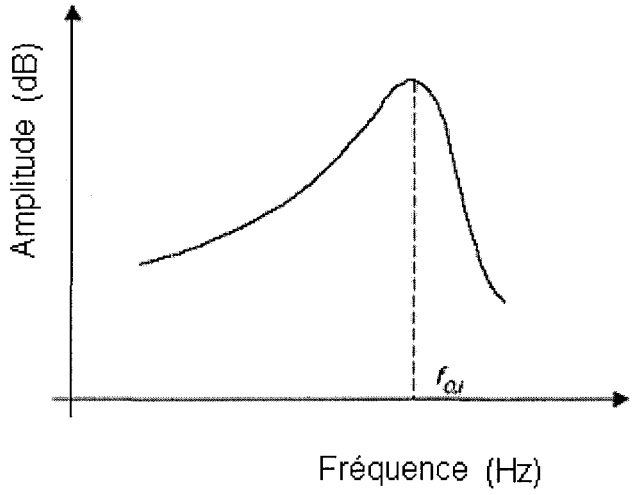

Figure 7 Réponse en fréquence d'une cellule ciliée [13]

1.4.2 Acoustique de l'audition

La gamme de fréquences couverte par le système auditif humain est principalement située entre 800 Hz et 8000 Hz. Les limites extrêmes peuvent s'étaler à des fréquences aussi basses que 20 Hz et aussi hautes que 20000 Hz [14], cependant il est pertinent de chercher à savoir comment l'information auditive est réellement perçue. C'est la psychoacoustique qui va essayer de répondre à cette question, une science qui s'intéresse à l'acoustique de l'audition, son objet est l'étude expérimentale des relations quantitatives entre les stimulus acoustiques mesurables physiquement tel que l'intensité, la fréquence, le spectre, et le temps, et les réponses du système auditif de l'être humain : sensations et perceptions auditives [2].

1.4.2.1 Les sensations auditives

Ces sensations sont [2] : le phone, la hauteur perçue ou tonie, le timbre et la durée.

La hauteur d'un son pur est liée à la fréquence de l'onde sonore, mais au delà de 1000Hz il faut plus que doubler la fréquence pour percevoir un doublement de la hauteur. L'échelle de tonie est graduée en mels. Un écart constant en mels est perçu comme un écart constant en hauteur.

Le phone permet de mesurer le niveau de l'intensité perçue d'un son pur, Les courbes isosoniques [13] (Figure 8) montrent que la courbe de réponse de l'oreille dépend de la fréquence et de la pression sonore. En effet elles révèlent un maximum de sensibilité dans la plage [50Hz, 10KHz], en dehors de cette plage les sons doivent être plus intenses pour être perçus.

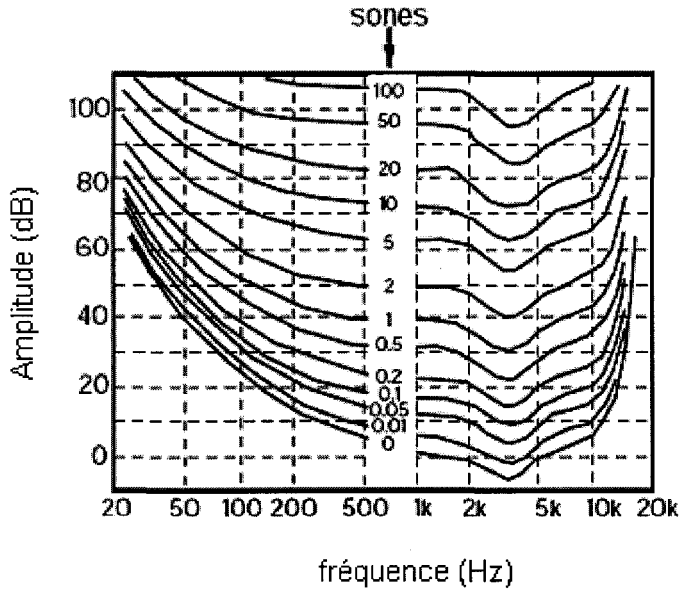

Figure 8 Courbes isosoniques [13]

Le timbre dépend de la répartition spectrale d'un son complexe, timbre clair pour une prédominance des fréquences hautes, timbre sombre pour une prédominance des basses fréquences.

1.4.2.2 Les bandes critiques

Le concept de bandes critiques se rapporte à une propriété importante en psychoacoustique appelée le phénomène de masquage [2], c'est une propriété qui stipule que lorsqu'on entend simultanément deux sons purs de fréquences différentes, il arrive que du point de vue perceptuelle l'un des deux domine l'autre, et fait que juste le son dominant soit audible, cet effet de masque dépend des intensités et fréquences relatives des deux sons. Pour prendre en considération ce phénomène, Fletcher [2] a suggéré que le système auditif se comporte comme un banc de filtres qui se chevauchent et dont les fréquences centrales s'échelonnent continûment. Il a modélisé cet ensemble par une série de filtres rectangulaires dont la largeur a été appelée bande critique, et un son dont la fréquence est à l'intérieur de cette bande peut influencer la perception des autres sons dans la même bande, mais pas en dehors de cette bande critique.

1.5 Conclusion

Dans ce chapitre nous avons passé en revue le mécanisme de la production de la parole, le principe de son audition ainsi que les caractéristiques générales du signal vocal.
Principalement on peut classer le signal vocal en deux catégories : les sons voisés, résultants de la vibration des cordes vocales, et les sons non voisés qui ne nécessitent pas l'intervention du larynx.

Il existe plusieurs modèles pour le traitement du signal vocal, le modèle le plus pratique et le plus utilisé est l'équivalent électrique du mécanisme de la production de la parole.

Nous étudierons dans le prochain chapitre les différents outils de traitements de la parole, qu'on appelle aussi analyse court-terme, nous présenterons aussi quelques méthodes de paramétrisation du signal vocal.

CHAPITRE 2

OUTILS POUR LE TRAITEMENT DE LA PAROLE

2.1 Introduction

Dans ce chapitre nous décrirons les différents outils nécessaires au traitement de la parole. Nous commencerons par un bref aperçu sur le traitement à court-terme, ensuite nous exposerons les concepts de base de la paramétrisation du signal vocal avec la présentation de l'analyse par prédiction linéaire pour les coefficients LPC (Linear Predictive Coding), et l'analyse homomorphique qui est à l'origine de deux types de coefficients, les MFCCs (Mel Frequency Cepstral Coefficients) et les coefficients LPCCs (Linear Predictive Cepstral Coefficients).

2.2 Traitement du signal court-terme

2.2.1 Définition

Le signal parole est un processus aléatoire non stationnaire, or les outils de traitements du signal conventionnels sous-entendent la stationnarité du signal, alors on va exploiter le fait que le signal parole soit quasi stationnaire sur des courts segments de parole appelés "frames" en anglais. Ces derniers sont des tranches temporelles de 10 à 45 ms, d'où l'appellation de l'analyse à cour-terme.

2.2.2 Les fenêtres

Généralement le découpage du signal dans le domaine temporel équivaut à multiplier le signal par une fonction rectangulaire, ce qui équivaut à une convolution dans le domaine fréquentiel entre le spectre du signal analysé et celui de la fenêtre. Dans la majorité des cas la fenêtre rectangulaire s'avère trop brutale. En effet il a été démontré [15] que toute variation rapide dans le domaine temporel correspond à des hautes fréquences dans le domaine fréquentiel qui se traduit par des ondulations sur le spectre. Alors on lui préfère d'autres fenêtres (voir Figure 9) plus douces. Parmi les fenêtres les plus utilisées on trouve [16]:

Rectangulaire:

$$w(n) = \begin{cases} 1 \text{ pour } 0 \le n \le N-1 \\ 0 \text{ ailleurs} \end{cases} \tag{2.1}$$

Bartlett :

$$w(n) = \begin{cases} 2n/(N-1) \text{ pour } 0 \le n \le (N-1)/2 \\ 2-2n/(N-1) \text{ pour } (N-1)/2 < n \le N-1 \\ 0 \text{ ailleurs} \end{cases} \tag{2.2}$$

Hanning :

$$w(n) = \begin{cases} 0.5-0.5\cos(2\pi n/(N-1)) \text{ pour } 0 \le n \le N-1 \\ 0 \text{ ailleurs.} \end{cases} \tag{2.3}$$

Hamming :

$$w(n) = \begin{cases} 0.54-0.46\cos(2\pi n/(N-1)) \text{ pour } 0 \le n \le N-1 \\ 0 \text{ ailleurs.} \end{cases} \tag{2.4}$$

Blackman :

$$w(n)=\begin{cases}0.42-0.5\cos(2\pi n/(N-1))+0.08\cos(4\pi n/(N-1)) & \text{pour } 0\le n\le N-1\\ 0 \text{ ailleurs.}\end{cases} \quad (2.5)$$

où N représente la longueur de la fenêtre, et n un échantillon du signal.

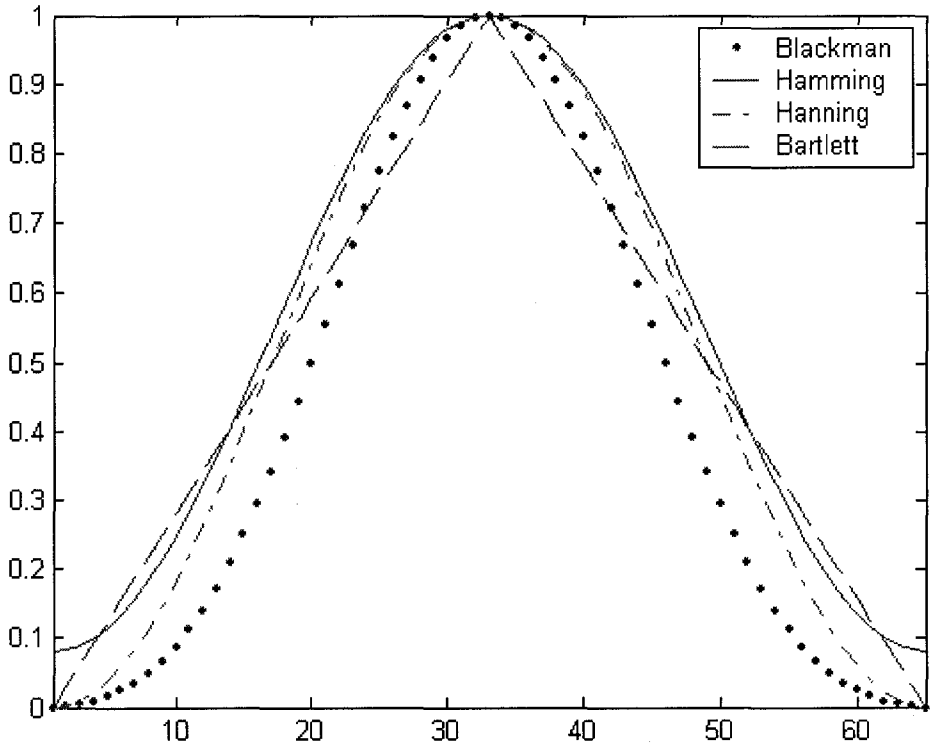

Figure 9 Exemples de fenêtres de pondération

En pratique la fenêtre de Hamming, est souvent la plus utilisée, la Figure 10 est une illustration de son spectre et celui de la fenêtre rectangulaire. Dans cette figure on voit

clairement que la fenêtre de Hamming permet une grande atténuation en dehors de la bande passante comparativement à la fenêtre rectangulaire d'où son avantage.

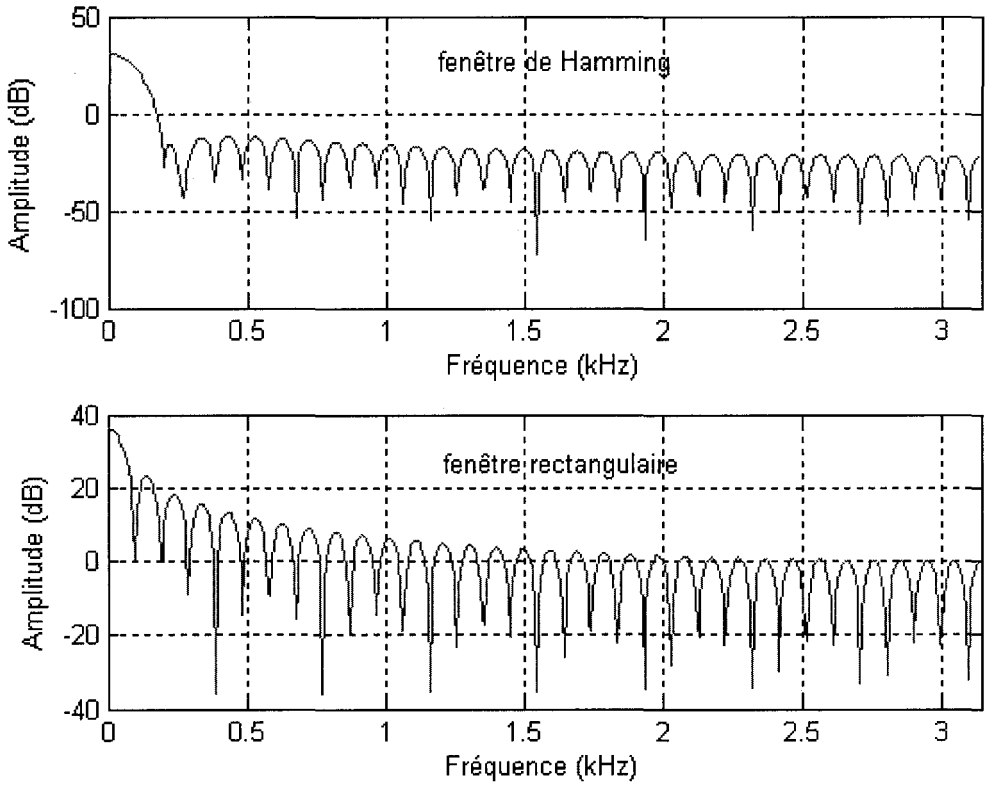

Figure 10 Spectre des fenêtres de Hamming et Rectangulaire

Lors du traitement du signal on peut prendre des fenêtres avec recouvrement (overlapped) ou non. La région de recouvrement peut varier de 0 à 75% de la taille de la fenêtre N.

2.2.3 Énergie à court- terme

Un des outils qui permettent de fournir une représentation fidèle des variations de l'amplitude du signal vocal $x(n)$ dans le temps est l'énergie court terme (voir Figure 11). En générale elle est définie par [17]:

$$E_n = \sum_{m=-\infty}^{\infty} \left[x(m)w(n-m)\right]^2 \tag{2.6}$$

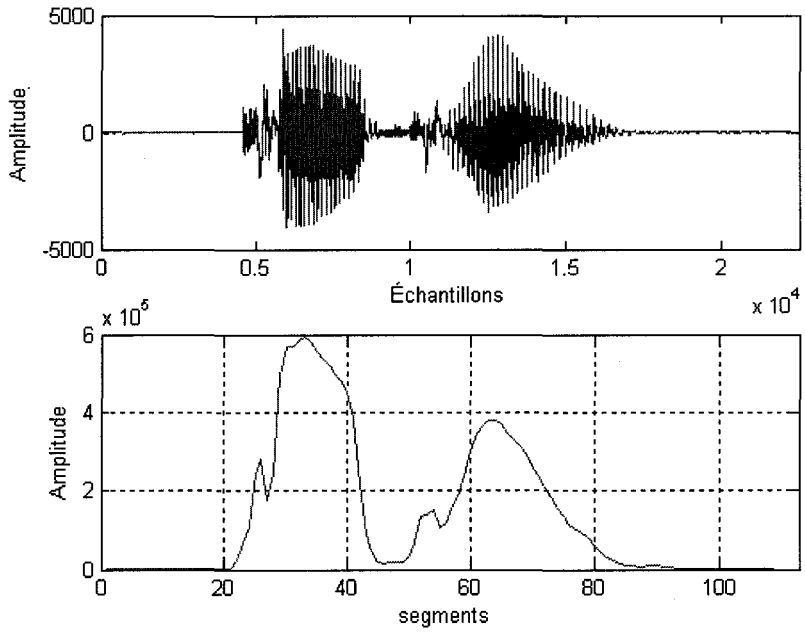

Figure 11 L'énergie à court terme d'un signal vocal

En absence du bruit de mesure, l'énergie s'avère un outil efficace pour séparer la parole du silence [18]

2.2.4 Amplitude moyenne

L'énergie à court terme avec une élévation au carré pour chaque échantillon, est très coûteuse en terme de temps de calcul. En pratique on préfère utiliser une autre forme qu'on appelle l'amplitude moyenne, elle est définie par :

$$M_n = \sum_{m=-\infty}^{\infty} |x(m)|w(n-m) \tag{2.7}$$

2.2.5 Puissance à court terme

La puissance à court terme d'un segment de parole de longueur N est définie par:

$$P_n = \frac{1}{N} \sum_{m=-\infty}^{+\infty} \left[x(m)w(n-m)\right]^2 \tag{2.8}$$

Il faut noter que l'énergie court terme et la puissance court terme fournissent à un facteur près (1/N) la même information.

2.2.6 Le taux de passage par zéro à court terme

Un autre outil très utile de traitement de la parole est le taux de passage par zéro (zero crossing rate en anglais). Pour un signal échantillonné, il y'a passage par zéro lorsque

deux échantillons successifs sont de signes opposés [14]. Le taux de passage par zéro court terme est estimé par la formule :

$$Z_n = \frac{1}{2}\sum_m \left|\mathrm{sgn}\left[x(m)\right] - \mathrm{sgn}\left[x(m-1)\right]\right| w(n-m) \tag{2.9}$$

avec

$$\mathrm{sgn}\left[x(m)\right] = \begin{cases} 1\,, si \;\; x(m) \geq 0. \\ -1\,, si \;\; x(m) < 0. \end{cases} \tag{2.10}$$

et

$$w(n) = \begin{cases} \dfrac{1}{N} & ,0 \leq n \leq N \\ 0 & ,\text{ailleurs.} \end{cases} \tag{2.11}$$

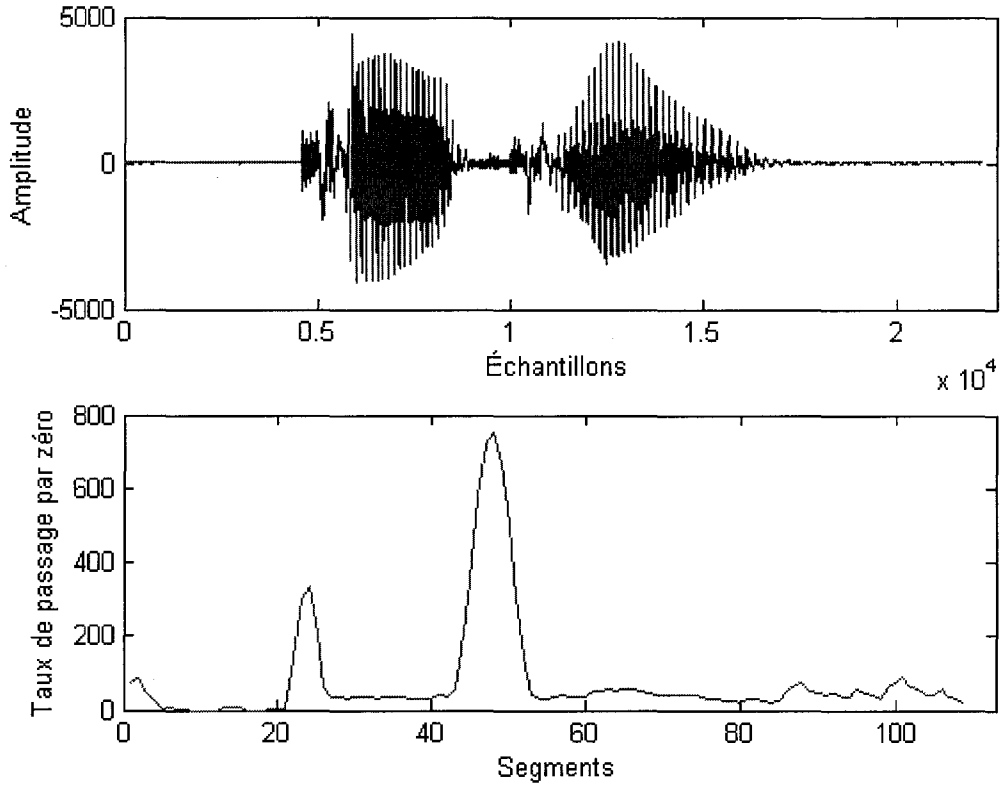

Figure 12 Le taux de passage par zéro d'un signal vocal

Une caractéristique pour le taux de passage par zéro, est qu'il est élevé pour le son non voisé et faible pour le son voisé. Le taux de passage par zéro constitue un outil important pour la classification voisé/non voisé, et pour la détection du début et la fin de la parole dans un signal vocal.

2.2.7 L'autocorrélation à court terme

La fonction d'autocorrelation d'un signal ergodique et stationnaire est définie par [14]:

$$\phi_{xx}(k) = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} x(n)x(n+k) \tag{2.12}$$

Sur un segment de durée finie, c'est le cas d'un segment (fenêtre) de parole, on peut estimer cette fonction par :

$$\hat{\phi}_{xx}(k) = \frac{1}{N-k} \sum_{n=0}^{N-k-1} x(n)x(n+k) \tag{2.13}$$

L'idée de l'utilisation de la fonction d'autocorrélation est de déterminer à quel point deux échantillons successifs d'un signal se ressemblent [15]. Parmi ses autres applications on peut citer son utilisation pour l'estimation de la fréquence du fondamental (ou pitch). En effet ce dernier se manifeste par un pic dans la fonction de l'autocorrélation qu'il faut isoler adéquatement pour pouvoir l'évaluer.

2.3 La paramétrisation du signal

La paramétrisation du signal parole, appelée aussi pré-traitement acoustique, se décompose en trois étapes, un filtrage analogique, une conversion analogique/numérique et un calcul de coefficients. Son rôle est de fournir et d'extraire des informations caractéristiques du signal pour produire une représentation moins redondante du signal.
En reconnaissance de la parole, les paramètres extraits doivent être précis, et leur nombre raisonnable, pour minimiser le temps de calcul ainsi que pour réduire la quantité de mémoire allouée.

Pour l'extraction des paramètres plusieurs méthodes existent :

- les méthodes fondées sur la décomposition fréquentielle du signal sans connaissance a priori de sa structure fine, dans cette catégorie la transformée de fourrier rapide ou FFT [18] est la plus utilisée pour l'obtention des spectres.
- les méthodes fondées sur la connaissance des mécanismes de production de la parole, parmi ces méthodes on peut citer [14] le codage prédictif linéaire LPC, et le cepstre où on tente simplement de déconvoluer la source et le conduit.
- les méthodes basées sur le modèle de perception [2], elles consistent à définir des bandes critiques de perception, correspondant à la distribution fréquentielle de l'oreille humaine. Les coefficients sont les sorties de bancs de filtres calibrées à partir de ces résultats.

Dans ce travail nous avons choisi la deuxième catégorie (les méthodes basées sur la connaissance des mécanismes de production de la parole) comme méthode de paramétrisation. Dans ce qui suit, nous citerons quelques unes de ces méthodes.

2.3.1 La méthode d'analyse par prédiction linéaire

Cette méthode [18] a pour objectif une représentation directe du signal vocal sous la forme d'un nombre limité de paramètres. Sa puissance provient du fait qu'elle est fondée sur un modèle simple de production de la parole qui s'approche du système phonatoire. La Figure 13 est une représentation d'un spectre vocal obtenu à l'aide de la méthode LPC.

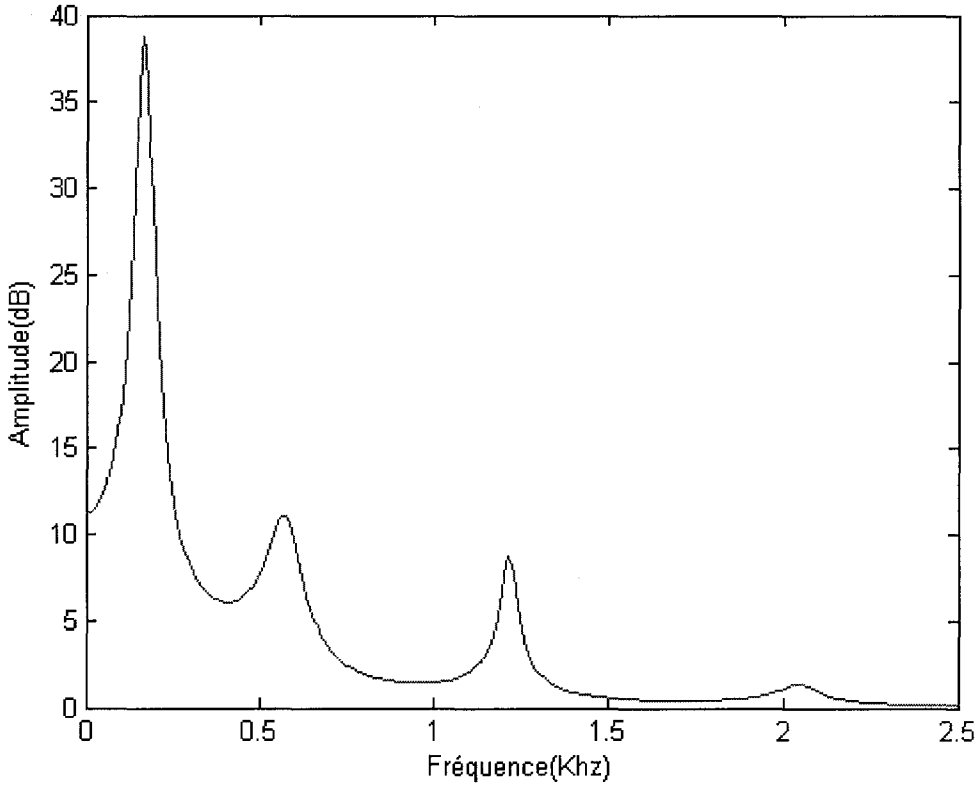

Figure 13 Exemple d'un spectre LPC

Le principe de cette méthode est fondé sur l'hypothèse selon laquelle un échantillon du signal de parole $x(nT_S)$, où T_S est la période d'échantillonnage, peut être prédit approximativement par une somme pondérée linéairement de p échantillons le précédant immédiatement, p est appelé l'ordre de prédiction.

On distingue deux modèles de prédiction, les modèles auto régressifs AR, et ceux appelés auto régressifs à moyenne ajustée ou ARMA.

2.3.1.1 Modèle auto régressif

Sommairement on peut représenter le mécanisme de production de la parole par le système suivant (Figure 14):

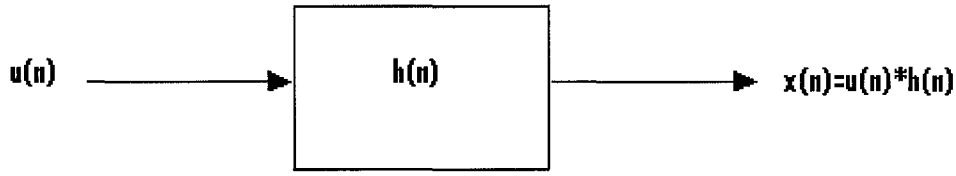

Figure 14 Modèle simplifié de la production de la parole

où $h(n)$ est la réponse impulsionnelle du filtre et $u(n)$ un signal d'excitation.

Pour un signal voisé l'excitation est un train périodique d'impulsions d'amplitude unité (2.14), par contre pour les sons non voisés, l'excitation est un bruit blanc de moyenne nulle et de variance unité.

$$u(n) = \sum_{k} \delta(n - kP) \tag{2.14}$$

Avec P la période du fondamental.

D'autre part en utilisant la transformée en Z de $x(n)$ on peut écrire:

$$X(z) = U(z).\frac{\sigma}{A(z)} \tag{2.15}$$

où

$$A(z) = 1 - \sum_{i=1}^{p} a(i) z^{-i} \tag{2.16}$$

Qui donne dans le domaine temporel :

$$x(n) = \sum_{i=1}^{p} a(i) x(n-i) + \sigma u(n) \tag{2.17}$$

Ainsi on peut prévoir l'échantillon $x(n)$ à partir d'une combinaison linéaire des p échantillons qui le précèdent, en plus de l'excitation $u(n)$. Ce modèle (Figure 15) est appelé modèle auto régressif (AR). Les coefficients a_i du filtre sont appelés les coefficients de prédiction [14].

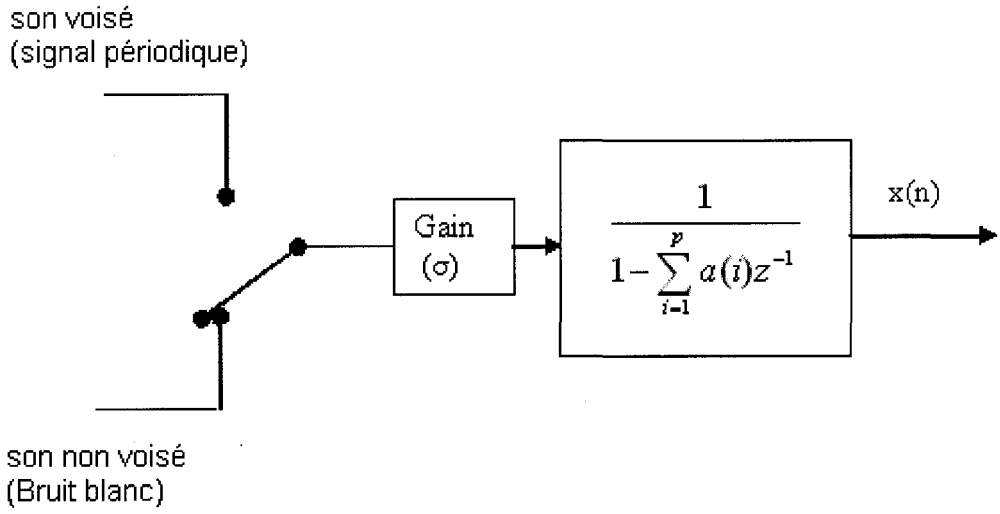

Figure 15 Modèle auto régressif

2.3.1.2 Le modèle ARMA [14]

La forme tous-pôles $\left(\frac{\sigma}{A(z)}\right)$ utilisée pour la transmittance du conduit vocal dans le modèle AR n'est pas vraiment une forme complète. Il s'agit juste d'une forme simplifiée ou d'une approximation. Dans le cas, par exemple, des sons nasalisés, la transmittance est plutôt de la forme :

$$X(z) = U(z)\frac{\sigma C(z)}{A(z)} \tag{2.18}$$

Avec :

$$C(z) = \sum_{i=0}^{q} c(i)z^{-i} \ , c(0) = 1 \tag{2.19}$$

ce qui donne dans le domaine temporel :

$$x(n)-\sum_{i=1}^{p} a(i)x(n-i)=\sigma\sum_{i=0}^{q} c(i)u(n-i) \tag{2.20}$$

C'est l'équation d'ordre (p,q) du modèle appelé auto régressif à moyenne ajustée ou ARMA.

2.3.1.3 L'estimation des coefficients de prédiction

Après avoir établi le modèle, reste le problème de déterminer ses paramètres optimaux, c'est à dire les coefficients du filtre tous pôles pour lesquels l'erreur de prédiction est minimale, avec la seule information à priori le signal de sortie, le signal de l'entrée lui étant inconnu.

Supposons qu'on a une estimation (ou prédiction) d'un échantillon $x(n)$ à partir des p échantillons qui le précèdent on peut écrire alors :

$$x(n)=\sum_{k=1}^{p} a_k x(n-k) \tag{2.21}$$

L'erreur de prédiction est définie par:

$$e(n)=x(n)-\tilde{x}(n) \tag{2.22}$$

D'autre part on définit l'énergie résiduelle de prédiction par [14]:

$$E_n=\sum_m e^2(m)=\sum_m \left(x_n(m)-\tilde{x}_n(m)\right)^2 \tag{2.23}$$

La minimisation de cette erreur est à la base de la détermination des coefficients de prédiction a_i. Ainsi les coefficients a_i optimaux seront tirés de l'équation suivante :

$$\frac{\partial E_n}{\partial a_k} = 0 \ , k = 1, 2, ..., p \tag{2.24}$$

Un système d'équations en découle appelée équations de Yule Walker dont la résolution a été approchée par plusieurs méthodes. Les plus importantes sont :

- la méthode de l'autocorrélation.
- la méthode de la covariance.

2.3.1.3.1 Méthode de l'autocorrélation [18]

Dans ce cas on suppose que le signal vocal est nul en dehors d'un certain intervalle, cela revient à multiplier le signal $x(m+n)$ par une fenêtre $w(m)$, en général une fenêtre de Hamming

$$x_n(m) = \begin{cases} x(m+n)w(m) \ , 0 \le m \le N-1 \\ 0 \ , \text{ailleurs} \end{cases} \tag{2.25}$$

Ainsi l'énergie résiduelle sera évaluée pour les valeurs de m, $0 \le m \le N-1+p$:

$$E_n = \sum_{m=0}^{N+p-1} e_n^2(m) \tag{2.26}$$

on aura alors à résoudre l'équation suivante :

$$\sum_{k=1}^{p} r_n(|i-k|)\hat{a}_k = r_n(i) \ ,1 \le i \le p \tag{2.27}$$

où

$$r_n(k) = \sum_{m=0}^{N-k-1} x_n(m)x_n(m+k) \tag{2.28}$$

p étant l'ordre de la prédiction, et r_n la fonction d'autocorrélation.

La forme matricielle de cette équation donne :

$$\begin{bmatrix} r_n(0) & r_n(1) & r_n(2) & \cdots & r_n(p-1) \\ r_n(1) & r_n(0) & r_n(1) & \cdots & r_n(p-2) \\ r_n(2) & r_n(1) & r_n(0) & \cdots & r_n(p-3) \\ \vdots & \vdots & \vdots & & \vdots \\ r_n(p-1) & r_n(p-2) & r_n(p-3) & \cdots & r_n(0) \end{bmatrix} \begin{bmatrix} \hat{a}_1 \\ \hat{a}_2 \\ \hat{a}_3 \\ \vdots \\ \hat{a}_p \end{bmatrix} = \begin{bmatrix} r_n(1) \\ r_n(2) \\ r_n(3) \\ \vdots \\ r_n(p) \end{bmatrix} \tag{2.29}$$

On peut remarquer que nous sommes en présence d'une matrice symétrique, dont tous les éléments de la diagonale sont égaux, il s'agit alors d'une matrice dite matrice de Toeplitz, une caractéristique très importante qui va faciliter la résolution de ce système d'équations. En effet il existe une méthode simple pour le résoudre, c'est l'algorithme de Levinson-Durbin [19], dont l'idée de base est l'utilisation d'une récursion pour arriver à

la $(i+1)^{ème}$ solution, à partir de la $i^{ème}$ solution, et ainsi de suite jusqu'à atteindre l'ordre p, et à chaque $i^{ème}$ niveau on obtient le coefficient a_i du filtre qui permet de minimiser l'énergie résiduelle, ou aussi l'erreur de prédiction.

2.3.1.3.2 La méthode de la covariance [18]

Dans cette méthode on fixe dès le départ une portion du signal [0, N-1], sur laquelle l'énergie résiduelle est évaluée :

$$E_n = \sum_{m=0}^{N-1} e_n^2(m) \tag{2.30}$$

Dans ce cas on ne fait pas la supposition (2.25) et on n'applique pas de fenêtrage au signal, alors contrairement à la méthode de l'autocorrélation où la covariance peut être réduite à une simple autocorrélation, la fonction de covariance définit par (2.31) est utilisée directement dans le système d'équations pour trouver les coefficients a_i.

$$\phi_n(i,k) = \sum_{m=-i}^{N-i-1} x_n(m)x_n(m+i-k) \ , 1 \le i \le p \ et \ 0 \le k \le p \tag{2.31}$$

Ainsi le système d'équations à résoudre aura la forme matricielle suivante :

$$\begin{bmatrix} \phi_n(1,1) & \phi_n(1,2) & \phi_n(1,3) & \cdots & \phi_n(1,p) \\ \phi_n(2,1) & \phi_n(2,2) & \phi_n(2,3) & \cdots & \phi_n(2,p) \\ \phi_n(3,1) & \phi_n(3,2) & \phi_n(3,3) & \cdots & \phi_n(3,p) \\ \vdots & \vdots & \vdots & & \vdots \\ \phi_n(p,1) & \phi_n(p,2) & \phi_n(p,3) & \cdots & \phi_n(p,p) \end{bmatrix} \begin{bmatrix} \hat{a}_1 \\ \hat{a}_2 \\ \hat{a}_3 \\ \vdots \\ \hat{a}_p \end{bmatrix} = \begin{bmatrix} \phi_n(1,0) \\ \phi_n(2,0) \\ \phi_n(3,0) \\ \vdots \\ \phi_n(p,0) \end{bmatrix} \tag{2.32}$$

C'est une matrice qui n'est pas du type Toeplitz, pour la résoudre on n'utilisera pas l'algorithme de Levinson Durbin, mais une autre méthode appelée la décomposition de Cholesky [14].

2.3.1.4 La méthode du treillis [14]

Les coefficients du modèle autorégressif peuvent aussi être estimés par la méthode dite du treillis. En effet à partir des récurrences de Levinson Durbin, on a pu établir pour le filtre inverse (dont les coefficients sont les coefficients de prédiction), une structure en treillis (voir Figure 16), composée de cellules. Pour chaque cellule on définit un coefficient k_m appelé coefficient de corrélation partielle ou parcor (Partial-Correlation). Le nombre de cellules est lui-même l'ordre de prédiction, et le filtre inverse d'ordre p est complètement défini par les paramètres k_m.

Il faut remarquer que le treillis d'ordre p inclu tous les treillis d'ordres inférieurs, on peut alors augmenter l'ordre de la prédiction p par l'ajout d'une nouvelle cellule sans modification des cellules précédentes.

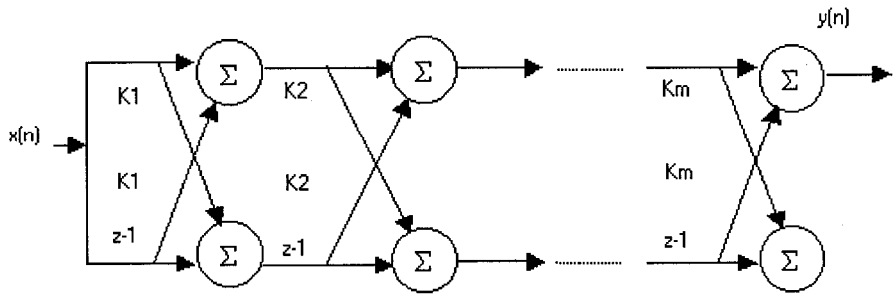

Figure 16 Structure en treillis d'un filtre

2.3.1.4.1 Calcul des coefficients k_m

Les coefficients k_m sont calculés par l'algorithme de Burg qui se résume par ce qui suit [14]:

L'acquisition

$$x(n)=\begin{cases} s(n) & n=0,1,\cdots,N-1 \\ 0 & n<0\ ,\ n>N-1 \end{cases}$$

$s(n)$ Étant un échantillon du signal parole pris sans pondération.

L'initialisation

$pour\ \ n=0,1,\cdots,N-1$

$f_0(n)=x(n)$

$g_0(n)=x(n-1)$

La récursion

pour $m=1,2,\ldots,p$

$$k_m=\frac{\sum_{n=0}^{N-2+m} f_{m-1}(n)g_{m-1}(n)}{\sum_{n=0}^{N-2+m} f_{m-1}^2(n)+\sum_{n=0}^{N-2+m} g_{m-1}^2(n)}$$

$pour\quad n=0,1,\ldots,N-2+m$

$f_m(n)=f_{m-1}(n)+k_m g_{m-1}(n)$

$g_m(n)=k_m f_{m-1}(n-1)+g_{m-1}(n-1)$

Pour cette méthode la matrice des fonctions d'autocorrélation (2.29) n'est pas nécessaire ainsi que la pondération des échantillons du signal, ce qui permet une réduction importante du volume de calcul.

2.3.2 L'analyse homomorphique

2.3.2.1 Définition

Dans le cas d'un signal de parole, tel que mentionné on peut supposer que le signal vocal $x(n)$, résulte de la convolution du signal de la source (de fréquence F_0 pour les sons voisés, et bruit blanc pour les sons non voisés) avec le filtre correspondant au conduit.

$$x(n) = u(n) * h(n) \tag{2.33}$$

Alors on peut approcher le problème de l'estimation des paramètres acoustiques avec une autre méthode d'analyse appelée analyse homomorphique. Son principe consiste à calculer le logarithme de la transformée en z du signal pour séparer les deux composantes de la convolution. Le résultat de cette méthode, des coefficients plus robustes et plus fiables pour la reconnaissance que les classiques LPC.

La Figure 17 est un exemple d'analyse homomorphique, on remarque la présence de la fréquence du fondamental sous formes de pics périodiques, notons aussi que seulement les premiers coefficients cepstraux (spectre vocal) sont nécessaires pour la reconnaissance.

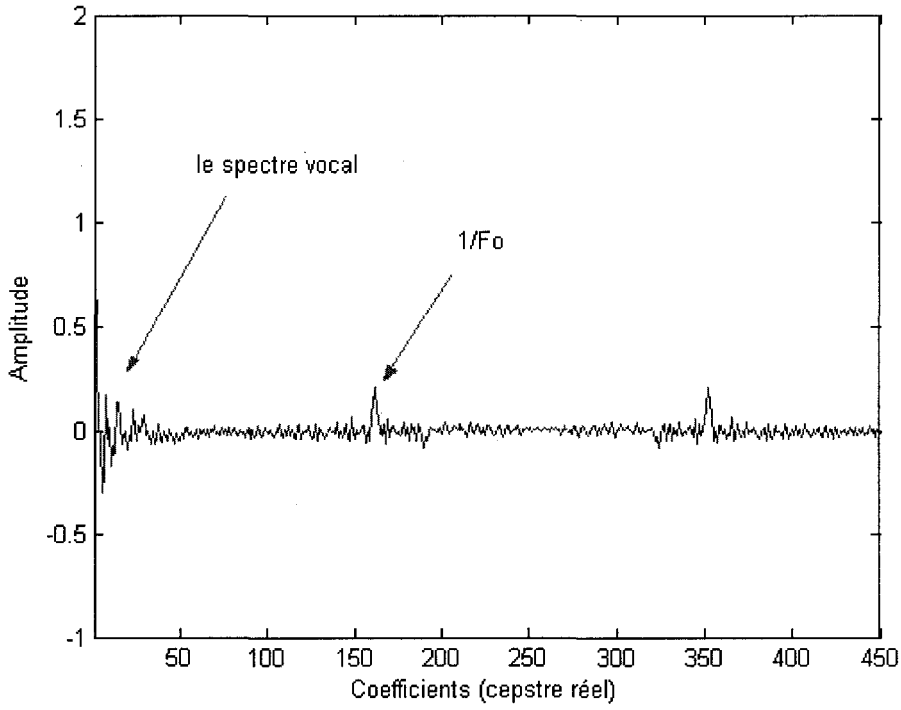

Figure 17 Exemple d'analyse homomorphique

2.3.2.2 Les coefficients cepstraux

La fréquence du fondamental F_0 est responsable de l'apparition de nombreuses harmoniques sur le spectre d'amplitude qui rendent difficile la reconnaissance, d'où la nécessité de l'isoler dans les applications de RAP.

Le but du lissage cepstral, est de découpler la source du conduit, en séparant leurs contributions. Les premiers coefficients cepstraux (voir Figure 18) contiennent l'information relative au conduit vocal se sont les paramètres qui seront utilisés en RAP.

Au-delà d'un échantillon n_0 (qui correspond à la fréquence du fondamental) cette contribution devient négligeable. On peut dire que les pics périodiques visibles pour tout échantillon $n > n_0$, reflètent les impulsions de la source.

Principalement on distingue deux types de coefficients cepstraux [20] :

- les coefficients MFCC.
- les coefficients LPCC.

On opte pour l'un ou l'autre des deux types selon des considérations pratiques dictées par les exigences de l'application ou selon les performances obtenues lors des expériences. Nous utiliserons les coefficients LPCCs dans le cadre de ce projet.

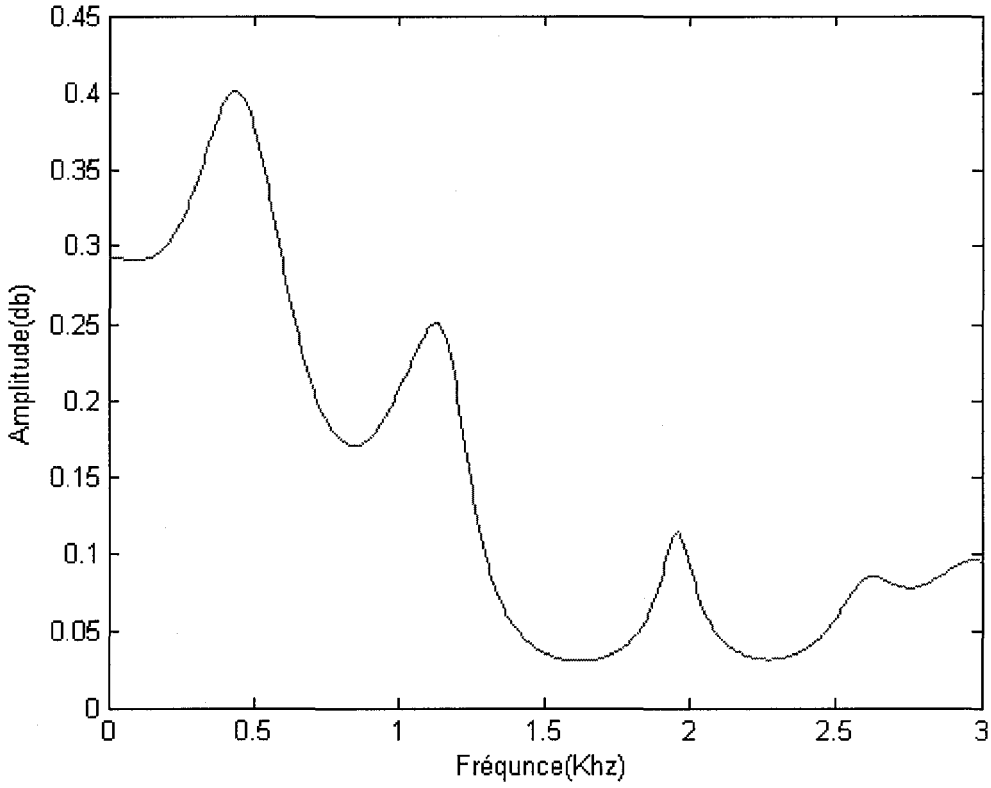

Figure 18 Lissage cepstral obtenu avec des coefficients LPCCs

2.3.2.3 Les coefficients MFCCs

Les coefficients MFCC (Mel Frequency Cepstral Coefficients) [20], sont basés sur une échelle de perception non linéaire (linéaire jusqu'à 1000Hz et logarithmique au-delà de 1000Hz). C'est une échelle qui consiste en la définition de bandes critiques de perception (à l'aide d'un banc de filtres). Elle correspond à la distribution fréquentielle de l'oreille humaine. Le passage de l'échelle fréquentielle à l'échelle de Mel est régit par l'équation suivante :

$$Mel(f) = x \log(1 + \frac{f}{y}) \tag{2.34}$$

Dans la littérature [2, 21]on trouve différentes valeurs pour x et y entre autre:

$x = 1000/\log 2$, $y = 1000$ [2]

$x = 2595$, y = 700 [21]

On utilise un banc de filtres triangulaire (ou rectangulaires), positionnés uniformément sur l'échelle Mel, c'est à dire non uniformément dans le domaine fréquentiel.

On peut résumer le processus pour déterminer les coefficients MFCC par le schéma suivant (Figure 19) :

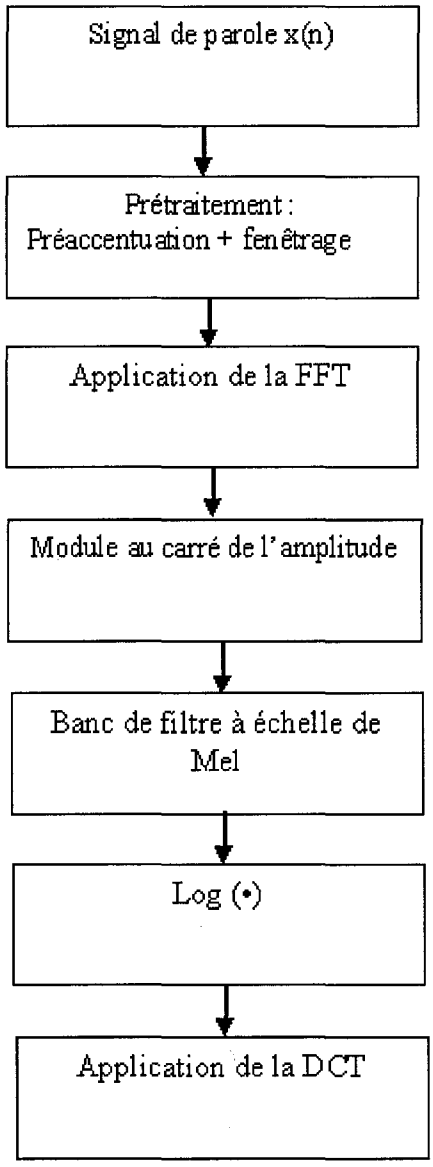

Figure 19 Processus pour l'obtention des coefficients MFCCs

Prétraitement : au début du traitement le signal est préaccentué, puis une fenêtre de type Hamming est utilisée pour décomposer le signal en un ensemble de segments d'échantillons.

FFT: en pratique la transformée en Z est remplacée par une transformée discrète de Fourrier (ou FFT) qui possède les même propriétés de linéarité que la transformée en Z.

Module au carré: la phase de la FFT du signal de parole ne contient pas d'informations suffisamment pertinentes pour la reconnaissance de la parole, alors il est judicieux de garder juste la partie réelle, c'est ce qu'on va faire on prenant seulement le module au carré de la FFT.

Banc de filtres: suite de filtres triangulaires appliqués selon l'échelle de Mel, qui permettent à la fois le lissage du spectre et la réduction de l'information à traiter.

Log(): le logarithme est appliquer pour transformer la multiplication en addition.

DCT (Discret Cosine Transform): à la fin du processus on revient dans l'espace temporel par une FFT inverse. Puisqu'on travaille juste avec la partie réelle du signal, une DCT (transformée cosinus discrète) peut aisément faire la transformée inverse. Si on pose K le nombre de filtres et L le nombre de coefficients qu'on veut avoir les coefficients MFCCs seront [22]:

$$\tilde{c}_n = \sum_{k=1}^{K} \left(\log \tilde{E}_k\right) \cos\left[n\left(k - \frac{1}{2}\right)\frac{\pi}{K}\right], \; n = 1, 2, \ldots, L. \tag{2.35}$$

avec

$\tilde{E}_k$ L'énergie à la sortie des filtres, k=1,2,…, K

On note que le coefficient $\tilde{c}_0$ a été écarté, cela en raison du fait qu'il correspond à la valeur moyenne du signal, qui n'est pas d'un apport important dans les applications de RAP.

2.3.2.4 Les coefficients LPCCs

Un inconvénient des coefficients MFCCs est qu'ils sont très coûteux en terme de temps de calcul du fait qu'ils incorporent le calcul de la FFT dans leurs processus. Une alternative: les coefficients LPCCs [3], obtenus eux par dérivation à partir des coefficients LPC, de la façon suivante :

Posons :

$$\log\left[\frac{1}{A(z)}\right]=\sum_{n=1}^{\infty}c_n Z^{-n},\ A(z)=1-\sum_{k=1}^{p}a_k z^{-k} \tag{2.36}$$

c_n étant les coefficients LPCC qu'on recherche

la dérivée des deux membres par rapport à z nous donne:

$$\begin{aligned}&\left(1-\sum_{k=1}^{p}a_k z^{-k}\right)\left(\sum_{n=1}^{\infty}nc_n z^{-n}\right)=\sum_{n=1}^{p}na_n z^{-n}\\ &\Rightarrow\sum_{n=1}^{\infty}nc_n z^{-n}-\sum_{k=1}^{p}\sum_{n=1}^{\infty}nc_n a_k z^{-(n+k)}=\sum_{n=1}^{p}na_n z^{-n}\end{aligned} \tag{2.37}$$

on remplace m par n-k on aura :

$$\sum_{n=1}^{\infty}nc_n z^{-n}=\sum_{k=1}^{p}\sum_{n=k+1}^{\infty}(n-k)c_{(n-k)}a_k z^{-n}+\sum_{n=1}^{p}na_n z^{-n} \tag{2.38}$$

notons que $n \geq k+1 \Rightarrow k \leq n-1$, alors

$$\begin{aligned} nc_n &= na_n + \sum_{k=1}^{\min(p,n-1)} (n-k)c_{(n-k)}a_k \\ \Rightarrow c_n &= a_n + \frac{1}{n} \sum_{k=1}^{\min(p,n-1)} (n-k)c_{(n-k)}a_k \end{aligned} \tag{2.39}$$

En ce qui concerne le nombre de coefficients cepstraux, il a été démontré qu'un nombre égal à trois fois l'ordre de prédiction p, est suffisant pour fournir une bonne représentation du spectre vocal.

2.4 Conclusion

Dans ce chapitre nous avons présenté quelques outils pour le traitement du signal vocal, qu'on appelle aussi analyse court-terme, en référence à l'utilisation des segments de parole de courte durée durant laquelle le signal est quasi-stationnaire pour pouvoir utiliser ces outils.

La parole est un signal redondant, ce qu'il lui confère une meilleure résistance au bruit, cependant les informations qu'il véhicule ne sont pas toutes pertinentes pour la reconnaissance de la parole, ainsi en pratique dans le but de réduire le nombre de données à traiter, le signal est représenté par un ensemble limité de paramètres, c'est la paramétrisation du signal.

Pour l'extraction des paramètres deux principales approches existent, l'analyse par prédiction linéaire, dont l'objectif est de prévoir un échantillon à partir des échantillons qui le précèdent, et l'analyse homomorphique dont l'idée est de séparer par déconvolution la contribution de la source de celle du conduit vocal, ce qui a pour effet un meilleur lissage du spectre. Selon l'application on opte pour l'un ou l'autre des types de paramètres. Dans les applications de reconnaissance vocale c'est la deuxième approche qui a donné les meilleurs résultats.

Dans le prochain chapitre, on décrira brièvement les méthodes de reconnaissance vocale, dont celle utilisée dans ce travail, soit la quantification vectorielle, quelques algorithmes de classification y seront aussi présentés.

CHAPITRE 3

LA RECONNAISSANCE DE LA PAROLE

3.1 Introduction

Avant d'aborder le problème de la reconnaissance, il est intéressant de mentionner les caractéristiques du signal vocal qui rendent cette tâche plus ou moins compliquée, ces caractéristiques sont :

- la redondance : ce qui nous oblige à extraire du signal les informations pertinentes pour réduire le traitement.

- la variabilité : on parle de la variabilité interlocuteur, liée aux caractéristiques anatomiques et aux habitudes linguistiques, et de la variabilité intra locuteur tel que le débit d'élocution, l'état émotionnel ...etc.

- la continuité : ainsi lorsqu'on écoute une personne parler, on perçoit une suite de mots alors que l'analyse du signal vocal ne permet de déceler aucun séparateur.

- la coarticulation : elle est provoquée par le fait que lors de la prononciation d'un phonème, l'appareil articulatoire se prépare pour la production du phonème suivant.

Dans ce chapitre nous décrirons les principales méthodes de reconnaissance, la métrique utilisée et les différentes mesures de distance nécessaires à la phase de décision. Nous

verrons aussi quelques notions de base d'apprentissage avec la présentation de quelques algorithmes de classification automatique.

3.2 Les distances dans l'espace acoustique

La reconnaissance de la parole est effectuée normalement au niveau acoustique. Le spectre du mot à reconnaître est comparé à ceux d'un ensemble de mots appelés mots de référence. Il est pertinent de se demander comment mesurer le degré de similarité entre une occurrence et une autre lors d'un processus de décision. En d'autres termes il faut établir une distance ou une mesure de dissemblance entre ces deux occurrences. Cependant il faut s'assurer de réduire au minimum la sensibilité de cette distance aux fluctuations des débits d'élocution.

3.2.1 La mesure de distorsion

Considérons un ensemble quelconque E de points. Nous dirons que E est un espace métrique réel s'il existe une fonction appelée distance, notée :

$d : E \times E \rightarrow R,$

Vérifiant les quatre propriétés suivantes :

- séparabilité : $\forall (a,b) \in E^2, \quad a \neq b \Rightarrow d(a,b) > 0,$
- réflexivité : $\forall a \in E, \ d(a,a) = 0,$
- symétrie : $\forall (a,b) \in E^2, \quad d(b,a) = d(b,a),$
- inégalité triangulaire : $\forall (a,b,c) \in E^3, \quad d(a,b) \leq d(a,c) + d(c,b).$

En parole ces conditions ne sont pas toutes satisfaites (c'est le cas par exemple de la symétrie), on parle plutôt de la mesure de dissemblance ou de mesure de distorsion.

Les distances utilisées pour comparer deux occurrences sont étroitement liées aux types de paramètres utilisés. Une définition particulière de la distance entre deux spectres doit être [14] :

- significative sur le plan acoustique.
- formalisable d'une façon efficiente sur le plan mathématique.
- définie dans un espace de paramètres judicieusement choisi.

3.2.2 La distance Euclidienne

Pour l'analyse spectrale ou cepstrale, le choix se porte généralement sur les distances associées à la norme dite de Hölder [14], pour des vecteurs à K composantes, cette norme est :

$$d_p(a,b) = \left[\sum_{k=1}^{K} |a_k - b_k|^p \right]^{\frac{1}{p}}, p = 1 \text{ à } \infty \qquad (3.1)$$

où a est un vecteur de paramètres du mot de référence, et b un vecteur de paramètres du mot à reconnaître.

Pour $p=2$, cette distance est connue sous le nom de la distance euclidienne, qu'on peut utiliser dans le domaine spectral ou cepstral.

3.2.3 La distance d'Itakura

La distance d'Itakura est utilisée pour comparer deux vecteurs a et b de $(p+1)$ coefficients de prédiction linéaire chacun, p étant l'ordre de la prédiction. Cette distance est définie par [14]:

$$d_I = \ln\left[\frac{aRa^T}{bRb^T}\right] \tag{3.2}$$

où a est le vecteur de référence et R est la matrice $(p+1)\times(p+1)$ des coefficients d'autocorrelation évalués sur le segment de signal correspondant à b, et a^T est le vecteur colonne transposé de a. Pour le numérateur il s'agit de l'énergie résiduelle on peut l'évaluer par :

$$aRa^T = r(0)r_a(0) + 2\sum_{n=1}^{p} r(k)r_a(k) \tag{3.3}$$

où les $r(k)$ sont les coefficients d'autocorrelation sur le segment du signal correspondant à b, et $r_a(k)$ sont les coefficients d'autocorrélation correspondant au vecteur a.

Une autre variante de la distance d'Itakura est la distance appelée rapport de vraisemblance (Likelihood Ratio) dont la forme est :

$$d_{LR} = \frac{aRa^T}{bRb^T} - 1 \tag{3.4}$$

3.2.4 La distance cepstrale

Soit deux vecteurs C_t et C_r qui contiennent respectivement les coefficients cepstraux d'un segment du mot de référence et d'un segment du mot à reconnaître. La distance cepstrale, d_{CEP} est la distance euclidienne entre ces deux vecteurs, elle est définie par :

$$d_{CEP} = \left[c_t(0) - c_r(0)\right]^2 + 2\sum_{k=1}^{\infty}\left[c_t(k) - c_r(k)\right]^2 \tag{3.5}$$

En pratique, on ne prend pas en considération le premier terme de la distance.

La distance cepstrale est généralement tronquée, elle est évaluée le long d'un nombre fini de coefficients typiquement 10 à 30 [3]. Cependant ce nombre ne doit pas être inférieur à l'ordre de prédiction p si les spectres sont issus d'un modèle tous pôles d'ordre p.

La distance cepstrale sera alors :

$$d_{CEP} = \sum_{k=1}^{L}\left[c_t(k) - c_r(k)\right]^2 \tag{3.6}$$

Avec L est le nombre de coefficients le long desquelles la distance est calculée.

Remarque :

Les coefficients cepstraux dans l'équation (3.6) peuvent être des coefficients MFCCs ou des coefficients LPCCs.

3.2.5 La distance cepstrale pondérée

Une autre variété de la distance cepstrale est la distance cepstrale pondérée, il a été démontré [23] que cette pondération permet d'améliorer considérablement le taux de reconnaissance, l'idée est d'ajouter une pondération $w(k)$ dans la formule (3.6), la forme générale de la distance sera alors :

$$d_{WCEP} = \sum_{k=1}^{L} \left(w(k)\left(c_t(k) - c_r(k)\right)\right)^2 \tag{3.7}$$

Plusieurs méthodes de pondération ont été introduites par des chercheurs, dans ce qui suit nous présenterons quelques unes.

3.2.5.1 La pondération par la quéfrence k

Cette pondération a été utilisée par Hanson et Wakita [24], elle permet de normaliser la contribution de chaque coefficient cepstral, elle est définie par :

$$\begin{aligned} w(k) &= k \\ d_{WCEP} &= \sum_{k=1}^{L} k^2 \left[c_t(k) - c_r\left(k\right)\right]^2 \end{aligned} \tag{3.8}$$

3.2.5.2 La pondération par la variance

Cette méthode de pondération a été utilisée par Tokhura [23] dans "la reconnaissance multi-locuteurs des chiffres isolés en utilisant les distances pondérées", son origine est la distance suivante :

$$d_{MCEP} = (c_t - c_r)V^{-1}(c_t - c_r)^T \tag{3.9}$$

Cette distance (3.9) connue sous le nom de distance de Mahanalobis [25], s'applique difficilement en reconnaissance de la parole, entre autre à cause de l'utilisation de la matrice de la covariance V, alors on l'a remplacée par la distance (3.10) où juste les éléments de la diagonale de la matrice de la covariance sont utilisés au lieu de la matrice V en entier :

$$\begin{aligned} w(k) &= \frac{1}{V_{kk}} \\ d_{WCEP} &= \sum_{k=1}^{L}\left(\frac{1}{V_{KK}}\right)^2 \left[c_t(k) - c_r(k)\right]^2 \end{aligned} \tag{3.10}$$

Où V_{kk} est le $k^{ème}$ élément de la diagonale de la matrice de la covariance.

3.2.5.3 La pondération par liftrage passe-bande (bandpass liftering)

Cette pondération a été introduite par Juang et al [26], son rôle est de réduire la variabilité spectrale en raison de la sensibilité du banc de filtres à tout changement de la fréquence du fondamental dans le cas des coefficients MFCCs ou les effets indésirables

sur les spectres causés par la représentation par un modèle tous pôles, dans le cas des coefficients LPCCs [26], trois types de fenêtres ont été avancées comme solution :

type1

$$w_1(k) = \begin{cases} 1, & k = 1,\ 2, \cdots, L \\ 0, & \text{ailleurs.} \end{cases} \tag{3.11}$$

type2

$$w_2(k) = \begin{cases} 1 + h \cdot (k-1)/(L-1), & k = 1,\ 2, \cdots, L \\ 0, & \text{ailleurs.} \end{cases} \tag{3.12}$$

type3

$$w_3(k) = \begin{cases} 1 + h \cdot \sin(k\pi / L), & k = 1,\ 2, \cdots, L \\ 0, & \text{ailleurs.} \end{cases} \tag{3.13}$$

Juang et al ont retenu la dernière forme (3.13) comme meilleure choix pour les applications de reconnaissance et ont suggéré comme valeur pour la variable h la valeur $\frac{L}{2}$, L étant le nombre des coefficients cepstraux.

3.3 Les méthodes utilisées pour la reconnaissance de la parole

D'une manière générale en parole on peut classer les méthodes de reconnaissance en deux [14] :

- la méthode globale : dans cette approche l'unité de base, est le mot considéré comme entité globale non décomposable. Elle a pour avantage d'éviter les effets

de coarticulation, dans ce type de méthode on compare globalement le mot ou la phrase, aux différentes références stockées dans un dictionnaire. Dans cette catégorie on trouve principalement la méthode DTW (Dynamic Time Warping), et la méthode HMM (Hidden Markov Models) et la méthode de la quantification vectorielle (VQ).

- la méthode analytique : dans cette approche est exploitée la structure linguistique du mot. En effet dans ce cas les unités de base sont les phonèmes, les syllabes, et les demi-syllabes, la reconnaissance dans cette méthode passe par la segmentation du signal de la parole en unités de décision, puis par l'identification de ces unités en utilisant des méthodes de reconnaissance des formes.

La première méthode est utilisée pour la reconnaissance des mots isolés, des mots enchaînés, alors que la deuxième méthode est beaucoup mieux adaptée pour les systèmes à grands vocabulaires et pour la parole continue.

3.3.1 La programmation dynamique

Lorsqu'un locuteur, même entraîné, répète plusieurs fois une phrase ou un mot, il ne peut éviter les variations du rythme de prononciation ou de la vitesse d'élocution, ces variations entraînent des transformations non linéaires dans le temps du signal acoustique, ce qui fait qu'on ne pourra comparer directement point à point (matching) deux formes acoustiques sans correction temporelle au préalable.

Pour établir une meilleure correspondance entre les axes temporels des deux mots, en même temps que leurs comparaisons, on utilise une technique appelée technique d'alignement temporel dynamique ou DTW [14]. C'est une technique basée sur la programmation dynamique qui consiste à trouver la trajectoire optimale entre le mot de

référence et le mot inconnu (voir Figure 20) en tenant compte de certains contraintes, appelés contraintes locales qui régissent le passage d'un point à un autre le long du chemin.

Supposons qu'on a un mot de référence R et un mot à tester T représentés par les vecteurs acoustiques suivants :

$$T:[T(i)\,;\, i=1,2,\cdots,I]$$
$$R:[R(j);\, j=1,2,\cdots,J]$$

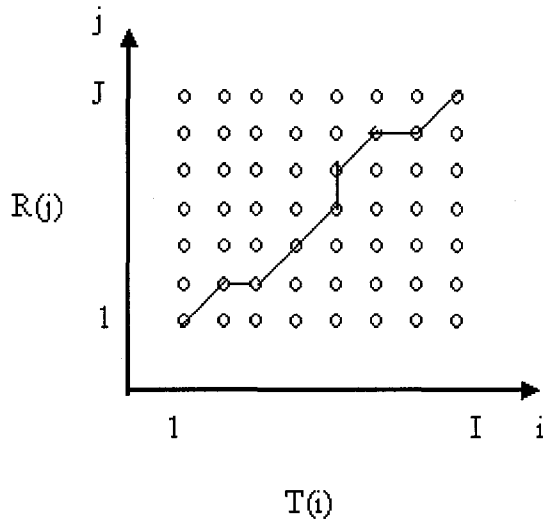

Figure 20 Alignement temporel entre les allocutions R et T

Pour ces deux vecteurs on va définir une distance locale *d*, qui représente la distance entre *T(i)*, un vecteur de *T*, et *R(i)* un vecteur de *R*, ensuite on définit une distance globale *D*, comme somme des distances locales, à lesquelles on peut ajouter une pondération, cependant cette distance doit être peu sensible aux distorsions temporelles,

pour cela on va l'évaluer le long d'un chemin optimal $w:[i(k), j(k)], k = 1, 2, \cdots K,$ en tenant compte des contraintes suivantes :

- faire coïncider les extrémités, le chemin doit alors commencer en [*T(1),R(1)*]et finir en [*T(I),R(J)*].
- progresser d'une manière monotone le long du chemin, $w_{k-1} < w_k$
- respecter la continuité du chemin

À cela il faut à jouter les contraintes dites locales tel que [14]:

- Contraintes du type A :

$$\widetilde{D}(i,j) = d(i,j) + \min\left[\widetilde{D}(i-1,j), \widetilde{D}(i-1,j-1), \widetilde{D}(i-1,j-2)\right]$$

- Contraintes du type B

$$\widetilde{D}(i,j) = d(i,j) + \min\begin{cases} \widetilde{D}(i-2,j-1) + 2\cdot d(i-1,j) \\ \widetilde{D}(i-1,j-1) + d(i,j) \\ \widetilde{D}(i-1,j-2) + 2\cdot d(i,j-1) \end{cases}$$

- Contraintes du type C

$$\widetilde{D}(i,j) = d(i,j) + \min\begin{cases} \widetilde{D}(i-1,j) \\ \widetilde{D}(i-1,j-1) + d(i,j) \\ \widetilde{D}(j,j-1) \end{cases}$$

où $d(i,j)$ la distance locale, et $\widetilde{D}$ la distance globale, ou l'accumulation des distances locales.

La distance minimale recherchée entre le mot de référence R et le mot inconnu T est :

$$\tilde{D}(T,R) = d(I,J)/N(g) . \qquad (3.14)$$

où $N(g)$ est un facteur de normalisation, qui pour les contraintes de type B ou C, vaut $I+J$.

3.3.2 Les modèles MMCs

Une autre approche différente consiste à remplacer l'ensemble des références acoustiques représentants les différentes prononciations par un modèle statistique. L'identification d'un mot revient alors à rechercher le modèle qui aurait pu produire ce mot avec la probabilité la plus élevée, c'est le principe des modèles MMCs.

Les modèles de Markov cachés (MMC) [3]sont des modèles doublement stochastiques dont la première composante est un processus stochastique non observable d'où le nom caché, mais qui peut l'être par le biais d'un second processus stochastique.

Un modèle MMC est aussi une machine à états fini (voir Figure 21), dont le nombre d'états est choisi d'une façon empirique et dans lesquelles les transitions et les sorties sont régies par des lois de probabilité, on distingue les MMCs discrets pour lesquels la distribution est discrète ou obtenue par quantification, et les MMCs continus où cette distribution est continue, généralement approximée par une mixture de gaussienne.

Les paramètres qui caractérisent un Modèle HMM sont :

- le nombre d'états N du modèle, $Q = (q_1, q_2, \ldots, q_N)$.

- le nombre M d'observations par état qu'on dénote $V = \{v_1, v_2, \ldots, v_M\}$.

- une matrice $A = \{a_{ij}\}$ qui permet de définir les probabilités de transition d'un état q_i vers un autre état q_j, ou d'un état vers lui même.

$$a_{ij} = p[q_{t+1} = j \mid q_t = i], \quad 1 \le i, j \le N.$$

Autrement dit, a_{ij} est la probabilité d'être à l'état j à t+1, sachant qu'on était à l'état i à t. Dans une chaîne de Markov d'ordre R la probabilité d'occurrence d'un symbole ne dépend que de la nature des R symboles qui le précèdent, or en RAP on fait souvent appelle à des chaînes de Markov d'ordre 1, alors la probabilité de passer à l'état suivant dépend seulement de l'état courant.

- une matrice $B = \{b_j(k)\}$ qui contient les probabilités d'émission des observations dans chaque état j

$$b_j(k) = P[o_t = v_k \mid q_t = j], \quad 1 \le k \le M \text{ et } j = 1,2,\ldots,N.$$

- une matrice $\pi = \{\pi_i\}$ qui donne la distribution de départ des états, c'est à dire la probabilité d'être à l'état j à l'instant initial t=1. Cet état possède la particularité de ne pouvoir émettre d'observations. $\pi_i = P[q_1 = i], \quad 1 \le i \le N.$

En conclusion un modèle MMC est caractérisé par un ensemble de mesures de probabilités qu'on regroupe par la notation $\lambda = (A, B, \pi)$

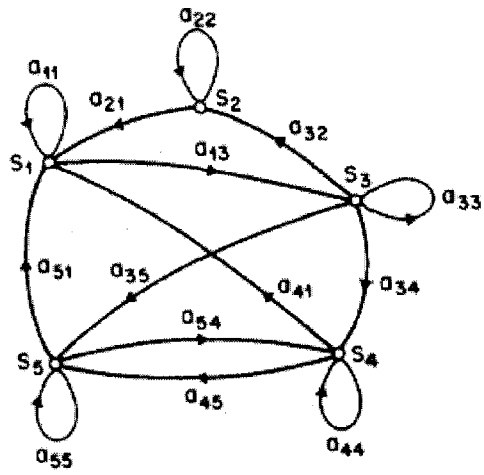

Figure 21 Modèle de Markov à 5 états [3]

3.3.3 La quantification vectorielle

La quantification vectorielle est une opération qui consiste en une partition de l'espace des vecteurs acoustiques en classes (voir Figure 22). Chaque classe est représentée par un vecteur particulier appelé centroïde ou noyau. Ce point représente la distance minimale intra-classe.

Le dictionnaire de prototypes appelé aussi code-book qui contient l'ensemble de tous les centroïdes, est obtenu par apprentissage ou entraînement sur un grand ensemble suffisamment représentatif de données. Pour sa construction on fait appel à la méthode de Llyod-Max (ou à une de ses variantes) très connue sous le nom de l'algorithme de k-means.

La quantification vectorielle tel que présentée ci-dessus est la quantification vectorielle statistique. En effet il existe une autre forme de quantification appelée la quantification

vectorielle algébrique dont le principe est d'utiliser un code-book qui présente une certaine structure mathématique. Cependant dans les applications RAP on lui préfère la première qui représente la plus simple interprétation de la quantification vectorielle.

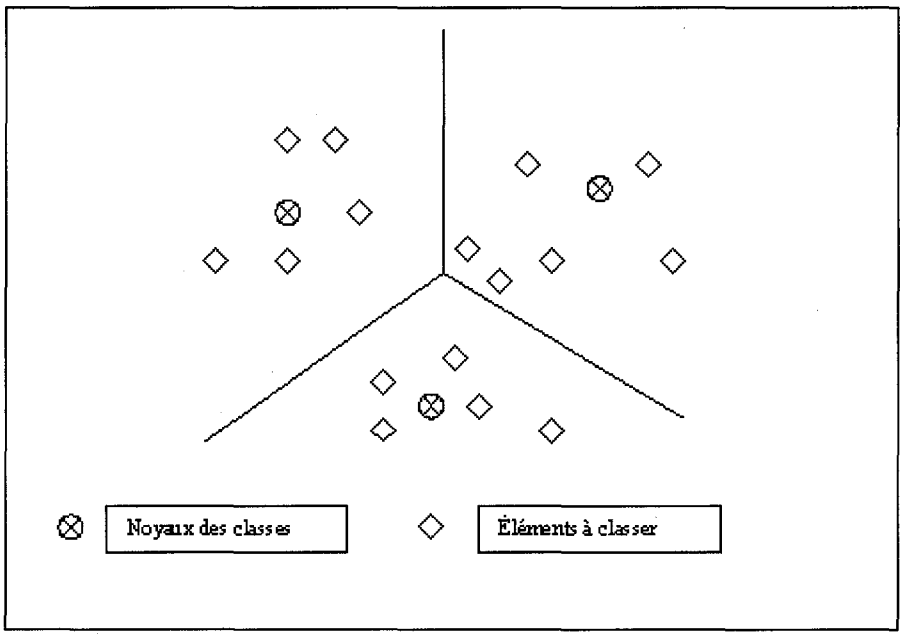

Figure 22 Partition d'un espace en trois classes

Dans le cadre de ce projet c'est la quantification vectorielle qui a été retenue pour la reconnaissance car elle ne nécessite que peu de calculs comparativement aux autres méthodes. En effet dans ce cas, chaque classe n'est représentée que par un seul point (noyau), ainsi lors de la décision le spectre du mot à reconnaître n'est comparé qu'à celui du représentant de la classe au lieu d'être comparer avec les spectres de la classe tout entière.

3.4 L'apprentissage

L'une des étapes les plus importantes dans le processus de reconnaissance est l'étape de la construction du dictionnaire de référence, appelée aussi l'étape d'apprentissage. En effet cette étape est d'une telle importance pour un système de reconnaissance, que même l'utilisation des plus puissants algorithmes lors de l'étape de décision ne peut compenser sa faiblesse éventuelle. La performance de tout le système dépend du soin apporté à cette étape.

Les méthodes utilisées en apprentissage diffèrent selon le type du système. Un dictionnaire de référence destiné à être utilisé dans un système RAP mono-locuteur ne se crée pas de la même manière qu'un autre destiné à un système multi-locuteurs, pour ce dernier on utilise des techniques de classification automatique appelée aussi clustering.

3.4.1 L'apprentissage mono locuteur

Pour la reconnaissance mono-locuteur généralement on distingue deux types d'apprentissage [2] :

3.4.1.1 Apprentissage simple

Pour un locuteur entraîné capable de garder le même rythme d'élocution, et dans des conditions idéales d'enregistrement, tel que l'absence de bruits, un procédé simple d'apprentissage consiste à utiliser chaque mot prononcé durant la session d'entraînement comme mot de référence [2].

3.4.1.2 Apprentissage robuste

Pour des mots qui ne sont pas acoustiquement voisins, la méthode précédente n'est pas suffisante pour espérer obtenir des résultats de reconnaissance consistants, alors une amélioration du procédé est suggérée. Dans ce cas le locuteur est invité à prononcer chaque mot du vocabulaire plusieurs fois et le dictionnaire de référence est conçu à partir de ces énoncés. C'est le cas par exemple de l'apprentissage appelé apprentissage robuste [2] où le locuteur prononce chaque mot plusieurs fois jusqu'à l'obtention de deux occurrences du mot suffisamment proches, le mot de référence est leur moyenne après alignement temporel effectué par programmation dynamique, ce moyennage est effectué dans le domaine spectral.

3.4.2 L'apprentissage multi-locuteurs

Le regroupement ou le "clustering " en anglais consiste à regrouper L occurrences prisent à partir de plusieurs locuteurs, en *N* classes ou cluster, Une bonne méthode de regroupement permet de garantir une grande similarité à l'intérieur de la même classe et une faible similarité avec les autres classes.

Pour les systèmes indépendants de locuteur, des algorithmes de classification sont utilisés pour la création du dictionnaire des mots de référence. Généralement on retient pour chaque mot plusieurs représentants de huit à douze [27], ce qui permet d'avoir une bonne image de l'ensemble des prononciations d'un même mot, et constitue un choix raisonnable pour tenir compte de la variabilité inter-locuteurs. Le dictionnaire peut être ensuite stocké en mémoire non-volatile dans le système de reconnaissance.
Plusieurs méthodes de classification ont été développées par les chercheurs, tel que les méthodes dites semi-automatique Ces dernières nécessitent la supervision d'un expert qui guide le processus de classification. Dans cette catégorie on retrouve entre autres des

algorithmes comme ISODATA, et chainmap [28]. Par la suite on a assisté à l'émergence d'une autre catégorie de classification celle-ci est totalement automatique, tel que l'algorithme UWA (Unsupervised Without Averaging) [29], ou MKM (Modified K-Means) [30].

3.4.2.1 Quelques algorithmes de classification

On distingue deux approches de classification automatique [2]:

- les algorithmes hiérarchiques: ils permettent de créer une décomposition hiérarchique de l'ensemble d'apprentissage selon certains critères, le résultat est un dendogramme (ou arbre hiérarchique) qu'il faudra ensuite interpréter.

- les algorithmes non hiérarchiques (ou de partitionnement): l'ensemble d'apprentissage est décomposer en sous-groupes (classes) selon certain critère, ces classes se distinguent les uns des autres soit par des frontières soit par le centre de gravité qui caractérise chaque classe appelée aussi centroïde.

La première approche étant peu utilisée en RAP, alors on va se contenter de citer quelques exemples de classification automatique non hiérarchique.

3.4.2.1.1 L'algorithme K-means (Llyod-Max)

Cet algorithme utilisé aussi dans la quantification vectorielle consiste à définir d'une manière itérative M classes à partir de L vecteurs de paramètres qui constitue l'ensemble d'apprentissage [3]. Chaque classe est concentrée autour d'un point appelé noyau, c'est ce point qui caractérisera cette classe et qui constituera avec les noyaux des autres

classes le dictionnaire de référence ou dictionnaire de prototypes qui sera utilisé dans la suite du processus.

L'algorithme peut être décrit de la manière suivante :

1. Initialisation : le nombre de classes M étant choisit à priori, alors on procède à leurs initialisations d'une manière aléatoire avec n_i noyaux (mots), $1 \le i \le M$
2. affectation : affecter chaque élément x_k ,$1 \le k \le L$ de l'ensemble d'apprentissage, à chacune des classes en utilisant la loi du k plus proche voisin (avec $k = 1$), qui consiste à choisir, pour chaque élément x_k le noyau le plus proche :

$$x_k \in C_i,\ ssi\ \ d(x_k, n_i) \le d(x_k, n_j),\ \text{avec}\ \ j \ne i\ \ et\ \ 1 \le j \le M$$

d étant la mesure de distorsion, en générale c'est la distance euclidienne.

3. mise à jour des centres de gravité : calculer les nouveaux centres de gravités n_i de chaque classe C_i, en tenant compte des éventuels nouveaux ajouts à cette classe, le but de cette étape est de minimiser la distorsion au sein de chaque classe, n_i est défini par:

$$ni = \frac{1}{N}\sum_{n=1}^{N} x_n\ ,N\ étant\ le\ nombre\ d'éléments\ de\ la\ classe\ i.$$

Cette façon de calculer n_i ne s'applique que si on utilise la distance euclidienne comme mesure de distorsion.

4. test de convergence : on va voir s'il n'y'a plus de nouvelles affectations qui s'ajoutent aux classes, et leurs contenus sont restés stables et inchangés entre deux itérations consécutives. Si c'est le cas alors c'est la fin de l'algorithme et les

n_i obtenus représente les mots de référence, dans le cas contraire l'algorithme continue en bouclant à l'étape 2.

Remarque

Il a été démontré que cet algorithme converge vers une situation localement optimum, qui dépend des valeurs initiales des centroïdes [31]. En effet un choix judicieux des valeurs initiales permet de réduire le temps de calcul, et aide l'algorithme à converger vers des bons résultats, d'où l'importance de l'initialisation.

La méthode classique consiste à prendre les valeurs initiales d'une manière aléatoire ce choix conduit à des résultats faibles en terme de représentativité de toutes les classes. Il peut même conduire à des classes vides, situation problématique qu'il faut éviter. Alors une solution optimale peut être approximée en appliquant l'algorithme plusieurs fois avec des valeurs initiales différentes et on retient la solution qui fournit les meilleurs résultats.

3.4.2.1.2 L'algorithme LBG (Linde-Buzo-Gray)

Une variante très utilisée de l'algorithme de k-means est l'algorithme de Linde, Buzo et Gray [32] appelé aussi Llyod généralisé. C'est un algorithme qui propose une solution pour le problème du choix du dictionnaire du départ par la réalisation d'une initialisation au cours du processus, en divisant (éclatant) d'une manière itérative l'ensemble d'apprentissage en $2,4,...,2^P$ classes, et en calculant le centre de gravité de chacune d'elles, d'où le nom de l'algorithme par éclatements binaires. Son principe se résume comme suit :

1. initialisation : posons $d = 0$, le nombre de classes initiales est égal à $2^d = 1$, on calcule le centre de gravité de cette classe, évidemment à ce niveau cette classe

est tout l'ensemble d'apprentissage, et par conséquent le dictionnaire de référence ne contient qu'un seule élément.

2. division (splitting) : diviser les 2^d classes en deux, ceci par exemple peut se faire par perturbation légère des centroïdes de la manière suivante :

$$n_{nouveau} = n_{ancien}(1 \pm \varepsilon)$$

3. Convergence (application de k-means) : à ce stade on a 2^{d+1} noyaux, ils vont servir comme dictionnaire initiale pour l'algorithme k-means qui leurs est appliqué, après convergence on obtient 2^{d+1} nouveaux centroïdes (noyaux) qui seront utilisés dans la suite du processus.

4. test d'arrêt : la valeur de d est incrémenté de 1, $d = d + 1$, ensuite si le nombre de classes désiré est atteint, alors c'est la fin de l'algorithme, sinon reprendre à partir de l'étape 2.

En terminant notons que d'autres méthodes ont été développées [30, 33, 34] pour l'initialisation des algorithmes et ceci dans le but de rendre ces algorithmes plus performants.

3.5 Conclusion

En reconnaissance on distingue Principalement deux grandes méthodes : la méthode globale et la méthode analytique, dans la première le mot (ou la phrase) à reconnaître est comparée dans son ensemble avec des mots (ou des phrases), sans décomposition en

entités plus élémentaire, alors que pour la deuxième méthode la reconnaissance est précédée d'une segmentation en unités de base tel que phonèmes, syllabes, ...etc.

Pour tout système RAP, deux phases sont nécessaires : une phase d'apprentissage, et une phase de reconnaissance, le but de la première est la construction d'un dictionnaire de référence qui servira de base dans la phase de reconnaissance. Dans certains cas des algorithmes de classification automatique sont utilisés pour réaliser l'apprentissage.
Au niveau acoustique la reconnaissance d'un mot se fait par une succession de comparaisons du spectre du mot à reconnaître avec ceux des mots de référence, pour ce faire une distance ou une mesure de dissemblance est définie d'une façon judicieuse entre deux mots. Dans le but d'améliorer les performances de reconnaissance, une pondération adéquate est ajoutée à ces distances.

Dans le chapitre suivant sera décrit notre système de reconnaissance vocal, ainsi que son implémentation à l'aide du système de développement du Texas Instrument.

CHAPITRE 4

LA RÉALISATION DU SYSTÈME DE RECONNAISSANCE

4.1 Introduction

Dans les chapitres précédents nous avons présenté les aspects généraux de la parole (production et audition), les différents outils nécessaires pour son traitement et sa paramétrisation, ainsi qu'un aperçu sur les principales approches de reconnaissance qu'on retrouve dans les différents systèmes RAP. Dans ce chapitre nous décrirons la réalisation d'un système de reconnaissance au moyen d'un processeur dédié au traitement numérique des signaux, le DSP TMS320C6711.

La réalisation est effectuée en utilisant un système de développement composé essentiellement de la carte DSK6711, et le logiciel Code Composer Studio. Ce logiciel supporte toutes les librairies du langage C. Tous les algorithmes utilisés seront programmés en langage C.

Dans ce chapitre nous commencerons par présenter les différents éléments qui composent notre système ainsi que les algorithmes utilisés pour les réaliser, suivra la présentation du système de développement utilisé, enfin nous présenterons quelques résultats de simulation.

4.2 Description du système de reconnaissance

Comme illustré par le schéma bloc (Figure 23), le processus conduisant à la reconnaissance commence par l'extraction de la parole du silence. La segmentation en

syllabes suivra (seulement pour les chiffres connectés), par la suite les paramètres LPCCs seront extraits pour chaque segment. Enfin le processus de décision basé lui sur la règle du plus proche voisin *Kpp* sera appliqué.

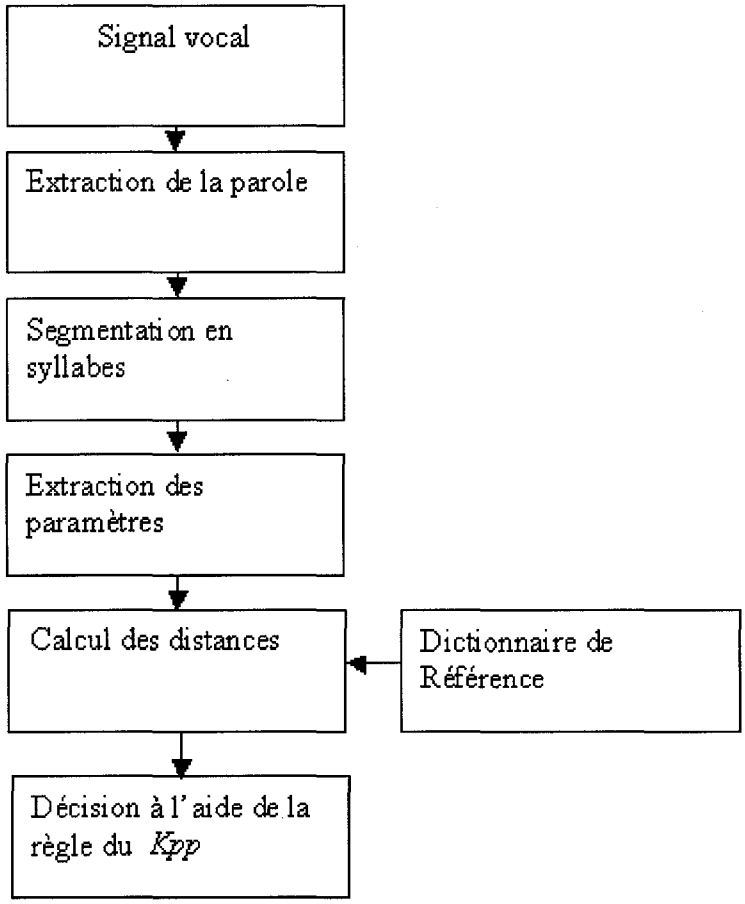

Figure 23 Schéma bloc du système RAP

4.2.1 L'extraction de la parole

Un des problèmes à résoudre en traitement de la parole, est la séparation de la parole du silence appelé aussi la détection du début et de la fin de la parole. En effet la solution de ce problème conduira à une réduction considérable du traitement, aussi bien en termes

de temps de calcul que de mémoire utilisée. Il faut noter également que l'imprécision dans la détermination de ces deux points (début et fin de la parole), est à l'origine de plusieurs cas d'erreur de reconnaissance.

Dans le cas d'un environnement avec un rapport signal sur bruit élevé, l'énergie des plus faibles niveaux de sons (faibles fricatives, segments de parole de faibles niveaux,... etc.), dépasse l'énergie du bruit de fond, de telle manière qu'une simple comparaison des niveaux d'énergies permet de différencier la parole du bruit (les bruits de mesures). Malheureusement ce cas idéal d'environnement d'enregistrement n'est pas toujours possible en RAP, et l'on ne le retrouve que rarement en pratique.

Ce problème appelé aussi "VAD" (Voice Activity Detection) [35] a été approché de plusieurs façon, comme par exemple la méthode qu'on retrouve dans la littérature sous le nom de la classification SUV, (Silence/Unvoiced/Voiced) [36, 37] . Cette méthode suggère l'étiquetage de la parole en trois catégories silence, voisé ou non voisé. En utilisant le taux de passage par zéro et l'énergie court terme. D'une manière générale pour un signal sans aucun bruit, on peut résumer les critères de classification de cette méthode par le tableau I, ces critères ont été évalués sur des segments de parole de 10 ms.

Tableau I

La classification SUV (Silence/Unvoiced/Voiced)

Taux de passage par zéro	Énergie court-terme	Étiquette
inférieur à 12	élevée	voisé
supèrieur à 50	faible	non voisé
0	0	silence

Une autre méthode plus performante, basée elle aussi sur deux mesures temporelles, l'énergie et le taux de passage par zéro, est la méthode proposée par Rabiner et Sambur [38], son principe général est illustré par la Figure 24. C'est la méthode que nous avons utilisé, car elle tient en compte un nombre de situations spéciales dans la séparation de la parole du silence tel que :

- les mots qui commencent ou finissent avec des phonèmes de faibles énergies (exemples les faibles fricatives).
- les mots qui finissent avec des plosives non-voisés.
- les mots qui finissent avec des nasales.
- locuteur finissant sa prononciation par un court souffle (bruit)

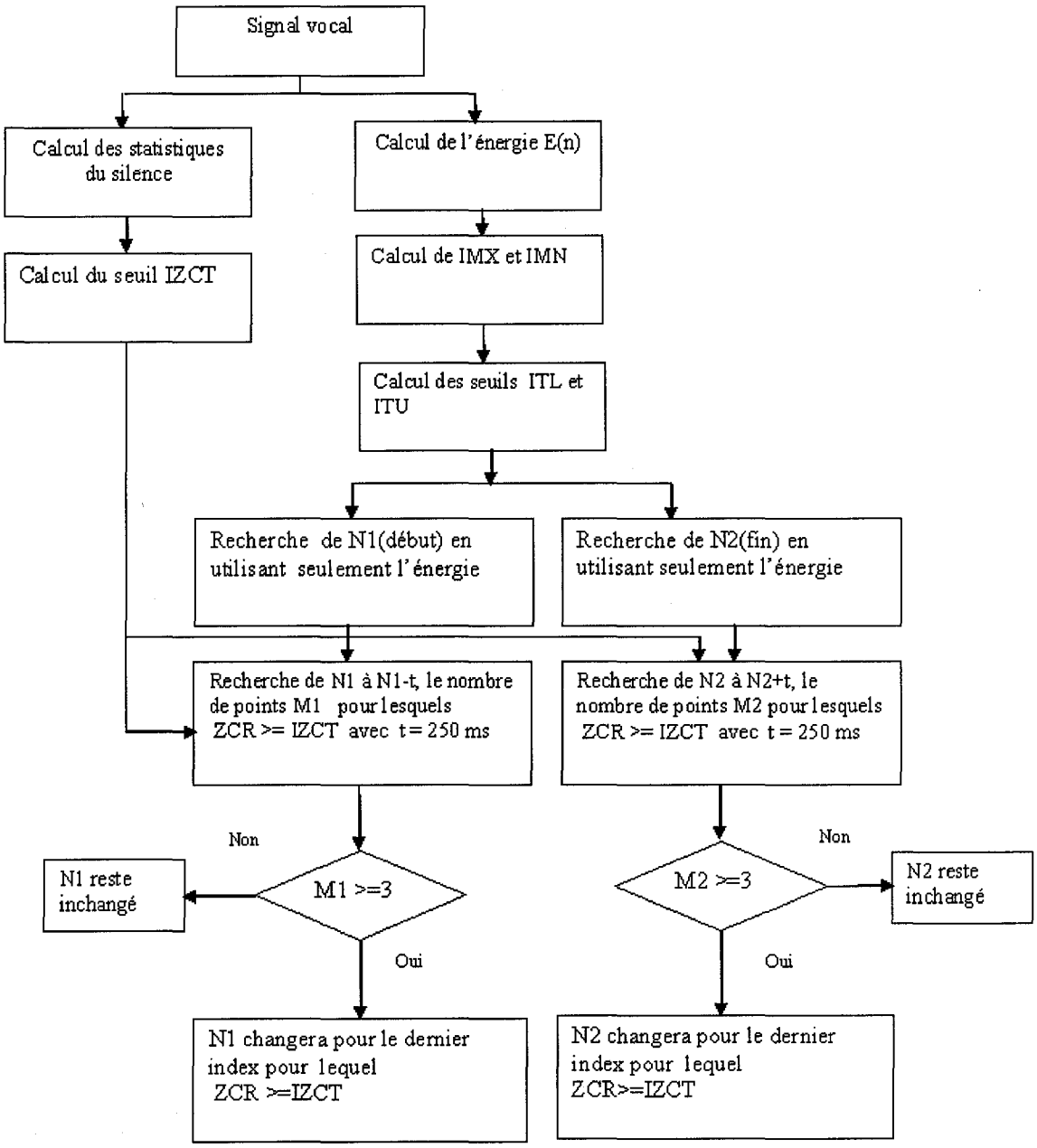

Figure 24 Organigramme général de la détection des points début et fin de la parole

4.2.1.1 Description de l'algorithme utilisé

On peut résumer cet algorithme par les étapes suivantes :

1. calcul de l'énergie et du taux de passage par zéro

Les deux paramètres, le taux de passage par zéro et l'énergie sont calculés chaque 10ms sur des fenêtres de 10ms. Pour des raisons de simplification de calcul, la formule utilisée pour la mesure de l'énergie est la suivante, évalué sur 10ms :

$$E(n) = \sum_{i=-40}^{40} |s(n+i)| \tag{4.1}$$

2. calcul des statistiques du silence

Dans cette étape, on émet l'hypothèse que pendant les premières 100 ms du signal il y'a absence totale de parole, alors on s'en sert pour calculer les statistiques du silence, qui vont servir pour la détermination du seuil du taux de passage par Zéro $IZCT$, et les seuils haut et bas de l'énergie ITU et ITL, ces paramètres sont :

$IF = 25 \text{ par } 10ms$
$I1 = 0.03(IMX - IMN) + IMN$
$I2 = 4IMN$
$ITL = \min(I1, I2)$
$ITU = 5ITL$
$IZCT = \min(IF, \overline{IZC} + 2\sigma_{IZC})$

Avec:

$IZCT$: Le seuil du taux de passage par zéro.

$\overline{IZC}$: La moyenne du taux de passage par zéro.

IMX : Le maximum de l'énergie.

IMN : La moyenne de l'énergie du silence.

ITL : Le seuil bas de l'énergie.

ITU : Le seuil haut de l'énergie.

3. détermination des points début et fin de la parole

Après la détermination des différents paramètres, on commence la recherche des points début et fin de la parole, cette partie de l'algorithme se fait en deux étapes :

1ere étape

Dans cette phase on travaille seulement avec l'énergie (voir Figure 25). Ainsi l'algorithme commence par la recherche du début de l'intervalle jusqu'à ce que le seuil inférieur de l'énergie (*ITL*) soit dépassé. Ce point alors est étiqueté le début de la parole, à moins que le niveau d'énergie chute en dessous de ce seuil (*ITL*) avant de dépasser le seuil supérieur (*ITU*). Si ceci se produit, un nouveau point sera obtenu en trouvant le premier point auquel l'énergie excède le seuil inférieur puis excède le seuil supérieur. De la même manière le point final est détecté.

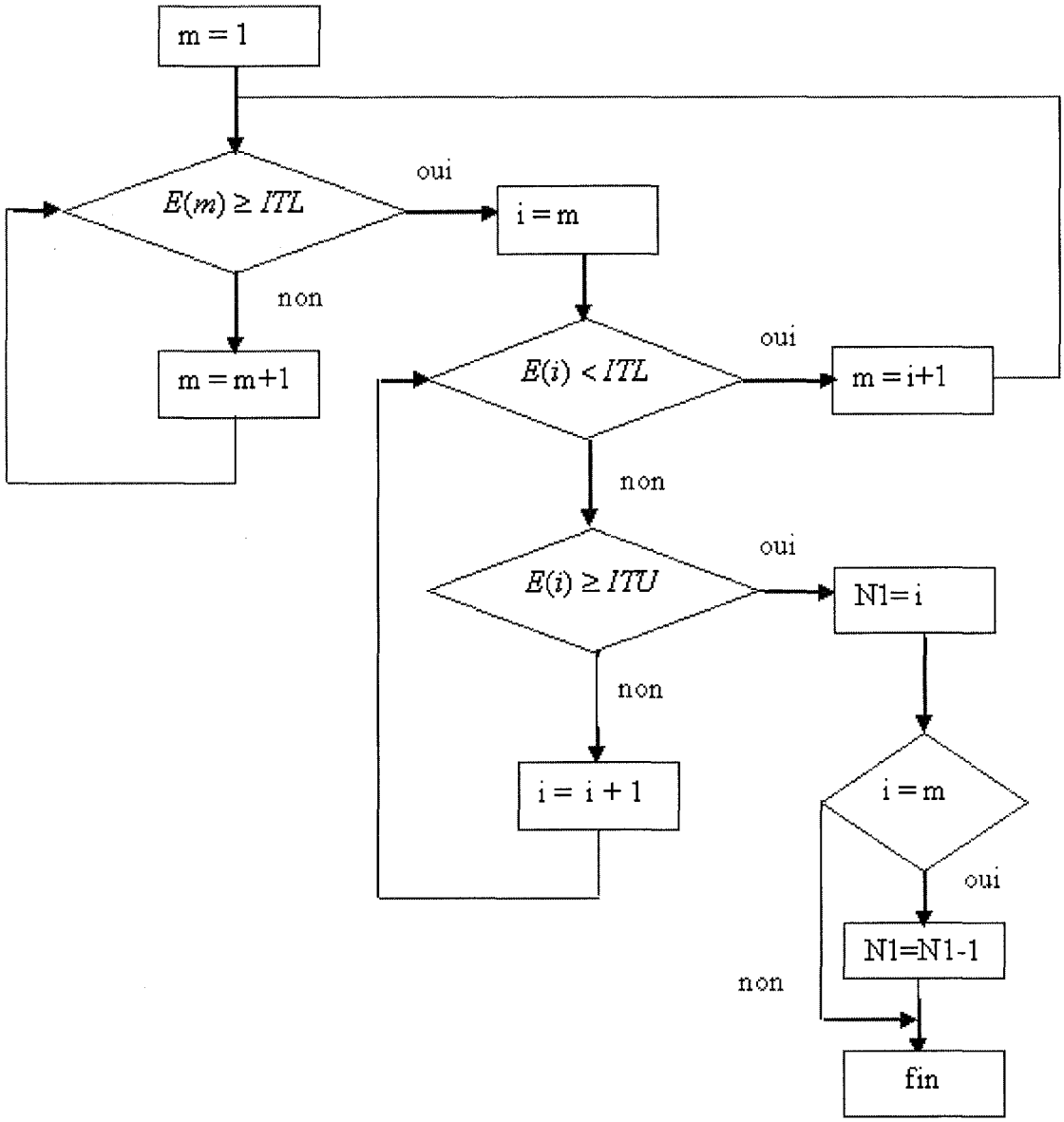

Figure 25 Recherche des points finals à l'aide de l'énergie

2ème étape

L'utilisation de l'énergie toute seule n'est pas toujours suffisante pour différencier la parole du silence, c'est le cas par exemple des fricatives (comme le /f/ dans le chiffre "four"). Ainsi on peut supposer qu'une partie de la parole puisse être en dehors de

l'intervalle trouvé en utilisant seulement les seuils d'énergie. Cet algorithme propose alors une vérification à l'aide du taux de passage par zéro pour confirmer l'intervalle trouvé où dans le cas contraire retrouver les nouveaux points début et fin de la parole. L'algorithme examine alors les 250 ms précédant le point du début et compte le nombre d'intervalles où le taux de passage par zéro excède le seuil (IZCT). Si ce nombre est supérieur ou égal à trois, le point du début est placé au premier point où le seuil a été dépassé, sinon, le point de début est gardé. De la même manière le point final est recherché en effectuant la vérification sur les 250ms suivant le point final trouvé dans la 1ère étape.

D'autres algorithmes existent dans la littérature pour la détection du début et la fin de la parole. Ces algorithmes [39, 40] suggèrent l'inclusion d'autres paramètres pour des résultats plus performants. Cependant les résultats obtenus avec cet algorithme qui ne fait appel qu'à l'énergie à court terme et le taux de passage par zéro dans le processus décisionnel, nous permettent de conclure que ce dernier constitue un bon choix pour une éventuelle implémentation sur DSP, à cause de son efficacité et de sa rapidité, ce qui concorde parfaitement avec notre objectif.

4.2.2 La segmentation

Après la détermination des points début et fin de la parole, une étape importante suivra, celle de la segmentation. Cette étape vise à extraire du continuum acoustique des unités sur lesquelles portera la décision dans l'étape de la reconnaissance.

Parmi les unités de segmentation on distingue les unités courtes infraphonémique (phones), les phonèmes et les unités longues (diphones, syllabes, mots, etc.), chacune de ces unités possède ses propres caractéristiques :

Les unités longues : un grand nombre de systèmes sont conçus autour de telles unités. Leurs grand avantage est qu'elles permettent de recouvrir le phénomène de la coarticulation.

Les phones : ils sont plus faciles à localiser mais n'ont aucun statut linguistique, il faut les assembler en unités plus larges au moyen de règles phonétiques, phonologiques, lexicales et syntaxiques.

Les phonèmes : leur nombre est limité quelle que soit la langue, cependant les localiser est une tâche très difficile. Ces unités sont utilisés en DAP (Décodage Acoustique Phonétique).

4.2.2.1 Le choix de l'unité de segmentation

Comme la majorité des chiffres en anglais sont monosyllabique, nous avons choisi d'effectuer la segmentation du signal en syllabes. L'algorithme utilisé pour réaliser cette segmentation, est celui proposé par Paul Mermelstein [41]. C'est un algorithme qui permet de déterminer les frontières des syllabes d'une manière automatique en se basant sur la différence entre la fonction "convexe hull " (une sorte d'enveloppe) de l'énergie, et l'énergie elle-même.

Linguistiquement la syllabe est définie comme un groupe formé de consonnes et de voyelles qui se prononcent d'une seule émission de voix. Par exemple, Paris possède deux syllabes. Une autre définition pratique de la syllabe: c'est une séquence de son de parole possédant un maximum de sonorité entre deux minimums de sonorités. Pour l'implémentation, cette définition n'est pas suffisante, nous devons alors matérialiser la sonorité en termes de mesures physiques prisent sur le signal parole. Dans notre cas on va utiliser l'énergie court terme avec lissage temporel et filtrage passe-bande, les

maximums de cette fonction seront interprétés comme des pics potentiels de syllabes, et les minimums comme des frontières potentielles de syllabes.

Les frontières syllabiques établies par cet algorithme, ne représentent pas nécessairement les syllabes telle que défini au niveau phonologique, alors on utilise le terme unité syllabique pour différencier entre la définition phonologique de la syllabe et sa définition phonétique.

4.2.2.2 Description de l'algorithme

Les principales étapes de cet algorithme sont le calcul de l'énergie, son lissage, le calcul de sa fonction convexe-hull, et enfin la segmentation en syllabes. Tous les filtres utilisés sont des filtres de type FIR dont la principale caractéristique est la stabilité.

Les filtres FIR

On peut définir un SLIT (Système Linéaire et Invariant dans le Temps) par une équation aux différences linéaire et à coefficients constants dont la forme générale est [16] :

$$y(n) = \sum_{k=0}^{M} \beta_k x(n-k) - \sum_{k=1}^{N} \alpha_k y(n-k) \tag{4.2}$$

Un filtre FIR est un SLIT dont les coefficients α_k, $1 \le k \le N$, de l'équation aux différences sont nuls, d'où l'appellation de systèmes non récursifs, ces systèmes ont le grand avantage d'être toujours stables.

Pour la synthèse des filtres RIF plusieurs méthodes existent, tel que [16]:

- la méthode de l'échantillonnage fréquentiel par transformée de Fourier inverse des coefficients d'un filtre discret idéal.
- la méthode de Parks et McClellan basée sur l'optimisation d'erreur entre courbe réelle et courbe idéale.
- la méthode du fenêtrage appliquée à un filtre idéal.

La dernière méthode est celle que nous avons utilisée dans ce travail pour sa simplicité, elle fait appelle à des fenêtres temporelles pour tronquer le nombre infini de termes (irréalisable en pratique) de la fonction de transfert en un nombre fini. La fenêtre peut être de type rectangulaire ou de type Hamming qu'on lui préfère, car elle permet d'avoir des bords doux et progressifs, et permet ainsi d'atténuer le phénomène connu sous le nom de l'effet de Gibs, ou l'apparition des ondulations au niveau des discontinuités de la réponse fréquentielle approximative.

Pour réaliser les filtres il suffit de calculer les coefficients selon le gabarit fixé. Ces coefficients sont ensuite implantés dans l'équation aux différences correspondante.

Calcul de l'énergie

Le signal parole est filtré par un filtre (FIR) passe-bande de 500Hz à 4Khz, avec une atténuation de 12dB/octave en dehors de cette bande, ensuite l'énergie est calculée sur des fenêtres de 30ms espacés de 10ms sur tout le signal.

Le lissage de l'énergie

L'énergie calculée, sera lissée par un filtre passe bas à 40Hz, le filtre utilisé est un filtre FIR, conçu par la méthode des fenêtres. Après lissage, l'énergie est normalisée :

$$energie\_normalisée(n) = \frac{energie(n) - \min(energie)}{\max(energie) - \min(energie)} \qquad (4.3)$$

La fonction Convex-Hull

La fonction convexe-hull (voir Figure 26), est définie comme une fonction non décroissante du début du segment jusqu'à son maximum, et non croissante du point maximum jusqu'à la fin du segment, elle est obtenue à partir de l'énergie du signal vocal.

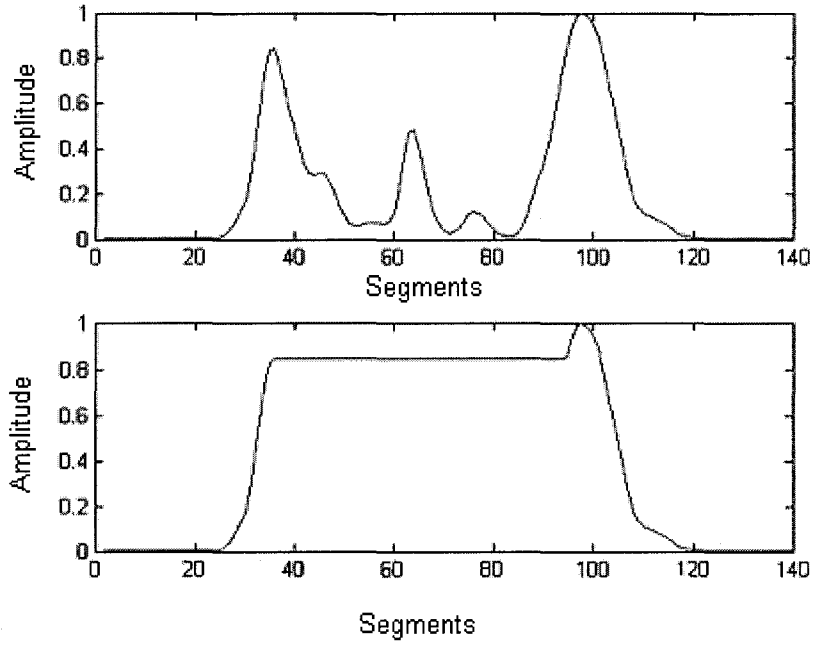

Figure 26 La fonction convexe-hull de l'énergie du signal

La segmentation en syllabes

La segmentation est effectuée d'une manière récursive. Au début on prend le segment en entier et on évalue la différence de la fonction "convexe- hull" de son énergie avec l'énergie de ce même segment. Le point qui correspond au maximum de cette différence est une frontière potentielle. Si ce maximum excède un seuil donné, alors le segment initial est divisé en deux sous segments, de son début jusqu'au point où la différence est supérieure au seuil, et de ce point vers la fin du segment initial. Une nouvelle fonction

"convexe- hull" alors est calculée et une nouvelle segmentation est appliquée pour chaque sous segment. Ce processus est répété jusqu'à atteindre à une différence de niveaux d'énergie inférieure au seuil.

4.2.2.3 La correction de la segmentation

À fin d'optimiser la segmentation, une étape de correction est nécessaire à fin de supprimer les segments non désirés. Selon le cas chacun de ces segments sera annexer au segment de parole qui le précède ou qui le suit. À la fin de cette opération tout segment représente un chiffre.

Dans le cas particulier des chiffres connectés plusieurs contraintes ont été ajoutées basées sur la durée, l'énergie et le taux de passage par zéro. Dans le cas où on obtient des segments plus courts que 120 ms une analyse supplémentaire est faite avant d'éliminer une frontière. Si ce segment présente un taux de passage par zéro élevé la frontière droite sera éliminée, sinon c'est la frontière gauche. Aussi nous avons imposé une durée entre les maximums de syllabes, ainsi la durée entre deux maximums doit être supérieure ou égale à 500 ms. Finalement une correction sera faite pour les frontières du chiffre 'six' basée sur le taux de passage par zéro (élevé) et le niveau d'énergie (faible niveau d'énergie).

4.2.3 Extraction des paramètres

Selon l'application il est important d'opter pour des paramètres pertinents, discriminants et robustes. Les coefficients cepstraux permettent un meilleur lissage du spectre vocal et sont plus fiables pour la reconnaissance. Pour notre application nous avons choisi les

coefficients LPCCs que nous dérivons à partir des coefficients LPCs. Le processus de l'extraction des paramètres est illustré par la Figure 27.

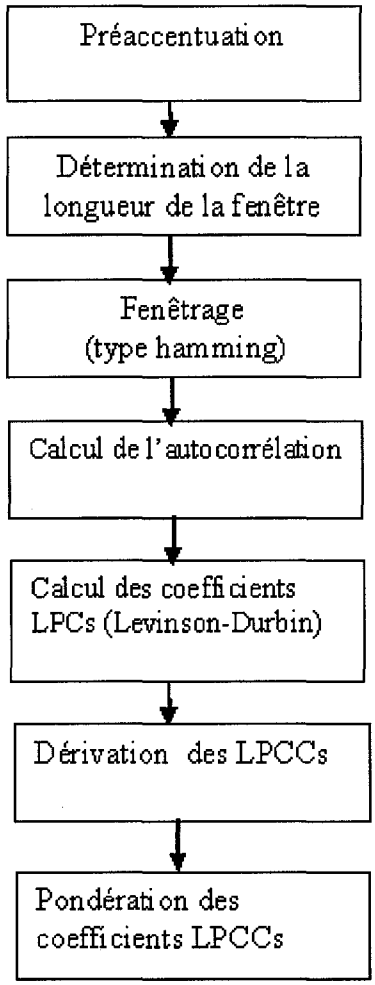

Figure 27 Processus de l'extraction des paramètres LPCCs pondérés

Préaccentuation

La préaccentuation est un procédé par lequel on amplifie les hautes fréquences, afin de leur donner une amplitude supérieure au bruit de fond.

La préaccentuation du signal consiste en un passage dans un filtre de transmittance

$1-\mu z^{-1}$, avec μ compris entre 0.9 et 1 (0.95 pour notre cas). Son but est d'accentuer la haute fréquence du spectre. (Figure 28)

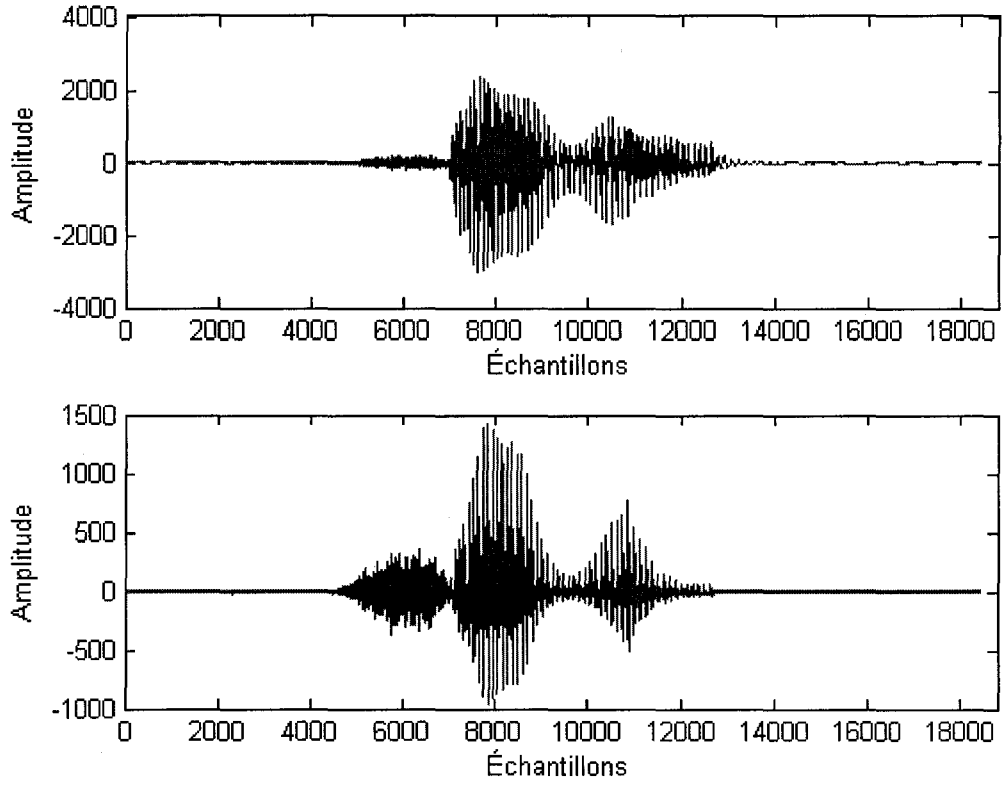

Figure 28 Signal vocal (source et préaccentué).

Ce prétraitement assure un bon conditionnement des algorithmes de résolution pour la détermination des coefficients LPCs, en particulier l'algorithmes de Levinson-Durbin [14].

Détermination de la longueur des fenêtres

Dans les systèmes RAP actuels on utilise toujours des fenêtres de longueur fixe pour le calcul des paramètres. Puisque le débit d'élocution n'est pas fixe et qu'il varie d'un locuteur à un autre voir même pour le même locuteur alors une fenêtre de longueur fixe,

ou un segment de parole de N échantillon pour un mot ne correspond pas forcément à un segment de N échantillon d'un autre mot. Ainsi il est judicieux de trouver un moyen d'adaptation entre les deux segments pour pouvoir les comparer d'où l'idée des fenêtres à longueur variable.

Dans notre cas la durée des fenêtres N n'est pas connue à priori, elle est définie lors de l'analyse à l'aide d'un seuil local ($seuil = const.\max_{local}$) prédéfini expérimentalement avec lequel on sélectionne la partie la plus énergétique pour chaque mot (voir Figure 29). Ce seuillage constitue aussi un moyen d'alignement entre les deux mots (référence, et à reconnaître) puisque nous sélectionnons la même zone d'énergie pour chacun d'eux, ensuite on divise cette longueur par un nombre de manière à respecter la stationnarité du segment. Dans notre cas le nombre dix s'est avéré un bon choix. La longueur obtenue constitue la longueur N de la fenêtre qu'on recherche.

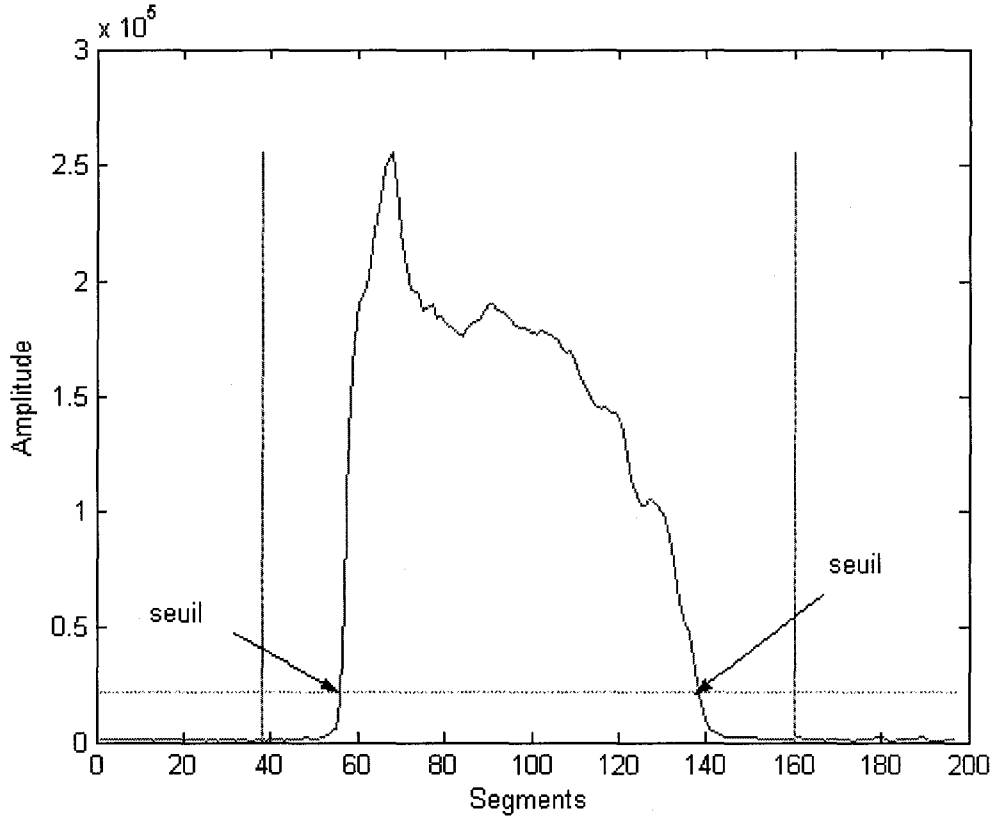

Figure 29 Sélection de la partie énergétique du signal à l'aide d'un seuil

Fenêtrage

Le résultat du fenêtrage donne:

$\widetilde{x}_l = x_l(n).w(n), \qquad 0 \le n \le N-1.$

$w(n)$: est une fenêtre de type hamming

$x_l(n)$: segment de parole

Calcul de l'autocorrélation

L'autocorrélation est évaluée pour chaque fenêtre l du signal et on aura :

$$r_l(m) = \sum_{n=0}^{N-1-m} \tilde{x}_l(n)\tilde{x}_l(n+m),\ m = 0,1,\ldots,p \tag{4.4}$$

p est l'ordre de prédiction.

Calcul des coefficients LPC

En général l'ordre de prédiction p est choisi en fonction de la fréquence d'échantillonnage [14], nous avons choisi $p = 8$.

Comme déjà avancé dans le chapitre 2, la méthode utilisée pour déterminer les paramètres LPC à partir des coefficients de l'autocorrélation est l'algorithme de Levinson-Durbin, son principe est le suivant [3] :

$$\begin{aligned}
&E^{(0)} = r(0)\\
&k_i = \left\{ r(i) - \sum_{j=1}^{L-1} \alpha_j^{(i-1)} r(|i-j|) \right\} / E^{(i-1)},\ 1 \le i \le p\\
&\alpha_i^{(i)} = k_i\\
&\alpha_j^{(i)} = \alpha_j^{(i-1)} - k_i \alpha_{i-j}^{i-1}\\
&E^{(i)} = (1-k_i^2)E^{(i-1)}
\end{aligned} \tag{4.5}$$

Cet ensemble d'équations est résolu d'une manière récursive. La solution finale donnera les coefficients LPCs a_m :

$$a_m = \alpha_m^{(p)}\ ,\ 1 \le m \le p \tag{4.6}$$

Calcul des coefficients cepstraux

En ce qui concerne les coefficients cepstraux, nous avons opté pour les LPCCs. Une étude comparative [20] a pu établir que les coefficients LPCCs surpassent les MFCCs en terme de performance de reconnaissance.

L'obtention de ces coefficients se fait par dérivation à partir des coefficients LPCs en utilisant les formules de conversion suivantes [42] :

$$c_n = \begin{cases} c_n = a_n, \ n = 1 \\ c_n = a_n \sum_{k=1}^{n-1} \left(\frac{n-k}{n} \right) c_{n-k} a_k, \ 2 \le n \le p \\ c_n = \sum_{k=1}^{p} \left(\frac{n-k}{n} \right) c_{n-k} a_k, \ p < n \le q \end{cases} \tag{4.7}$$

Pour le nombre de coefficients cepstraux, nous avons choisi $q = 18$.

La pondération des coefficients ou liftrage

Lorsqu'on utilise des coefficients cepstraux en RAP, on peut améliorer la performance de la reconnaissance par l'application d'une pondération appropriée sur ces derniers [26], une forme générale de cette pondération est :

$\hat{c}_n = w_n c_n, \ 1 \le n \le q.$

Une méthode appropriée de pondération est le liftrage passe-bande ou filtrage dans le domaine cepstral. La fonction utilisé pour effectuer la pondération est celle retenue par Juang et al [26] comme meilleure choix soit :

$$w_n = \left[1 + \frac{q}{2} \sin\left(\frac{\pi . n}{q} \right) \right], \ 1 \le n \le q. \tag{4.8}$$

4.2.4 La création du dictionnaire de référence

Cette phase appelée aussi phase de l'apprentissage, son but est la création du dictionnaire de référence qui va servir de base dans la phase de décision dans le système de reconnaissance. C'est un ensemble représentatif des différentes classes qui existent dans l'ensemble d'apprentissage, pour cela on va utiliser l'algorithme de classification de k-means (Liyod-Max) que nous avons déjà présenté.

4.2.4.1 Choix du nombre de classe M

Pour les systèmes de reconnaissance multi-locuteurs, généralement on suggère de prendre plusieurs mots de référence par mot [27, 43-45] jusqu'à 12 par mots, spécialement lorsqu'il s'agit d'utiliser des mots isolés pour la reconnaissance des mots connectés. Pour notre application nous avons choisi de prendre quatre références par mot.

4.2.4.2 L'initialisation de l'algorithme

D'après l'algorithme de k-means les M points initiaux doivent être pris d'une façon aléatoire, dans le cas de la parole il est déconseiller de procéder de cette manière car l'algorithme risque de converger vers des résultats très pauvres en terme de représentativité, plusieurs idées ont été développés pour pallier à cet inconvénient, pour notre cas nous avons utilisé la méthode dite des bins. Cette méthode suggère la répartition de l'ensemble d'apprentissage en groupe (bins), et de prendre aléatoirement des points de chacun de ces groupes, ce qui permet d'avoir des points initiaux qui reflètent la variété de l'ensemble d'apprentissage.

Comme nous savons à priori les différentes classes qu'on veut avoir (les chiffres de 0 à 9 et oh), alors on va prendre d'une manière aléatoire un nombre (quatre) de mots de chaque classe, et on va s'en servir pour initialiser notre algorithme. D'autre part on sait que les solutions obtenues par l'algorithme k-means sont localement optimales, alors on va répéter le procédé plusieurs fois. Il est impossible de faire toutes les combinaisons possibles dont le nombre est donné par l'équation(4.9), alors on va essayer de répéter l'opération jusqu'à l'obtention des résultats acceptables.

$$S(k,N)=\frac{1}{k!}\sum_{i=0}^{k}(-1)^{k-i}\binom{k}{i}i^{N} \tag{4.9}$$

Avec k le nombre de classes voulues, et N le nombre d'éléments dans le groupe d'apprentissage.

4.2.4.3 La distance utilisée

Les performances de l'algorithme de k-means dépendent fortement de la distance utilisée pour décider de la classe d'appartenance de chaque élément du groupe d'apprentissage. Dans la littérature souvent on retrouve la distance de Manhattan (4.11) ou la distance Euclidienne (4.10), nous avons utilisé la deuxième.

$$d(x,y)=\sqrt{\sum_{i}(x_i-y_i)^2} \tag{4.10}$$

$$d(x,y) = \left|(x-y)\right| = \sum_i \left|(x_i - y_i)\right| \tag{4.11}$$

Avec x_i et y_i les $i^{ème}$ élément respectivement de x et y.

4.2.5 La phase de reconnaissance

La phase de reconnaissance est la dernière étape dans le processus de reconnaissance, elle se déroule en deux étapes, le calcul de distance et le processus de décision.

4.2.5.1 Le calcul de distance

L'évaluation de la distance entre le mot de référence et celui à reconnaître est effectuée à l'aide d'une distance locale et d'une distance globale.

La distance locale : elle est évaluée entre deux segments représenté chacun par un vecteur de paramètres. La distance que nous avons utilisée est la distance cepstrale pondérée. C'est cette distance qui a donné les meilleurs résultats de reconnaissance [23].

La distance globale : La distance globale entre deux mots sera une somme pondérée des distances locales entre le mot de référence et celui à reconnaître.

$$D(X,Y) = \sum_i d_i \tag{4.12}$$

D La distance globale.

d La distance locale.

4.2.5.2 Le processus de décision

La décision est l'ultime étape de reconnaissance, dans les systèmes de RAP, généralement la règle de décision utilisée est la règle du plus proche voisin *Kpp* [27], appelée aussi *K-nn* (K-Nearest Neighbour) .

La méthode de décision *K-nn* est habituellement liée à la notion de ressemblance entre les observations [25], l'idée de base de cette méthode est fort intuitive. Considérons une observation $x^T = (x_1, x_2, \ldots, x_N)$ comme un point dans l'espace à N dimensions $\left(\Re^N\right)$, l'ensemble des observations forme des nuages de points répartis dans l'espace des formes, la notion de ressemblance peut être réduite à la notion de distance, selon cette méthode, on calcule les distances entre x et chacun des représentants des classes existantes, qu'on appelle aussi les références, et on identifie la classe de chacun des K prototypes les plus proches de x, ensuite l'appartenance de l'observation x est décidée selon la classe la plus représentée par les K prototypes, la figure 30 est une illustration de la décision par K plus proche voisins dans ce cas $K=4$.

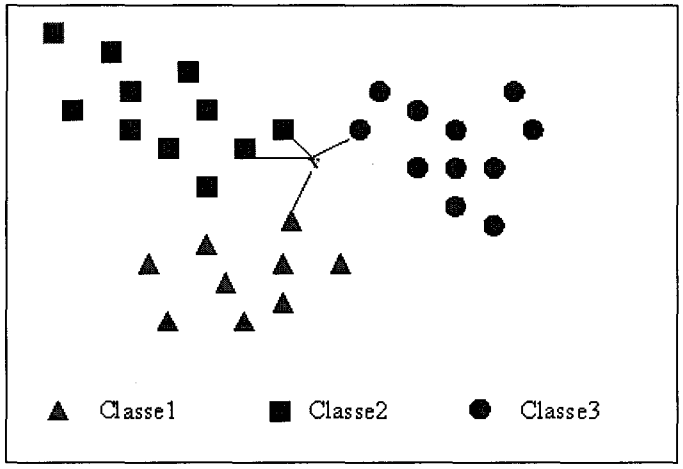

Figure 30 Principe du plus proche voisin : l'élément x est affecté à la classe2

4.3 Le DSP

4.3.1 Introduction

Certaines applications nécessitent des calculs en temps réel sur des échantillons, c'est le cas par exemple pour :

- le filtrage numérique,
- la reconnaissance de la parole,
- le contrôle (moteur, etc.).

Pour ces applications on utilise des processeurs spécifiques dédiés au traitement du signal appelés DSP (Digital Signal Processor).

Les DSPs sont similaires aux microprocesseurs d'usage général à l'exception d'être mieux adaptés pour effectuer les opérations d'addition et de multiplication et ceci d'une manière plus optimale. Les DSPs ont aussi l'avantage de la faible consommation d'énergie et du coût moindre.

La plupart du temps, on retrouve le DSP dans une chaîne de traitement du signal (Figure 31) composée essentiellement d'un CAN (Convertisseur Analogique Numérique) pour numériser le signal, et d'un CNA (convertisseur Numérique Analogique) pour restituer le signal après traitement.

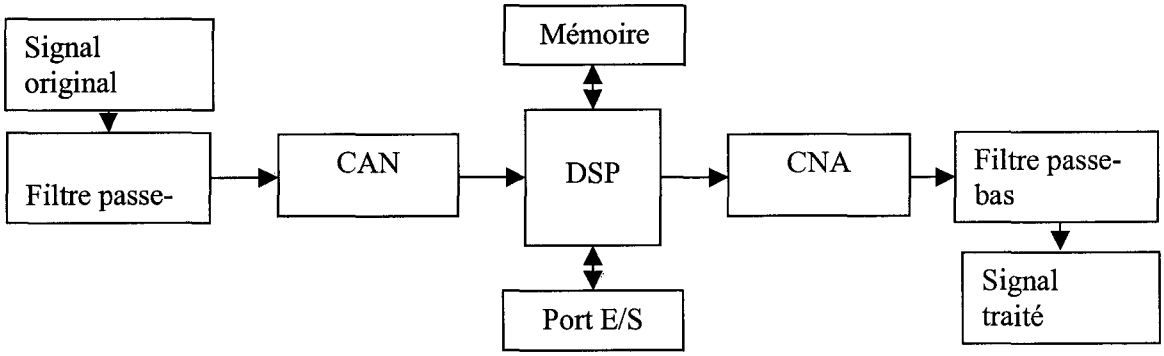

Figure 31 Chaîne de traitement à base d'un DSP

Vu d'ensemble on peut classer les DSPs en deux grandes familles:

- les DSPs à virgule fixe: qu'on retrouve dans la plupart des applications où le coût est un facteur important (ils sont moins chers que les DSP à virgule flottante), cependant ils sont plus compliqués à programmer.
- les DSPs à virgule flottante : sont plus faciles et plus souples à programmer que les DSPs à virgule fixe, dans ce cas on fait intervenir une mantisse et un exposant et on bénéficie d'une dynamique plus importante. On retrouve ce type de DSP dans les applications Audio.

Dans le cadre de ce projet, le DSP utilisée est le TMS320C6711 de la compagnie Texas Instrument.

4.3.1.1 Mesure des performances des DSPs

Pour évaluer les performances d'un DSP, des unités de mesure ont été établies dont les plus courantes sont :

- MFLOPS (Million Floating Point Operations Per Second) : mesure le nombre d'opérations à virgule flottante (multiplications, additions, soustractions, etc.) que le DSP à virgule flottante peut réaliser en une seconde.

- MOPS (Million Operations Per Second): mesure le nombre total d'opérations (les transferts de données et les opérations d'E/S) que le DSP peut effectuer en une seconde. Cette définition permet de mesurer les performances globales d'un DSP, plutôt que ses seules capacités de calcul.

- MIPS (Million Instructions Per Seconde) : mesure le nombre de codes machines (instructions) que le DSP peut effectuer en une seconde.

- MBPS (Mega Bytes Per Second) : cette unité permet de mesurer le taux de transfert d'un Bus particulier ou d'un dispositif d'E/S du DSP.

4.3.2 Description du DSP TMS320C6711

Le TMS320C6711 est un processeur à virgule flottante de la famille TMS320C6000, c'est une nouvelle génération de DSPs développé par Texas Instrument, qui utilise une version améliorée de l'architecture VLIW (Very Long Instruction Word) appelée TI VelociTI [46] comme pour les processeurs conventionnels, le TMS320C6711 est composé de parties principales suivantes : l'unité centrale (CPU), les mémoires et les périphériques, le tout est relié par un bus interne(voir Figure 32).

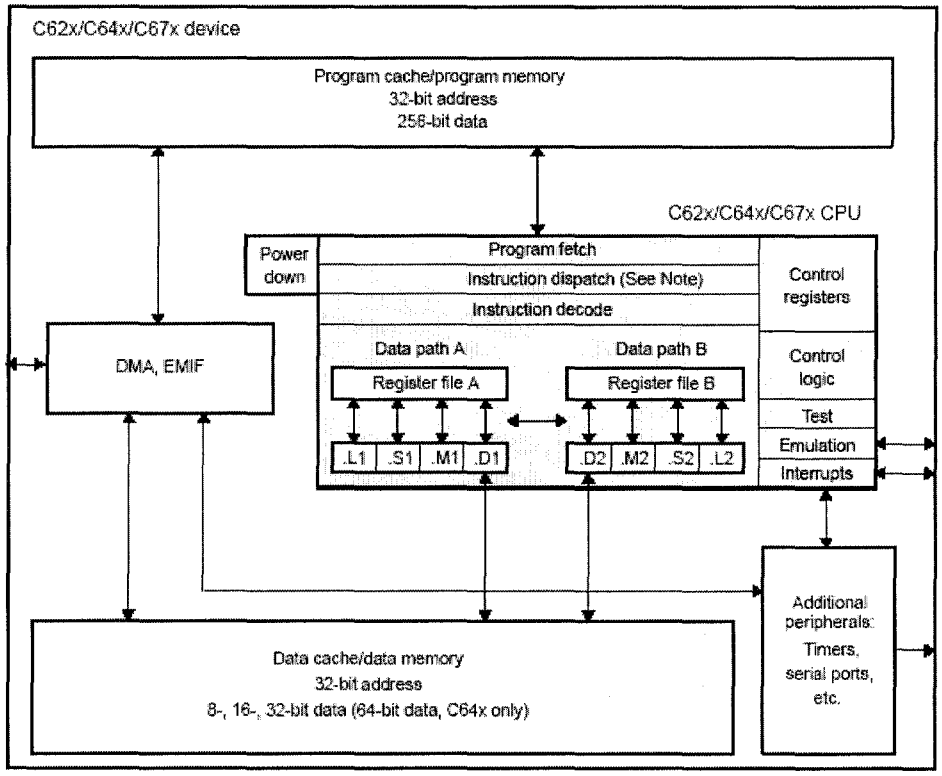

Figure 32 Structure interne du DSP TMS320C6711 [52]

4.3.2.1 L'unité centrale (CPU)

C'est le cœur du processeur, il est composée principalement des éléments suivants :

L'unité de contrôle du programme : elle contient trois composantes (Instruction Fetch, Instruction Dispatch, Instruction Decode) nécessaires à l'exécution d'instructions en pipeline, c'est un moyen qui permet de gagner du temps lors de l'exécution en effectuant plusieurs taches en même temps en les parallélisant.

Les unités fonctionnelles : Le CPU par son architecture VLIW permet le traitement par paquets de 8 instructions pouvant être traitées par les unités fonctionnelles (huit unités: 6 UALs et 2 Multiplieurs). Ces unités sont organisées en deux blocks d'unités (Data Path A, Data Path B), chaque block contient les éléments suivants :

- l'unité (.M) : Utilisée pour les opérations de multiplications.
- l'unité (.L) : Utilisée pour les opérations logiques et arithmétiques.
- l'unité (.S) : Utilisée pour effectuer un branchement ou pour la manipulation des bits.
- l'unité (.D) : Utilisée pour rechercher ou stocker (load/store) des données, également utilisée pour les opérations arithmétiques.
- Les registres : on compte plusieurs registres dont 32 registres à usages générale à 32 bits, 16 registres pour l'unité A (A0 jusqu'à A15), et 16 pour l'unité B (B0 jusqu'à B15), aussi on compte 13 registres à usage spéciale appelés registres de contrôle dont huit registres dédiés à la gestion et le contrôle d'interruptions.

4.3.2.2 Les mémoires

Avec un bus de données de 32 bits, le TMS320C6711 est capable d'adresser jusqu'à 4 GOctets d'adresses, soit 2^{32}, réparties en adresses internes et adresses externes.

La mémoire interne

C'est une mémoire cache à deux niveaux L1/L2 répartis comme suit :

- 4 Koctets de mémoire programme cache notée L1P.
- 4 Koctets de mémoire donnée cache notée L1D.

- 64 Koctets de mémoire RAM/cache qu'on alloue en programme ou donnée notée L2.

La mémoire externe

Se sont quatre espaces de mémoire (CE0, CE1, CE2, CE3) reliés au processeur via un interface de mémoire externe (EMIF), et dont la location dépend du type da la cartographie mémoire utilisée (MAP 0 ou MAP 1).

Les espaces de mémoires CE0, CE2 et CE3 peuvent supporter des mémoires Asynchrones (SRAM et EPROM) et synchrones (SBSRAM et SDRAM) à lecture/écriture à 8 ou 16 bits, et à 32 bits pour lecture seulement, par contre CE1 supporte des mémoires à lecture/écriture à 32 bits, et des mémoires à lecture seulement à 8 et 16 bits.

4.3.2.3 Les périphériques

Le TMS320C6711 contient les périphériques suivants :

- EMIF : c'est une interface de mémoire externe à 32 bits, elle permet l'interfaçage avec différents types de composants: SBSRAM, SDRAM, SRAM, ROM,...etc. L'EMIF du TMS320C6711 nécessite une horloge externe fournie par le système, et fournit un signal d'horloge à lequel toutes les mémoires externes doivent être synchronisées.

- EDMA : elle permet les transferts de données depuis un élément externe libérant ainsi la CPU pour des tâches de calcul, elle contient 16 canaux de transmission programmables, et permet des transferts de données depuis la mémoire cache L2, les périphériques du TMS320C6711 et de la mémoire externe.

- HPI : c'est un port parallèle à 16 bits qui peut être adressé directement à la mémoire, que celle ci soit interne ou externe, permettant ainsi des accès directs dans l'espace mémoire du DSP.

- McBSPs : se sont deux ports (McBSP0 et McBSP1) dédiés à la communication série entre le processeur et le monde extérieur.

- 2 timers : c'est des compteurs programmables à 32 bits, ils permettent de mesurer la durée ou de compter des évènements, de générer des impulsions, de générer des interruptions CPU et de synchroniser les échanges lors des accès directs à la mémoire.

- Générateur d'horloge par PLL.

4.3.2.4 Performances du TMS320C6711

Avec une utilisation optimale (les 8 unités fonctionnelles en fonction), le TMS320C6711 est capable d'atteindre 900 MFLOPS, à une fréquence d'horloge de 150 MHz c'est 1200 MIPS avec un temps de 6.67-ns par cycle d'instruction.

4.3.3 Les outils de développement

Pour les besoins de notre application nous avons utilisé l'outil de développement de Texas Instrument composé essentiellement d'un outil matériel le Kit DSK6711, et d'un outil logiciel le Code Composer Studio.

4.3.3.1 Le kit DSK6711

Le DSK6711 est un système à DSP complet, c'est une carte conçue autour d'un processeur TMS320C6711 cadencé à une fréquence de 150Mhz et composée essentiellement de :

- circuit codec à 16 bits de type TLC320AD535 qui assure la conversion Analogique/Numérique et Numérique/Analogique, et connecté à une prise jack d'entrée mono (J7) et une prise jack de sortie stéréo (J6), sa communication avec le TMS320C6711 se fait via la liaison série McBSP0, le AD535 est cadencé à 4Mhz ce qui engendre une fréquence d'échantillonnage fixe des données de 8Khz. C'est la contrainte qu'on doit respecter dans notre application. Il faut noter que la liaison série McBSP0 n'est pas dédiée seulement au circuit codec, en effet elle est partagée via multiplexeur entre la partie codec et un module d'extension de périphérique, alors que la l'autre liaison série McBSP1 est entièrement destinée à une extension de périphérique disponible sur le connecteur J3.
- 16Mo de SDRAM, et 128Ko de flash ROM ont été rajoutés sur les espaces CE0 et CE1, la mémoire peut encore être étendue par le connecteur J1 dans les espaces CE2 et CE3 qui sont non utilisés.
- le port hôte est accessible par le connecteur J2 de même que le module JTAG (pour le test et l'émulation) permettant le développement sur PC en utilisant Code Composer Studio.

Cet outil est complété par tout une gamme de documentation [47-57] qui détaillent le fonctionnement et l'exploitation, et qui permettent ainsi de tirer le meilleur parti des ressources du processeur et profiter de tout la puissance qu'offre la carte DSK6711 .

4.3.3.2 Le Code Composer Studio

L'évaluation des performances des algorithmes sur le DSP est effectuée en utilisant Code Composer Studio (CCS), son fonctionnement est illustré par la Figure 33

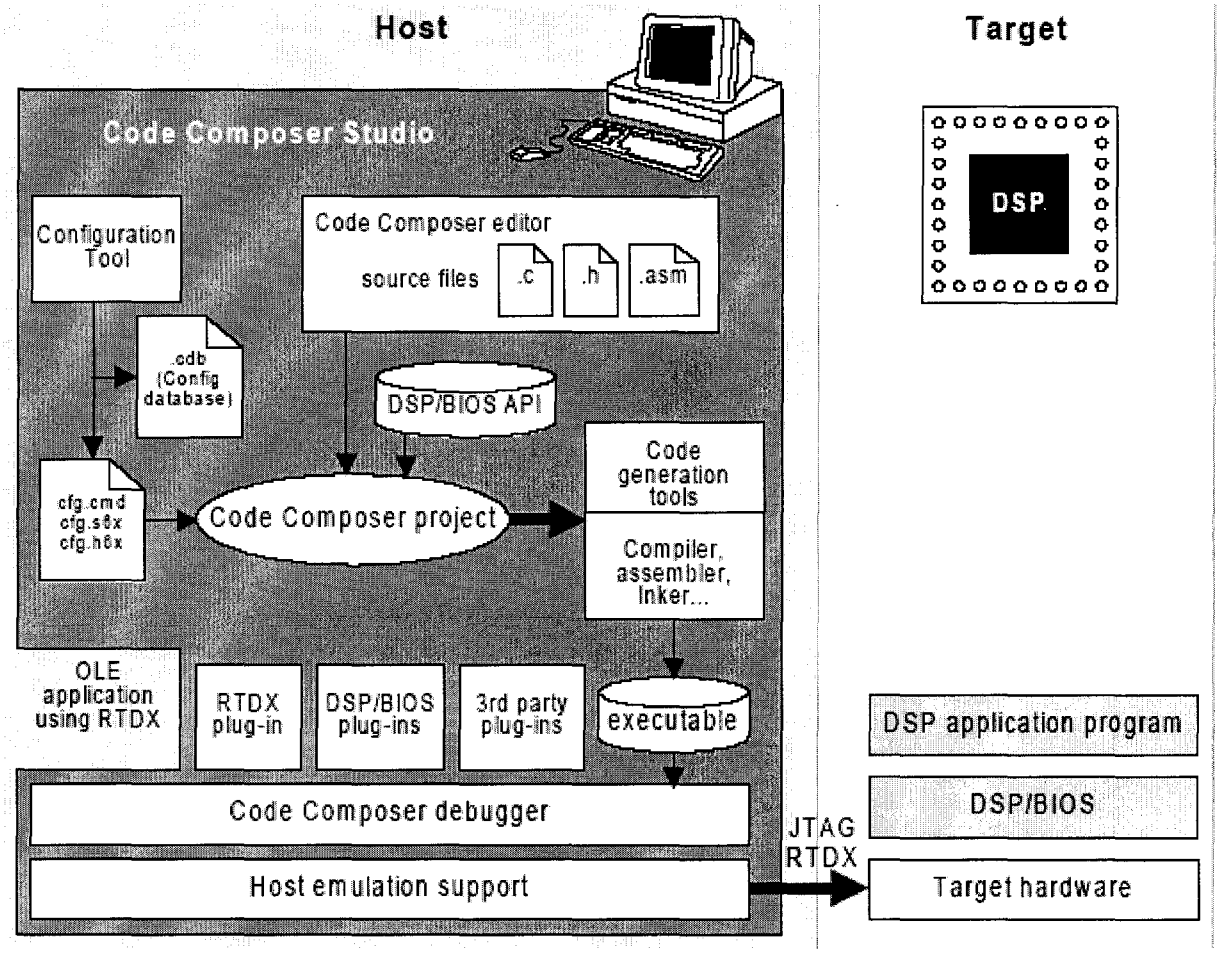

Figure 33 Structure du système du développement du TMS320C6711 [57]

Code Composer Studio est un logiciel de développement qui inclut les éléments suivants:

- environnement de développement intégré (IDE) : il permet l'édition, le 'built', et la correction 'debug' des programmes destinés au DSP.

- les outils de génération du code pour le TMS320C6000 : ces outils sont le compilateur, l'assembleur et l'éditeur de lien. Le compilateur C/C++ permet de compiler le programme source xxx.c pour le convertir en assembleur xxx.asm, l'assembleur reçoit le fichier xxx.asm et le convertit en langage machine ou fichier objet xxx.obj, enfin l'éditeur de liens (linker) qui combine les fichiers objet et les fichiers librairies et le fichier xxx.cmd pour produire un fichier exécutable avec une extension .out, c'est ce fichier qui sera chargé sur le processeur C711 pour être exécuter.
- le (DSP/BIOS) : c'est un outil d'analyse en temps réel, pour s'en servir, on doit créer un fichier de configuration 'xxx.cdb', où seront définis les objets utilisés par l'outil DSP/BIOS (voir Figure 34), ce fichier permet aussi de faciliter l'organisation de la mémoire et la gestion du vecteur des interruptions, en offrant la possibilité de les faires sur un environnement visuel via la section de gestion de la mémoire MEM (Memory Section Manager), et via HWI (Hardware Interrupt Service Routine Manager) pour les interruptions.

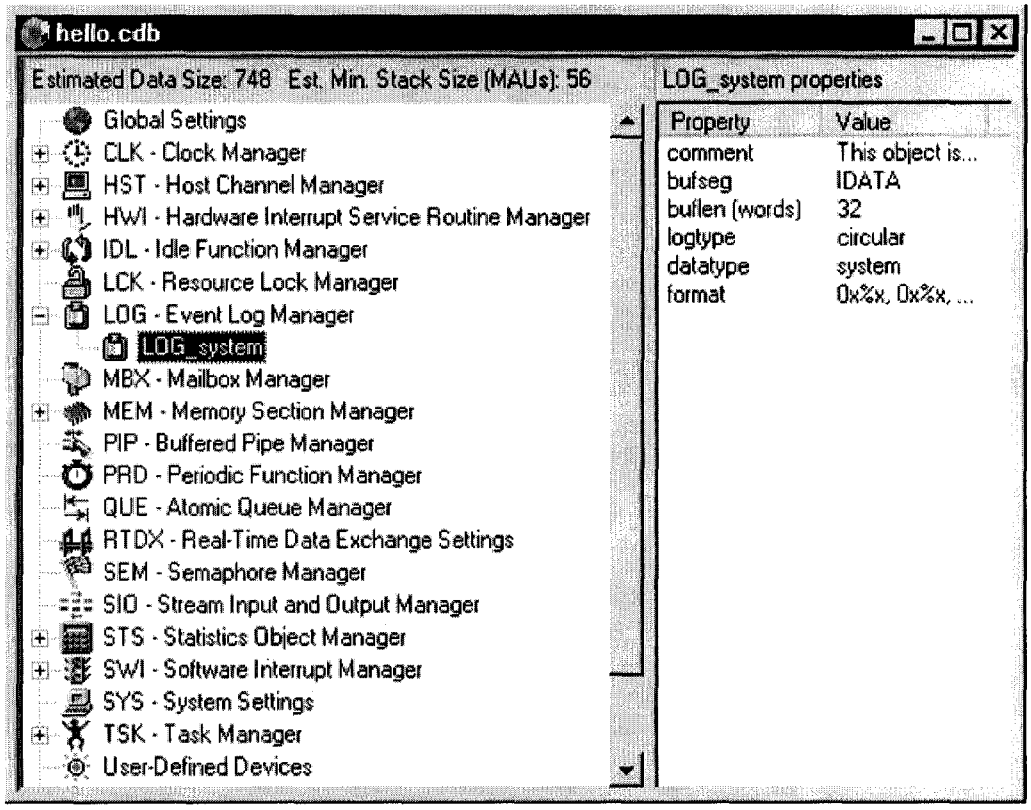

Figure 34 L'outil DSP/BIOS

- Lien JTAG (Joint Team Action Group) et le RTDX(Real Time Data Exchange) (voir Figure 35):Le RTDX permet un échange de données en temps réel entre l'hôte(PC par exemple) et la destination (la carte DSK dans notre cas), il permet aussi l'analyse et la visualisation des données au cours de l'exécution du programme, alors que le lien JTAG est utilisé pour atteindre l'émulateur (qui se trouve à l'intérieure du DSP), c'est ce dernier qui permet au CCS de contrôler en temps réel l'exécution du programme

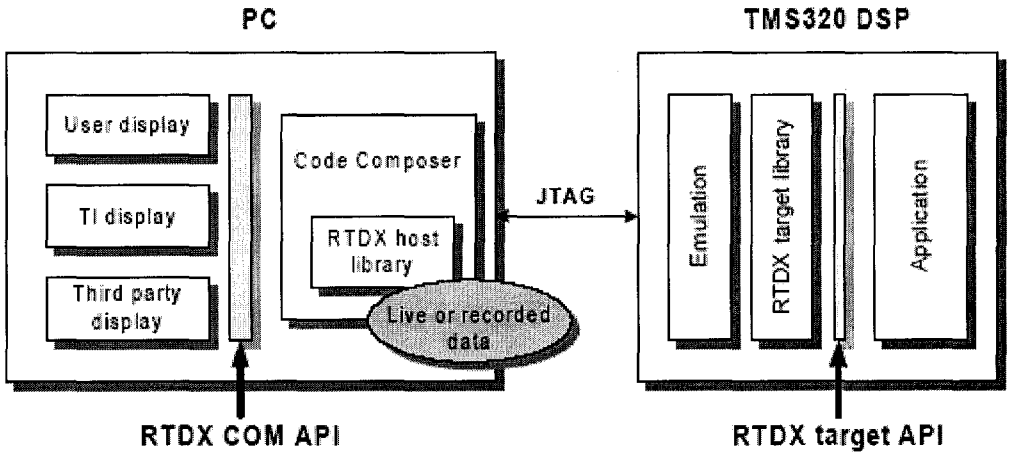

Figure 35 L'outil JTAG [57]

- un simulateur intégré : en effet Code Composer Studio offre la possibilité de tester des programmes pour DSPs sans utiliser la carte DSK à l'aide d'un simulateur intégré.

4.3.3.3 La Création d'un projet en CCS

La réalisation d'une application avec CCS se fait par la création d'un projet (un fichier avec extension .pjt) suivi de la configuration des différentes options nécessaires à son exécution. Dans un projet on retrouve plusieurs fichiers regroupés selon leurs types dans des répertoires différents (Figure 36), ces répertoires sont :

- Include : contient les fichiers de l'en-tête *.h
- Libraries : contient les fichiers librairies *.lib
- Source : contient les fichiers sources du projet, ces fichiers peuvent être des programmes en langage C, *.c, ou/et des programmes en assembleur *.asm. Dans le cadre de ce projet nous avons préféré utiliser le langage C à cause de sa portabilité d'abord et de sa simplicité. En effet traduire un algorithme en

assembleur optimisé dans le cas du TMS320C6711 est une tâche extrêmement difficile.

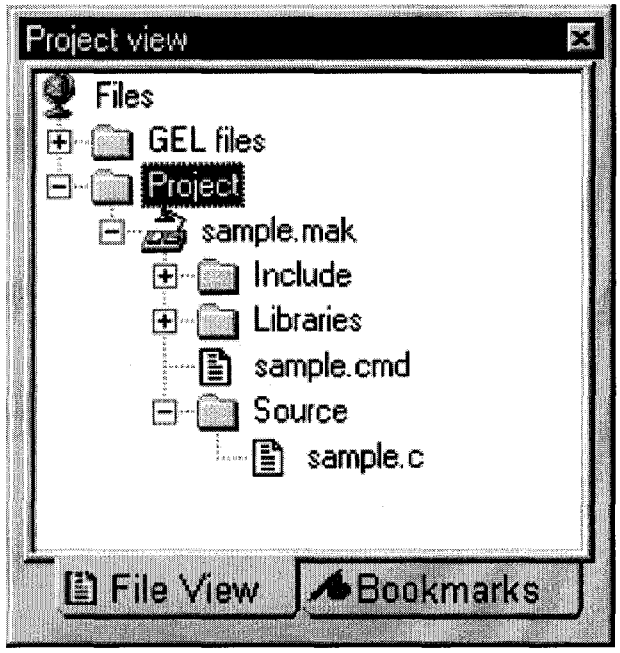

Figure 36 La structure d'un projet en Code Composer Studio

4.3.3.4 L'organisation de la mémoire

Pour réussir tout projet, il est indispensable de bien organiser la mémoire, par organisation de la mémoire, on entend définir l'emplacement physique des différents éléments du programme. Lors de la compilation le compilateur C crée plusieurs sections de code ou de données, dans ce qui suit on va définir ces sections et leurs contenus :

- .text : code exécutable

- .cinit : tables d'exécution des variables globales et statiques explicitement initialisées.
- .const : chaînes de caractères constantes et variables globales et statiques de type const et explicitement initialisées.
- .switch : contient les tables pour les lignes de switch.
- .bss : variables globales et statiques.
- .far : variables globales et statiques déclarés de type far.
- .stack : ou pile pour les variables locales.
- .sysmem : zone de mémoire dynamique ou (heap en anglais) pour les fonctions d'allocation mémoire (malloc, calloc ,...) .
- .cio : les fonctions standards d'E/S du langage C. (printf, ...ect.) .

L'organisation de la mémoire est réalisée par un type spécial de fichier appelé fichier de commande (xxx.cmd) générer par le fichier de configuration CDB via le gestionnaire de mémoire MEM, son rôle est d'assurer trois fonctions : Allouer aux sections une zone mémoire appartenant à la mémoire physique du système cible, reloger les symboles et les sections en leur attribuant une adresse finale et résoudre les références externes entre les fichiers objets d'entrée.

Dans notre cas on a utilisé la SDRAM de la carte DSK6711 adressable par le DSP via CE0 et CE1 comme emplacement physique de la mémoire dynamique et des différentes sections du programme (voir Figure 37).

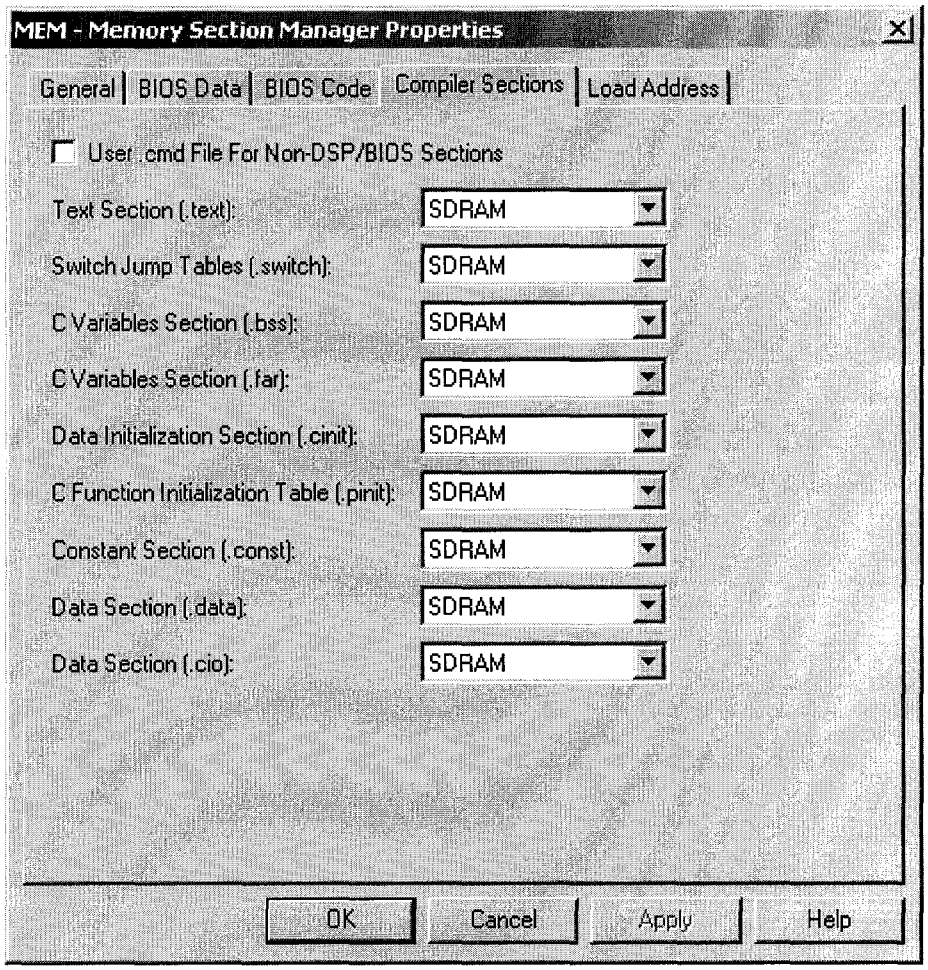

Figure 37 L'organisation de la mémoire via le gestionnaire de mémoire MEM

4.3.3.5 La configuration du projet

Après la création du projet vient l'étape de la configuration de celui-ci, elle permet de définir les options nécessaires au développement du programme, entre autres les options de compilation, et les options de l'éditeur de liens (linker), comme par exemple le niveau d'optimisation,...ect, elle se fait à partir du gestionnaire du projet via le ' Build Options.' (voir Figure 38).

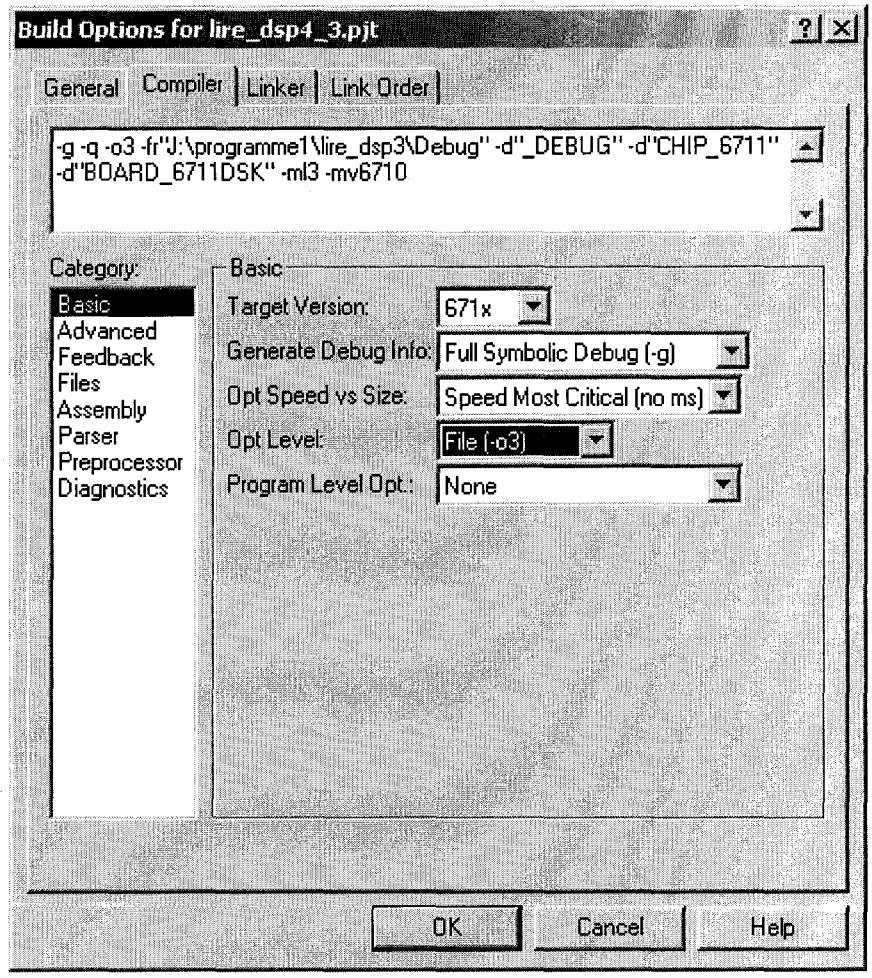

Figure 38 La configuration du projet

4.4 Méthodologie de l'implémentation

Cette application a été conçue de la manière suivante (Figure 39) : d'abord on réduit la fréquence d'échantillonnage du signal à 8 kHz. Ensuite le DSP est utilisé pour le traitement de ce signal, et la restitution du résultat de la reconnaissance en effectuant la comparaison du mot à reconnaître et les mots qui constituent le dictionnaire de référence. Ce dernier a été conçu à l'aide du logiciel Matlab.

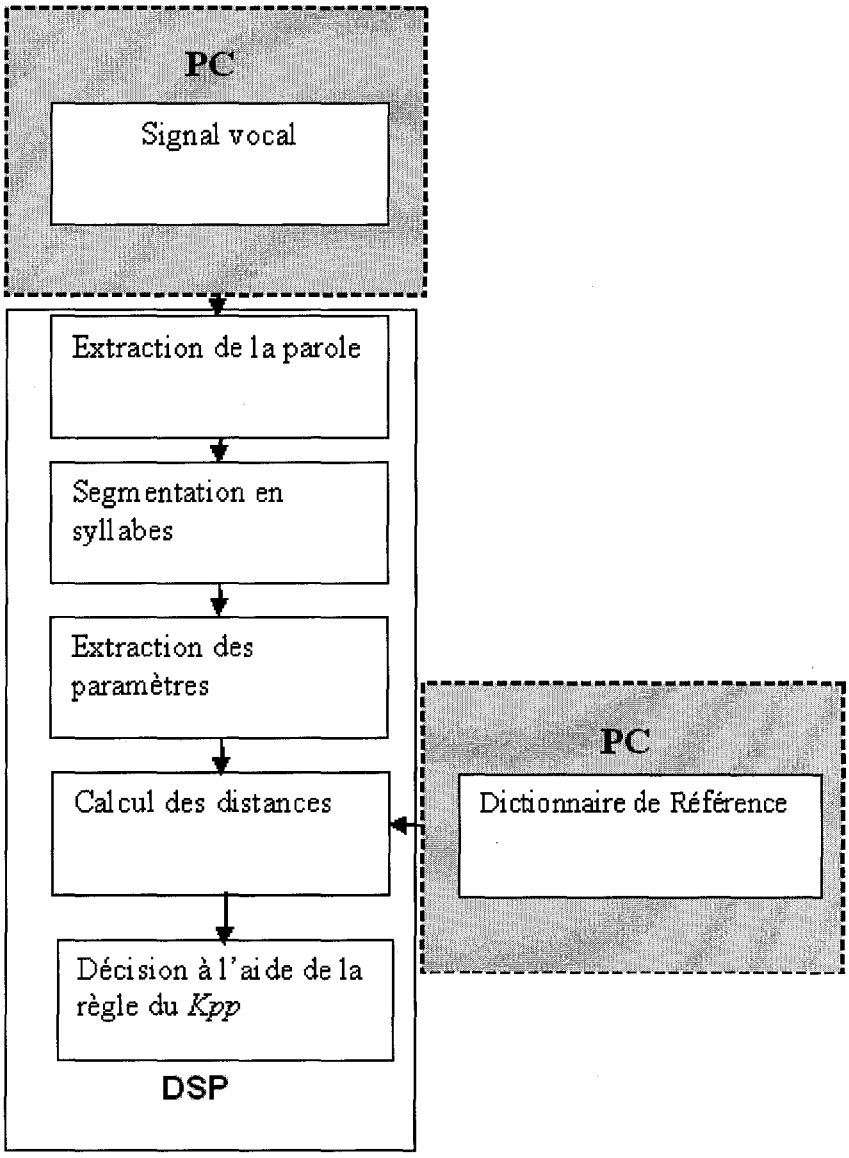

Figure 39 Méthodologie de l'implémentation

4.4.1 Description de la base de donnée utilisée

Pour tester les performances de notre système de reconnaissance, nous avons utilisé un corpus de 112 locuteurs :55 Hommes et 57 Femmes, à partir de La base de donnée

TIDIGITS [58]. Cette dernière est collectée par la compagnie Texas Instruments (TI) pour le développement et l'évaluation des systèmes de reconnaissance automatique des chiffres. Se sont des fichiers enregistrés dans un environnement sans bruit (salle acoustique RE-244B), le microphone (type RE-16 Dynamic cardioid) a été disposé de 2 à 4 pouces des lèvres, les signaux ont été numérisés à 20 kHz sur 16 bits et convertis en format NIST SPHERE, la production et la distribution de cette base de donnée a été réalisé par l'Institut National de Standards et Technologie (NIST).

L'organisation dans le CD-ROM respecte le format suivant:

/tidigits/<USAGE>/<LOCUTEUR>/<IDENTIFICATION>/<IDENTIFICATION>/<SEQUENCE><VERSION>.WAV

avec:

USAGE = entraînement ou test
LOCUTEUR = homme, femme.
IDENTIFICATION= le code de chaque locuteur (deux lettres)
SEQUENCE = le code de chaque chiffre –1,2, …,9, o(oh) et Z(zero)
VERSION = deux versions existent la version a ou la version b

4.4.2 Passage de 20 kHz à 8 kHz

La base de donnée utilisée (TDIGIT) a été conçue avec une fréquence d'échantillonnage de 20 kHz, pour la réduction de cette fréquence à 8 kHz, soit la fréquence d'échantillonnage de l'ADC de notre système à DSP, nous avons appliqué sur le signal

une interpolation suivi d'une décimation, deux opérations que nous allons voir le principe.

4.4.2.1 La décimation

La décimation (Figure 40) appelée aussi sous échantillonnage est une opération qui consiste à réduire la fréquence d'échantillonnage par un rapport entier k, pour la réaliser, il suffit de ne garder qu'un échantillon sur chaque k échantillons.

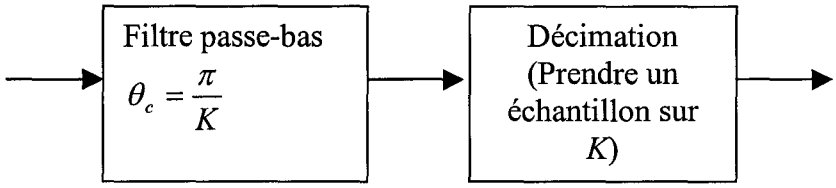

Figure 40 La décimation par un facteur K

En résumé:

un signal de fréquence f_s se retrouve après décimation par k à la fréquence $\frac{f_s}{k}$.

pour éviter tout repliement (aliasing), il faut filtrer passe-bas avant la décimation, avec une fréquence de coupure $F_c = \pi / k$

4.4.2.2 L'interpolation

L'interpolation par un facteur entier L (voir Figure 41), appelée aussi sur-échantillonnage, est l'opération qui consiste à multiplier la fréquence d'échantillonnage

f_s par L, elle peut se faire par l'insertion de L-1 valeurs nulles après chaque échantillon, puis d'un filtrage passe-bas pour supprimer les spectres dupliqués.

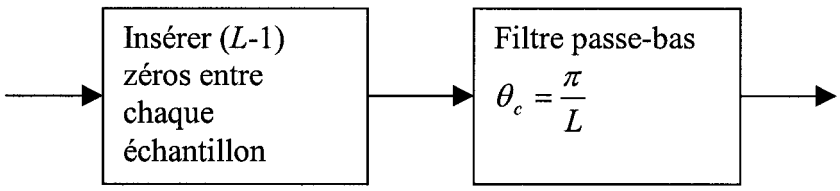

Figure 41 L'interpolation par un facteur L

En résumé:

Un signal de fréquence f_s se retrouve après interpolation par L à la fréquence $f_s.L$.

Pour éviter la duplication du spectre, il faut filtrer passe-bas après interpolation avec une fréquence de coupure $F_c = \pi / L$.

4.4.2.3 Changement de fréquence par un facteur fractionnaire de la forme *L/K*

Pour parvenir à un changement de fréquence d'échantillonnage f_s à $f_s' = f_s.L/K$, c'est à dire la multiplication de la fréquence d'échantillonnage par un facteur rationnel L/K, la solution consiste à faire à une interpolation par un facteur L, suivit d'une décimation par un facteur K, ces opérations sont illustrées par la Figure 42.

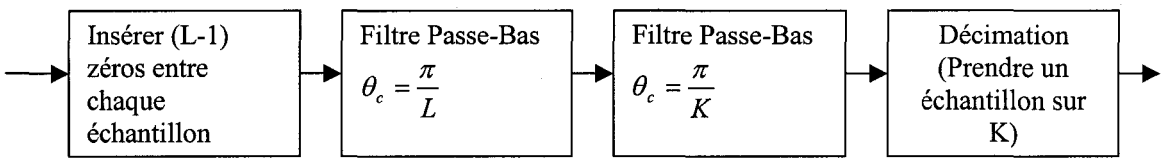

Figure 42 Réduction de la fréquence par un rapport L/K

En pratique les deux filtres passe-bas sont combinés en un seul filtre passe-bas, dont la fréquence de coupure est la plus basse des deux, le schéma deviendra alors (Figure 43):

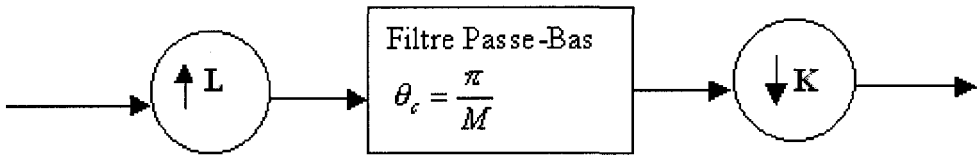

Figure 43 Schéma final de la réduction de fréquence

Où $M = \max(K, L)$, et F_c est légèrement inférieur à π/ M comme précaution, car les filtres en pratique ne sont pas idéals.

Application à notre cas :

Dans notre cas nous avons une fréquence d'échantillonnage de 20 kHz, qu'on veut réduire à 8 kHz, alors les paramètres utilisés seront:

- $K = 5 \quad L = 2$.

- Filtre passe-bas de largeur $\frac{\pi}{5}$

Pour le filtrage, on utilise un filtre FIR conçu avec la méthode des fenêtres, la fenêtre utilisée est une fenêtre de Hamming

4.4.3 L'acquisition du signal et restitution du résultat

Le système de développement CCS offre plusieurs possibilités pour l'échange des données de et vers le monde extérieur.
Principalement cet échange peut se faire de deux manières :

- via l'outil RTDX : dans ce cas les données à traiter sont obtenues à partir de dispositifs (microphone, appareil, ...etc.) reliés à la carte DSK.
- via un fichier de données : dans ce cas CCS permet de lire ou d'écrire les données selon l'un des deux formats suivants, le format COFF (Common Object File Format) ou le format binaire, soit le format du CCS qui est un fichier texte avec une ligne d'entête qui contient entre autres le type de données continues dans le fichier.

C'est l'acquisition de données via fichier que nous avons utilisé dans le cadre de ce projet, elle se fait à l'aide de sondes (probe points) à lesquelles les fichiers à lire ou écrire sont connectés. La lecture se fait par des pointeurs (voir Figure 44).

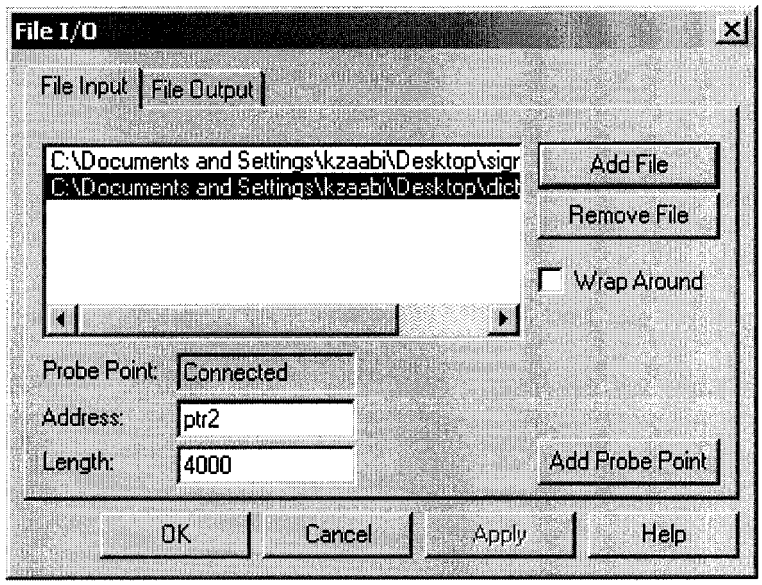

Figure 44 Lecture/Écriture des fichiers à l'aide de sondes

Après traitement du signal vocal, le résultat de la reconnaissance est affiché (voir Figure 45), dans cette figure apparaît le résultat de reconnaissance pour le fichier 'man\ae\8397261a.wav'.

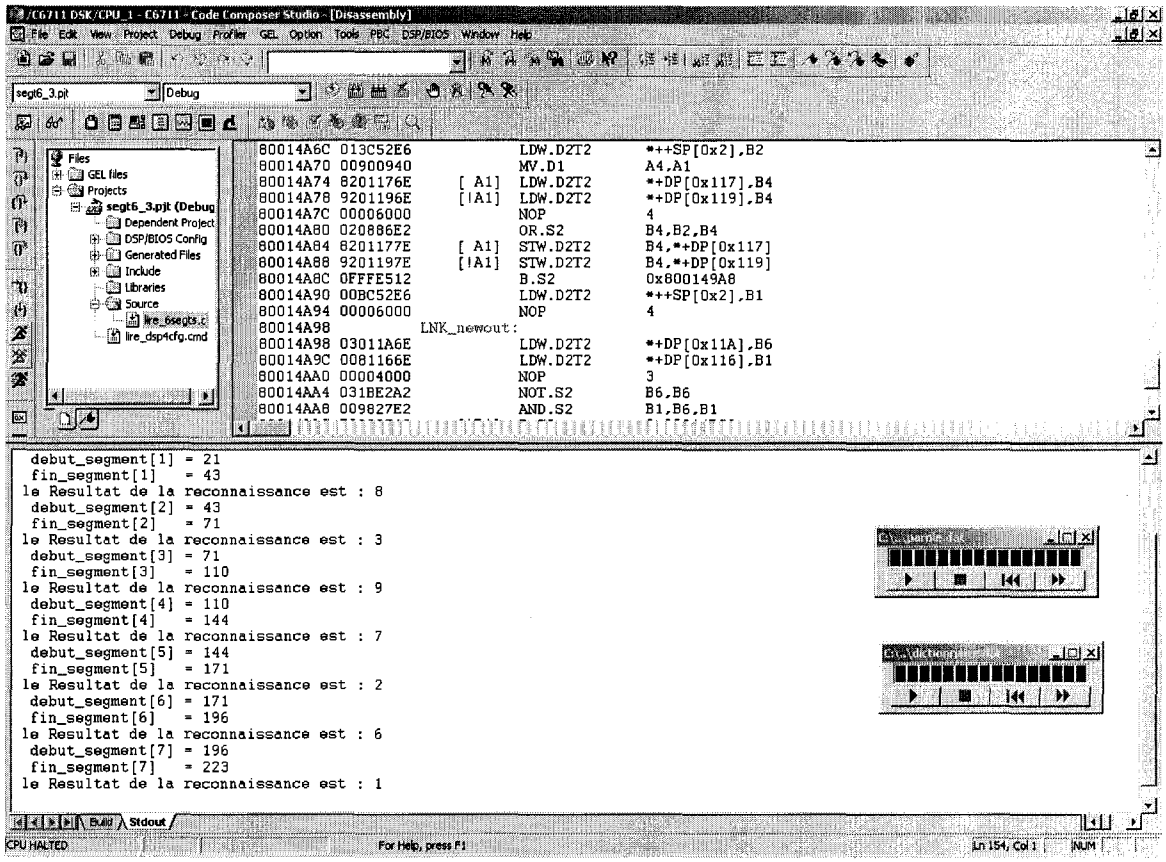

Figure 45 Affichage du résultat de reconnaissance après traitement

4.4.4 Résultats expérimentaux

4.4.4.1 Résultat de la détection début fin

Avec l'algorithme utilisé pour la détection du début et fin du signal, nous avons pu obtenir des résultas précis (voir Figure 46). Cependant nous avons remarqué que la valeur mentionnée par l'auteur [38]comme condition sur le taux de passage par zéro (supérieur ou égale à 3) n'est pas toujours efficace alors nous avons pris des valeurs supérieures.

Aussi nous avons apporté quelques modifications sur l'algorithme en effectuant la recherche de la fin du signal à partir du point début trouvé (N1) et non pas à partir de la fin du signal, ceci dans le but de supprimer le silence même à l'intérieur du signal (voir Figure 47).

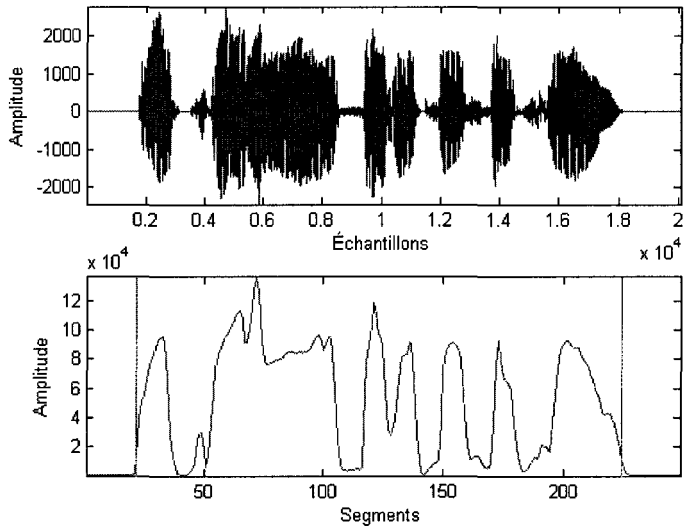

Figure 46 Détection des points début et fin

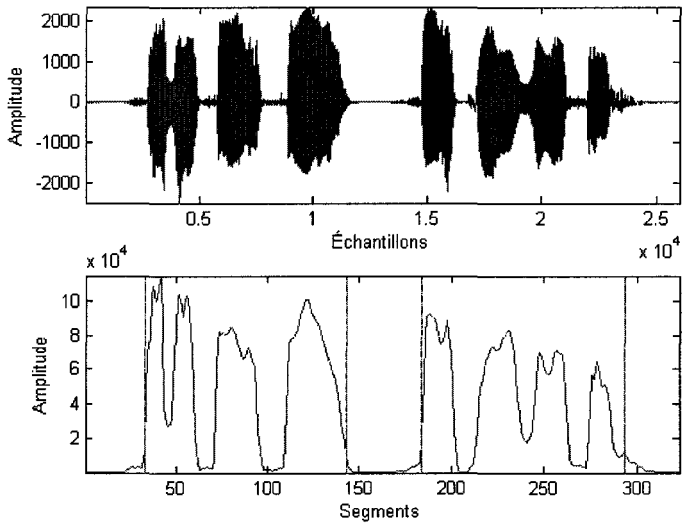

Figure 47 Suppression du silence à l'intérieur du signal

4.4.4.2 Résultat de la segmentation

La Figure 48 est une illustration de la segmentation, dans cette figure on remarque que le nombre de chiffres contenus dans le signal (sept chiffres), dépasse le nombre de segments trouvés après segmentation (soit neuf segments). Idéalement est d'avoir un segment pour chaque chiffre. On ajoute alors une étape de correction pour annexer les segments en plus aux segments qui les précèdent. Dans ce cas pour les chiffres "six "et "seven". Le résultat final est illustré par la Figure 49

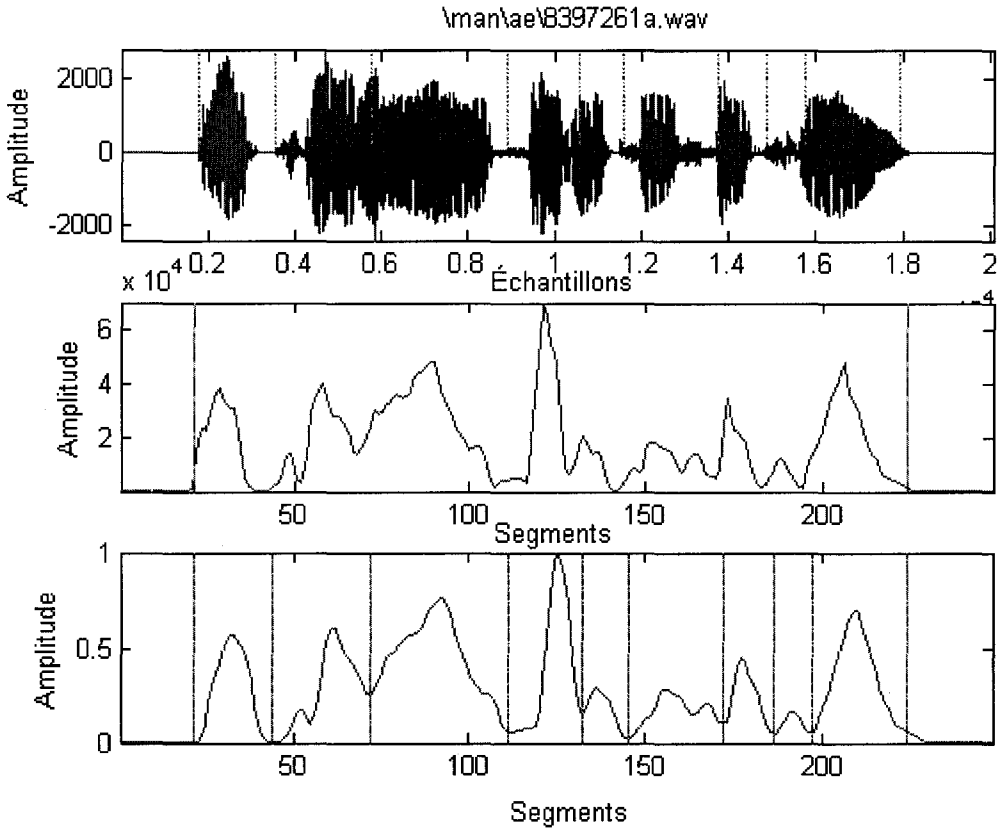

Figure 48 Segmentation en syllabes sans correction

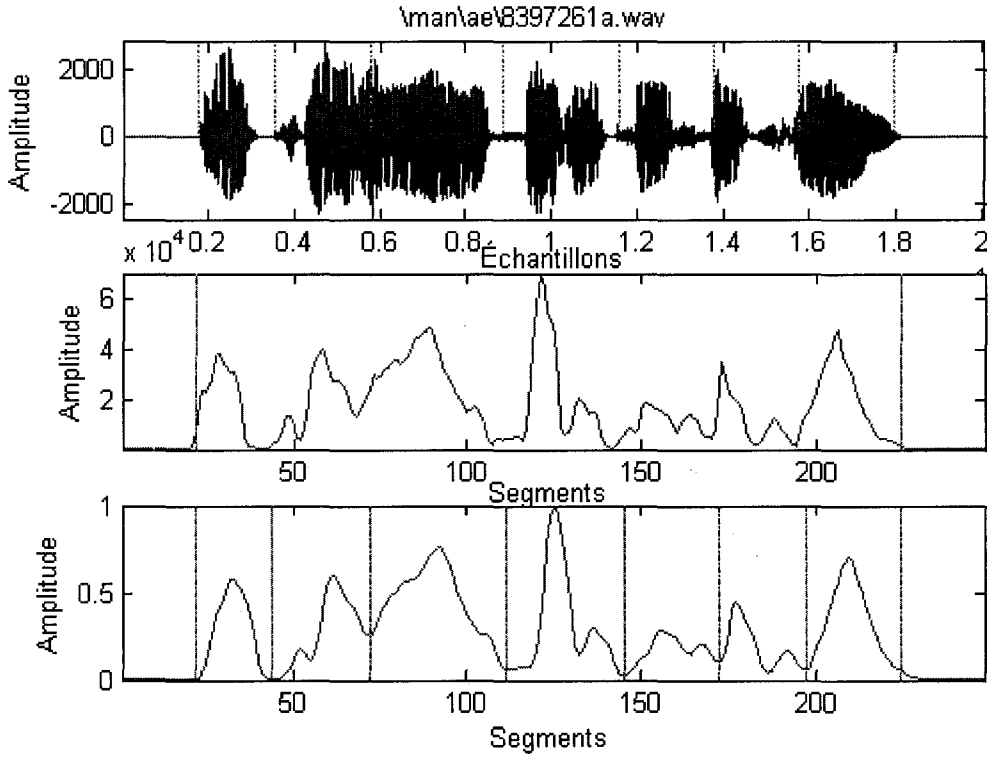

Figure 49 Segmentation en syllabes et correction

4.4.4.3 Résultats de reconnaissance

Le système de reconnaissance conçu a été testé sur des chiffres isolés (sans segmentation), et sur les chiffres connectés. Dans tous les cas l'apprentissage a été effectué sur 40% de l'ensemble des locuteurs et le test sur les 60% restants. Pour les chiffres isolés nous avons utilisé différents nombre de segments, cinq, six, huit et dix segments .Pour chaque segment 18 coefficients LPCCs dérivés à partir de 8 coefficients LPCs sont utilisés. Pour le premier test nous avons utilisé cinq segments (deux segments de chaque extrémité du chiffre et un segment de son milieu) Nous avons obtenu les résultats indiqués dans le tableau II.

Tableau II

Performance de reconnaissance en utilisant 5 segments

Locuteurs	Taux de Reconnaissance
Hommes + Femmes	96.7 %
Hommes	97.10 %
Femmes	97.24 %

Pour le deuxième test nous avons utilisé six segments (deux segments de chaque extrémité du chiffre et deux segments de son milieu) Nous avons obtenu les résultats indiqués dans le tableau III.

Tableau III

Performance de reconnaissance en utilisant 6 segments

Locuteurs	Taux de Reconnaissance
Hommes + Femmes	97.58 %
Hommes	98.20 %
Femmes	96.96 %

Pour le troisième test nous avons utilisé huit segments (deux segments de chaque extrémité du chiffre et quatre segments de son milieu) Nous avons obtenu les résultats indiqués dans le tableau IV.

Tableau IV

Performance de reconnaissance en utilisant 8 segments

Locuteurs	Taux de Reconnaissance
Hommes + Femmes	97.65 %
Hommes	98.48 %
Femmes	96.96 %

Pour le dernier test pour les chiffres isolés, nous avons utilisé tous les segments du chiffre (les dix segments). Nous avons obtenu les résultats indiqués dans le tableau V.

Tableau V

Performance de reconnaissance en utilisant 10 segments

Locuteurs	Taux de Reconnaissance
Hommes + Femmes	96.76 %
Hommes	98.07 %
Femmes	96.6 %

Pour les chiffres connectés nous avons effectué le test en utilisant huit segments de 18 coefficients LPCCs chacun (deux segments de chaque extrémité du chiffre et quatre segments de son milieu). Nous avons obtenu les résultats indiqués dans la table VI.

Tableau VI

Performance de reconnaissance des chiffres connectés en utilisant 6 segments

Locuteurs	Taux de Reconnaissance
Hommes + Femmes	87.6 %

4.5 Conclusion

Nous avons présenté dans ce dernier chapitre, la méthode de réalisation d'un système de reconnaissance vocale, c'est une méthode basée essentiellement sur le seuillage de l'énergie et l'utilisation de fenêtres de durée variable, le cœur de ce système est un processeur dédié au traitement numérique des signaux le TMS320C6711 de Texas Instrument, le système de développement utilisé avec ses deux composantes le Code Composer Studio et la carte DSK6711, a été présenté, tous les algorithmes ont été programmés en langage C avec l'ajout de quelques modifications pour l'adapter au Code Composer Studio.

Des chiffres isolés et connectés ont été utilisés pour tester les performances de ce système. Les résultats expérimentaux que nous avons obtenu montrent que cette méthode bien que simple ait permis des résultats remarquables pour les chiffres isolés. Pour les chiffres connectés, bien que l'idée soit attrayante il est clair qu'il reste un chemin à parcourir pour améliorer cette méthode et par conséquent les résultats. Car malgré une simplicité apparente, la tâche de la reconnaissance des chiffres connectés est d'une extrême complexité et ceci en raison du fait qu'elle dépend de plusieurs facteurs

que nous allons essayer de cerner dans la conclusion générale de ce mémoire, où nous présenterons aussi nos analyses concernant les éléments à améliorer.

CONCLUSION

Dans ce travail nous avons proposé un système de reconnaissance vocale indépendant du locuteur, notre objectif était la reconnaissance automatique d'un vocabulaire limité de mots (chiffres isolés et chiffres connectés), une procédure simple fondée sur la segmentation en syllabes et le seuillage d'énergie a été développée, dans un deuxième temps nous avons implémenté l'algorithme à l'aide du système de développement du DSP TMS320C6711.

Des tests de reconnaissance ont été effectués sur la base de données TIDIGITS de Texas Instrument, nous avons obtenu un taux de reconnaissance de 97.65 % pour les chiffres isolés et de 87.6 % pour les chiffres connectés. À partir des résultas obtenus, nous avons constaté une meilleure performance de cette méthode pour la reconnaissance des chiffres isolés que pour les chiffres connectés.

Les résultats de reconnaissance obtenus pour les chiffres isolés montre clairement qu'avec un seuillage d'énergie et avec juste quelques segments (six) judicieusement choisis (deux au début, deux au milieu et deux à la fin), nous avons presque égaler les performances des autres méthodes qui elles utilisent des algorithmes plus compliqués et plus coûteux en temps de calcul, comme la programmation dynamique (TDW), ou les modèles statistiques (MMCs). Avec quelques améliorations sur la classification, nous serons même en mesure d'atteindre un taux de reconnaissance de 100 %.

Les résultats expérimentaux de reconnaissance pour les chiffres connectés bien qu'encourageants, n'étaient pas aussi concluants que ceux obtenus pour les chiffres isolés, ceci à cause de plusieurs facteurs:

1. La segmentation

Une des difficultés majeures de la parole continue, est la segmentation d'un son en une suite de mots et ce que cela peut engendrer comme erreurs sur la reconnaissance car il n'est guère possible de détecter avec une totale sécurité les débuts et fins de ces mots, d'ailleurs c'est la raison pour laquelle on retrouve dans la littérature deux approches différentes pour la reconnaissance des chiffres connectés, la première est sans segmentation (segmentation-free approach) et la série de chiffres est apparier en entier avec des autres séries de chiffres. Dans la deuxième approche (classification-after-segmentation), l'appariement (matching) est précédé par une segmentation.

L'algorithme que nous avons utilisé effectue la segmentation en se basant sur les maximums et les minimums d'énergie, pour les hauts débits de locutions, cela se traduit parfois par des chiffres non segmentables, nous proposons alors le raffinement de cette méthode de segmentation par l'ajout d'autres critères pour la rendre plus puissante.

2. La classification

Pour la construction du dictionnaire des mots de référence nous avons utilisé seulement des chiffres isolés dans notre algorithme de classification. Cependant une série de chiffres connectés n'est pas forcément équivalente à une concaténation de ces mêmes chiffres pris isolément, alors procéder ainsi va forcément influer sur le taux de reconnaissance car ce dernier dépend largement de jusqu'à quel point un chiffre isolé est capable de représenter le même chiffre dans une série de chiffres, alors nous recommandons l'introduction d'autres méthodes telle que l'utilisation d'arbres pour la classification.

3. La vitesse du débit

Une des problématiques des systèmes de RAP actuels est la vitesse du débit d'élocution. En effet la plupart des systèmes ont tendance à performer mieux avec des faibles débits (100 à 130 mots par minute), et voient leurs performances se dégrader pour des débits

plus rapides (de l'ordre de 180 à 300 mots par minute), et ceci en raison entre autres du problème de la coarticulation. Même chose dans notre cas puisque notre système permet de reconnaître sans difficulté les chiffres connectés quand ils sont parfaitement prononcés et séparés par des silences.

Pour alléger l'impact de ce problème, et spécialement lorsque des chiffres isolés sont utilisés pour reconnaître des chiffres connectés, des idées ont été avancées suggérant par exemple le renforcement du dictionnaire des mots de référence par la combinaison des chiffres isolés avec des chiffres tirés à partir des chiffres connectés, en prenant des chiffres situés au milieu des séries de trois chiffres[43].

4. Le passage de 20 KHz à 8 KHz

Le système de développement utilisé exigeait une fréquence d'échantillonnage de 8 KHz, soit celle de l'ADC de la carte DSK6711, la base de données utilisée (TIDIGIT) étant conçue avec une fréquence d'échantillonnage de 20Khz, nous avons alors réalisé l'adaptation à l'aide d'une décimation suivie d'une interpolation, ce qui nous prive d'informations qu'aurait pu contribuer à améliorer le taux de reconnaissance. En effet la décimation se traduit par l'élimination d'échantillons, et l'interpolation par l'ajout de zéros entre deux échantillons. L'effet est plus visible pour les chiffres connectés que pour les chiffres isolés, ces derniers sont parfaitement prononcés.

Pour les perspectives futures, on peut envisager l'élargissement du vocabulaire à reconnaître par ce système avec l'ajout de quelques mots de commande, et dans ce cas la distinction entre chiffres et mots de commande peut se faire par un seuillage sur la durée de la parole. Aussi nous suggérons l'introduction de la méthode de l'ACP (Analyse en Composantes Principales) qui est un outil puissant pour la réduction du nombre des caractéristiques d'une forme, et ceci pour la sélection des segments les plus significatifs et qui représentent le mieux un mots.

Aussi on peut envisager de tester le système sur des données issues des conversations téléphoniques , et dans ce cas il faut prévoir l'ajout d'un module de débruitage, ainsi que l'utilisation d'autres types de paramètres comme par exemple les paramètres LSF (Line Spectral Frequencies), qui sont des paramètres très utilisés dans le domaine de la téléphonie.

BIBLIOGRAPHIE

[1] M. Ferretti et F. Cinare, Synthèse, Reconnaissance de la Parole, éditests,Paris, 1983.

[2] Calliope, la Parole et son Traitement Automatique, Masson, Paris , 1989.

[3] L. Rabiner and B. Juang, Fundamentals of Speech Recognition, Prentice Hall, 1993.

[4] S. E. Bou-Ghazale and A. O. Asadi, "Hands-free voice activation of personal communication devices,", ICASSP 2000, pp. 1735-1738, Istanbul, 2000.

[5] S. Oh, V. Viswanathan, and P. Papamichalis, "Hands-free voice communication in an automobile with a microphone array,", ICASSP 92, pp.281-284, San Francisco, 1992.

[6] G. Bendelac and I. D. Shallom, "Eyes free dialing for cellular telephones," presented at Vehicular Technology Conference, 1991. 'Gateway to the Future Technology in Motion', 41st IEEE, 1991.

[7] L. R. Rabiner, J. G. Wilpon, and F. K. Soong, "High performance connected digit recognition using hidden Markov models," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 37, pp. 1214-1225, 1989.

[8] J. G. Wilpon, C.-H. Lee, and L. R. Rabiner, "Improvements in connected digit recognition using higher order spectral and energy features," , ICASSP 91,pp.349-352, Toronto, 1991.

[9] C. Myers and L. Rabiner, "Connected digit recognition using a level-building DTW algorithm," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 29, pp. 351-363, 1981.

[10] M. A. Franzini, M. J. Witbrock, and K.-F. Lee, "Speaker-independent recognition of connected utterances using recurrent and non-recurrent neural networks,", IJCNN 89, pp 1-6, Washington, 1989.

[11] K. Zaabi et M. Gabrea, " Reconnaissance robuste de chiffres enchaînés avec des ressources limitées," IEEE CCECE 2003, vol. 2, pp 1155-1158, Montréal,

[12] Robert Paul, Le petit Robert, Paris, 1976.

[13] T. Dutoit, Introduction au traitement automatique de la parole, Faculté polytechnique de Mons, Belgique, 2000.

[14] R. Boite et M. Kunt, Traitement de la parole, Presses Polytechniques Romandes, Lausanne,1987.

[15] T. Parsons, Voice and Speech Processing, McGraw-Hill, 1986.

[16] C.-S. Gargour, Traitement numérique des signaux, École de technologie supérieure, 2001.

[17] T.-F. Quatieri, Discrete-Time Speech Signal Processing Principles and Practice, Prentice Hall PTR, 2001.

[18] L. Rabiner and R. W. Schafer, Digital Processing of Speech Signals, Prentice-Hall, 1978.

[19] J. Makhoul, "Linear prediction : A tutorial review," Proceeding of the IEEE, vol. 3, pp. 561-580, 1975.

[20] E. Wong and S. Sridharan, "Comparison of linear prediction cepstrum coefficients and mel-frequency cepstrum coefficients for language identification," presented at Intelligent Multimedia, Video and Speech Processing, 2001. Proceedings of 2001 International Symposium on, 2001.

[21] J.-C. Junqua and J.-P. Haton, Robustness in Automatic Speech Recognition, Kluwer Academic Publishers, 1996.

[22] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," IEEE Transactions on, Acoustics, Speech, and Signal Processing, vol. 28, pp. 357-366, 1980.

[23] Y. Tohkura, "A weighted cepstral distance measure for speech recognition," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 35, pp. 1414-1422, 1987.

[24] B. Hanson and H. Wakita, "Spectral slope distance measures with linear prediction analysis for word recognition in noise," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 35, pp. 968-973, 1987.

[25] M. Cheriet, Reconnaissance des Formes et Inspection, notes de cours, vol. 1, École de Technologie supérieure, 1997.

[26] B. Juang, L. Rabiner, and J. Wilpon, "On the use of bandpass liftering in speech recognition," ICASSP 86, pp. 765-768, 1986.

[27] L. Rabiner, S. Levinson, A. Rosenberg, and J. Wilpon, "Speaker-independent recognition of isolated words using clustering techniques," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 27, pp. 336-349, 1979.

[28] S. Levinson, L. Rabiner, A. Rosenberg, and J. Wilpon, "Interactive clustering techniques for selecting speaker-independent reference templates for isolated word recognition," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 27, pp. 134-141, 1979.

[29] L. Rabiner and J. Wilpon, "Considerations in applying clustering techniques to speaker independent word recognition,", ICASSP '79, pp.578-581, New Jersey, 1979.

[30] J. Wilpon and L. Rabiner, "A modified K-means clustering algorithm for use in isolated work recognition," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 33, pp. 587-594, 1985.

[31] S. Easwaran and J. N. Gowdy, "An improved initialization algorithm for use with the K-means algorithm for code book generation," Southeastcon '92, pp. 471-474, Birmingham, 1992.

[32] Y. Linde, A. Buzo, and R. Gray, "An Algorithm for Vector Quantizer Design," IEEE Transactions on Communications, vol. 28, pp. 84-95, 1980.

[33] D. Lee, S. Baek, and K. Sung, "Modified K-means algorithm for vector quantizer design," Signal Processing Letters, IEEE, vol. 4, pp. 2-4, 1997.

[34] I. Katsavounidis, C.-C. Jay Kuo, and Z. Zhang, "A new initialization technique for generalized Lloyd iteration," Signal Processing Letters, IEEE, vol. 1, pp. 144-146, 1994.

[35] S. G. Tanyer and H. Ozer, "Voice activity detection in nonstationary noise," IEEE Transactions on Speech and Audio Processing, vol. 8, pp. 478-482, 2000.

[36] T. Ghiselli-Crippa and A. El-Jaroudi, "Voiced-unvoiced-silence classification of speech using neural nets,", IJCNN-91, pp.851-856, Seattle,1991.

[37] Y. Qi and B. R. Hunt, "Voiced-unvoiced-silence classifications of speech using hybrid features and a network classifier," IEEE Transactions on Speech and Audio Processing vol. 1, pp. 250-255, 1993.

[38] L. Rabiner and M. R. Sambur, "An Algorithm for Determinig the Endpoints of Isolated Utterances," The Bell System Technical Journal, vol. 54, pp. 297-315, 1975.

[39] A. Hussain, S. A. Samad, and L. B. Fah, "Endpoint detection of speech signal using neural network,", TENCON 2000, pp.271-274, Kuala Lumpur, 2000.

[40] S. E. Bou-Ghazale and K. Assaleh, "A robust endpoint detection of speech for noisy environments with application to automatic speech recognition,", ICASSP '02, pp. IV-3808 - IV-3811, New Port Beach,2002.

[41] P. Mermelstein, "Automatic Segmentation of Speech into Syllabic Units," *JASA*, vol. 58, pp. 880-883, 1975.

[42] J. Markel and A. Gray, Jr., Linear prediction of speech. New York: Springer-Verlag, 1976.

[43] L. Rabiner, J. Wilpon, A. Quinn, and S. Terrace, "On the application of embedded digit training to speaker independent connected digit recognition," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 32, pp. 272-280, 1984.

[44] L. Rabiner, J. Wilpon, and B. Juang, "A continuous training procedure for connected digit recognition,", ICASSP '86, ,1986.

[45] L. Rabiner, "On creating reference templates for speaker independent recognition of isolated words," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 26, pp. 34-42, 1978.

[46] R. Chassing, DSP Applications Using C and the TMS320C6x DSK, Wiley, 2002.

[47] TMS320C6000 Programmer's Guide, SPRU198D, Texas Instruments, Dallas, 2000.

[48] TMS320C6000 Assembly Language Tools User's Guide, *SPRU186I,* Texas Instruments, Dallas, 2001.

[49] TMS320C6000DSK Board Support Library API User's guide, SPRU432A, Texas Instruments, Dallas, 2001.

[50] TMS320C6x Csource Debugger User's guide, SPRU 188D,Texas Instruments, Dallas, 1998.

[51] TMS320C6000 Chip Support Library API User's Guide, SPRU401D, Texas Instruments, Dallas, 2002.

[52] TMS320C6000 CPU and Instruction Set Reference Guide, *SPRU189F*, Texas Instruments, Dallas, 2000.
[53] TMS320C6000 Instruction Set Simulator User's guide, *SPRU546*, Texas Instruments, Dallas, 2001.
[54] TMS320C6000 Peripherals Reference Guide, SPRU190D, Texas Instruments, Dallas, 2001.
[55] TMS320C6000 Optimizing Compiler User's Guide, *SPRU187G*, Texas Instruments, Dallas, 2000.
[56] Code Composer Studio User's Guide, SPRU509, Texas Instruments, Dallas, 2001.
[57] Code Composer Studio Tutorial, SPRU301C, Texas Instruments, Dallas, 2000.
[58] R. Leonard, "A database for speaker-independent digit recognition,",ICASSP '84, pp.328-331,1984.