**République Algérienne Démocratique et Populaire**

**Ministère de L'Enseignement Supérieur et de la Recherche Scientifique**


**UNIVERSITE MOHAMED BOUDIAF-M'SILA**

**FACULTE DE TECHNOLOGIE**

**DEPARTEMENT DE L'ELECTRONIQUE**

***MEMOIRE DE MASTER***

**DOMAINE : sciences et technologies**

**FILIERE : Electronique.**

**OPTION : STN**

## Thème :

# REALISATION D'UNE APPLICATION EN VUE DE LA RECONNAISSANCE AUTOMATIQUE DE LA PAROLE TRANSCODEE SPEEX ET G.729

**Présenté par :**

Bachiri Salah Eddine.

**Encadré par :**

Dr.Yessad.D

**Promotion : JUIN 2016**

# Remerciement

*Tout d'abord on remercie le bon dieu puissant de la bonne santé, la volonté et de la patience qu'il nous a donnée tout au long de notre étude.*

*Avant de commencer la présentation de ce travail, je profite de l'occasion pour remercier toutes les personnes qui ont contribué de près ou de loin à la réalisation de ce projet de fin d'études.*

*Nous remercions Très sincèrement Mme.Yessad.D notre encadreure de ce travail, pour ses conseils pertinents, et ses orientations judicieuses sa patience et diligence, et par ses suggestions a grandement facilité ce travail.*

*Nous tenons à exprimer notre gratitude aux membres de jury qui ont bien voulu examiner ce travail.*

*Nos remerciements vont aussi à tous les enseignants du département d'électronique qui ont contribués à notre formation.*

*Enfin nous tenons à exprimer notre reconnaissance à tous nos amis et collègues pour leur le soutien moral et matériel.*

بسم الله الرحمن الرحيم

Au nom de dieu clément et miséricordieux

# Dédicaces 

*Avant tout, je tiens à remercier le bon dieu, et l'unique qui m'offre le courage et la volonté nécessaire pour affronter les différentes difficultés de la vie,*

*Je dédie ce modeste travail :*

*A ceux qui sont les plus chers du monde, Mon père, et ma mère, à qui je n'arriverai jamais à exprimer ma gratitude et ma reconnaissance, pour ses amours ses soutiens tout au long de mes études.*

*A mes frères Djamil ,Omar ,Samr, Halim ,Mohamed , et mes sœurs Nasima,Wafa et « Aya » . A toute ma famille.*

*A mes amis et collègue Hako, Hicham, Walid, Ibrahim , Nacer, Lehbib ,Fares ,Abdou .*

# Sommaire

**CHAPITRE III : Modèles de Markov Cachés.**



**CHAPITRE IV : Expériences et résultats .**

# Liste des Tableaux

# Liste Des Figures

# Introduction générale

Les systèmes de communications actuels évoluent de plus en plus vers des réseaux alliant la voix, la vidéo et les données. A ce titre la VoIP (Voice over Internet Protocol) connait un essor fulgurant et est en passe de gagner des parts de marché importantes au détriment de la téléphonie fixe. Si actuellement le codec G.711 à 64 Kbits/s reste encore utilisé en raison de l'intelligibilité de la parole à ce débit, il sera supplanté par des codecs nécessitant un moindre débit, en particulier le G729 dédié à la VoIP.

Par ailleurs, les applications en reconnaissance vocale se tournent actuellement vers la reconnaissance distribuée (Distributed speech recognition and Network speech recognition) qui empruntent nécessairement des réseaux de communications pour des applications distantes, avec souvent des architectures clients serveur. L'utilisation d'un codec, et par conséquent le transcodage de la parole est une opération essentielle dans ce processus.

Notre étude consiste à évaluer l'influence des codeur G.729 et Speex sur la reconnaissance automatique de la parole, puis la réalisation d'une interface graphique vue de la RAP transcodée. Pour cela, nous avons utilisé la plateforme open source HTK (Hidden ToolKit) basée sur une le HMM (Hidden Markov Models). L'application a porté sur reconnaissance de mots isolés (Chiffres arabes de la base de données ARADIGIT) .

Dans le premier chapitre, nous décrivons l'ensemble du système vocal chez l'être humain, ainsi que les facteurs qui entrent en jeu lors de la production de la parole et les différentes étapes de prétraitement du signal parole, ainsi que les méthodes et les techniques de la reconnaissance de la parole et son modélisation, les outils d'analyse de la parole.

Le second chapitre est consacré pour l'étude des distincts codecs exploités sur la VoIP comme le G.729 et Speex.

Dans le troisième chapitre nous présentons la méthode de modélisation HMM (principe de modélisation HMM pour la parole, les algorithmes utilisés, etc.).

Tandis qu'au quatrième chapitre nous traitons la mise en œuvre et l'implémentation de la reconnaissance avec la plateforme HTK en utilisant la base de données ARADIGIT transcodée Speex et G.729.

Le cinquième chapitre est consacré pour la représentation de notre application développée en vue de la RAP transcodée Speex et G.729.

Enfin nous terminerons ce mémoire par une conclusion qui résume les résultats obtenus au cours de notre travail.

# CHAPITRE I :

# Reconnaissance automatique de la parole

## I.1 Introduction

La parole est, depuis toujours, le moyen privilégié de communication de l'Homme. Avec la révolution des machines et notamment des ordinateurs, chercheurs et industriels se sont efforcés d'étendre l'usage de la parole à la communication homme-machine. Si les systèmes actuels basés sur le langage naturel sont loin d'assimiler toutes les finesses d'une langue, le traitement automatique de la parole a considérablement progressé ces dix dernières années, notamment dans les domaines de la reconnaissance automatique de la parole (RAP) et de la synthèse.

Dans ce chapitre, nous nous intéresserons aux bases de la reconnaissance automatique de la parole (RAP) et nous verrons quels sont les fondements théoriques des différents algorithmes utilisés, les différentes caractéristiques d'un système de reconnaissance de la parole, la structure générale de ce dernier, les méthodes d'analyse du signal pour une paramétrisation efficiente et enfin les approches de reconnaissance en insistant sur celles les plus utilisées actuellement.

## I.2 Mécanismes de production de la parole et perception des sons

Pour développer un système de reconnaissance automatique du la parole, il est nécessaire de connaitre les paramètres acoustiques caractérisant la parole humain. Pour cela, une bonne compréhension du processus de production de la parole est nécessaire. La parole est considérée parmi les principaux moyens de communication de l'être humain. Elle contient essentiellement le sens du message prononcée par le locuteur, ainsi que des informations individuelles concernant l'identité et parfois l'émotion du locuteur.

### I.2.1 Production de la parole

Le processus de production de la parole est un mécanisme très complexe qui repose sur une interaction entre les systèmes neurologique et physiologique. La parole commence par une activité neurologique [1]. Après que soient survenues l'idée et la volonté de parler, le cerveau dirige les opérations relatives à la mise en action des organes phonatoires. Le fonctionnement de ces organes est bien, quant à lui, de nature physiologique [2].

Une grande quantité d'organes et de muscles entrent en jeu dans la production des sons des langues naturelles. Le fonctionnement de l'appareil phonatoire humain repose sur l'interaction entre trois entités: les poumons, le larynx et le conduit vocal (voir figure I.1).

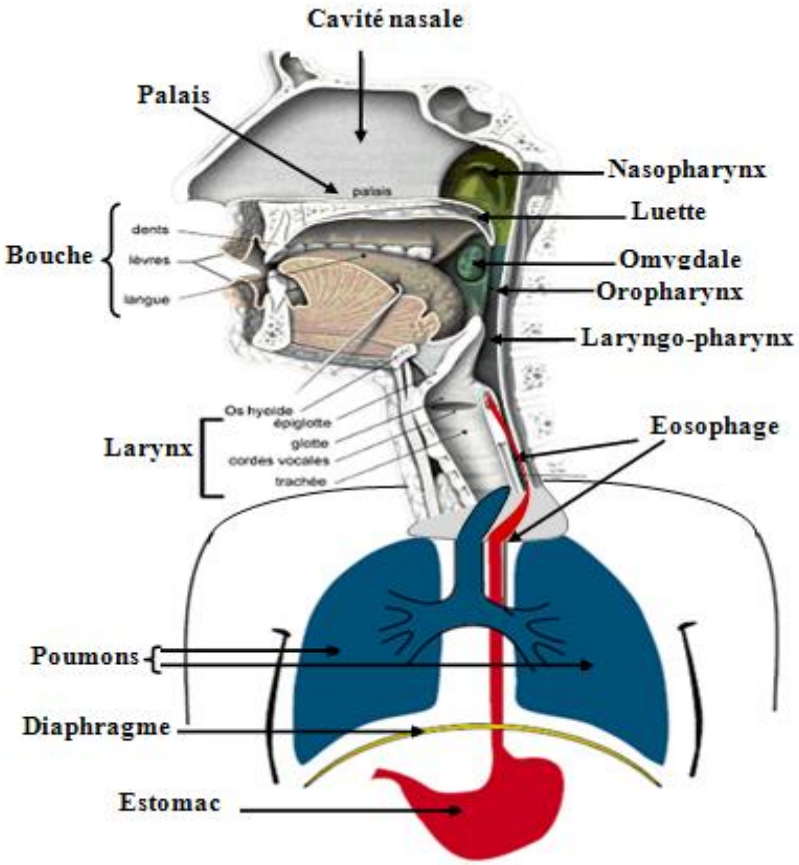



**Figure I.1:** Organes de production de la parole [3].

Le larynx est une structure cartilagineuse qui a notamment comme fonction de réguler le débit d'air via le mouvement des cordes vocales. Le conduit vocal s'étend des cordes vocales jusqu'aux lèvres dans sa partie buccale, et jusqu'aux narines dans sa partie nasale [2].

L'air des poumons est comprimé par l'action du diaphragme. Cet air sous pression arrive ensuite au niveau des cordes vocales. Si les cordes sont écartées, l'air passe librement et permet la production de bruit. Si elles sont fermées, la pression peut les mettre en vibration et l'on obtient un son quasi-périodique, dont la fréquence fondamentale correspond généralement à la hauteur de la voix perçue. L'air mis ou non en vibration poursuit son chemin à travers le conduit vocal et se propage ensuite dans l'atmosphère. La forme de ce conduit, déterminée par la position des articulateurs tels que la langue, la mâchoire, les lèvres ou le voile du palais, détermine les particularités des différents sons de la parole. Le conduit vocal est ainsi considéré comme un filtre pour les différentes sources de production de la parole telles que les vibrations des cordes vocales ou les turbulences engendrées par le passage de l'air à travers les constrictions du conduit vocal [2], [3]. Le son résultant peut être classé comme voisé ou non voisé, selon que l'air émis a fait vibrer les cordes vocales ou non [2], [3], [4].

### I.2.2 Perception de la parole dans la bande téléphonique

L'oreille humaine ne peut percevoir que certains sons, le niveau d'intensité acoustique moyenne produit par un son de parole, mesurée à 1 mètre, est compris entre 30 et 110 dB, ce qui correspond respectivement à une voix chuchotée et à une voix criée [1]. L'étendue spectrale de la parole humaine est comprise en général entre 80 et 8000 Hz (certaines cantatrices peuvent atteindre 15 kHz dans le chant d'opéra), la puissance sonore de la parole dans une conversation normale se situe de 60 à 70 dB et le niveau d'intensité acoustique est maximal lorsque la fréquence se situe aux alentours de 500Hz [5]. La figure I.2 donne une représentation du domaine audible pour un être humain. On remarque tout d'abord que le niveau de perception dépend grandement de la plage de fréquences considérée ainsi que du niveau sonore. On définit alors deux courbes dans le plan fréquence-intensité : un seuil d'audibilité et un seuil de confort. La zone ainsi définie est le domaine dans lequel les sons peuvent être perçus. Tout signal en dehors de cette plage est inaudible, gênant ou même dangereux.

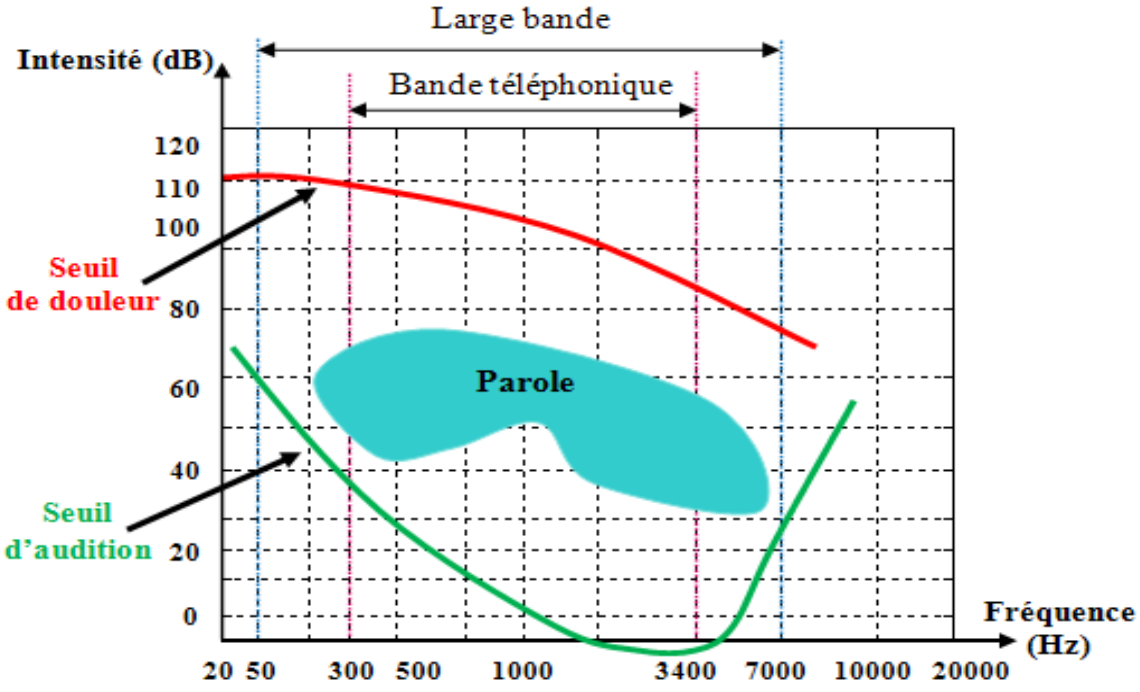



**Figure I.2:** Perception auditif [3].

La bande d'audition est composée des fréquences audibles par une oreille humaine situées entre 20 Hz et 20 kHz : ce qui à première vu semble très loin de la bande étroite de la téléphonie correspond à la plage de fréquence 300-3400 Hz, cette bande téléphonique est juste suffisante pour conserver l'intelligibilité du langage ainsi que les paramètres propres au

locuteur (voix, émotion, etc.). En pratique la plage de 100-4000 Hz permet de positionner correctement les premiers formants (trois ou quatre) et de contenir toutes les premières harmoniques du pitch, qui sont les informations utiles à la bonne compréhension de la parole humaine [1]. Cependant, l'intelligibilité de parole est intégralement contenue dans la bande téléphonique, sauf peut-être pour quelques harmoniques où la fréquence de pitch est relativement faible (allant de 50 Hz pour les hommes à 600 Hz pour les voix les plus aiguës telle une voix d'enfant) [4]. Ainsi l'addition des fréquences inférieures à 300 Hz donne une meilleure représentation des premières harmoniques (phonèmes voisé). On remarque surtout cela pour un locuteur masculin pour lequel la fréquence de pitch est assez faible. La plage des hautes fréquences, supérieures à 3400 Hz n'a de l'importance que pour les phonèmes complexes [4].

La plage de la bande téléphonique, où le système auditif est le plus sensible, justifie le taux d'échantillonnage de 8 kHz pour les codecs de parole qui traitent en général le signal sur la bande étroite. Bien que la bande téléphonique soit suffisante pour la bonne compréhension du message, l'élargissement de la bande passante des codecs de 50 à 7000 Hz (large bande) permet de rendre la parole reconstituée plus naturelle avec une qualité se rapprochant d'une communication face à face [1], [6]. Les fréquences inférieures à 50 Hz et supérieures à 7000 Hz n'apportent pas plus d'information à la perception de la parole, surtout sur les phonèmes, l'unité principale constituant le langage parlé [7].

En conclusion, la parole humaine produit des fréquences qui, en partie, ne sont pas comprises dans la bande téléphonique, et qui sont nécessaires pour obtenir une voix humaine naturelle. Ce qui complique le processus de reconnaissance de la parole à travers un canal téléphonique ou un réseau IP.

### I.3 Variabilité du signal de la parole

Le signal de parole est très complexe où se mêlent informations linguistiques, informations caractéristiques du locuteur, informations relatives au matériel utilisé pour la transmission ou l'enregistrement du signal, etc. En outre, le signal de parole est très redondant. Cette caractéristique est d'ailleurs reconnue pour faciliter la communication entre deux personnes dans un environnement très bruyant. Par ces différents aspects, le signal de parole présente une très grande variabilité.

La capacité des systèmes de reconnaissance de locuteur à différencier plusieurs individus repose essentiellement sur la variabilité inter-locuteur : la disposition du signal de parole à varier entre différents individus. Néanmoins, le signal de parole renferme d'autres

types de variabilité qui rendent problématique la tâche de reconnaissance, telles que la variabilité intra-locuteur ou la variabilité dûe au matériel.

### I.3.1 Variabilité intra-locuteur

La variabilité intra-locuteur est une variabilité propre au locuteur qui ne peut pas reproduire exactement le même signal. Cette variabilité intra-locuteur est dépendante de état physique et psychologique pour un même individu, ainsi les facteurs de variabilités sont multiples, on cite:

- L'état pathologique : Des variations peuvent être induites involontairement sur la voix d'une personne, de type fatigue, rhume et stress, etc., ou les variations émotionnels. Ces facteurs provoquent des altérations momentanées dans la voix. Dans ce sens, la voix peut changer entre le début et la fin de la journée [8], [9].Plus généralement, il est impossible pour une personne de répéter à l'identique le même signal de parole deux fois de suite. Une légère variation est toujours observée.
- Dans le cas d'une interaction volontaire et consciente avec un système de reconnaissance, comme par exemple dans le cadre d'un accès sécurise, le comportement d'un individu se modifie au fur et à mesure de son utilisation du système. L'individu devient de plus en plus confiant ainsi sa voix évolue dans ce sens et s'en trouve modifiée.
- Enfin, à plus long terme, la voix change au fur et à mesure du vieillissement d'une personne.

L'influence de ces divers facteurs varie selon l'application visée. Des travaux ont montré l'importance des variations à long terme et que les performances d'un système se dégradent en augmentant le temps qui sépare les sessions des références et les tests [10], [11]. Plus ce temps augmente, plus les performances se dégradent. Néanmoins, même les variations à court terme (émotion, état pathologique) peuvent être très préjudiciables aux systèmes de la reconnaissance.

### I.3.2 Variabilité inter-locuteurs

Les signaux de parole véhiculent plusieurs types d'informations. Parmi eux, la signification du message prononcé est d'importance primordiale. Cependant, d'autres informations telles que le style d'élocution ou l'identité du locuteur jouent un rôle important dans la communication orale. Ecouter un interlocuteur permet d'avoir des indications

concernant son sexe, son état émotionnel et bien souvent de l'identifier si on l'a déjà entendu. Dans notre vie quotidienne, ces informations, sont très utiles. Elles nous permettent, par exemple, de différencier les divers messages que nous entendons selon le locuteur et leur degré d'importance. Si toutes les voix étaient perçues de la même façon, il serait par exemple impossible de suivre une émission radio faisant participer des personnes différentes.

La grande variabilité entre les locuteurs est due, d'une part, à l'héritage linguistique et au milieu socioculturel de l'individu, et d'autre part aux différences physiologiques des organes responsables de la production vocale. L'expression acoustique de ces différences peut être traduite par une variation de la fréquence fondamentale, dans l'échelle des formants (plus haute chez les femmes et les enfants que chez les hommes) et dans le timbre de la voix (richesse en harmoniques due à la morphologie du locuteur et au mode de fermeture des cordes vocales).

### I.3.3 Variabilité due au matériel

La transmission du signal de parole au système de reconnaissance chargé de l'analyser nécessite plusieurs étapes et emprunte divers types de supports. A chacune de ces étapes, le media utilisé (ex : microphone, combiné téléphonique) pour transporter ce signal y imprime sa marque. Ces empreintes apparaissent le plus souvent sous la forme de déformations/dégradations du signal de parole. Ces déformations sont différentes selon le type de matériel utilisé.

## I.4 Système de reconnaissance du parole RAP

On peut décomposer la structure générale d'un système de reconnaissance automatique de la parole en trois modules :

- Un module d'analyse acoustique, dont la fonction est de transformer le signal brut en une suite de vecteurs de coefficients acoustiques;
- Un module de modélisation, utilisé pendant la phase d'apprentissage pour extraire un modèle de la distribution des vecteurs acoustiques obtenus à partir d'un ou plusieurs énoncés;
- Un module de décision, dont le rôle est d'évaluer la similarité entre un énoncé de test et le modèle de référence.

### I.4.1 Analyse acoustique

La paramétrisation du signal de parole consiste en l'extraction d'un ensemble de vecteurs acoustiques. Le but de cette opération est d'obtenir une nouvelle représentation qui est plus compacte et plus appropriée à la modélisation statistique. Les paramètres les plus utilisés dans les systèmes de reconnaissance de la parole reposent sur une représentation cepstral du signal de parole.

Avant l'extraction des paramètres, un prétraitement qui consiste à détecter les zones de silence est effectué afin de n'utiliser que les zones d'activité acoustique. Cette opérationest très difficile à mener à cause de la présence de bruit qui change les caractéristiques du signal de parole. Les techniques les plus utilisées pour résoudre se problème se basent sur le taux de passage par zéros et l'amplitude moyenne des trames courtes. Cette technique est une version modifié de l'algorithme de Rabiner et Sambur [12]. L'amplitude moyenne Am et le taux de passage par zéros TPZ sont donnés par :

$$\mathrm{Am} = \frac{1}{\mathrm{T}}\sum_{\mathrm{t}=1}^{\mathrm{N}}|\mathrm{X}(\mathrm{t})| \tag{I.1}$$

$$\mathrm{TPZ} = \sum_{\mathrm{i}=0}^{\mathrm{l}}|\mathrm{sgn}[\mathrm{X}(\mathrm{t}+\mathrm{i}+1)] - \mathrm{sgn}[\mathrm{X}(\mathrm{t}+\mathrm{i})]| \tag{I.2}$$

La description de cet algorithme est comme suit:

- Découper le signal de parole en plusieurs trames non chevauchantes;
- Calculer l'amplitude moyenne est le taux de passage par zéros de chaque trame suivant les équations données par (I.1 et I.2);
- Si l'amplitude moyenne d'une trame Am est supérieure à un seuil maximal SMX, la trame est considérée une trame de signal de parole.
- Si l'amplitude moyenne d'une trame Am est inférieur à SMX et supérieur à SMN, et l'amplitude moyenne de la trame précédent est supérieur à SMX, la trame est considérée une trame de signal de parole;
- Si l'amplitude moyenne d'une trame Am est inférieur à SMX et supérieur à SMN, l'amplitude moyenne de la trame précédent est inférieur à SMX et le taux de passage par zéros de cette trame est supérieur à un seuil SPZ la trame est considérée une trame de signal de parole.
- Dans les autres cas, la trame est considérée comme une zone de silence. Avec SMX et SMN sont les seuils maximal et minimal respectivement pour l'amplitude moyenne et SPZ est le seuil pour le taux de passage par zéros. Ces différents seuils sont donnés par les équations (I.3 et I.7) :

$$SPZ = \min(IFS, \overline{TPZ} + 2\sigma_{TPZ}) \quad (I.3)$$

Avec TPZ est la moyenne de TPZ pendant le silence, ¾TPZ est l'écart type de TPZ, IFS est un facteur choisi suivant l'expérience.

.

$$E1 = 0.03\times(EMX-EMN)+EMN \quad (I.4)$$

Avec EMX et EMN sont l'énergie maximale et minimale des trames pendant le silence. Ils sont estimé à partir des premiers 75 msec du signal de parole.

$$E2 = 4\times EMN \quad (I.5)$$

$$SMN = \min (E1, E2) \quad (I.6)$$

$$SMX = 5\times SMN \quad (I.7)$$

La figure I.3 montre un signal de parole après l'application de l'algorithme de détection des zones de silence.

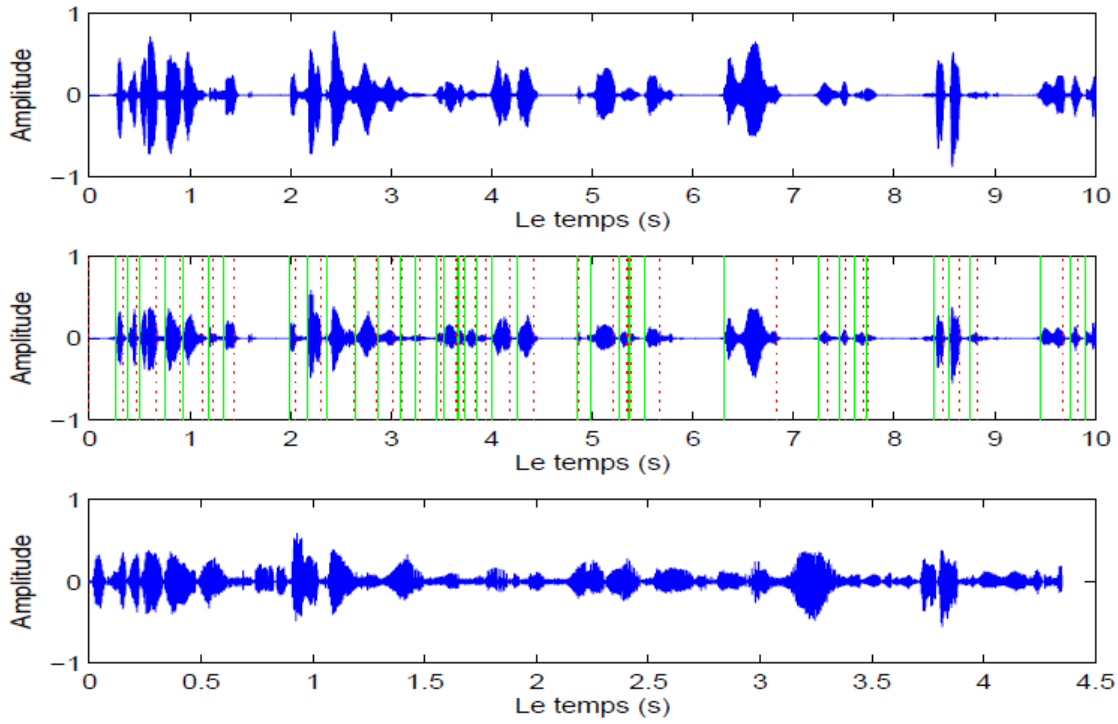



**Figure I.3:** Signal de parole après le passage par un détecteur de silence.

### I.4.1.1 Mel Frequency Cepstral Coefficients (MFCC)

Le signal de parole résulte de l'intermodulation entre un signal émis par une source et le conduit vocal, ce qui rend difficile, sur le spectre de Fourier, la séparation entre la contribution de la source (la fréquence fondamentale F0) et celle du conduit. Dans le modèle acoustique de production de la parole, le signal résulte de la convolution de la source par le conduit. Pour déconvoluer ce signal, il est intéressant de transposer le problème par homomorphisme dans un espace où la convolution est remplacée par une somme.

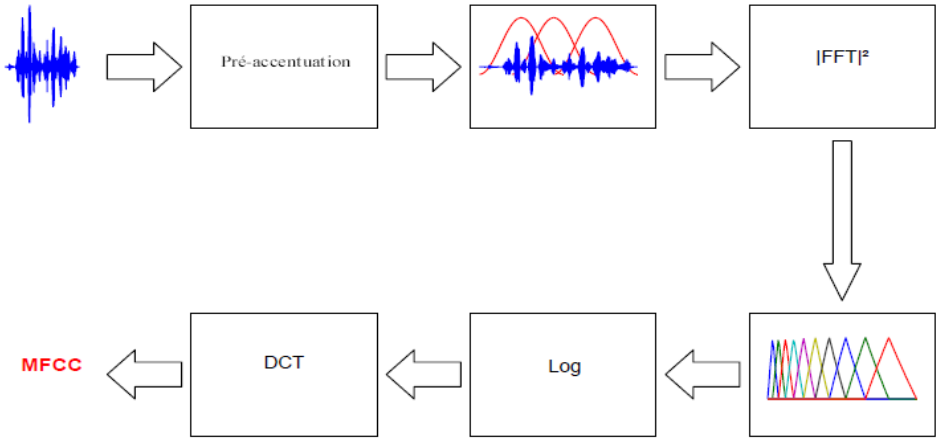



**Figure I.4:** Extraction des paramètres MFCC.

La figure I.4 montre une représentation cepstral modulaire basé sur un banc de filtre. Le signal de parole est pré-accentué en premier lieu, en appliquant un filtrage de type passe-haut. Le but de ce filtre est de rehausser les hautes fréquences du spectre, qui sont généralement réduites par le procède de production de la parole. La préaccentuation du signal est obtenue en appliquant le filtre suivant :

$$x_p(t) = x(t) - a.x(t-1) \qquad (I.8)$$

Les valeurs de a sont généralement prises dans l'intervalle [0:95; 0:98]. Ce filtre n'est pas toujours appliqué. Cette opération obéit à une expérimentation empirique. La réponse fréquentielle de ce filtre est donnée par la figure I.5.

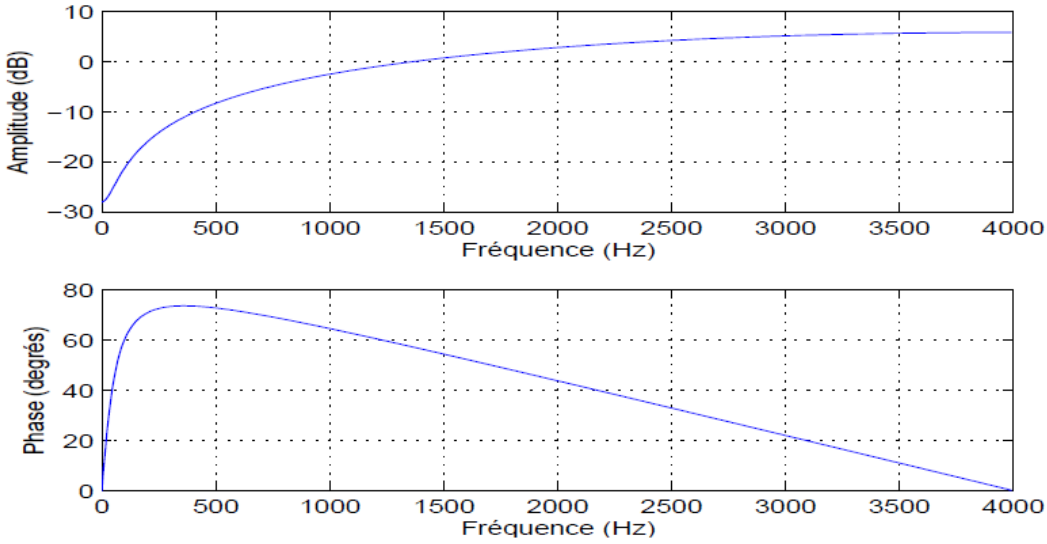



**Figure I.5:** Réponse fréquentielle, amplitude et phase.

Pour voire l'effet de ce filtre sur le signal de parole, la figure I.6 représente l'enveloppe spectrale du signale de parole avant et après l'application de ce filtre.

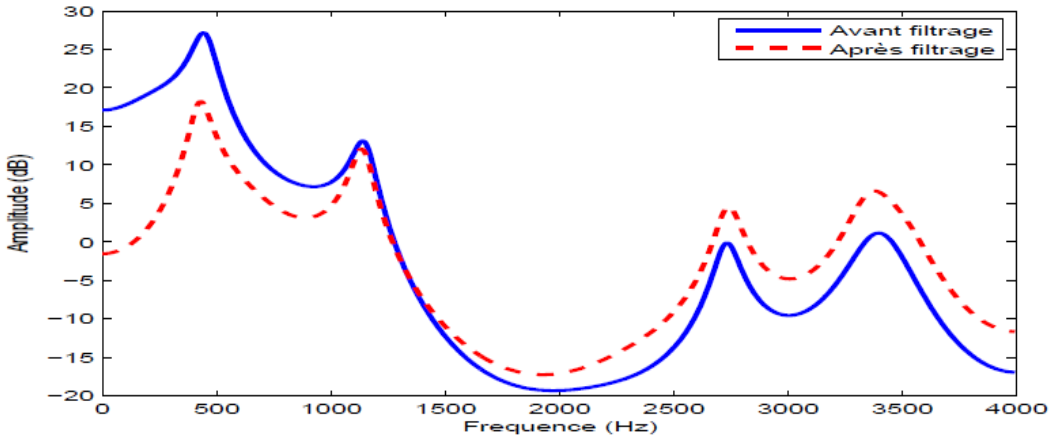



**Figure I.6:** L'enveloppe spectrale du signale de parole avant et après la préaccentuation.

Le signal de parole est ensuite analysé par une fenêtre glissante de durée courte de l'ordre de 25 ms avec un recouvrement de 50% où le signal de parole peut être considéré quasi stationnaire.

Les fenêtres les plus utilisées dans la reconnaissance de la parole sont la fenêtre de Hamming et la fenêtre de Hanning. On emploie habituellement une fenêtre de Hamming ou une fenêtre de Hanning plutôt qu'une fenêtre rectangulaire pour effiler le signal original des cotés et d'éviter la formation d'artéfacts liés aux effets de bord durant la transformation du domaine temporel au domaine fréquentiel:

$$\text{Hmming} = \begin{cases} 0.54 - 0.46.\cos\left(\frac{2\pi n}{N}\right) & 0 \le n \le N-1 \\ 0 & \text{aeilleurs} \end{cases} \quad \text{(I.9)}$$

Après la préaccentuation et le fenêtrage du signal de parole, la transformée de Fourier est calculé pour chaque trame pour obtenir le spectre du signal. Il y a de nombreux algorithmes pour calculer la FFT [13] [14]. Le spectre présente beaucoup de fluctuations, l'intérêt est porté seulement sur l'enveloppe du spectre. Une autre raison de lisser le spectre est la réduction de la taille des vecteurs spectraux. Pour réaliser ceci, nous multiplions le spectre précédemment obtenu par un banc de filtres tenant compte de la réponse acoustique de l'oreille humaine. Un banc de filtre est une série de filtres à bande passante réparti d'une façon équidistante dans l'échelle Mel. Il est défini par la forme des filtres et par la localisation des fréquences gauche, centrale et droite de chaque filtre. Les filtres peuvent être triangulaires, ou ayant d'autres formes, et ils peuvent être différemment placé sur l'échelle de fréquence. La localisation des fréquences centrales des filtres est donnée par:

$$f_{mel} = 1000.\frac{\log(1+f/1000)}{\log 2} \quad \text{(I.10)}$$

Ou f est la fréquence en Hz.

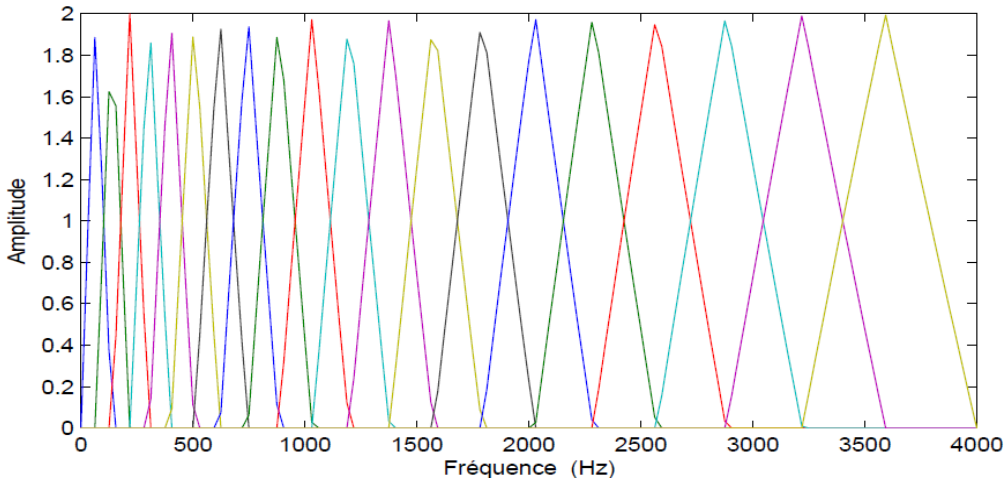



**Figure I.7:** Banc de Filtres Triangulaires équidistance en échelle Mel.

Finalement, nous prenons le logarithme de cette enveloppe spectrale et nous multiplions chaque coefficient par 20 afin d'obtenir l'enveloppe spectrale en dB. Ensuite, les coefficients cepstraux sont obtenus par une transformée en cosinus discrète à partir des logarithmes des énergies issues du banc de filtres. L'avantage de la transformation cepstrale

est de fournir des coefficients peu corrélés et de mieux séparer l'influence du locuteur [14], [15], l'expression de ces coefficients est données par:

$$c_n = \sum_{K=1}^{K} S_k . \cos\left[n\left(k - \frac{1}{2}\right)\frac{\pi}{K}\right] \qquad n = 1,2, \dots \dots L \qquad (I.11)$$

Où K est le nombre de coefficients spectraux calculés précédemment, $S_k$ sont les coefficients spectraux, et L est le nombre de coefficients cepstraux que nous voulons calculer (L·K). Finalement, nous obtenons des vecteurs cepstraux pour chaque fenêtre.

La figure IV.8 présente l'évolution de 16 paramètres MFCC sur le chiffre «واحد».

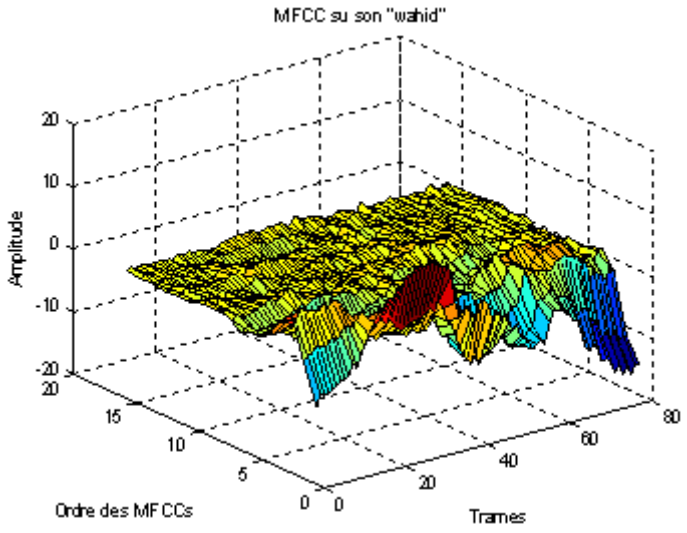



**Figure I.8:** Paramètres MFCC obtenues sur le chiffre «واحد».

### I.4.1.2 Les paramètres dynamiques

Jusqu'ici aucune information sur l'évolution de temps n'est incluse dans les MFCC. L'information dynamique dans le signal de parole est également différente d'un locuteur à l'autre. Cette information est souvent donner par les dérives cepstrales. La première dérivée des coefficients cepstraux s'appelle les coefficients Delta, et la dérivée deuxième des coefficients cepstraux s'appelle les coefficients Delta-Delta. Les coefficients delta nous donnent quelques informations sur la variation de ces vecteurs dans le temps, et les coefficients Delta-Delta nous donnent des informations sur l'accélération de la parole. Ces coefficients sont données par:

$$\Delta cm = \frac{\sum_{k=-1}^{l} k \,.c_{m+k}}{\sum_{k=-1}^{l} k^2} \quad \text{(I.12)}$$

$$\Delta\Delta cm = \frac{\sum_{k=-1}^{l} k^2.c_{m+k}}{\sum_{k=-1}^{l} k^2} \quad \text{(I.13)}$$

### I.4.2 Modélisation de la parole

Ce paragraphe parcourt les techniques les plus utilisées en reconnaissance de la parole. Différentes approches ont été développées, néanmoins on peut les classer en cinq grandes familles :

- L'approche vectorielle où le locuteur est représenté par un ensemble de vecteurs de paramètres (MFCC, PLP, etc.) dans l'espace acoustique. Ses principales techniques sont la reconnaissance à base de DTW et la reconnaissance par quantification vectorielle.
- L'approche statistique qui consiste à représenter chaque signal de parole par une densité de probabilité dans l'espace des paramètres acoustiques. Elle couvre les techniques de modélisation par les modèles de Markov cachés, par les mélanges de gaussiennes et par des mesures statistiques du second ordre.
- L'approche connexionniste qui consiste, principalement, à modéliser les signaux par des réseaux de neurones.
- L'approche prédictive.
- L'approche discriminante

#### I.4.2.1 L'approche vectorielle

Dans l'approche vectorielle, un modèle de signal parole est un ensemble de vecteurs de paramètres représentatifs de l'espace acoustique construit lors de la phase de paramétrisation a partir des signaux d'apprentissage. Lors de la reconnaissance, une distance entre cet ensemble de vecteurs et les vecteurs de paramètres (MFCC, PLP, etc.) issus des signaux de test est calculée.

L'approche vectorielle compte deux grandes techniques : la programmation dynamique et la quantification vectorielle.

- **Reconnaissance à base de DTW**

L'algorithme DTW (Dynamic Time Warping) appliqué à la reconnaissance vocale [16] [17], consiste à aligner temporellement une séquence de vecteurs de paramètres (MFCC, PLP, etc.) de test avec une séquence de vecteurs d'apprentissage. Dans ce cas, le modèle de

locuteur est tout simplement l'ensemble des vecteurs de paramètres obtenus après paramétrisation des signaux d'apprentissage. Une distance est calculée entre vecteurs d'apprentissage et vecteurs de test et moyennée sur l'ensemble de la séquence.

❖ **Quantification vectorielle**

Il s'agit de représenter l'espace acoustique par un nombre fini de vecteurs acoustiques. Cela consiste à faire un partitionnement de cet espace en régions, qui seront représentées par leur vecteur centroïde [18] [19]. Pour déterminer la distance d'un vecteur acoustique à cet espace, on effectue une mesure de distance avec chacun des centroïdes des régions et on retient la distance minimale. Si le vecteur acoustique provient du même signal pour lequel on a établi le dictionnaire de quantification, la distance sera en général moins grande que si ce vecteur provient d'un autre signal. Ainsi, on va représenter un signal par son dictionnaire de quantification.

### I.4.2.2 L'approche statistique

L'approche statistique repose sur la modélisation de la distribution des vecteurs de paramètres correspondant à un signal de parole.

❖ **Méthodes Statistiques du Second Ordre**

Le principe des Méthodes Statistiques du Second Ordre (MSSO) est de représenter une séquence de vecteurs acoustiques par une distribution gaussienne multi-dimensionnelle. Le modèle d'un signal parole se résume alors par le triplet $\{\mu, \sum, X\}$ où $\mu$ est un vecteur moyen, $\sum$ est une matrice de covariance, qui sont tous les deux estimés à partir de la séquence de $X$ vecteurs acoustiques. Les MSSO sont généralement associées à des mesures de similarité particulières en vue de la reconnaissance. Ces mesures ont pour particularité de faire intervenir le triplet$\{\mu_0, \sum_0, \gamma\}$. Ce dernier est estimé sur la séquence de vecteurs de test de manière analogue au triplet$\{\mu, \sum, X\}$. Les mesures reposent ainsi essentiellement sur une ressemblance entre les matrices $\sum$ et $\sum_0$[50].

❖ **Modèles de Markov cachés**

Les modèles de Markov cachés (ou HMM, Hidden Markov Models) s'appliquent parfaitement à la reconnaissance automatique de la parole [20]. Dans cette approche, où ne s'intéresse pas à mesure de distance d'une forme acoustique à une référence, mais de la probabilité que la forme acoustique ait été engendrée par le modèle. Le modèle d'un signal est

constitué de l'association d'une chaîne de Markov, une succession d'états avec des probabilités de transition d'un état à l'autre, et de lois de probabilités (probabilités d'observation d'un vecteur acoustique dans un état). Nous détaillerons cette méthode dans le chapitre trois.

❖ **Mélanges de gaussiennes**

La reconnaissance de la parole par mélanges de gaussiennes (ou GMM pour Gaussian Mixture Models) consiste à modéliser un signal par une somme pondérée de composantes gaussiennes [21]. Ainsi une large gamme de distributions peut être parfaitement représentée. Chaque composante des gaussiennes est supposée modéliser un ensemble de classes acoustiques. Les mélanges de gaussiennes est considéré comme un cas particulier des HMM et une extension de la quantification vectorielle [22].

### I.4.2.3 L'approche connexionniste

Les réseaux de neurones ont été assez largement utilisés en reconnaissance de la parole [23]. Ces outils de classification permettent de séparer des classes, dans un espace de représentation donné, de façon non linéaire. On peut aussi utiliser les réseaux de neurones en les couplant à d'autres techniques, comme par exemple les modèles de Markov cachés. On parle alors de méthodes hybrides.

### I.4.2.4 L'approche prédictive

L'approche prédictive repose sur le principe qu'une trame de signal peut être prédite par la seule observation des trames précédentes. De par ce concept, cette approche est considérée dans la littérature comme une approche dynamique. Une approche tenant compte des informations dynamiques véhiculées par le signal de parole. Elle s'appuie principalement sur l'estimation d'une fonction de prédiction, propre à chaque signal et apprise sur les signaux d'apprentissage. Lors de la reconnaissance, une erreur de prédiction peut être calculée entre une trame prédite (par la fonction de prédiction) et la trame réellement observée dans la séquence de test [24]. L'erreur de prédiction moyenne constitue alors la mesure de similarité entre le signal de test et le modèle (fonction de prédiction). Une autre solution envisageable est d'estimer une fonction de prédiction sur la séquence de test et de la comparer, à l'aide d'une distance, à la fonction de prédiction estimée lors de l'apprentissage [25].

### I.4.2.5 L'approche discriminante

Les Support Vector Machine (SVM) ont été conçus comme une fonction discriminante permettant de séparer au mieux des régions complexes dans des problèmes de classification à 2 classes. Cette approche donne aujourd'hui des performances similaires à l'approche HMM. Ces deux méthodes sont aussi combinées dans un nouveau formalisme.

### I.4.3 Prise de décision

La reconnaissance de parole consiste à reconnaître un mot parmi un ensemble de mots en comparant son signal à des références connues. Les performances du système de reconnaissance sont données en termes de taux de classification correcte $I_c$ ou incorrecte $I_i$, soit :

$$I_c = \frac{Nombre\ de\ tests\ correctement\ identifiés}{Nombre\ total\ de\ tentatives} \tag{I.14}$$

Et

$$I_i = \frac{Nombre\ de\ tests\ mal\ identifiés}{Nombre\ total\ de\ tentatives} \tag{I.15}$$

## I.5 Conclusion

Dans ce chapitre, nous avons exposé de façon générale le processus de production de la parole à travers les mécanismes mis en jeu lors de la phonation. Ensuite, nous avons énoncé la perception de la parole dans la bande téléphonique, ainsi que variabilité du signal de la parole. Nous avons présenté, également le système de la reconnaissance de la parole, qui peut se subdiviser en trois étapes qui sont l'extraction des paramètres acoustiques du signal de parole, la modélisation puis une dernière étape de décision.

# CHAPITRE II :

# Codecs de la parole

## II.1 Introduction

La VoIP (Voice over Internet Protocol), ou Voix sur IP est une technique permettant de transporter la voix sur un réseau IP. Le principe de cette technique consiste à encapsuler un signal audio numérisé dans le protocole IP pour le transporter sur un réseau. Pour une communication en VoIP, le signal vocal doit être compressé et codé à l'aide d'un codec dédié à la voix sur IP, ensuite, l'information à transmettre est découpée en paquets par une procédure de paquétisation avant l'envoi sur le réseau IP. Les paquets d'informations, qui circulent sur Internet, empruntent des chemins différents et arrivent fréquemment dans le désordre. Les paquets sont alors stockés dans des mémoires tampons, ou buffer, pour être ré-séquencés et permettre la décompression de l'information et sa transformation en signal sonore.

Dans ce chapitre, nous allons voir un aperçu sur les différent critères relatifs au codage, les codecs dédie à la VoIP et les différents méthodes de mesure la qualité de la voix.

## II.2 Critères relatifs au codage

Dans ce paragraphe, nous allons énumérer les différents critères couramment utilisés pour juger et classer les méthodes de codage [26]. Une liste non exhaustive d'attributs, considérés dans tout système de codage de parole, est présentée par la suite. D'autres critères peuvent être importants, selon le type d'applications, tels que la détection d'activité de parole ou la reconnaissance vocale.

### II.2.1 Débit de transmission

Le débit de transmission détermine le nombre de bits par seconde alloué au codeur pour la représentation de l'information. L'objectif d'un algorithme de codage est de réduire ce débit en maintenant la bonne qualité du signal. La largeur de bande disponible dans le système de communication détermine la limite supérieure du débit envisageable du codeur de parole. Lors de la conception d'un système de codage, un choix sera effectué parmi les codeurs à débit de transmission fixe ou variable.

### II.2.2 Qualité de parole

Une considération importante dans tout codage de parole est la qualité du signal reconstruit. Les recherches sur les différents types de codage essaient toujours de trouver un bon compromis entre la qualité du signal de parole restitué et le débit de transmission [27]. Pour un débit fixé, le critère de qualité pourra alors être employé pour évaluer un système de

codage. Deux types de mesures, objective et subjective, peuvent permettre l'évaluation de la qualité de parole:

- La qualité de restitution peut être déterminée par des tests d'écoute du signal de la parole, codé et décodé dans les conditions désirées, où des auditeurs jugeront, subjectivement, de la qualité globale et de l'intelligibilité de la parole. Pour ce genre d'étude, un grand nombre de personnes est nécessaire pour effectuer une analyse statistique de leur opinion moyenne (MOS: Mean Opinion Score).
- Les mesures objectives utilisent des fonctions ou des critères mathématiques pour comparer les formes d'onde codées et originales telles que des mesures de distorsion ou de gain [28] .Certaines mesures donnent des informations utiles selon le type de codage testé. Par exemple, le rapport signal à bruit (SNR : Signal to Noise Ratio) est représentatif pour les codeurs temporels et certains codeurs hybrides, tels que les codeurs de type CELP, qui incorporent des mécanismes de modélisation de forme d'onde.

### II.2.3 Complexité de calcul et d'implémentation

Baisser le débit de transmission en maintenant une bonne qualité du signal se fait généralement au détriment de la complexité des algorithmes mis en place. Dans pratiquement toutes les applications de codage et de décodage de la parole, une exécution temps réel sur DSP (Digital Signal Processor) est exigée. Ces processeurs étant limités au niveau de leur mémoire (RAM : Random Access Memory) et de leur vitesse (MIPS : Million d'Instructions par Seconde), les algorithmes de codage ne doivent pas être trop complexes et leurs exigences ne pas dépasser les capacités des DSP.

### II.2.4 Robustesse face aux erreurs

Dans certaines applications, la robustesse au bruit de fond et/ou aux distorsions de canal est essentielle. Un codeur de parole devra maintenir ses performances malgré un environnement bruyant [28]. Le bruit de fond entraînant une plus ou moins grande distorsion de la parole, les techniques de traitement de signal employées pour extraire les paramètres de modélisation peuvent échouer lorsque le niveau de bruit est élevé. Différentes stratégies devront alors être utilisées dans les algorithmes de codage pour compenser les erreurs possibles dues au bruit et conserver la qualité du signal reconstruit.

### II.2.5 Délai de codage

Le délai de codage est très important dans les transmissions en temps réel. Le délai global est engendré par le temps de traitement du codage et du décodage mais aussi par le délai de transmission qui dépend du canal utilisé et les différents temps d'attente liés à l'application choisie. De ce temps de restitution du signal dépendra la qualité d'écoute de la communication.

## II.3 Codecs dédie à la VoIP

Les codecs vocaux utilisés dans la VoIP comprennent ceux proposés par l'ITU-T telles que G.711, G.729 et G.723.1 ; par ETSI tels que AMR; les codecs open-source tels que les codecs iLBC et Speex ; et les propriétaires tels que le codec Silk de Skype. Ces codecs ont un débit variable dans la gamme de 6 à 40kbit/s et une fréquence d'échantillonnage variable sur une bande étroite à une bande super large. Certains codecs ne peuvent fonctionner qu'à un débit binaire fixe, tandis que de nombreux codecs avancés peuvent avoir des débits binaires variables qui peuvent être utilisées pour l'adaptation afin d'améliorer la qualité de la voix. Le choix du codec est un compromis entre la qualité de vocale et la capacité de l'infrastructure IP à délivrer une bande passante et des paramètres de QoS qui vont impacter cette qualité. En général, plus le débit binaire de la parole est grand, plus la qualité de la parole est bonne et plus l'application est gourmande en bande passante et en stockage.

### II.3.1 ITU G.711

G.711 [29] est un codec qui a été mis en place par l'ITU en 1972 pour la téléphonie numérique. Le codec a deux variantes : A-Law est utilisé en Europe et lors des communications internationales, μ-Law est utilisé dans les États-Unis d'Amérique et le Japon. G.711 utilise une quantification à 8 bits et une compression logarithmique. Le débit résultant, pour une seule direction, est de 64 kbit/s, donc un appel consomme 128 kbit/s. Ce codec peut être librement utilisé (open-source) dans des applications Voix sur IP. Les meilleures performances de ce codec sont obtenues dans les réseaux locaux où nous avons beaucoup de bande passante disponible. De plus, ce codec est caractérisé par une très bonne qualité audio perçue (un MOS de 4,2 sur 5) et par une simple implémentation, et donc il n'a pas besoin d'un processeur puissant.

### II.3.2 ITU G.729 (voir annexe A)

Le standard G.729 [30] décrit un algorithme pour le codage de signaux vocaux à 8 kbit/s au moyen de la prédiction linéaire à excitation par séquences codées à structure algébrique conjuguée (CS-ACELP) (Conjugate-Structure Algebraic-Code-Excited Linear-Prediction). Les annexes A, B du standard G.729 étendent les fonctionnalités du codec tel que le codage à un taux de 6,4 kbit/s et 11,8 kbit/s, une multi-cadence de fonctionnement, Le codec G.729 est généralement utilisé dans les applications VoIP.

### II.3.3 ITU G.723.1

Le codec G.723.1 [31] est aussi généralement utilisé dans les applications VoIP. Le codeur est fondé sur les principes du codage prédictif linéaire (Linear Predictive Coding LPC) par analyse et synthèse, en vue de minimiser un signal d'erreur pondéré par une courbe de perception. Le G.723.1 possède deux débits binaires associés: 5,3 kbit/s et 6,3 kbit/s. Pour le débit supérieur, on fait appel à l'excitation par quantification d'impulsions multiples selon le critère du maximum de vraisemblance (MP-MLQ, Multi-Pulse Maximum Likelihood Quantization). Pour le débit inférieur, on fait appel à l'excitation par séquences codées à structure algébrique (ACELP, Algebraic-Code-Excitation).

### II.3.4 GSM-FR

Le codec GSM 06.10 Full-Rate [32] décrit le transcodage dans la télécommunication cellulaire. Le schéma de codage est basé sur Regular Pulse Excitation – Long Term Prediction (RPE-LTP) paradigme de codage de la parole.

### II.3.5 GSM-HR

GSM 06.20 Half Rate (HR) [33] nécessite moins de la moitié de la bande passante du GSM-FR au prix d'une moins bonne qualité audio. Le codec utilise l'algorithme Vector-Sum Excited Linear Prediction (VSELP).

### II.3.6 AMR

Le codec audio Adaptive Multi-Rate (AMR) [34] est largement utilisé dans les réseaux cellulaires GSM et UMTS. Le codec encode le signal à huit différents débits, de l'ordre de 4,75 kbit/s à 12,2 kbit/s. Le débit le plus élevé de 12,2 kb/s est compatible avec le standard GSM Enhanced Full Rate (GSM-EFR). Le système de codage est basé sur l'algorithme Algebraic Code Excited Linear Prediction (ACELP).

### II.3.7 iLBC

Internet Low Bitrate Codec (iLBC) [35] est un codec conçu pour la communication voix sur IP. L'algorithme utilise Block-Independent Linear Predictive Coding (BI-LPC) et prend en charge des débits binaires de 13,3 et 15,2 kbit/s. Généralement les codecs à bas débit exploitent les dépendances entre les trames. Le traitement indépendant des trames appliquées par iLBC offre une meilleure robustesse du codec, similaire à celle du codec G.711 avec le masquage de pertes de paquets (Packet Loss Concealment PLC).

### II.3.8 Speex (voir annexe B)

Speex [36] est un codec libre (Open source) ciblée pour la VoIP. Le codec utilise CELP comme technique de codage. Speex supporte un large intervalle de débits : de 3,95 à 24,6 kbit/s pour les signaux à bande étroite (narrow band). L'encodage est contrôlé par le paramètre de qualité qui varie de 0 à 10. Le mode de plus faible qualité 0 (correspondant à un débit de 2,15 kbit/s) est principalement utilisé pour le bruit de confort (confort noise).

### II.3.9 Silk

Le codec Silk [37] est utilisé par l'application Skype. Le débit binaire pour les signaux à bande étroite (narrowband) peut être réglé entre 6 et 20 kbit/s. Le codec fournit également la transmission discontinue DTX (discontinuous transmission), un générateur de bruit de confort CNG (Comfort Noise Generator) et les mécanismes de masquage de pertes de paquets (Packet Loss Concealment PLC).

## II.4 Qualité de la voix

### II.4.1 MOS (Mean Opinion Score)

Le MOS est la mesure la plus bien connue de qualité de voix. Il y a deux méthodes de tester; conversation opinion et écouter opinion, Le but est de juger la qualité du système de transmission de voix par réaliser une conversation ou par écouter des échantillons de la parole. MOS est exprimé dans un certain nombre (tableau II.1), de 1 à 5, 1 étant le pire et 5 (la meilleur). MOS est tout à fait subjectif, car il est à base de chiffres qui résultent de ce qui est perçu par les gens lors des essais. Cependant, il existe des applications logicielles qui MOS mesure sur les réseaux.

| MOS | Qualité | Dépréciation |
|---|---|---|
| **5** | Excellent | Imperceptible |
| **4** | Bon | Perceptible mais pas ennuyeux |
| **3** | Moyen assez bon | Légèrement gênant |
| **2** | Médiocre | Ennuyeux |
| **1** | Mauvais | Très ennuyeux |

**Tableau II.1:** Note moyenne d'opinion (MOS).

Le principe de calcul du MOS est basé sur un sondage d'un échantillon supposé représentatif de la population des utilisateurs. Les personnes constituant l'échantillon sont invitées à écouter un signal (souvent de la voix), puis son équivalent codé-décodé. Après chaque écoute, l'auditeur donne une note sanctionnant la qualité qu'il a perçue. La moyenne des notes fournies par la population constitue le MOS. Le tableau ci-dessous montre les résultats obtenus par quelques codecs courants.

| Codec VOIP | Débit (kbps) | Score MOS |
|---|---|---|
| **G.729** | 8 | 3.92 |
| **G.711** | 64 | 4.1 |
| **G.726** | 32 | 3.85 |
| **G.723.1** | 6.4 | 3.9 |
| **G.723.1** | 5.3 | 3.65 |
| **iLBC** | 15,2 | 4,14 |

**Tableau II.2:** Scores moyens d'opinion des différents codecs.

### II.4.2 PSQM (Perceptual Speech Quality Measure)

Le modèle **PSQM** (figure II.1) a été défini en 1996 dans la recommandation P.861 de l'UIT-T. Il s'agit d'un processus d'évaluation automatique, qui utilise un algorithme permettant d'obtenir des scores liés à ceux du **MOS**. Il fournit une valeur de sortie comprise entre 0 et 6,5 où le 0 indique le bon canal et le 6,5 le plus mauvais, Si le signal de sortie est identique au signal d'origine, l'algorithme émet alors un score dit "parfait". De même, si des différences sont mesurées mais qu'elles ne sont pas audibles, le score n'est pas dégradé.

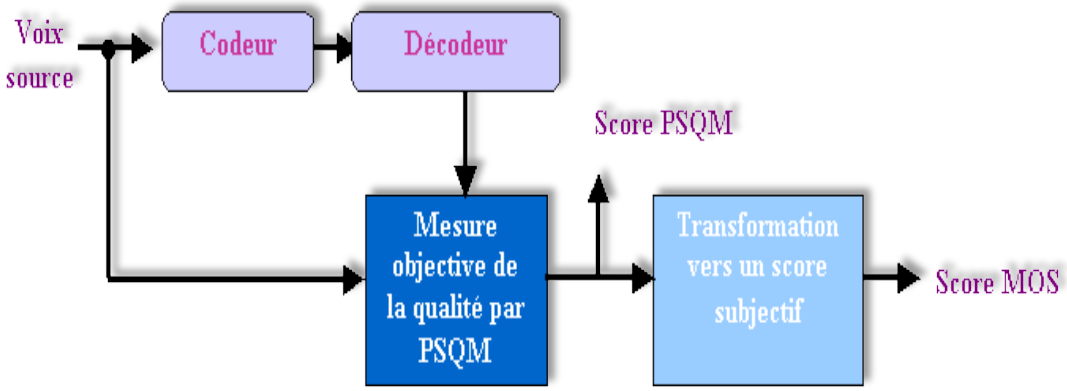



**Figure II.1:** Principe d'algorithme de PSQM.

Le délai de propagation dans un réseau IP, il peut parfois être important (de 30 à 300 ms). Or, si un décalage trop important apparait sur le signal reçu, il ne sera pas aligné proprement avec le signal de référence. La comparaison ne pourra alors se faire correctement et la mesure de la qualité obtenue ne sera pas significative. C'est pour cette raison, que le **PSQM** n'est pas encore totalement efficace pour les applications de voix sur IP.

### II.4.3 E-modèle

L'E-Model consiste à effectuer des mesures réseau sur les qualités objectives directement à partir des trames réseau (délais, gigue…). Il a été standardisé par l'UIT. On complète ces données avec des informations issues d'une table d'évaluation des codecs, on obtient ainsi une note R comprise entre 0 et 100, une table de correspondance permet d'établir un lien calculé entre l'échelle R et l'échelle MOS comprise entre 1 et 5, on obtient a la fin des résultats très proches de ceux obtenus dans les mêmes conditions par le modèle MOS d'origine.

L'E-Modèle fournit un modèle utile pour l'analyse prédictive. L'équation du modèle est comme suit :

$$R = R_0 - Is - Id - Ie + A \qquad \text{(II.1)}$$

- $R_0$: Coefficient initial signal / bruit, « capital initial de QoS », égal à 94,3 en VOIP.
- Is: Coefficient de dommages simultanés avec l'émission de la voix (bruit de fond...).
- Id: Coefficient de dommages dus au délai de transmission et de transport.

- Ie: Coefficient de dommage de distorsion causée par les équipements.
- A: Coefficient d'amélioration.

Le facteur R ainsi calculé de 0 à 100 permet de déduire directement un coefficient MOS de 0 à 5.

## II.5 Conclusion

Dans ce chapitre, nous avons décrit les critères relatifs au codage, ainsi que les différents codecs utilisés sur la VoIP, et nous avons terminé par les distinctes méthodes exploités pour mesurer la qualité de la voix transcodée.

# CHAPITRE III :

# Modèles de Markov Cachés

## III.1 Introduction

Les modèles de Markov caché sont des outils statistiques permettant de modéliser des phénomènes stochastiques. Ces modèles sont utilisé dans de nombreux domaines [38] tels que la reconnaissance et la synthèse de la parole, la biologie, l'ordonnancement, l'indexation de document, la reconnaissance d'images, la prédiction de séries temporelles, pour pouvoir utiliser ces modèles efficacement, il nécessaire d'en connaitre les principes.

Ce chapitre a pour objectif d'établir les principes, les notations utiles et les principaux algorithmes qui constituent la théorie des modèles de Markov cachés (MMC).

## III.2 Présentation et applications des modèles de Markov cachés

Les modèles de Markov cachés (Hidden Markov Models ou HMMs) ont été introduits par Baum et al, à la fin des années 60. Ce modèle est fortement apparenté aux automates probabilistes, définis par une structure composée d'états et de transitions, et par un ensemble de distributions de probabilité sur les transitions. A chaque transition est associé un symbole d'un alphabet fini. Ce symbole est généré à chaque fois que la transition est empruntée. Un HMM se définit également par une structure composée d'états et de transitions et par un ensemble de distributions de probabilité sur les transitions [39].

Les modèles de Markov cachés sont utilisés pour modéliser des séquences d'observations. Ces observations peuvent être de nature discrète (par exemples les caractères d'un alphabet fini) ou continue (fréquence d'un signal, température). Sans prétendre à une présentation exhaustive des modèles de Markov cachés, l'objectif de ce chapitre est de dresser un portrait général de ce modèle et de son utilisation en apprentissage. Le signal parole peut être assimilé à une succession d'unités acoustiques, qui sont modélisées par des modèles de Markov Cachés (MMC). A chaque état du modèle de Markov est associée une distribution de probabilité modélisant la génération des vecteurs acoustique via cet état. Un HMM est caractérisé par plusieurs paramètres:

- Son nombre d'états N;
- L'ensemble des états du modèle: $\mathbf{e = (e_i)}$, avec $\mathbf{(1 \leq i \leq N)}$;
- Une matrice de transition entre les états: $\mathbf{A = (a_{ij})}$, avec $\mathbf{(1 \leq i, j \leq N)}$ de taille $\mathbf{N \times N}$;
- La probabilité d'occupation (vecteur de probabilités initiales) d'un état à l'instant initial: $\mathbf{(\pi i)}$ avec $\mathbf{(1 \leq i \leq N)}$: $\mathbf{\pi i = P( e_1 = e_i)}$.
- La densité de probabilité d'observation associée à l'état $e_i$ **:** $b_i.b_i$ qui est généralement modélisée par un modèle à mélange de Gaussiennes (GMM).

Lorsque le processus fournit en sortie une suite d'observations et non pas la séquence des états par lesquels il est passé, on dit que le modèle de Markov est **caché.** A chaque instant **t**, le processus se trouve dans un état donné et une observation est générée par la fonction aléatoire qui lui est associée. Un MMC est donc représenté par un ensemble de paramètres :

$$\theta_{MMC} = (N, A, \{\pi_i\}, \{b_i\}) \quad \text{(III.1)}$$

Les paramètres du MMC sont estimés empiriquement sur de grands corpus de parole annotés. On peut classer les principales applications des HMMs en deux catégories. La première traite les problèmes de reconnaissance ou de classification, la seconde a trait aux problèmes de segmentation de séquences, c'est-à-dire au d'coupage d'une séquence en sous-séquences de différents types.

### III.3 Modélisation de la parole par un HMM

Pour simplifier les choses nous supposons qu'un modèle HMM modélise un mot du vocabulaire, dans un cas plus général, la modélisation d'un mot est construite par la concaténation de plusieurs modèles HMMs, où chaque modèle HMM modélise une unité acoustique de base telle que le phonème.

#### III.3.1 Principe de la modélisation [40]

Un modèle HMM va modéliser un signal parole d'une telle façon que chaque segment supposé stationnaire de signal va correspondre à un état dans le modèle HMM. Chaque état HMM est caractérisé par une distribution de probabilité des différents vecteurs acoustiques associés au segment attribué à cet état. La transition d'un segment à un autre segment du signal est modélisée par la transition entre les états, laquelle est supposée être instantanée et caractérisée par la probabilité de transition de l'état (Figure III.1).

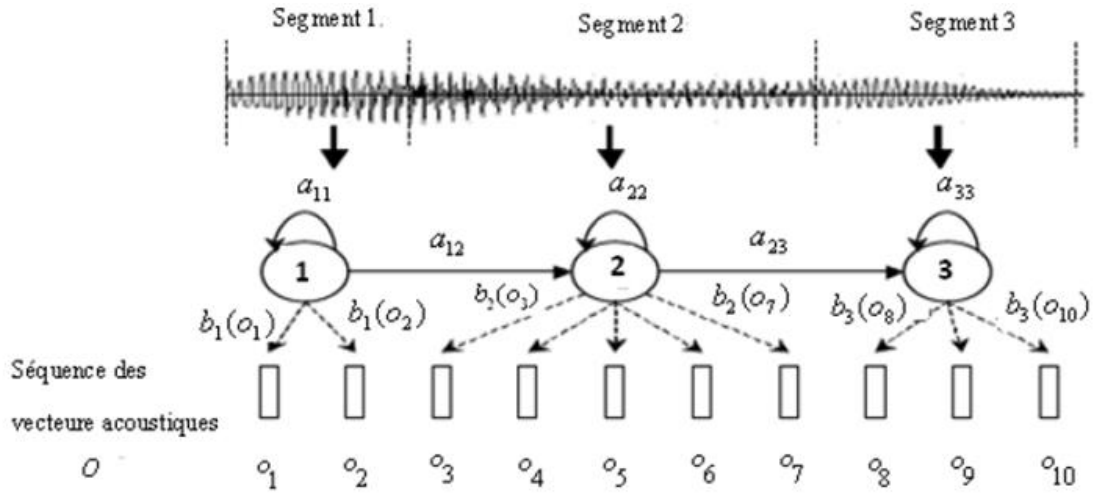



**Figure III.1:** Un exemple de HMM à 3 états modélisant un signal contenant 10 vecteurs acoustiques.

### III.3.2 Topologie des HMMs utilisés pour la parole

La parole est un phénomène dont la dimension temporelle ne peut être ignorée. Les HMM utilisés pour la représenter sont des modèles "gauche-droite" qui ne permettent pas de "retour en arrière" et qui démarrent toujours depuis l'état initial (i=1). C'est-à-dire que leurs probabilités vérifient :

$$i > j \quad \Rightarrow a_{ij} = 0 \qquad 2 \le i \le N, \qquad 1 \le j \le N-1$$

$$\pi_i = P(q_i = i) = \begin{cases} 1 pour i=1 \\ 0 pour i \le N \end{cases} \tag{III.2}$$

Dans ce cadre, R. Bakis [40] a proposé un modèle type pour représenter un mot qui permet le bouclage sur l'état courant (progression acoustique stationnaire) et le passage à l'état suivant (progression acoustique standard), voir figure III.2. Le nombre d'états du modèle est normalement proportionnel à la durée moyenne du mot. La plupart des systèmes de reconnaissance utilisent des modèles à trois états.

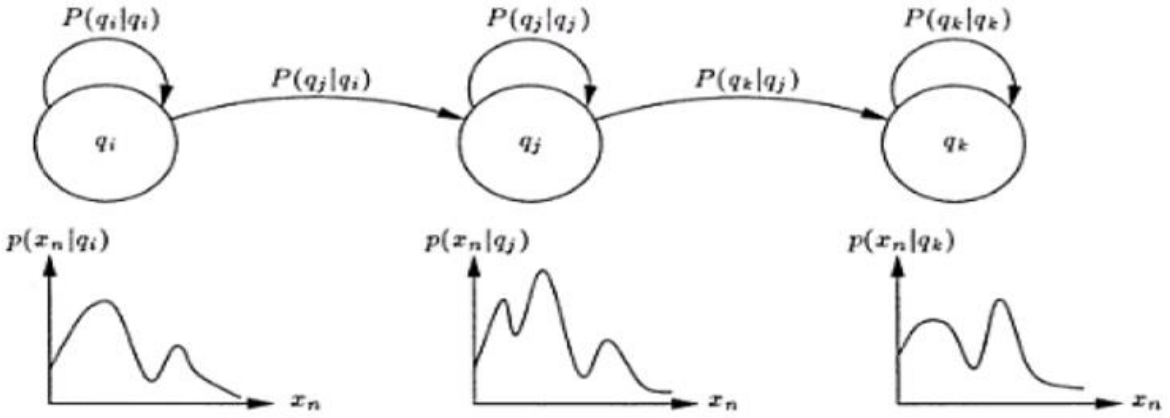



**Figure III.2:** Exemple d'un HMM avec une topologie de type Bakis à 3 états.

Ce type de modèle est devenu générique dans le domaine de la RAP. Il est utilisé dans de nombreux systèmes pour modéliser les unités acoustiques de base.

### III.3.3 Modélisation des observations acoustiques

Les observations émises lors des transitions représentent la succession des trames acoustiques au cours de la prononciation du mot. Ces observations peuvent être décrites par un nombre fini de symboles au moyen de la quantification vectorielle dans le cas des modèles discrets, ou de modéliser leurs probabilités d'émission par des densités de probabilité continues dans le cas des modèles continus.

### III.3.4 Problèmes rencontrés avec les MMC [41]

L'utilisation des MMC conduit à résoudre essentiellement trois types de problèmes.

#### III.3.4.1 Problème de l'évaluation

Soit une séquence d'observation O = $\{o_1, o_2,\ldots, o_t\}$ et un modèle M (A,B,Π). Comment évaluer la probabilité de générer la séquence d'observations par le modèle M : **P(O/M) ?**

#### III.3.4.2 Le chemin optimal

Comment choisir une séquence d'états $S_1, S_2, \ldots..S_n$ qui soit optimale au sens d'un certain critère.

### III.3.4.3 Estimation des modèles

Comment estimer les paramètres du modèle M ( A,B,Π ) de manière à maximiser la probabilité P(O/M).

### III.3.5 Evaluation de P (O/M)

Soit une séquence d'observations O= $O_1$, $O_2$, …., $O_T$ est un modèle M = (A,B,Π). P(O/ M) représente la probabilité de générer la séquence d'observations O par le modèle M peut être évaluée soit par la variable FORWARD ou BACKWARD, soit par l'alignement de VITERBI.

- **Evaluation de P (O/M) par la variable FORWARD**

La variable FORWARD est définie comme suit :

$$\alpha_t(i) = P(O_1, O_2, \ldots\ldots, Ot = i / M) \qquad \text{(III.3)}$$

C'est la probabilité d'arriver à l'état i à l'instant t après avoir observer la séquence $O_1$, $O_2$,…….. $O_t$. Elle est donnée par l'équation récurrente suivante :

$$\alpha_{t+1}(i) = \left( \sum_{i=1}^{N} \alpha_t(i) a_{ij} \right) * b_j(O_{t+1}) \qquad \text{avec } \alpha_{t=0}(0) = 1 \text{ et } \alpha t=0\ (i) = 0 \text{ pour } i \neq 0 \qquad \text{(III.4)}$$

Voici l'algorithme qui permet de la calculer:

**Algorithme :**

Début
  Initialisation: $\alpha_1(i) = \pi_i\ b_i$;  $1 \leq i \leq N$
  Pour t = 1,2, …. , T-1
    Faire
      Pour j=1, 2, ….. , N
        Faire
          $\alpha_{t+1}(i) = \left( \sum_{i=1}^{N} \alpha_t(i) a_{ij} \right) * b_j(O_{t+1})$
        Fait
    Fait
  $P(O / M) = \sum_{j=1}^{N} \alpha_T(j)$
Fin

- **Evaluation par la variable Backward**

Cette variable est à son tour définie comme suit :

$$B_t(i) = P( O_{t+1}, O_{t+2}, ....., O_T, O_t = i / M ) \quad \text{(III.5)}$$

C'est la probabilité d'observer la séquence $O_{t+1}$ , $O_{t+2}$, ……, $O_T$ sachant qu'a l'instant t on est à l'état i. Elle est donnée par l'équation récurrente suivante :

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{i+1}) \beta_{t+1}(j) \quad \text{(III.6)}$$

Voici l'algorithme d'évaluation suivant décrire les étapes de traitement:

**Algorithme :**

Début

Initialisation: $B_T(i) = 1, \quad 1 \le i \le N$

Pour t = T-1, T-2, …. …. ,1

Faire

Pour i=1, 2, ….. , N

Faire

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)$$

Fait

Fait

$$\beta_t(i) = \sum_{i=1}^{N} \pi_i \beta_i(i)$$

Fin

- **Algorithme de VITERBI**

L'algorithme de VITERBI consiste à calculer la probabilité P (O/M) et de donner le chemin optimal pour une suite d'observation. Cette probabilité P (O/M) n'est rien d'autre que la variable $\delta_t(j)$ définie par :

$$\delta_t(j) = \max ( \delta_{t-1}(i) a_{ij} b_j( O_t)) \quad \text{(III.7)}$$

Avec $\delta_{t=0}(0) = 1$ et $\delta_{t=0}(j) = 0$ pour $j \neq 0$

**Algorithme :**

Début

Initialisation: $\delta_1(i) = \pi_i b_i(O1)$ ; $1 \le i \le N$

Pour t = 2,3 …. , T

Faire

Pour j =1, 2, ….. ,N

Faire

Pour i = 1, 2, ……,N

Faire

$\delta_t(j) = \max ( \delta_{t-1}(i)\, a_{ij}\, b_j( O_t))$

$\psi_t(j) = \arg\max ( \delta_{t-1}(i) a_{ij} )$

Fait

Fait

Fait

$P(O / M) = \max (\delta_T(i) )$

$1 \le i \le N$

$I= \max (\delta_T(i))$

$1 \le i \le N$

Chemin optimal: chemin [T] = $\psi_{t+1}$[chemin[T+1]]

Fin

### III.3.6 Ré-estimation des modèles

Pour ajouter les paramètres du modèle (A, B, $\pi$) on introduit deux variables $\varepsilon_t( i , j)$ et $\gamma_t( i)$ telles que : $\varepsilon_t( i , j)$ : est la probabilité d'être à l'état i à l'instant t et d'être à l'état j à l'instant t+1.

$$\varepsilon_t( i , j) = P(O_t = i, O_{t+1} = j / O, M ) = \alpha_t (i)\, a_{ij}\, b_j(O_{t+1})\beta_{t+1}(j) / P( O /M) \qquad \text{(III.8)}$$

$\gamma_t( i)$ : représente la probabilité d'être à l'état i à l'instant t.

$\sum_{t=1}^{T} \gamma_t(i)$ : compte le nombre de transitions à partir de l'état i

$$\sum_{t=1}^{T} \gamma_t(i) = \sum_{i=1}^{N} \varepsilon_t(i, j) \qquad \text{(III.9)}$$

$\sum_{i=1}^{N} \varepsilon_t(i, j)$ : compte le nombre de transitions de l'état i à l'état j.

### III.4 Paramètres d'évaluation

Les performances des systèmes d RAP sont évaluées en comparant le résultat de la reconnaissance obtenue sur un nombre de phases de test avec l'étiquetage de référence de ces phases. La précision de cette évaluation dépende du nombre de tests réalisés. Habituellement, les taux de reconnaissance sont représentés par le pourcentage d'identification (percent correct en anglais) et le pourcentage de reconnaissance (percent accuracy en anglais). Le pourcentage d'identification (Ident) correspond à l'équation suivante :

$$Ident = (N\text{-}O\text{-}S) \times 100 / N \qquad (III.10)$$

Le pourcentage de reconnaissance (Reco) correspond à l'équation suivante : pourcentage de mots ou de phrase reconnus correctement.

$$Reco\ (N - O\text{-}S\text{-}1) \times 100 / N \qquad (III.11)$$

Avec :

N: le nombre total d'unités.

O: le nombre d'omissions (le nombre d'unités non détectés).

S: le nombre de substitutions (le nombre d'unités pour lesquels le système a commis une erreur).

I: le nombre d'insertions (le nombre d'unités reconnus alors qu'aucune unité n'a été prononcé).

Comme le montre l'équation III.11, le pourcentage d'identification ne prend pas en compte le nombre d'insertions. C'est le pourcentage de reconnaissance qui le prend en compte et pour cela il est considéré comme le paramètre le plus indicatif pour évaluer les performances d'un système de RAP. Ces deux équations d'évaluation peuvent être résumées dans une simple équation (III.12) pour le mode de reconnaissance de mots isolés, étant donné que le nombre d'omissions (O) et le nombre d'insertion (I) sont nuls.

$$Ident = Reco = (N - S) \times 100 / 100 \qquad (III.12)$$

### III.5 Conclusion

Dans ce chapitre nous avons présenté le concept des modèles de Markov cachés (HMMs), leur formalisme ainsi que leur emploi dans le domaine de la reconnaissance de la parole. Le chapitre suivant, sera consacré à décrire notre mise en pratique des modèles de Markov cachés pour de la RAP en mode des mots isolés, sous la plate forme HTK, l'une des plus utilisée actuellement dans le domaine de la RAP.

# CHAPITRE IV :

# Expériences et résultats

## IV.1 Introduction

Ce chapitre traite de la mise en œuvre du système de reconnaissance de parole basé sur les modèles de Markov cachés (HMMs) avec de la parole codée G.729 et Speex. Pour cela, nous commençons par décrire la base de données utilisée. Nous exposons également la technique utilisée pour l'extraction des caractéristiques. Ensuite, nous examinons les performances du modèle HMM en termes de taux de reconnaissance correcte; en fonction du nombre de gaussiennes.

Les performances sont évaluées également en fonction de type de base de données utilisée à savoir: i) base de données clean ; ii) la base de test transcodée G.729 puis ; iii) la base transcodée Speex, donc utilisant de la parole reconstituée.

A noté que notre système est mis au point à partir de la plate-forme HTK (Hidden Markov Model Toolkit), la boîte à outils de modèles de Markov cachés de l'Université de Cambridge [42].

## IV.2 Description des bases de données utilisées

La base de données utilisée dans ce travail est la base de données ARADIGIT [43]. Elle est constituée de prononciations des 10 chiffres de la langue Arabe, de zéro jusqu'à neuf, prononcés par 60 locuteurs des deux sexes avec trois répétitions pour chaque chiffre. Cette base a été enregistrée par des locuteurs algériens de différentes régions âgés entre 18 et 50 ans dans un environnement calme avec un niveau de bruit ambiant inférieur à 35 dB, sous format WAV, avec une fréquence d'échantillonnage égale à 8 KHz.

**( صفر, واحد, اثنان, ثلاثة, أربعة, خمسة, ستة, سبعة, ثمانية, تسعة )**

La base de données ARADIGIT8K est passée à travers le codec G.729 où Speex pour aboutir les bases de données transcodées de type G.729 et Speex.

Dans ce travail nous exploitons trois bases de données :

1. ARADIGIT8K: La base de données originale et son codage;
2. G.729-ARADIGIT8K: La base de données transcodée via le codec G.729;
3. Speex-ARADIGIT8K: La base de données transcodée via le codec Speex.

## IV.3 Outils de programmation utilisés

### IV.3.1 Logiciel Matlab

Matlab est l'outil de référence pour la simulation numérique. Il permet, de manière plus générale, de résoudre une grande diversité de problèmes de simulation dans des domaines aussi variés que le traitement du signal, les statistiques ou la vision.

### IV.3.2 Langage C

Langage de programmation, très utiles en domaine scientifique pour traite et manipuler des fichiers ou des dossiers, notamment des fichiers de parole. Ici nous allons utiliser C pour implémenté le codeur G.729, ce dernier à été exploité pour coder et décoder notre base de données ARADIGIT8K.

### IV.3.3 Perl

Perl est un langage de programmation reprenant des fonctionnalités du langage C et des langages de scripts comme le Shell (sh), ce langage étant particulièrement adapté au traitement et à la manipulation de fichiers texte.

## IV.4 Développement d'un système de reconnaissance de la parole sous HTK

### IV.4.1 La plate-forme HTK

La plate-forme HTK est constituée d'un ensemble d'outils logiciels qui permettent de construire des systèmes de reconnaissance de la parole à base de modèles de Markov cachés. HTK offre une très grande liberté de choix tout au long de la construction du système de reconnaissance. Les modèles peuvent représenter des mots ou tout type d'unité sub-lexicale, et leur topologie est librement configurable. Les densités de probabilité d'émission, qui sont associées aux états, sont décrites par des multi-gaussiennes. Les modèles sont initialisés avec l'algorithme de Viterbi, puis ré-estimés par l'algorithme optimal de Baum-Welch. Le décodage est réalisé par l'algorithme de Viterbi, sous la contrainte d'un réseau syntaxique défini par l'utilisateur et éventuellement d'un modèle de langage de type bi-gramme dans la plupart de cas. Les résultats sont enfin évalués par alignement dynamique avec la chaîne phonétique ou lexicale de référence [44].

L'ensemble de ces outils est écrit en langage C, ce qui rend l'outil HTK largement répandu dans le monde de la recherche. En 1992, ses concepteurs revendiquaient déjà plus d'un centaine d'utilisateurs. Tous ces avantages nous ont encouragés à construire notre système de reconnaissance avec HTK.

### IV.4.2 Présentation d'HTK

HTK dans sa version 3.4 est structuré comme le montre la figure IV.1.

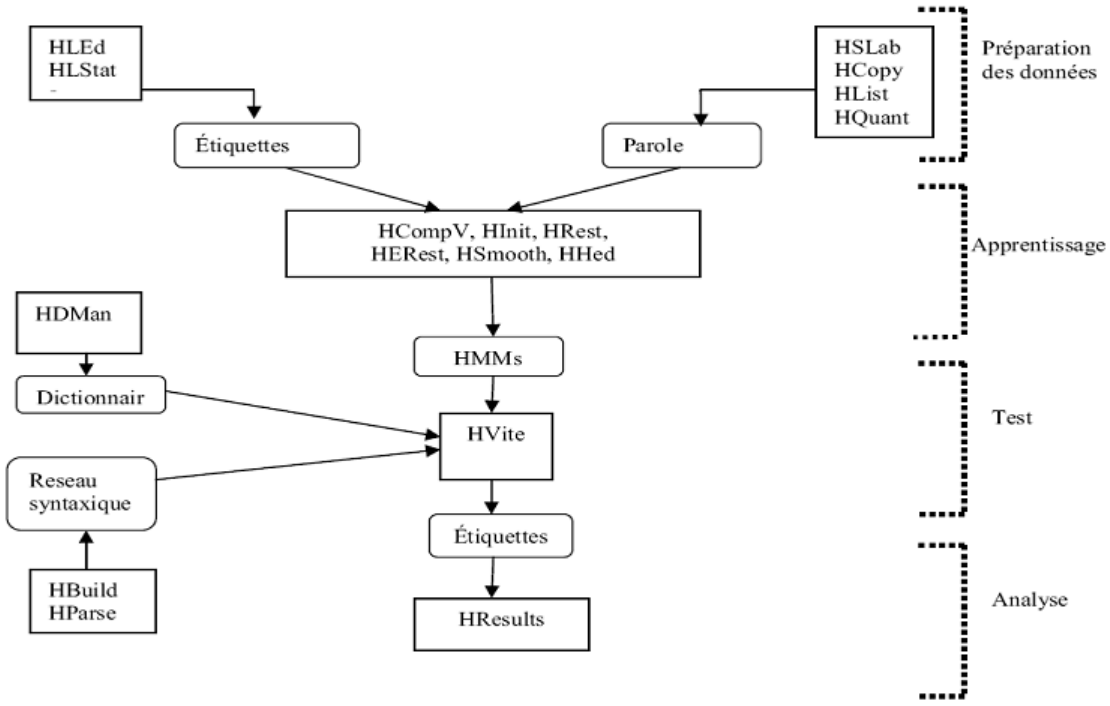



**Figure IV.1:** Structure d'un système de reconnaissance avec HTK.

Toutes les fonctionnalités de HTK sont définies dans des modules constituant la librairie qui assure l'interfaçage avec les objets extérieurs et constitue la ressource commune aux outils permettant :

- L'analyse du signal de parole;
- La manipulation de dictionnaires de prononciation ;
- La définition de modèles du langage ;
- L'apprentissage et l'adaptation des modèles acoustiques ;
- Le décodage acoustico-phonétique de parole ;
- L'alignement de parole sur des transcriptions linguistiques.

Les principaux outils de base de HTK s'enchaînent naturellement pour réaliser les différentes étapes d'un système de reconnaissance, ces outils ainsi que leurs descriptions sont donnés dans le tableau IV.1 suivant:

| **Outils** | **Rôle** |
|---|---|
| Hbuild | Conversion de modèles de langage dans différents types de format. |
| HcompV | Calcul de la moyenne et de la variance sur un ensemble de données d'apprentissage. |
| Hcopy | Calcul des paramètres de fichiers signaux. |
| HDMan | Édition des dictionnaires. |
| HERest | Phase d'apprentissage - Ré-estimation des HMM en continu (Baum-Welch). |
| HHEd | Édition des HMM. |
| Hinit | Phase d'apprentissage - Initialisation d'un HMM. |
| Hled | Édition des fichiers d'étiquettes. |
| Hlist | Visualisation en format texte des fichiers de données |
| HLStats | Calcul de statistiques sur les étiquettes. |
| Hparse | Génération du graphe de décodage. |
| Hquant | Quantification vectorielle pour HMM discret. |
| Hrest | Phase d'apprentissage - Ré-estimation d'un HMM (Baum-Welch). |
| Hresults | Résultats du décodage (alignement dynamique entre les fichiers de résultats et de références). |
| HSGen | Générateur automatique de phrase en fonction d'une grammaire. |
| HSLab | Affichage du signal et des étiquettes. |
| Hsmooth | Lissage des paramètres des HMM. |
| Hvite | Décodage parole continue (Viterbi). |

**Table IV.1:** Outils logiciels de base de HTK (Version 3.4).

### IV.4.3 Système de la reconnaissance de la parole sous HTK

Nous avons développé, dans le cadre de ce mémoire, le système de la reconnaissance de la parole fondés sur les modèles de Markov cachés à partir de la plateforme HTK (Hidden Markov ToolKit) sur la base de données de parole ARDIGIT. La boîte à outils HTK est efficace, flexible (liberté du choix des options et possibilité d'ajout d'autres modules) et complète dans le sens où elle fournit une documentation très détaillée, le livre HTK [45], est une encyclopédie dans le domaine de reconnaissance de la parole.

Afin de concevoir notre système, on se base sur des mots chiffres de la langue Arabe. On commence par définir les ressources nécessaires dont on a besoin par la suite. On définit,

alors, le modèle de langage, appelé aussi lexique ou dictionnaire (Figure IV.2), qui décrit l'enchainement des mots.

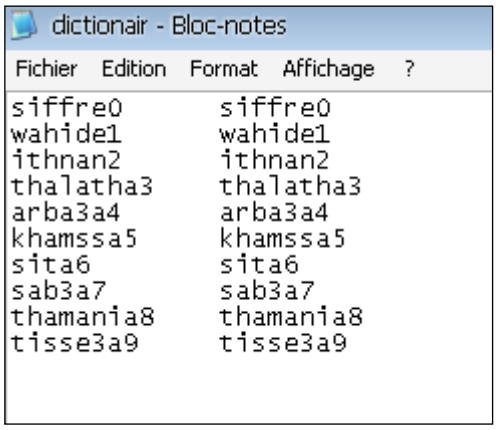



**Figure IV.2 :** Dictionnaire de la base ARADIGIT.

Ensuite, on construit le dictionnaire (Figure IV.3). Pour la base de données ARADIGIT, qui est une base de chiffres en arabe, composée d'un vocabulaire de 10 chiffres arabes (de zéro à neuf) est assez limité, d'où la simplicité de définir le dictionnaire et la grammaire (Figure IV.2) et (Figure IV.3).

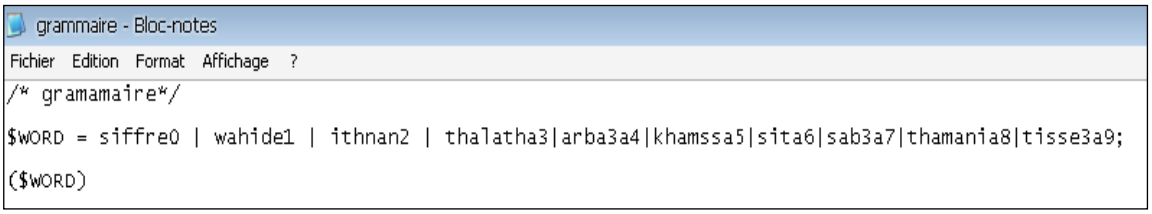



**Figure IV.3 :** Grammaire de la base ARADIGIT

### IV.4.3.1 Extraction des coefficients MFCC sous HTK

Une fois qu'on a défini le dictionnaire, la grammaire, on passe à l'extraction des coefficients MFCC exploités par les modèles de Markov cachés. Le fichier de configuration, appelé dans notre cas config (Figure IV.4)**,** permet de définir les paramètres indispensables pour la phase de l'analyse acoustique. Ces coefficients sont extraits des fichiers **wav** et sur des fenêtres de 25ms grâce à l'outil **HCopy** en se servant du fichier de configuration comme paramètre d'entrée.

config - Bloc-notes

Fichier Edition Format Affichage ?

```
#
# Example of an acoustical analysis configuration file
#
#SOURCEFORMAT = TIMIT # same as -F TIMIT
#SOURCEFORMAT = HTK # Gives the format of the speech files
SOURCEFORMAT = WAV


# Unit = 0.1 micro-second :

WINDOWSIZE = 250000.0 # = 25 ms = length of a time frame
TARGETRATE = 100000.0 # = 10 ms = frame periodicity
ENORMALISE=F
USEHAMMING = T # Use of Hamming function for windowing frames
ZMEANSOURCE=T
PREEMCOEF = 0.97 # Pre-emphasis coefficient
NUMCHANS = 26 # Number of filterbank channels
CEPLIFTER = 22 # Length of cepstral liftering


TARGETKIND = MFCC_D_A
NUMCEPS = 12
```

**Figure IV.4:** Fichier de configuration pour la phase de l'analyse acoustique.

### IV.4.3.2 Modélisation par MMC sous HTK

La topologie MMC choisie est de type gauche droit à 5 états dont les transitions autorisées sont décrites dans la Figure IV.5 et initialisées dans la matrice de transition.

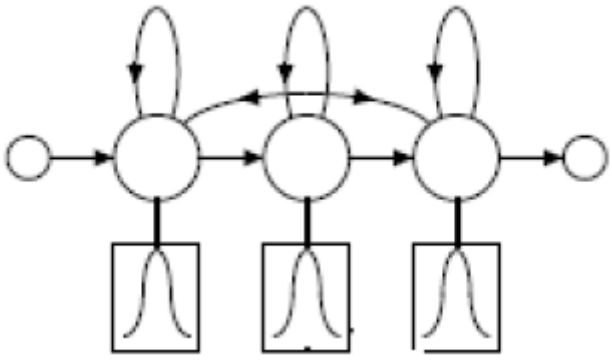


**Figure IV.5:** Modèle de Markov Cachés utilisé.

La moyenne est initialisée à 0 et la variance à 1, comme le montre le fichier prototype d'initialisation (Figure IV.6). Ces paramètres du modèle MMC seront ré-estimées par la suite lors de la phase d'apprentissage.

sab3a7

```
~o <VecSize> 36 <MFCC_D_A>
~h "sab3a7"
<BeginHMM>
  <NumStates> 7
  <State> 2 <NumMixes> 1
  <Mixture> 1 1.0000
    <Mean> 36
      0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
    <Variance> 36
      1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
  <State> 3 <NumMixes> 1
  <Mixture> 1 1.0000
    <Mean> 36
      0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
    <Variance> 36
      1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
  <State> 4 <NumMixes> 1
  <Mixture> 1 1.0000
    <Mean> 36
      0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
    <Variance> 36
      1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
  <State> 5 <NumMixes> 1
  <Mixture> 1 1.0000
    <Mean> 36
      0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
    <Variance> 36
      1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
  <State> 6 <NumMixes> 1
  <Mixture> 1 1.0000
    <Mean> 36
      0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
    <Variance> 36
      1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
  <TransP> 7
   0.0   1.0   0.0   0.0   0.0   0.0   0.0
   0.0   0.6   0.4   0.0   0.0   0.0   0.0
   0.0   0.0   0.6   0.4   0.0   0.0   0.0
   0.0   0.0   0.0   0.6   0.4   0.0   0.0
   0.0   0.0   0.0   0.0   0.6   0.4   0.0
   0.0   0.0   0.0   0.0   0.0   0.6   0.4
   0.0   0.0   0.0   0.0   0.0   0.0   0.0
<EndHMM>
```

**Figure IV.6:** Fichier prototype d'initialisation du mot «**سبعة**».

La phase d'apprentissage permet de constituer la base de données des modèles de référence du système. La qualité de cette modélisation conditionne en grande partie les résultats du système de reconnaissance de la parole. L'apprentissage est réalisé sous HTK en deux étapes majeures : l'initialisation et la ré-estimation. Pour cela, On utilise deux outils: **HCompV** et **HERest**. La phase d'initialisation des modèles MMC par l'outil **HCompV**, permet de mettre à jour la moyenne et la variance qui valent, avant cette étape, respectivement, 0 et 1. Cette mise à jour est réalisée sur l'ensemble des données du corpus d'apprentissage permettant d'aboutir, à la fin, à des valeurs globales qui seront clonées pour chaque état des modèles MMC. Ensuite, on obtient dans le répertoire **hmm0** un nouveau fichier prototype contenant des valeurs globales de la moyenne et de la variance.

Tous les mots seront ainsi initialisés aux mêmes valeurs de moyenne et de variance. A noter également que la mise à jour des variances est effectuée par défaut avec la commande **HCompV,** tandis que pour ré-estimer la moyenne, l'option **–m** devient indispensable. Le raffinement des modèles MMC consiste à ré-estimer leurs paramètres (moyenne et variance) suivant l'algorithme de Baum Welch grâce à l'outil **HERest** (la ré-estimation des modèles MMC contenu dans le répertoire **hmmi** est sauvegarde dans le répertoire **hmm i +1** à chaque

itération **i**). Ensuite, on généré un autre fichier **modeles0** dans un autre répertoire**.** Les modèles contenus dans ce fichier seront ré-estimés suite à deux itérations de l'algorithme de Baum Welch représenté par l'outil **HERest**. Les derniers paramètres estimés, à ce stade, sont sauvegardés dans le répertoire **hmm-final**. Deux itérations de l'algorithme de Baum Welch permettent de ré-estimer les modèles. Ainsi s'achève la phase d'apprentissage des modèles **MMC** avec une seule gaussienne.

Les modèles obtenus peuvent être améliorés par utilisation de densités de probabilités d'émission multi-gaussiennes au lieu de se contenter d'une simple loi normale. Cela permet d'´eviter certaines hypothèses grossières sur la forme de la densité si le nombre de gaussiennes est suffisant. En effet, le choix du nombre optimal de gaussiennes est un problème difficile. Un outil **d'HTK, HHEd** réalise l'augmentation du nombre de gaussiennes, où on augmente progressivement le nombre de gaussiennes (1, 2, 4, 8, 12, 16). Chaque augmentation de gaussienne est suivie de deux ré-estimations des modèles avec **HERest, HERest**. Suite à cette procédure les modèles sont de plus en plus précis. Le seul inconvénient est la charge des calculs qui augmente à son tour.

#### IV.4.3.3 Reconnaissance

Le processus de décodage consiste à comparer le signal de mot à identifier avec ce lui de la base de référence. Le module de décodage de la parole, **HVite**, utilise l'algorithme de Viterbi pour trouver la séquence d'états la plus probable correspondant aux paramètres observés et en déduire les mots acoustiques correspondantes. Le décodage est réalisé par l'algorithme de Viterbi.

## IV.5 Evaluation des résultats

### IV.5.1 Mesures de la qualité de la parole par logiciel PESQ

La qualité des signaux vocaux (parole) numérique a été fortement réduite par des algorithmes du codage hautement spécialisés, sachant que les codages et décodage G.729 ne fournissent plus le signal d'origine contrairement au codec Speex. Une solution est offerte par la « Perceptual Evaluation of Speech Quality » (PESQ) avec la méthode de mesure publiée en 2001 par l'Union Internationale des Télécommunications sous la recommandation ITU-T P.862. Cette solution a été conçue sous la forme d'un algorithme permettant une évaluation des signaux vocaux par comparaison avec le signal de référence.

Notre comparaison a été faite sur la base de données ARADIGIT originale et codée par G.729. Les résultats obtenus sont illustrés dans le Table IV.2.

| | |
|---|---|
| 1er test | MOS = 2.974 |
| 2eme test | MOS = 3.405 |
| 3eme test | MOS = 3.178 |
| **MOSmoy** | **MOS = 3.185** |

**Table IV.2:** Qualité de la parole transcodée G.729 par PESQ.

Nous remarquons d'après les résultants de test, la qualité décrite par PESQ (MOS moy = 3.185) varie dans les normes de l'UIT, soit entre 2.5 à 4.3.

### IV.5.2 Influence des codecs sur le taux de la RAP

Avant de procéder à la phase d'évaluation des performances de la RAP transcodée nous examinons d'abord le chemin optimal qui donne les meilleures performances de reconnaissance de la RAP sans codec.

Les résultats obtenus après l'exécution du HTK avec la base de données ARADIGIT sans codec montrent que les performances de la RAP, par rapport à la base ARADIGIT, sont meilleures dans le cas de 36 coefficients MFCC, une gaussienne et 5 états d'HMM le taux de reconnaissance correct est de 97.5%, comme le montre le tableau suivant.

```
====================== HTK Results Analysis ==================
  Date: Sun May 01 15:11:00 2016
  Ref : Base_données\Test\Lab\test_référence.txt
  Rec : Reconnaissance\reconnaissance.txt
------------------------ Overall Results --------------------
SENT: %Correct=97.50 [H=39, S=1, N=40]
WORD: %Corr=97.50, Acc=97.50 [H=39, D=0, S=1, I=0, N=40]
------------------------ Confusion Matrix -------------------
         s    w    i    t    a    k    s    s    t    t
         i    a    t    h    r    h    i    a    h    i
         f    h    h    a    b    a    t    b    a    s
         f    i    n    l    a    m    a    3    m    s
         r    d    a    a    3    s    6    a    a    e   Del [ %c / %e]
siff     4    0    0    0    0    0    0    0    0    0     0
wahi     0    4    0    0    0    0    0    0    0    0     0
ithn     0    0    4    0    0    0    0    0    0    0     0
thal     0    0    0    3    0    0    0    1    0    0     0 [75.0/2.5]
arba     0    0    0    0    4    0    0    0    0    0     0
kham     0    0    0    0    0    4    0    0    0    0     0
sita     0    0    0    0    0    0    4    0    0    0     0
sab3     0    0    0    0    0    0    0    4    0    0     0
tham     0    0    0    0    0    0    0    0    4    0     0
tiss     0    0    0    0    0    0    0    0    0    4     0
Ins      0    0    0    0    0    0    0    0    0    0
==============================================================
```

**Table IV.3:** Résultat d'exécution du HTK avec ARADIGIT.

Le tableau IV.3 montre un exemple du résultat de la reconnaissance généré par HTK. La ligne appelée (SENT) donne le taux de reconnaissance de la phrase (de % Corr =97.50), la ligne surnommée (WORD) donne le taux de reconnaissance des mots (% Corr = 97.50). Dans ce travail le taux de reconnaissance dans les deux cas (phrase et mot) est le même, parce que notre système de reconnaissance est orienté vers tâche reconnaissance de la parole isolée, ainsi la grammaire ne permet la reconnaissance «phrases» qu'avec un seul mot (en dehors des silences), donc seulement la première ligne (SENT) doit être considéré ici. H = 8 donne le nombre de données de test correctement reconnu, S = 2 le nombre d'erreurs de substitution et N = 10 le nombre total de données de test.

### IV.5.2.1 Influence du Speex sur le taux de la RAP

On applique le HTK sur la base de données ARADIGIT transcodée Speex . Les résultats obtenus sont illustrés dans la **Table IV.4**

```
====================== HTK Results Analysis ================
  Date: Tue May 03 13:17:23 2016
  Ref : Base_données\Test\Lab\test_référence.txt
  Rec : Reconnaissance\reconnaissance.txt
------------------------ Overall Results --------------------
SENT: %Correct=97.50 [H=39, S=1, N=40]
WORD: %Corr=97.50, Acc=97.50 [H=39, D=0, S=1, I=0, N=40]
------------------------ Confusion Matrix -------------------
       s   w   i   t   a   k   s   s   t   t
       i   a   t   h   r   h   i   a   h   i
       f   h   h   a   b   a   t   b   a   s
       f   i   n   l   a   m   a   3   m   s
       r   d   a   a   3   s   6   a   a   e   Del [ %c / %e]
siff   4   0   0   0   0   0   0   0   0   0     0
wahi   0   4   0   0   0   0   0   0   0   0     0
ithn   0   0   4   0   0   0   0   0   0   0     0
thal   0   0   0   4   0   0   0   0   0   0     0
arba   0   0   0   0   4   0   0   0   0   0     0
kham   0   0   0   0   0   4   0   0   0   0     0
sita   0   0   0   0   0   0   4   0   0   0     0
sab3   0   0   0   1   0   0   0   3   0   0     0 [75.0/2.5]
tham   0   0   0   0   0   0   0   0   4   0     0
tiss   0   0   0   0   0   0   0   0   0   4     0
Ins    0   0   0   0   0   0   0   0   0   0
=============================================================
```

**Table IV.4:** Résultat d'exécution du HTK avec ARADIGIT transcodée Speex.

Les paramètres le plus important c'est le paramètre (H) qui représente le nombre de données de test correctement reconnues, (S) qui représente le nombre d'erreurs de substitution et le nombre total de tests données (N).

Les résultats montrent que les performances obtenus avec de la parole avec Speex transcodée égal à les performances obtenus sans codec

### IV.5.2.2 Influence du G.729 sur le taux de la RAP

On applique le HTK sur la base de données ARADIGIT transcodée G.729 . Les résultats obtenus sont illustrés dans la **Table IV.5** :

```
====================== HTK Results Analysis ==================
  Date: Tue May 03 13:06:45 2016
  Ref : Base_données\Test\Lab\test_référence.txt
  Rec : Reconnaissance\reconnaissance.txt
------------------------ Overall Results -------------------
SENT: %Correct=95.00 [H=38, S=2, N=40]
WORD: %Corr=95.00, Acc=95.00 [H=38, D=0, S=2, I=0, N=40]
------------------------ Confusion Matrix -------------------
       s   w   i   t   a   k   s   s   t   t
       i   a   t   h   r   h   i   a   h   i
       f   h   h   a   b   a   t   b   a   s
       f   i   n   l   a   m   a   3   m   s
       r   d   a   a   3   s   6   a   a   e   Del [ %c / %e]
siff   3   0   0   0   0   0   1   0   0   0     0 [75.0/2.5]
wahi   0   4   0   0   0   0   0   0   0   0     0
ithn   0   0   4   0   0   0   0   0   0   0     0
thal   0   0   0   4   0   0   0   0   0   0     0
arba   0   0   0   0   4   0   0   0   0   0     0
kham   0   0   0   0   0   4   0   0   0   0     0
sita   0   0   0   0   0   0   4   0   0   0     0
sab3   0   0   0   0   0   0   0   4   0   0     0
tham   0   0   0   0   0   1   0   0   3   0     0 [75.0/2.5]
tiss   0   0   0   0   0   0   0   0   0   4     0
Ins    0   0   0   0   0   0   0   0   0   0
=============================================================
```

**Table IV.5:** Résultat d'exécution du HTK avec ARADIGIT transcodée G.729.

le Table IV.5 représente le taux de reconnaissance de la base ARADIGIT transcodée G.729 La première ligne (SENT) donne le taux de la reconnaissance (% Corriger = 95.00), la deuxième (WORD) donne le taux de reconnaissance de mots (% Corr = 95.00). Dans notre cas, les 2 tarifs sont les mêmes parce que notre tâche la grammaire ne permet «phrases» avec un seul mot , est une tâche de reconnaissance des mots isolée. Seule la première ligne (SENT) doit être considéré ici. H = 38 donne le nombre de données de test correctement reconnues, S = 2 le nombre d'erreurs de substitution et N = 40 le nombre total de tests données, d'après cette résultat , les performance du système de reconnaissance avec le G.729 codec est diminué

## IV.6 Conclusion

Dans ce chapitre nous avons présenté en premier temps l'évaluation de notre système de reconnaissance de la parole. Les expériences d'évaluation sur la base de données ARADIGIT non codée, la base de données transcodée G.729 et Speex ont montré que les performances du système de reconnaissance se dégradent avec les caractéristiques de codec utilisé (type de Filtre utilisé, durée de trame de composition du signal échantillonné, etc.).

# CHAPITRE V :

# Présentation du logiciel

## V.1 Introduction

Afin de simplifier la manipulation des différentes techniques développées précédemment. Nous avons réalisé une application software (Application en vue de la reconnaissance automatique de la parole transcodée Speex et G.729), sous l'environnement Matlab.

## V.2 Architecture du logicielle

L'application offre plusieurs fonctionnalités rencontrées dans le domaine du traitement du signal de parole (enregistrement du son, visualisation du signal de parole, aussi la paramétrisation du signal vocal...etc.). Il permet également de manipuler les différentes tâches relatives à la parole transcodée Speex et G.729, comme codage, décodage du signal parole et la reconnaissance de la parole transcodée. La Figure V.1 montre la fenêtre principale de l'application développée.

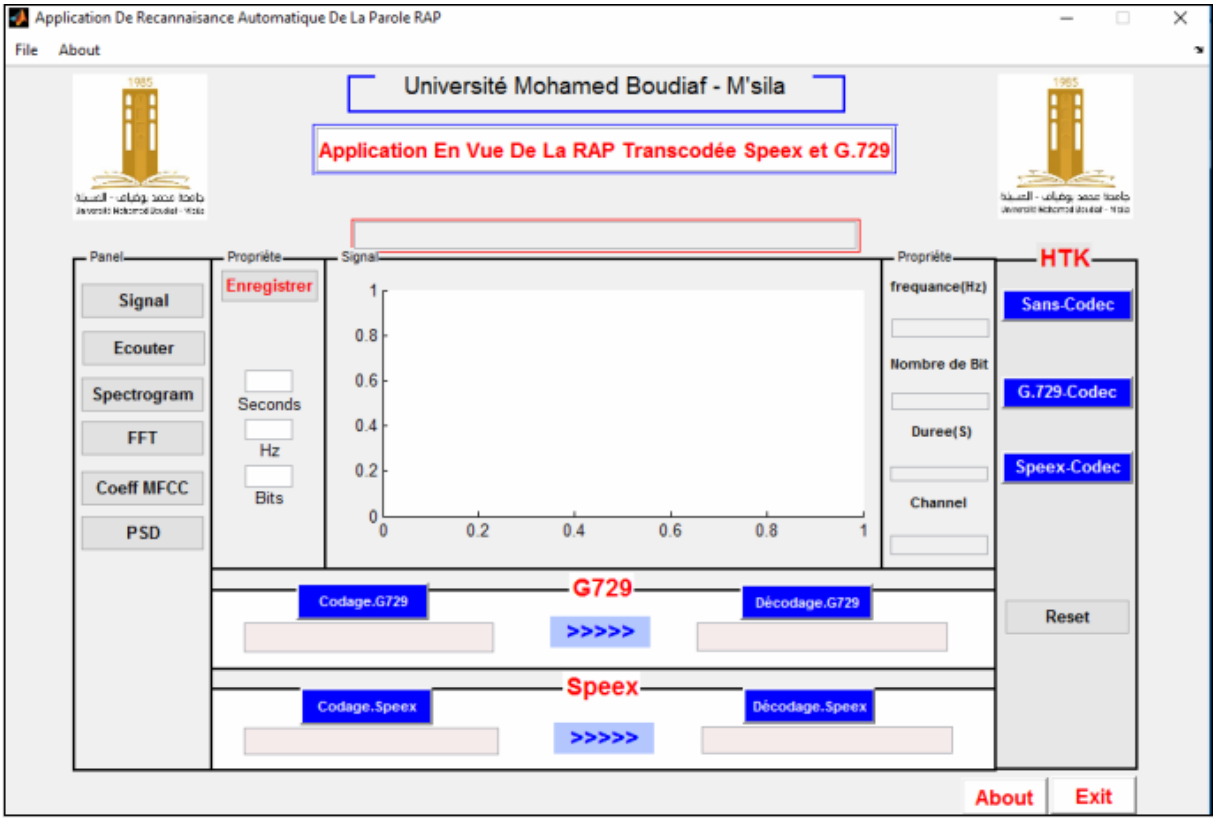



**Figure V.1**: Logiciel de la RAP transcodée Speex et G.729.

La figure V.1 montre que notre interface permet l'acquisition, la visualisation, l'écoute du signal de parole, l'enregistrement des signales audio, l'analyse du signal de parole. Elle permet aussi l'extraction des paramètres MFCC, le codage et décodage de la parole en

utilisant G.729 et Speex, et la compilation du système RAP sous HTK, En plus, l'interface fait appel à une autre fenêtre du bouton About (voir figure V.2).

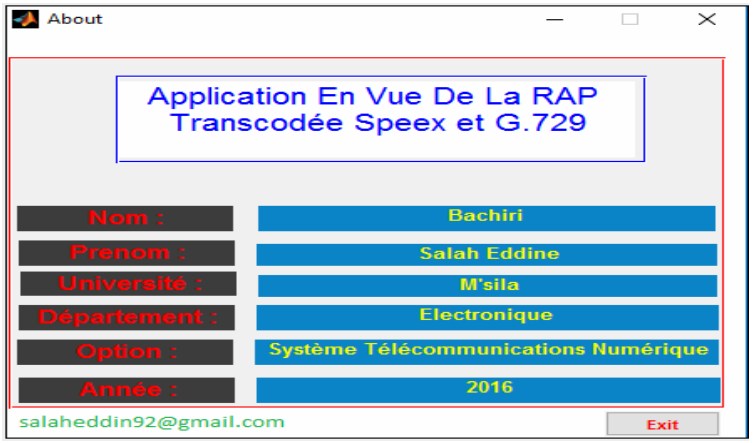



**Figure V.2:** Fenêtre correspondante au bouton About.

## V.3 Différentes applications du logiciel

Pour mieux comprendre le fonctionnement de notre application, on sélectionne sur la fenêtre principale deux parties importantes, l'une est la partie de l'analyse du signal surnommée FP_Partie_01 (Fenêtre Principale Partie 01) colorée en rouge, et l'autre est la partie de la reconnaissance sous l'outil HTK avec les codecs G.729 et Speex appelée Partie_02 (Fenêtre Principale Partie 02) en ver, comme le montre la figure V.3.

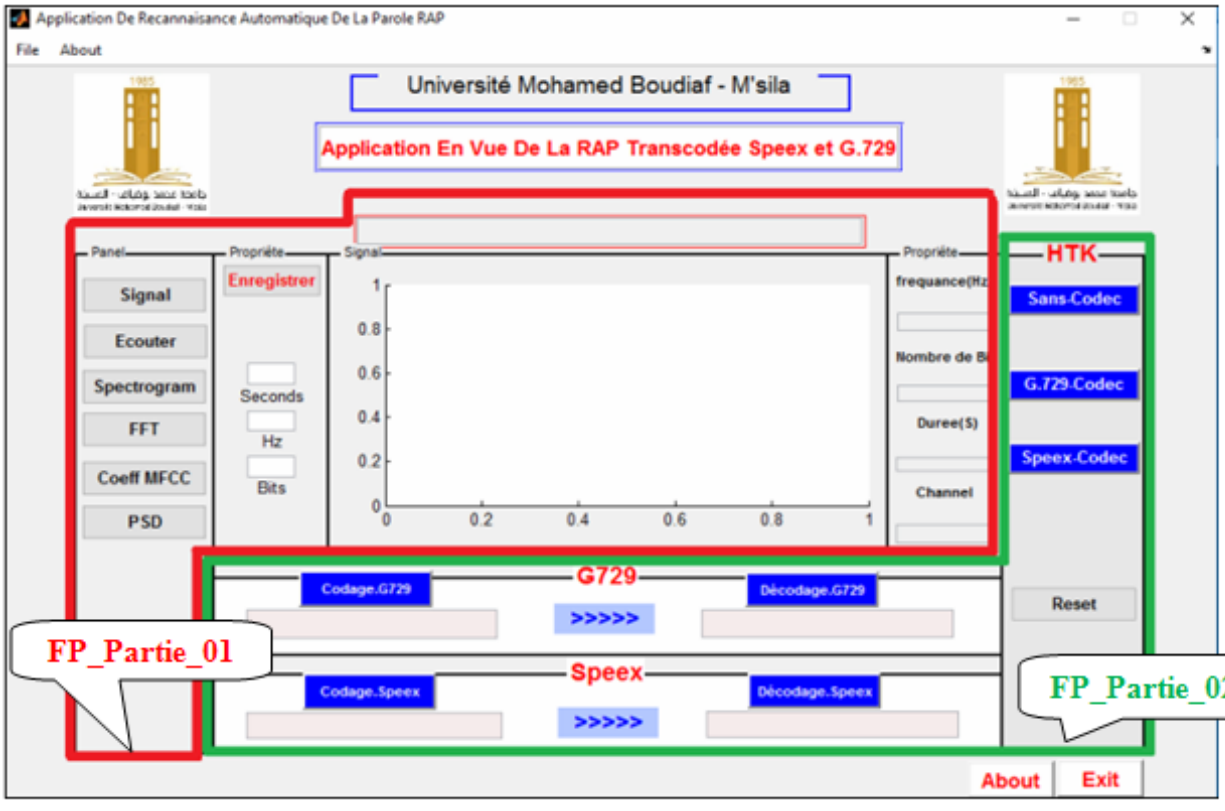



**Figure V.3:** Deux parties essentielles de la fenêtre principale.

### V.3.1 Partie 01 de la fenêtre principale

La figure suivante représente les différents boutons de la partie 01de la fenêtre principale.

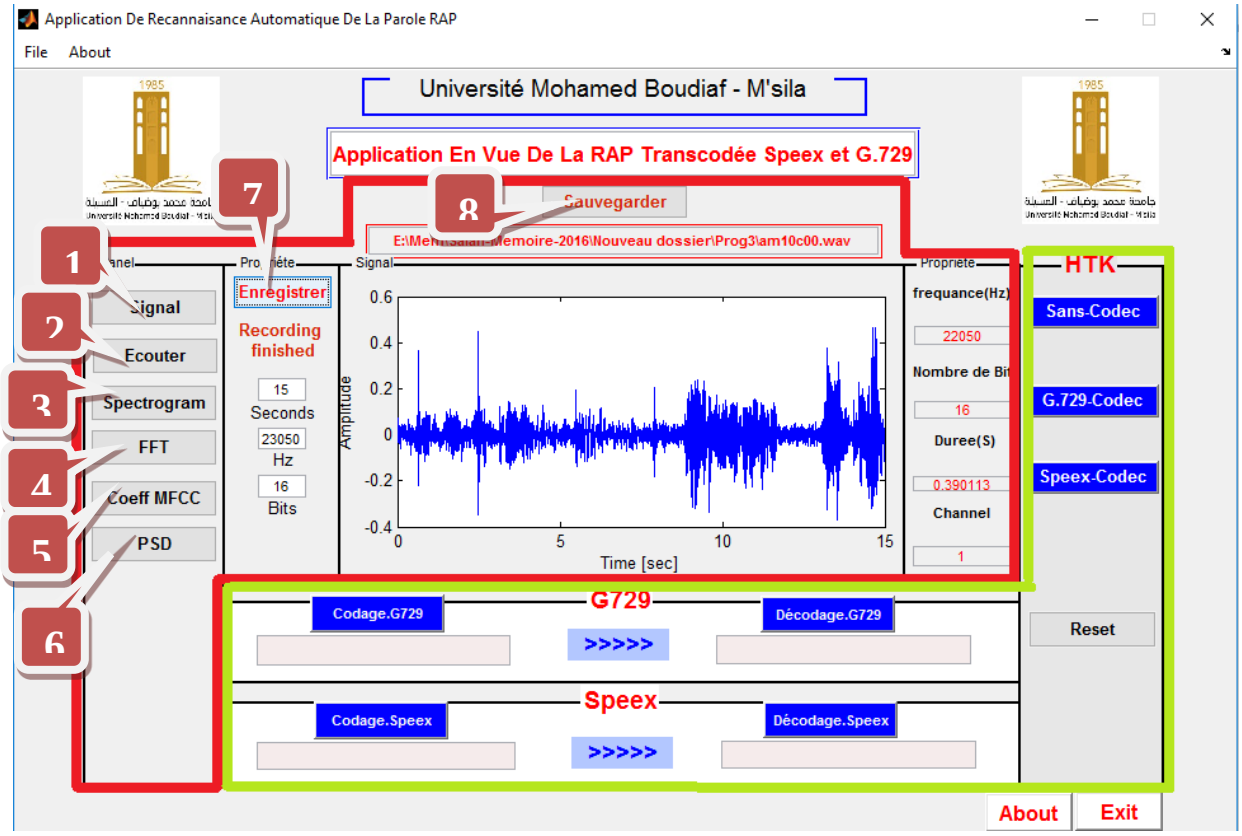



**Figure V.4:** Partie 01 de la fenêtre principale.

Dans cette partie de la fenêtre principale on trouve les différents boutons suivants:

1. **Signal:** Pour charger un signal de parole, cliquer sur le bouton Signal, puis choisir le fichier qui vous voulez ouvrir. L'extension du fichier est par défaut (**.**wav), c'est-à-dire qu'on peut appeler que les fichiers audio de type .wav. Après le chargement du fichier son et la visualisation du signal audio d'autres paramètres sont affichés automatiquement comme; la fréquence, le nombre de bit, la durée de signal et le nombre de chanel, ainsi que le chemin de répertoire où se trouve le fichier.
2. **Ecoute du signal:** L'écoute du signal de parole permet d'apprécier la qualité du signal, et de détecter les problèmes induits par le système de prise de son automatique. Ceci se fait en cliquant sur le bouton «Ecouter».
3. **Spectrogramme:** Ce bouton génère le spectrogramme du signal parole, qui permet de visualiser le signal (amplitude/temps) comme une forme

tridimensionnelle (amplitude/fréquence/temps), où la fréquence est représentée comme une fonction du temps alors que l'amplitude est donnée par le niveau de gris (ou une variation de couleur) de chaque point. Il utilise la technique de la transformée de Fourier à court terme et donc du calcul de spectres. La figure IV.6 suivante illustre le spectrogramme du chiffre «ستة».

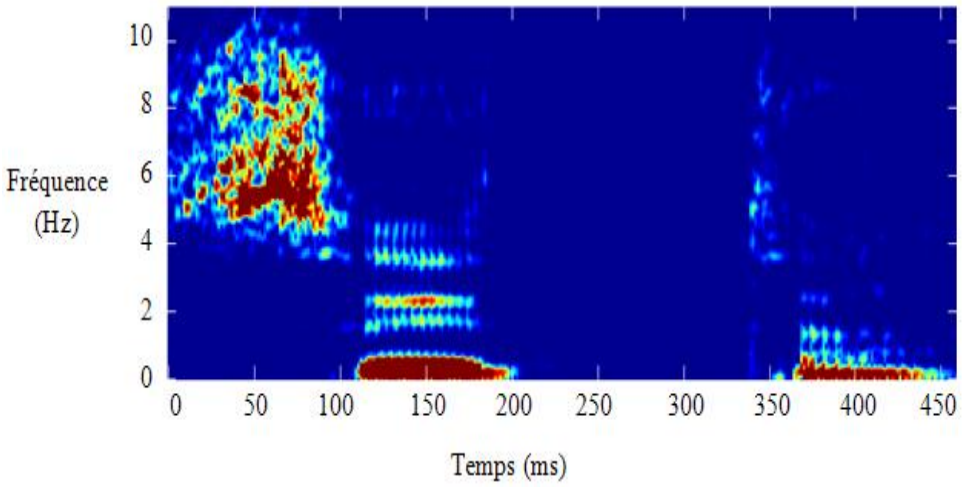



**Figure V.5:** Spectrogramme du chiffre «ستة».

4. **FFT:** La transformation de Fourier rapide (en anglais, Fast Fourier Transform) est un algorithme de calcul de la transformation de Fourier discrète (TFD). Le bouton (FFT) exécute et visualise la FFT de signal parole choisie.
5. **Coeff_MFCC:** La procédure d'extraction des coefficients MFCC est comme suite:
    a) Charger un signal de parole, en cliquant sur le bouton (Signal) ;
    b) Choisir les différents paramètres, préaccentuation du signal, nombre des coefficients MFCC, nombre des filtres Mel, le type de la fenêtre et sa taille;
    c) Appuyez sur le bouton (Coeff_MFCC), pour visualiser la représentation acoustique des coefficients MFCC.

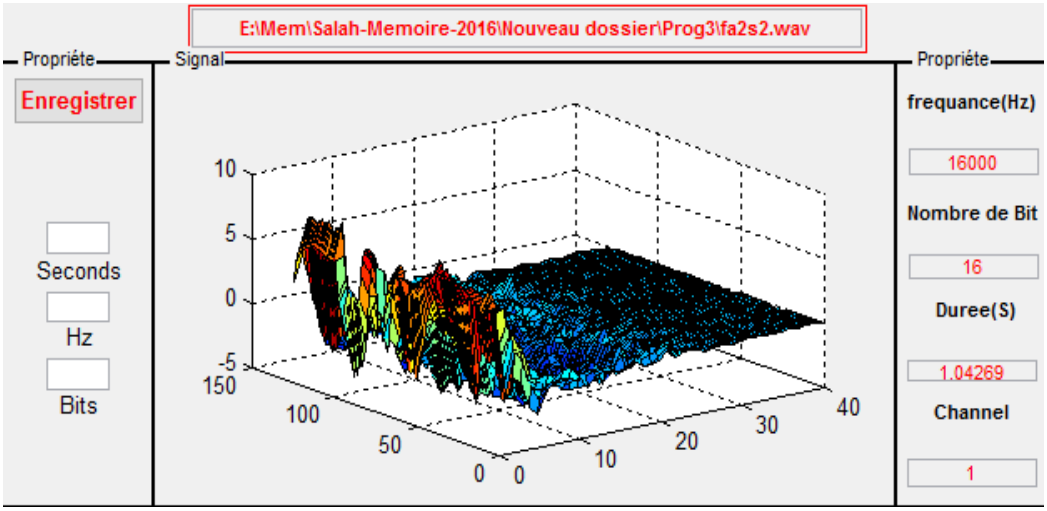



**Figure V.6:** Coefficient MFCC du fichier du chiffre «ستة».

6. **PSD :** On définit la densité spectrale de puissance (en anglais, Power Spectral Densité) comme étant le carré du module de la transformée de Fourier , pour visualise le PSD du signal audio en appuyant sur le bouton (PSD).
7. **Enregistrer:** Le bouton 7 appelé (Enregistrer) offre la possibilité d'enregistrer un signal de parole, après la modification des paramètres comme; La durée, la fréquence et le nombre de bit. Lors de la validation de ce bouton, un texte box apparaitre indique que d'enregistrement est en cours. Après l'enregistrement la visualisation du signale st automatique.
8. **Sauvegarder:** Après l'enregistrement du signal on peut le sauvegarder comme un fichier (.wav), en cliquant sur le bouton (Sauvegarder), l'utilisateur est libre pour le choix d'emplacement d'enregistrement du fichier audio.

### V.3.2 Partie 02 de la fenêtre principale

Les différents boutons de la partie 02 de la fenêtre principale sont représentés par figure suivante.

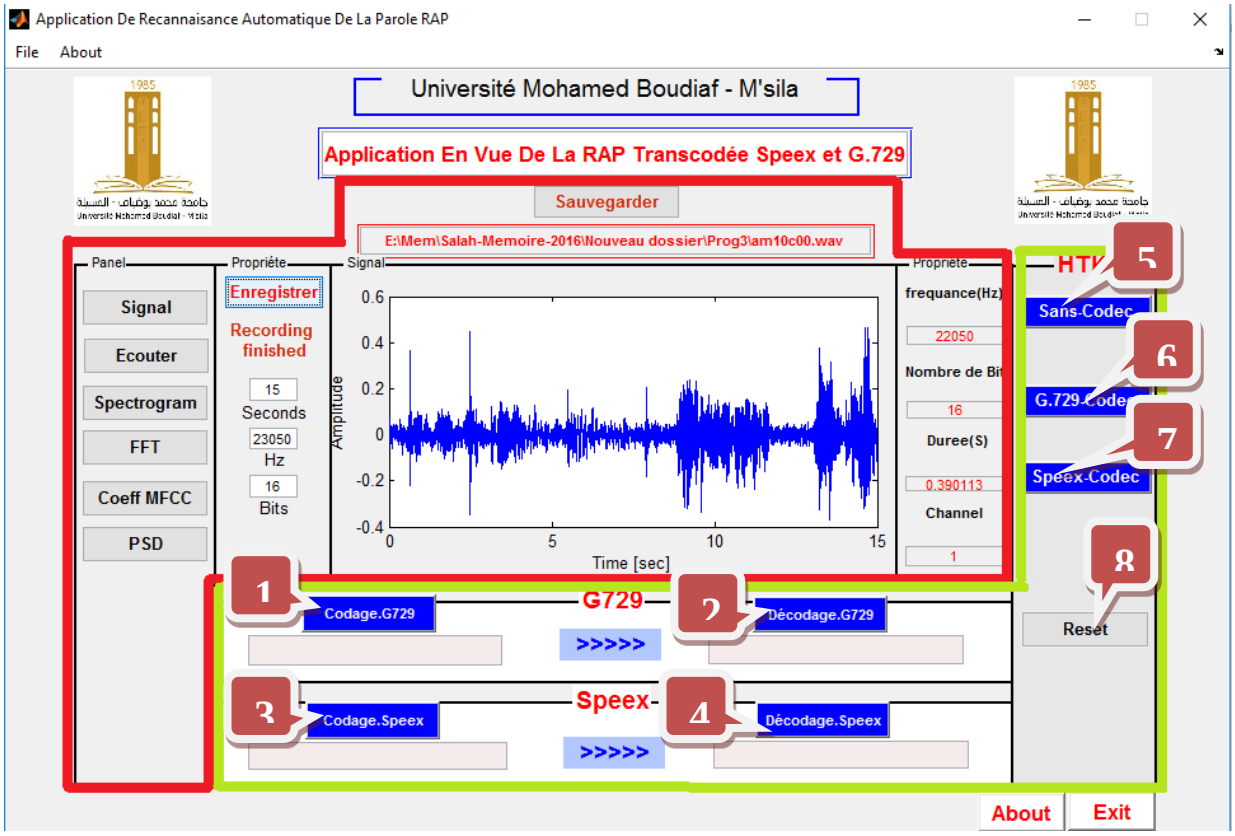



**Figure V.7:** Partie 02 de la fenêtre principale.

Dans la partie 02 de la fenêtre principale on trouve les différents boutons suivants:

1. **Codage.G729:** Ce bouton exécute le fichier d'encodeur G.729.exe, crée en langage perl, pendant l'exécution différents répertoires et liens sont combinés pour lancer l'algorithme d'encodage, le fichier obtenue après la compilation du G.729 code source est illustré par la figure suivante.

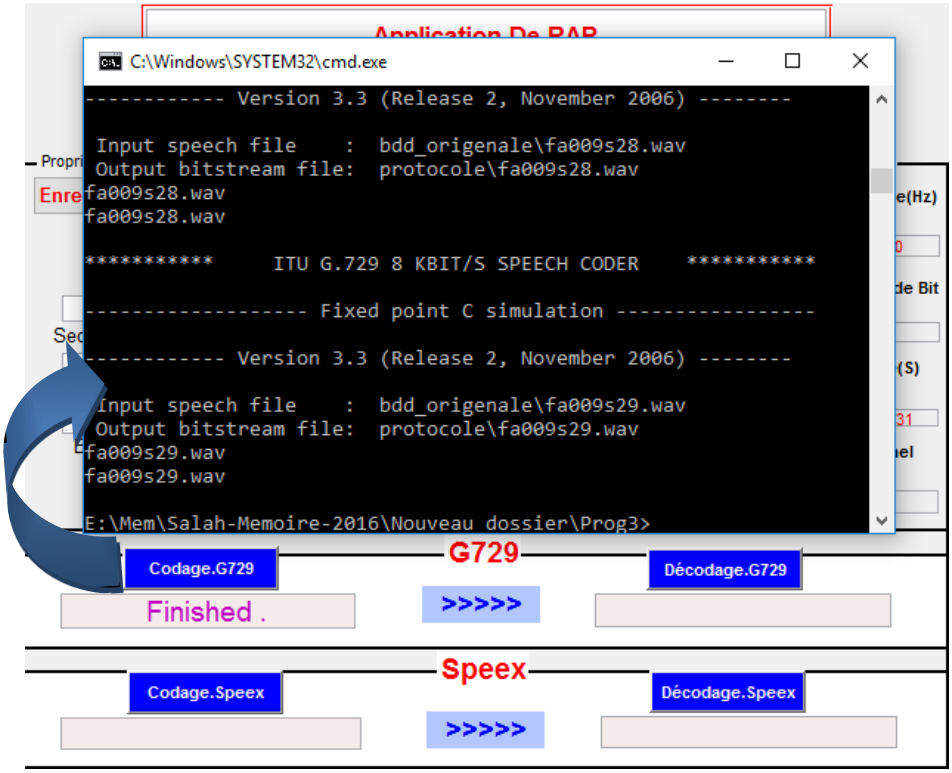



**Figure V.8 :** Encodage de la base des données ARADIGIT par G.729.

2. **Décodage.G729:** Ce bouton exécute le fichier perl pour décoder les fichiers encodés par le bouton 1 (Codage.G729).

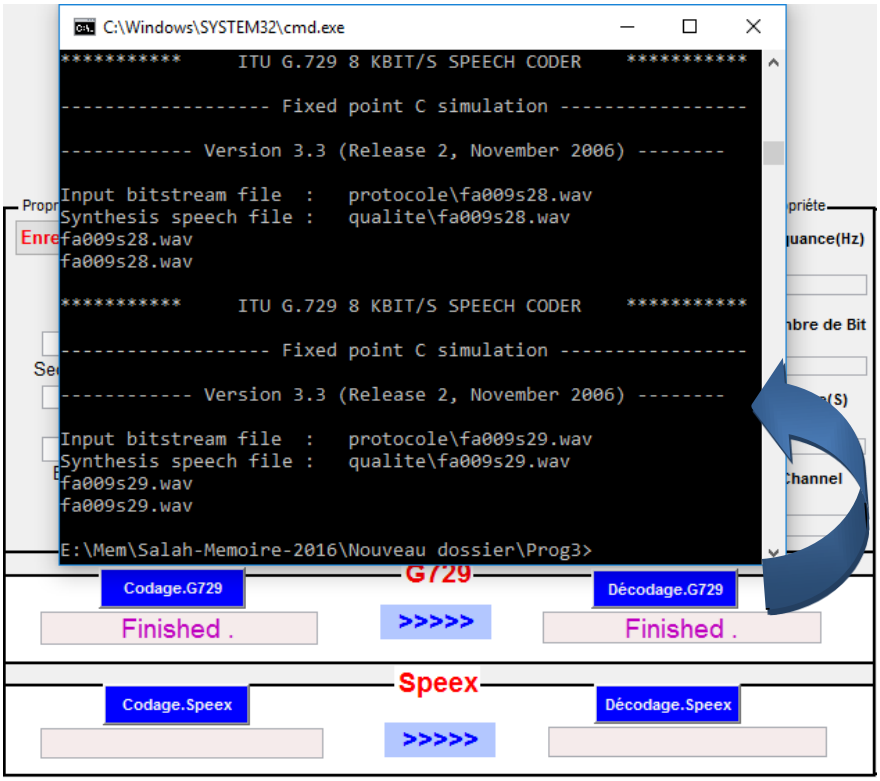



**Figure V.9:** Décodage des fichiers encoder par le décode G729.

3. **Codage.Speex :** Bouton 3 représente l'encodeur Speex, qui est un codec open source, la compilation et l'exécution de Speex est effectué sous langage perl la figure ci-dessous est la représentation du Speex pendant l'encodage de la base de données ARADIGIT.
4. **Decodage.Speex:** C'est le bouton 4 sur l'interface graphique, qui exécute le decodeur.Speex.exe sous perl pour décoder les fichiers encodés le bouton 3.
5. **HTK.Sans.Codec:** Bouton 5 sur l'interface exécute une boîte à outils HTK (Hidden Markov Model Toolkit) pour la construction du system RAP sans codecs et la manipulation de modèles de Markov cachés.
6. **HTK.G729:** Ce bouton fait appel à la plateforme HTK pour la construction du system RAP transcodée G.729, donc ce bouton manipule les fichiers déjà codés et décodés par le codec G.729 en vue de la RAP sous HTK (voir figure V.10).

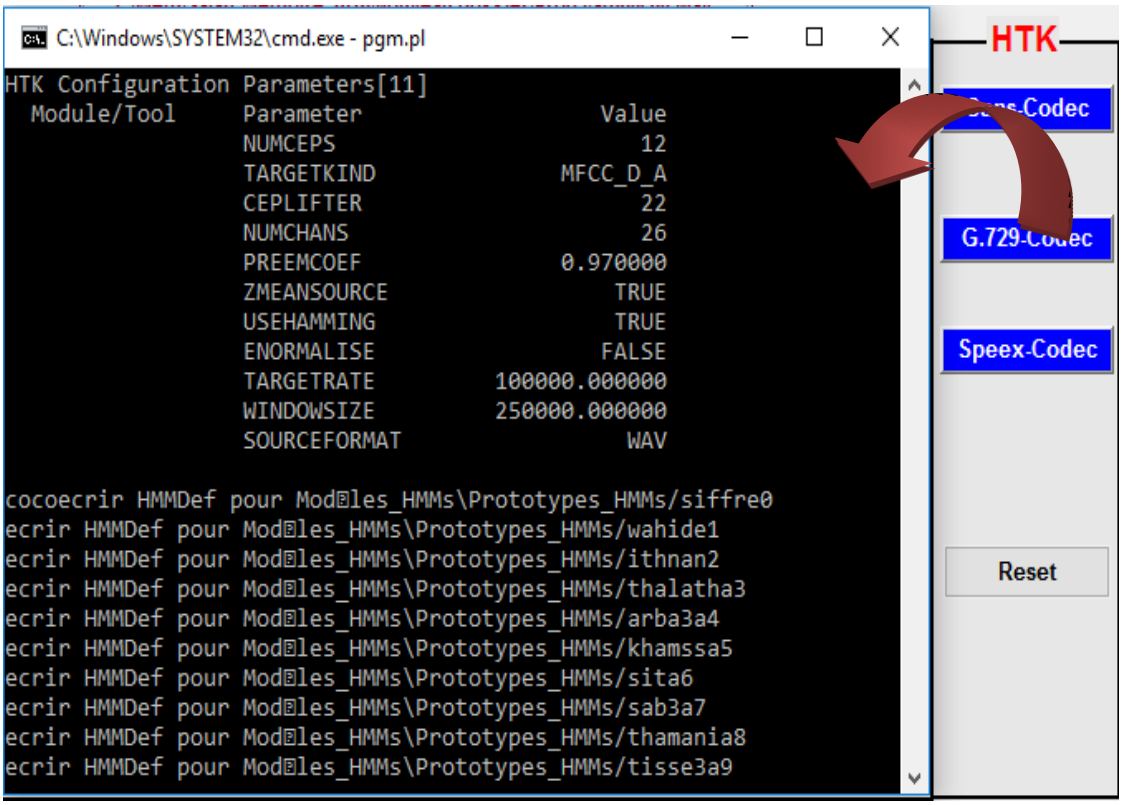
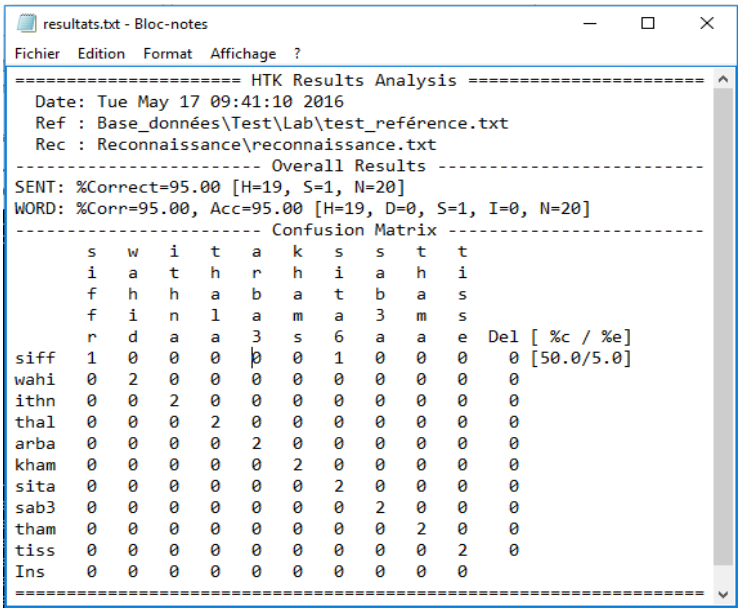



**Figure V.10:** Compilation du HTK avec le codec G.729.

Le résultat obtenu après la compilation du bouton HTK.G729 est représenté par la figure V.11.



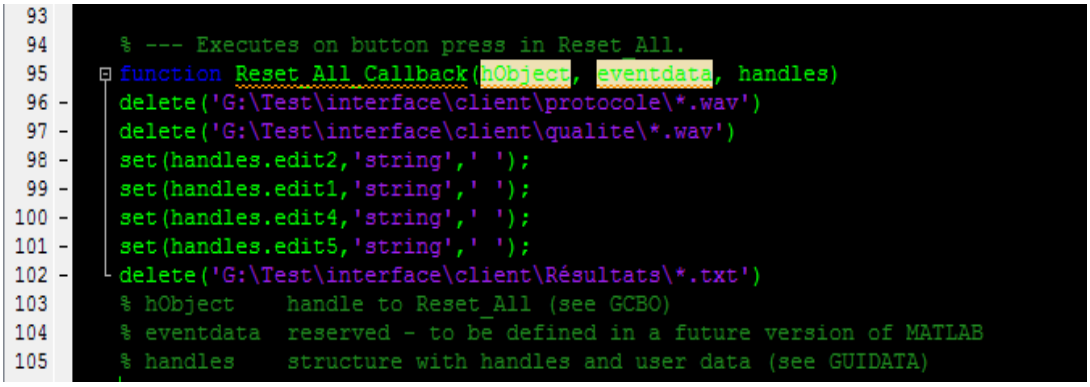
resultats.txt - Bloc-notes

Fichier Edition Format Affichage ?

```
====================== HTK Results Analysis =======================
  Date: Tue May 17 09:41:10 2016
  Ref : Base_données\Test\Lab\test_reférence.txt
  Rec : Reconnaissance\reconnaissance.txt
------------------------ Overall Results --------------------------
SENT: %Correct=95.00 [H=19, S=1, N=20]
WORD: %Corr=95.00, Acc=95.00 [H=19, D=0, S=1, I=0, N=20]
------------------------ Confusion Matrix -------------------------
       s   w   i   t   a   k   s   s   t   t
       i   a   t   h   r   h   i   a   h   i
       f   h   h   a   b   a   t   b   a   s
       f   i   n   l   a   m   a   3   m   s
       r   d   a   a   3   s   6   a   a   e  Del [ %c / %e]
siff   1   0   0   0   0   0   1   0   0   0    0 [50.0/5.0]
wahi   0   2   0   0   0   0   0   0   0   0    0
ithn   0   0   2   0   0   0   0   0   0   0    0
thal   0   0   0   2   0   0   0   0   0   0    0
arba   0   0   0   0   2   0   0   0   0   0    0
kham   0   0   0   0   0   2   0   0   0   0    0
sita   0   0   0   0   0   0   2   0   0   0    0
sab3   0   0   0   0   0   0   0   2   0   0    0
tham   0   0   0   0   0   0   0   0   2   0    0
tiss   0   0   0   0   0   0   0   0   0   2    0
Ins    0   0   0   0   0   0   0   0   0   0
===================================================================
```

**Table V.1:** Résultat d'exécution de HTK avec codec G.729.

1. **HTK.Speex :** Ce bouton exécute la plateforme HTK avec de la parole transcodée Speex.
2. **Reset:** Bouton 8 à pour but de nettoyer et vider les dossiers et effacer les résultats précédemment obtenue, comme le montre sur la figure suivant.



```
93
94      % --- Executes on button press in Reset_All.
95      function Reset_All_Callback(hObject, eventdata, handles)
96 -    delete('G:\Test\interface\client\protocole\*.wav')
97 -    delete('G:\Test\interface\client\qualite\*.wav')
98 -    set(handles.edit2,'string',' ');
99 -    set(handles.edit1,'string',' ');
100 -   set(handles.edit4,'string',' ');
101 -   set(handles.edit5,'string',' ');
102 -   delete('G:\Test\interface\client\Résultats\*.txt')
103     % hObject    handle to Reset_All (see GCBO)
104     % eventdata  reserved - to be defined in a future version of MATLAB
105     % handles    structure with handles and user data (see GUIDATA)
```

**Figure V.11 :** Code source du bouton reset.

## V.4 Conclusion

Ce chapitre montre l'application développée au cours de notre travail. Cette application n'exige aucune condition particulière pour son utilisation, sauf l'utilisation d'un ordinateur. Elle donne des outils graphiques permettant de manipuler plusieurs tâches pour le traitement du signal de parole, et de compiler le system de la RAP sous la plateforme HTK avec de la parole transcodée Speex et G.729.

# Conclusion Générale

# Conclusion générale

L'objectif principal de ce travail est la conception d'une interface graphique après d'évaluer l'influence des codecs G.729 et Speex utilisés en VoIP, sur la reconnaissance automatique de la parole. L'application a porté sur reconnaissance de chiffres arabes, en utilisant l'approche de reconnaissance statistique basée sur les HMMs au moyen de la plateforme open source HTK (Hidden Markov Model Toolkit).

Pour atteindre cet objectif, une base de données transcodée G.729 et Speex a été obtenue à partir des chiffres arabes de la base de données ARADIGIT. Une série d'expériences portant sur la reconnaissance en milieu calme puis de la parole transcodée.

Nous avons représenté le signal de parole par les coefficients acoustiques suivants : MFCC et leurs dérivées premières et secondes. Pour modéliser les mots (chiffres arabes), nous avons utilisé les modèles de Markov caches HMM. Les paramètres de ce modèle sont estimés par l'algorithme itératif. Les performances de cette méthode sont meilleures lorsqu'on augmente le nombre de MFCC, le taux de reconnaissance augmente aussi avec l'augmentation de l'ordre du modèle et le nombre d'états.

Le taux de reconnaissance le plus élevé égal à 97.5% pour la base de données non codée en milieu calme. Il est de 95% pour la base de données de parole reconstituée après codage G.729. Les résultats montrent que les performances obtenus avec de la parole avec Speex transcodée égal à ceux sans codec est restent supérieurs à ceux obtenus avec de la parole transcodée G.729.

Cette étude montre que des efforts sont encore à faire sur le chemin de la reconnaissance distribuée. Les résultats restent également à améliorer concernant l'aspect application en temps réel. En effet, dans un processus à temps réel, il faudra tenir compte du temps de calcul qui peut être élevé. Il faudrait donc faire appel à des langages plus performants, tel que le langage C, pour permettre d'implémenter l'interface graphique pour des applications en reconnaissance distribuée.

# Annexes

# Annexe A :
# Description et Schémas de principe du Codec CS-ACELP G.729

## A.1 Introduction

La recommandation G.729de l'Union Internationale des Télécommunications normalise un codeur de parole CS-ACELP (Conjugate Structure - Algebraic Code Excited Linear Prediction)à8kbits/s [ITUG729]. Les deux figures suivantes donnent respectivement les schémas de principe du codeur et du décodeur de cette recommandation.

### A.2 Codeur G.729

Le codeurG.729, dont la circulation des signaux est donnée en figure A.1, opère sur des trames de parole de 10 millisecondes, qui correspondent à 80 échantillons numérisés sur 16 bits pour une fréquence d'échantillonnage de 8kHz. Le signal de parole est analysé à chaque trame pour extraire les coefficients du filtre de Prédiction Linéaire (LP) du 10 eme ordre, qui sont convertis en lignes de raies spectrales (LSP) et numérisés par quantification vectorielle prédictive à deux étapes. Par la suite, les paramètres d'excitation tels que la période de pitch, les index ainsi que les gains des dictionnaires, fixe et adaptatif, sont estimés sur la base de sous-trames de 40 échantillons, soit 5 millisecondes.

Le signal d'excitation est choisi au moyen d'une procédure de recherche par analyse et synthèse dans laquelle l'erreur, entre le signal vocal original et celui reconstruit, est minimisée en fonction d'une mesure de distorsion pondérée.

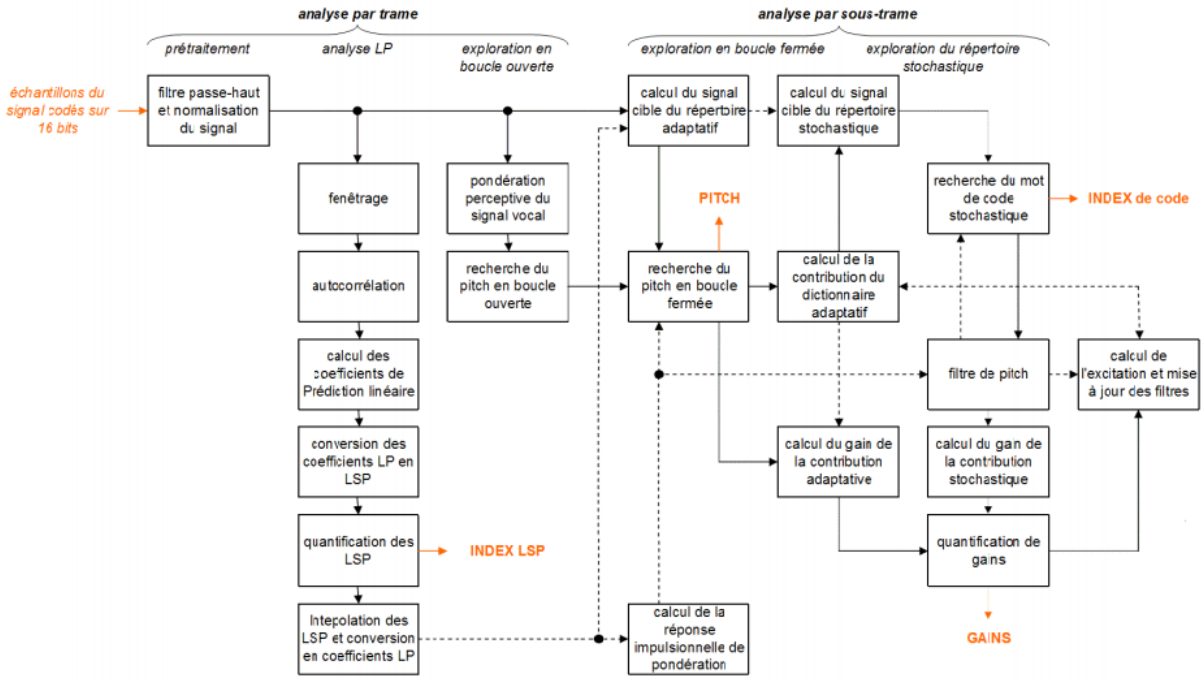



**Figure A.1** : Codeur CS-ACELP G.729.

## A.3 Décodeur G.729

Les paramètres de codage, correspondant à une trame vocale de 10 millisecondes, sont ensuite traités par le décodeur G.729 décrit par la figure A.2. Ainsi, Les coefficients LSP sont interpolés et reconvertis en coefficients de filtre de prédiction linéaire pour chaque sous-trame de 5millisecondes; l'excitation est construite par combinaison des codes vectoriels adaptatif et fixe, multipliés par leur gain respectif; enfin, le signal vocal est reconstitué par le filtre de synthèse LP et amélioré à l'aide d'un bloc de post-traitement suivi d'un filtre passe-haut et d'un échantillonneur-normalisateur.

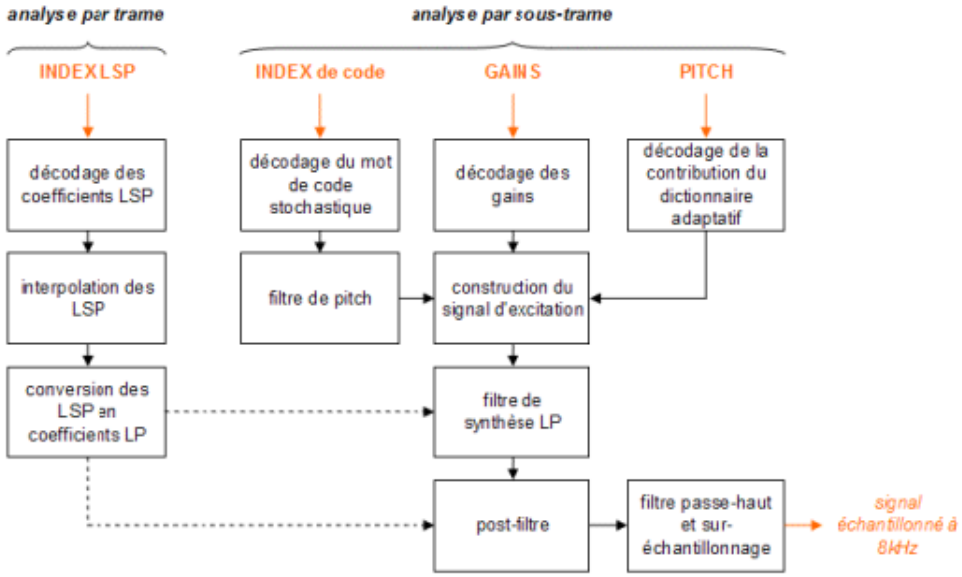



**Figure A.2** : Décodeur CS-ACELP G.729.

# Annexe B

# Codec Speex

## B.1 Introduction

Speex est un codec libre , a la différence de la plupart des autres codecs dédiés à la parole, Speex n'est pas fait pour une utilisation sur téléphone portable mais plutôt pour la VoIP et la compression dans des fichiers. Les principaux buts ont été de créer un codec optimisé pour la parole, qui associe une bonne compréhension du dialogue transmis, ainsi qu'un fort taux de compression des données possibles. Speex peut utiliser de nombreux débits et bande passante possibles

## B.2 Codeur Speex

Cette section présente des exemples de code pour la parole de codage et de décodage en utilisant l'API Speex. Les commandes peuvent être utilisées pour coder et décoder un fichier en appelant:

```
% sampleenc in_file.sw | sampledec out_file.sw
```

où les deux fichiers sont raw (sans en-tête) des fichiers codés à 16 bits par échantillon (dans la machine endianness naturelle).

sampleenc prend un fichier RAW 16 bits / échantillon, code et délivre un flux Speex à stdout. Notez que l'emballage utilisé est pas compatible avec celle de speexenc / speexdec.

## sampleenc.c

```
#include <speex/speex.h>
#include <stdio.h>

/*The frame size in hardcoded for this sample code but it doesn't have to be*/
#define FRAME_SIZE 160
int main(int argc, char **argv)
{
   char *inFile;
   FILE *fin;
   short in[FRAME_SIZE];
   float input[FRAME_SIZE];
   char cbits[200];
   int nbBytes;
   /*Holds the state of the encoder*/
   void *state;
   /*Holds bits so they can be read and written to by the Speex routines*/
   SpeexBits bits;
   int i, tmp;

   /*Create a new encoder state in narrowband mode*/
   state = speex_encoder_init(&speex_nb_mode);

   /*Set the quality to 8 (15 kbps)*/
   tmp=8;
   speex_encoder_ctl(state, SPEEX_SET_QUALITY, &tmp);

   inFile = argv[1];
   fin = fopen(inFile, "r");

   /*Initialization of the structure that holds the bits*/
   speex_bits_init(&bits);
   while (1)
   {
      /*Read a 16 bits/sample audio frame*/
      fread(in, sizeof(short), FRAME_SIZE, fin);
      if (feof(fin))
         break;
      /*Copy the 16 bits values to float so Speex can work on them*/
      for (i=0;i<FRAME_SIZE;i++)
         input[i]=in[i];

      /*Flush all the bits in the struct so we can encode a new frame*/
      speex_bits_reset(&bits);

      /*Encode the frame*/
      speex_encode(state, input, &bits);
      /*Copy the bits to an array of char that can be written*/
      nbBytes = speex_bits_write(&bits, cbits, 200);

      /*Write the size of the frame first. This is what sampledec expects but
       it's likely to be different in your own application*/
      fwrite(&nbBytes, sizeof(int), 1, stdout);
      /*Write the compressed data*/
      fwrite(cbits, 1, nbBytes, stdout);

   }

   /*Destroy the encoder state*/
   speex_encoder_destroy(state);
   /*Destroy the bit-packing struct*/
   speex_bits_destroy(&bits);
   fclose(fin);
   return 0;
}
```

## B.3 Decodeur Speex

sampledec lit un flux Speex à partir de stdin, décode et délivre à un 16 bits fichier brut / échantillon. Notez que l'emballage utilisé est pas compatible avec celui de speexenc / speexdec.

sampledec.c

```
#include <speex/speex.h>
#include <stdio.h>
/*The frame size in hardcoded for this sample code but it doesn't have to be*/
#define FRAME_SIZE 160
int main(int argc, char **argv)
{
   char *outFile;
   FILE *fout;
   /*Holds the audio that will be written to file (16 bits per sample)*/
   short out[FRAME_SIZE];
   /*Speex handle samples as float, so we need an array of floats*/
   float output[FRAME_SIZE];
   char cbits[200];
   int nbBytes;
   /*Holds the state of the decoder*/
   void *state;
   /*Holds bits so they can be read and written to by the Speex routines*/
   SpeexBits bits;
   int i, tmp;
   /*Create a new decoder state in narrowband mode*/
   state = speex_decoder_init(&speex_nb_mode);
   /*Set the perceptual enhancement on*/
   tmp=1;
   speex_decoder_ctl(state, SPEEX_SET_ENH, &tmp);

   outFile = argv[1];
   fout = fopen(outFile, "w");
   /*Initialization of the structure that holds the bits*/
   speex_bits_init(&bits);
   while (1)
   {
      /*Read the size encoded by sampleenc, this part will likely be
        different in your application*/
      fread(&nbBytes, sizeof(int), 1, stdin);
      fprintf (stderr, "nbBytes: %d\n", nbBytes);
      if (feof(stdin))
         break;
      /*Read the "packet" encoded by sampleenc*/
      fread(cbits, 1, nbBytes, stdin);
      /*Copy the data into the bit-stream struct*/
      speex_bits_read_from(&bits, cbits, nbBytes);
      /*Decode the data*/
      speex_decode(state, &bits, output);
      /*Copy from float to short (16 bits) for output*/
      for (i=0;i<FRAME_SIZE;i++)
         out[i]=output[i];
      /*Write the decoded audio to file*/
      fwrite(out, sizeof(short), FRAME_SIZE, fout);
   }
   /*Destroy the decoder state*/
   speex_decoder_destroy(state);
   /*Destroy the bit-stream truct*/
   speex_bits_destroy(&bits);
   fclose(fout);
   return 0;
}
```

# Liste D’abréviation :

**IP** : Internet Protocol .

**HMM** : Hidden Markov Models .

**VoIP** : Voice over Internet Protocol

**MMC** :Model de Markov Cachés .

**HTK** : Hidden ToolKit .

**RAP**: Reconnaissance Automatique de la parole .

**SMX**: Seuils Maximale.

**SMN**: Seuils Minimale .

**EMN**: Energie Minimale .

**EMX**: Energie Maximale .

**TPZ**: Taux de Passage par Zéros .

**SPZ:** Seuil pour le taux de Passage par Zéros .

**MFCC**: Mel Frequency Cepstral Coefficients .

**PLP**: prédiction linéaire perceptuelle .

**DTW**: Dynamic Time Warping .

**SVM**: Support Vector Machine .

**MSSO**: Méthodes Statistiques du Second Ordre .

**CS-ACELP** : Conjugate Structure - Algebraic Code Excited Linear Prediction .

**FFT** :Fast Fourier Transform .

**SNR**: Signal Noise Ratio .

**MIPS**: Million d’Instructions par Seconde .

**DSP**: Digital Signal Processor .

**MOS**: Mean Opinion Score .

**CNG**: Comfort Noise Generator .

**PLC**: Packet Loss Concealment .

**PSD**: Power Spectral Densité GSM .

**ITU** : Union Internationale des Télécommunications .

**ARADIGIT** :Base de données des chiffres arabe .

**PESQ** : Perceptual Evaluation of  Speech Quality .

**PSQM** : Perceptual Speech Quality Measure.

## Abstract

The main goal of our final project is the realization of an application for speech recognition transcoded Speex and G.729. Isolated word recognition using the open source platform based on HMMs HTK (Hidden ToolKit) has been performed. a transcode database and Speex G729 was obtained from the Arabic numerals of the database ARADIGIT. A set of recognition experiments was conducted: i) on the speech recognition in quiet environment, ii) transcoded speech G.729 and Speex, and iii) reconstructed speech. MFCC and their first and second derivatives are used for extracted features vectors. The results show that the performances obtained with the speech transcoded by Speex Codec are better than those obtained with the G.729 codec. This study shows that efforts are still needed in the way of distributed recognition.



Keywords : HMM, HTK, ARADIGIT, MFCC .



## Résumé

L'objectif principal de notre projet de fin d'études est la réalisation d'une application en vue de la reconnaissance automatique de la parole transcodée Speex et G.729. L'application a porté sur la reconnaissance de chiffres arabes, en utilisant l'approche statistique HMM basée sur HTK. Pour atteindre cet objectif, une base de données transcodée G.729 et Speex a été obtenue à partir des chiffres arabes de la base de données ARADIGIT. Une série d'expériences portant sur la reconnaissance en milieu calme, puis de la parole transcodée G.729 et Speex. L'extraction des caractéristiques discriminantes se base sur les MFCC et leurs dérivées premières et secondes. Les résultats montrent que les performances obtenus avec de la parole transcodée par Speex Codec et supérieurs à ceux obtenus avec de la parole transcodée par G.729 codec. Cette étude montre que des efforts sont encore à faire sur le chemin de la reconnaissance distribuée.



mots clés : HMM, HTK, ARADIGIT, MFCC .

# ملخص

الهدف الرئيسي من الدراسة هو تصميم تطبيق للتعرف الآلي على الكلام بعد استعمال كل من Speex وG.729 في عملية الترميز التطبيق موجه للتعرف على الأرقام العربية، وذلك باستخدام HMM المنهج الإحصائي الذي يرتكز على منصة HTK. ولتحقيق هذا الهدف، قاعدة بيانات محولة تم الحصول عليها من خلال G.729 و Speex من الأرقام العربية من قاعدة بيانات ARADIGIT. والتي تحتوي على سلسلة من التسجيلات في بيئة هادئة، ثم ترميز المقاطع بواسطة G.729 و Speex. ويستند استخراج الخصائص المميزة باستعمال MFCC ومشتقاتها الأولى والثانية. وأظهرت النتائج المحصل عليها مع المقاطع المحولة عن طريق الترميز Speex احسن من تلك التي المحصل عليها مع المقاطع المحولة عن طريق الترميز G.729.