REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

ECOLE NATIONALE POLYTECHNIQUE

Département d'électronique
Projet de fin d'études

En vue de l'obtention du diplôme d'ingénieur d'état en électronique

Thème :

RECONNAISSANCE AUTOMATIQUE DE LA PAROLE CONTINUE EN UTILISANT LES HMM.

Proposé et dirigé par :

-Mr BOUSSEKSOU

Etudié par :

-Mr HADJLOUM Massinissa

-Mr HAMROUN Aghilas

Promotion : Juin 2012

Ecole Nationale Polytechnique, 10 Av. Hassan Badi, El Harrach, Alger, Algérie.

Remerciements

Nous tenons à remercier en premier lieu Dieu le tout puissant, qui nous a donné la force, le courage et la volonté pour mener à bien ce modeste travail.

Nous exprimons notre profonde gratitude, notre grand respect et notre sincère reconnaissance à notre promoteur monsieur B. BOUSSEKSOU qui nous a encadrés le long de ce travail. Pour tous ses conseils et critiques sur le plan scientifique qui nous ont permet de bien orienter notre travail.

Nous tenons à remercier nos parents, frères et sœurs ainsi que tous nos proches qui nous ont encouragés, soutenus et aidés sur tous les plans, le long de nos études.

Nos remerciements vont également à tous les enseignants de l'Ecole Nationale Polytechnique qui ont contribué à notre formation.

Nous remercions tous ceux, qui de prés ou de loin, nous ont soutenus et aidés dans la réalisation de ce travail.

Dédicaces

A mes très chers parents.

A mes très chers grands parents.

A mes deux sœurs KAHINA et FEROUDJA.

A mon petit frère HICHAM.

A tous mes amis.

Aghilas

Dédicaces

A mes très chers parents.

A ma sœur YASSINA.

A mes frères DRISS et YOUBA.

A la mémoire de ma chère grand-mère.

A tous ceux que j'aime.

Massinissa

Table des matières

Liste des figures

Liste des tableaux

Liste des abréviations

ACP	**A**nalyse en **C**omposantes **P**rincipales
ALD	**A**nalyse **L**inéaire **D**iscriminante
API	**A**lphabet **P**honétique **I**nternational
DAP	**D**écodage **A**coustico-**P**honétique
DTW	**D**ynamic **T**ime **W**arping
EM	**E**xpectation-**M**aximisation
FFT	**F**ast **F**ourier **T**ransform
HMM	**H**idden **M**arkov **M**odels
HTK	**H**idden Markov Model **T**ool**K**it
IPA	**A**lphabet **P**honétique **I**nternational
LFCC	**L**inear **F**requency **C**epstral **C**oefficients
LPC	**L**inear **P**rediction **C**oefficients
LPCC	**L**inear **P**rediction **C**epstral **C**oefficients
MFCC	**M**el **F**requency **C**epstral **C**oefficients
MLE	**M**aximum **L**ikelihood **E**stimation
QV	**Q**uantification **V**ectorielle
RAP	**R**econnaissance **A**utomatique de la **P**arole
TFD	**T**ransformés de **F**ourier **D**iscrète

Résumé et mots clés

ملخص

الهدف من هذا العمل هو إنشاء نظام أوتوماتيكي للتعرف على الكلام المستمر لذلك، قمنا بإنشاء قاعدة بيانات متكونة من 18 كلمة، تلفظ بها 25 متحدثا، تم إجراء التحليل الصوتي لقاعدة البيانات هذه باستخدام نوعين من المعاملات، .LPC وMFCC. هذا النظام أنجز باستعمال النماذج الإحصائية من خلال استخدام نماذج مركوف ألمختفتة HMM.
كلمات البحث: التعرف الأوتوماتيكي على الكلام، HMM، LPC، MFCC، نموذج صوتي، نموذج اللغة.

Résumé

L'objectif de ce mémoire est la réalisation d'un système de reconnaissance automatique de la parole continue. Pour ce faire, on a réalisé une base de données de dix-huit mots prononcés par 25 locuteurs. L'analyse acoustique de cette base a été effectuée en utilisant deux types de coefficients, les MFCC et les LPC. Pour mettre en œuvre notre système, nous avons choisi une modélisation statistique par l'utilisation des modèles HMM.

Les mots-clés : reconnaissance de la parole, MFCC, LPC, HMM, décodage acoustico-phonétique, modèle acoustique, modèle linguistique.

Abstract

The objective of this thesis is the achievement of an automatic recognition system for continuous speech .Therefore; we conducted a database of 18 words uttered by 25 speakers. The acoustic analysis of this database was performed using two types of coefficients: the MFCC and the LPC. In order to implement our system, we chose a statistical modeling through the use of HMMs.

Keywords: speech recognition, MFCC, LPC, HMM, acoustic-phonetic decoding, acoustic model, linguistic model.

Introduction générale

Nous assistons de nos jours à de grands progrès technologiques qui proposent des outils de plus en plus sophistiqués tout en étant de plus en plus compacts. Le clavier risque de devenir obsolète dans quelques années, c'est pourquoi on cherche de nouveaux moyens de communication plus intuitifs et moins encombrants. Aujourd'hui, après le clavier, la souris et les écrans tactiles, la parole s'impose comme l'alternative la plus directe et la plus naturelle pour communiquer avec les machines.

Les travaux présentés dans ce mémoire se situent dans le cadre de la reconnaissance automatique de la parole continue. Un système de reconnaissance a pour objectif la transcription d'un signal de parole. Pour ce faire, deux types de traitements sont nécessaires : un traitement acoustique accompli par un module de décodage acoustico-phonétique suivi d'un traitement linguistique dont la responsabilité est déléguée à un modèle de langage. La conception d'un système de reconnaissance est confrontée à plusieurs difficultés. En effet, la production de la parole est un processus continu et donc l'identification univoque d'unités symboliques dans ce flux n'est pas toujours possible. Par ailleurs, la variabilité du signal de parole inter et intra locuteurs constitue une autre source de complexité.

Dans ce travail, nous nous intéressons plus particulièrement à l'approche statistique. Cette approche a fait preuve de simplicité et d'efficacité. Dans cette dernière, le problème de reconnaissance est divisé en deux parties : l'apprentissage et la reconnaissance. La reconnaissance proprement dite est réalisée au cours de la deuxième étape où on associe à chaque phrase sa transcription correspondante.

Ce mémoire est organisé comme suit :

Le premier chapitre présente une étude générale sur l'anatomie du conduit vocal et du système auditif humain, ainsi que les mécanismes de production et de perception de la parole.

Dans le deuxième chapitre nous présentons une introduction à la reconnaissance de la parole. La structure générale d'un système de RAP, ainsi que les paramètres les plus efficaces pour représenter le signal de parole et les différentes approches de reconnaissance.

Le troisième chapitre est consacré pour le formalisme des modèles de Markov cachés, leur théorie, leurs principes pour la modélisation de la parole ainsi que les algorithmes d'apprentissage et de décodage permettant la construction d'un système de RAP. Et ceci pour les deux modes de reconnaissance : reconnaissance de mots isolés et reconnaissance de la parole continue.

Le dernier chapitre est le chapitre noyau de ce mémoire, il présente la description de notre application : le principe de fonctionnement des différentes fonctions constituantes notre système de RAP et les étapes suivies pendant sa réalisation, ainsi que l'évaluation de ses performances en utilisant différentes méthodes d'analyse (MFCC et LPC).

CHAPITRE I

Production et perception de la parole

Introduction

La parole constitue le moyen de communication le plus simple et le plus utilisé par l'homme. Ce mode de communication met en action un certain nombre d'organes et de facultés humaines.

Dans ce chapitre nous allons voir en détail le mécanisme et les différents organes utilisés depuis la production du signal vocal jusqu'à sa perception, ainsi que l'analyse effectué par le système auditif.

I.1 Processus de production et de perception de la parole

La production de la parole commence quand le locuteur formule un message (dans son esprit) qu'il veut transmettre à l'auditeur par la parole.

La prochaine étape dans le processus est la conversion du message dans un code d'une langue. Une fois la langue est choisie, le locuteur doit exécuter une série de commandes neuromusculaires qui mettent les cordes vocales en vibration, produisant ainsi un signal acoustique.

Les commandes neuromusculaires doivent contrôler simultanément tous les aspects du mouvement articulatoire y compris le contrôle des lèvres, mâchoire, la langue, et le voile (une «trappe» qui contrôle le flux acoustique dans la cavité nasale). Une fois que l'onde acoustique est générée, elle se propage vers un auditeur.

Le processus de perception (reconnaissance de la parole) se déroule de la manière suivante :

- ✓ le signal acoustique sera traité dans la membrane basilaire au niveau l'oreille interne qui fournit une analyse spectrale.
- ✓ le processus de transduction neuronale convertit le signal spectral à la sortie de la membrane basilaire en potentiel d'action nerveux, correspondant approximativement à un processus d'extraction de paramètres.
- ✓ le potentiel d'action nerveux est converti dans le code de la langue aux plus élevés centres de traitement dans le cerveau, et finalement la compréhension du message (compréhension de la signification) est atteinte.

La figure suivante représente un schéma du processus de production et perception de la parole.

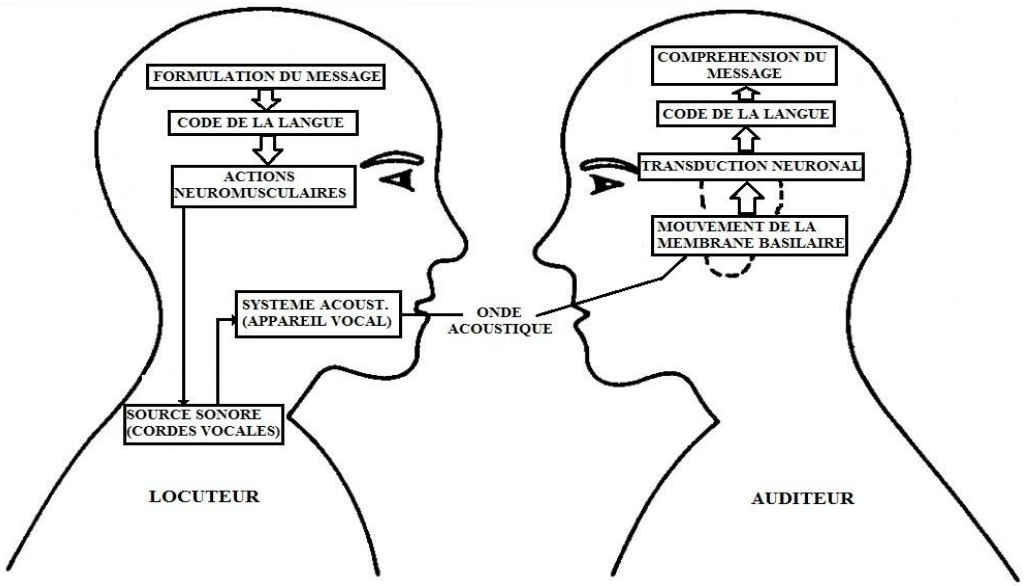

Figure I.1 : schéma du processus de production et perception de la parole

I.2 Production de la parole

La parole est le résultat de l'action volontaire et coordonnée des appareils respiratoire et masticatoire. Cette action se déroule sous le contrôle du système nerveux central qui reçoit en permanence des informations par rétroaction auditive.

L'appareil respiratoire fournit l'énergie nécessaire lorsque l'air est expiré par la *trachée-artère*. Au sommet de celle-ci se trouve le *larynx* où la pression de l'air est modulée avant d'être appliquée au *conduit vocal*, qui s'étend du *pharynx* jusqu'aux *lèvres* (figure I.2).

Le larynx est un ensemble de muscles et de cartilages mobiles qui entourent une cavité située à la partie supérieur de la trachée (figure I.3(a)).

Les cordes vocales sont en fait deux lèvres symétriques placées en travers du larynx ; ces lèvres peuvent fermer complètement le larynx et, en s'écartant, déterminer une ouverture triangulaire appelée *glotte* (figure I.3(b)). L'air y passe librement pendant la respiration et la *voix chuchotée*, et aussi pendant la phonation des sons non voisés. Les sons voisés résultent au contraire d'une vibration périodique des cordes vocales ; des impulsions périodiques de pression sont ainsi appliquées au conduit vocal [1].

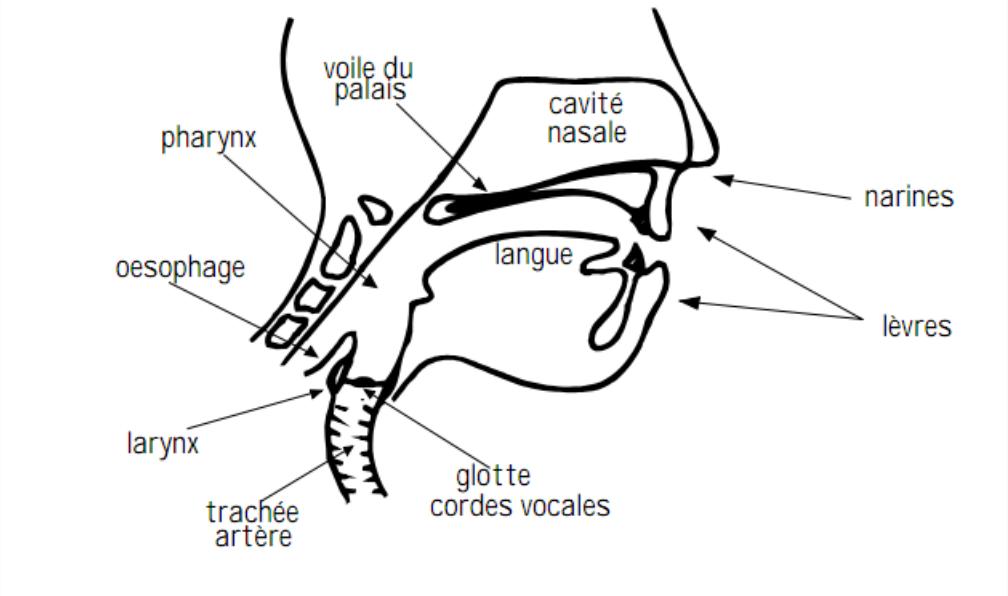

Figure I.2 : L'appareil phonatoire

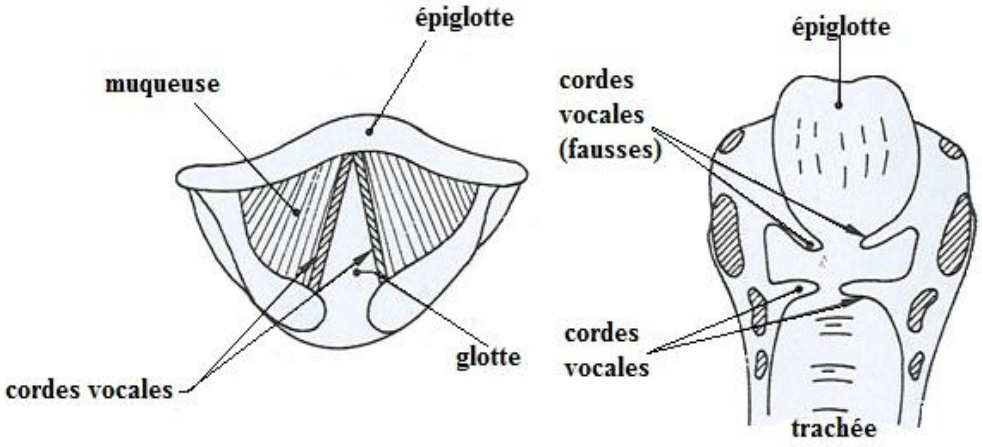

Figure I.3 : vue du larynx : (a) vue de haut ; (b) coupe verticale.

Ce dernier est un ensemble de cavités situées entre la glotte et les lèvres ; on peut sur la figure I.2 distinguer la *cavité pharyngienne*, la *cavité buccale* et la *cavité nasale*.

Le conduit vocal peut être considéré comme une succession de tubes ou cavités acoustiques de sections diverses [4].

Les sons voisés résultent donc de l'excitation du conduit vocal par des impulsions périodiques de pression liées aux oscillations des cordes vocales : l'ouverture brusque de la glotte libère la pression accumulée en amont ; elle se referme en suite plus graduellement.

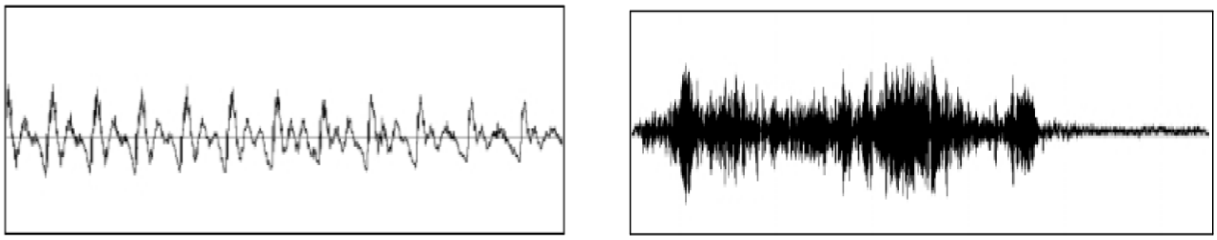

Figure I.4 : (a) son voisé ; (b) son non voisé

I.2.1 La parole d'un point de vue physiologique

La parole est une séquence de sons qui correspond à une succession d'états de l'appareil phonatoire.

Les états de l'appareil phonatoire sont définis par :

- ✓ État des cordes vocales: tendues / relâchées ;
- ✓ Position, forme, taille des diverses cavités (pharynx, bouche, nez) et de leurs mécanismes d'occlusion.

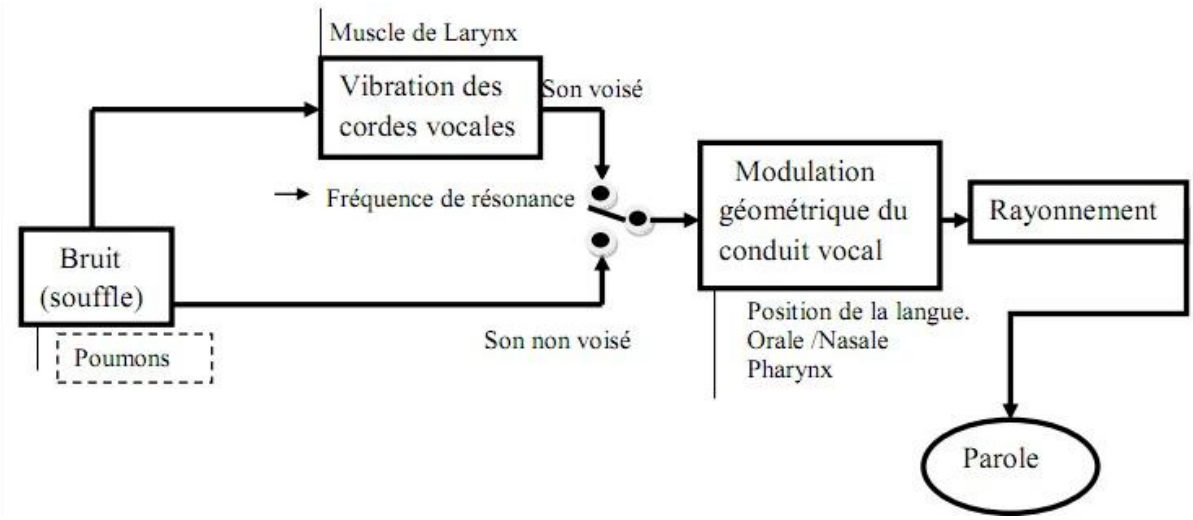

Figure I.5 : Fonctionnement général de l'appareil phonatoire.

I.2.2 Notions phonétiques

En linguistique, un phonème est la plus petite unité distinctive (c'est-à-dire permettant de distinguer des mots les uns des autres) que l'on puisse isoler dans la chaîne parlée.

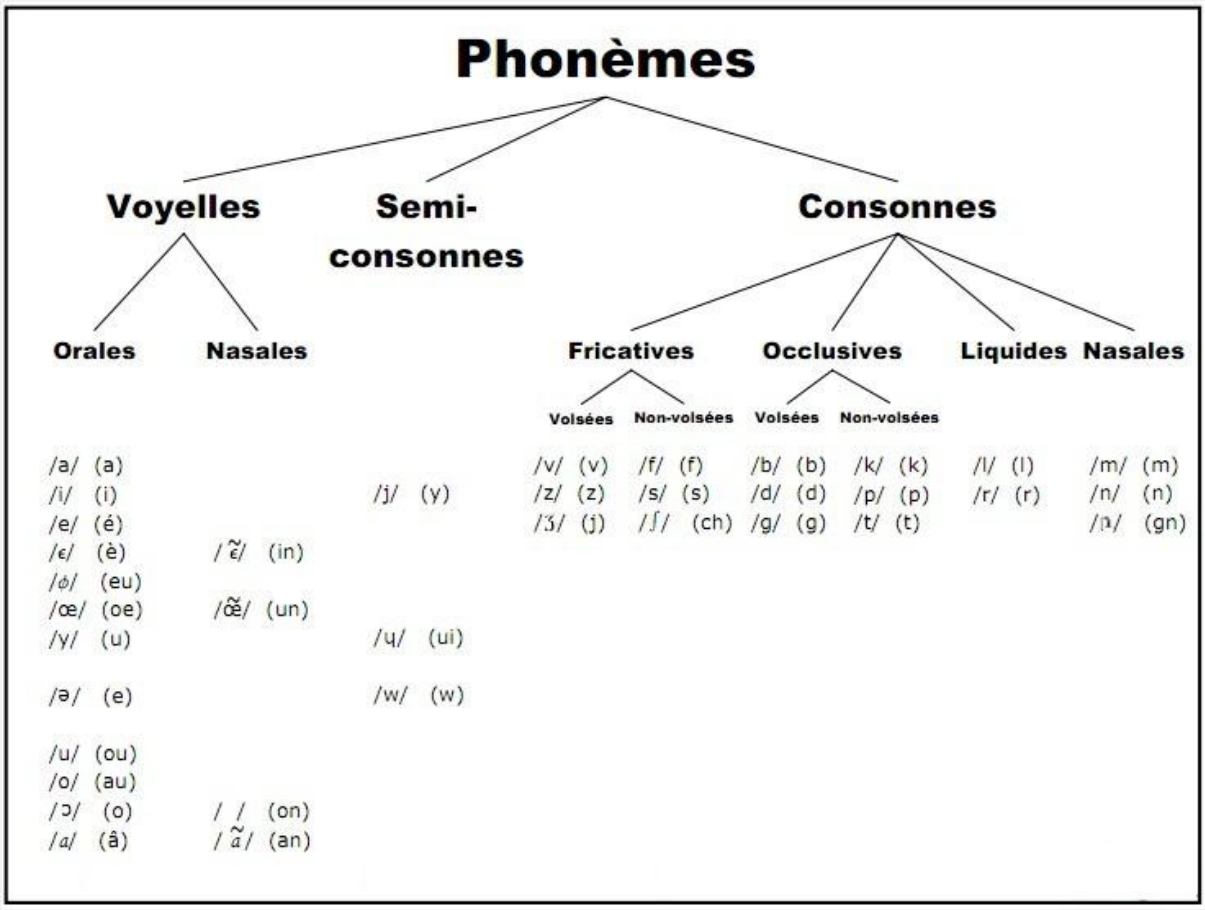

Figure I.6 : Les phonèmes de la langue française.

I.2.2.1 L'alphabet phonétique international

L'*alphabet phonétique international* (IPA) associe des symboles phonétiques aux sons, de façon à permettre l'écriture compacte et universelle des prononciations (voir tableau I.1 pour le français).

IPA	EXEMPLES	IPA	EXEMPLES
i	idée, ami	p	patte, repas, cap
e	ému, ôté	t	tête, ôter, net
ɛ	perdu, modèle	k	carte, écaille, bec
a	alarme, patte	b	bête, habile, robe
ɑ	bâton, pâte	d	dire, rondeur, chaud
ɔ	Obstacle, corps	g	gauche, égal, bague
o	auditeur, beau	f	feu, affiche, chef
u	coupable, loup	s	sœur, assez, passe
y	punir, élu	ʃ	chanter, machine, poche
ø	creuser, deux	v	vent, inventer, rêve
œ	malheureux, peur	z	zéro, raisonner, rose
ə	petite, fortement	ʒ	jardin, manger, piège
ɛ̃	peinture, matin	l	long, élire, bal
ɑ̃	vantardise, temps	ʁ	rond, chariot, sentir
ɔ̃	rondeur, bon	m	madame, aimer, pomme
œ̃	lundi, brun	n	nous, punir, bonne
j	piétiner, briller		agneau, peigner, règne
w	oui, fouine	ŋ	jumping, smoking
ɥ	huile, nuire	h	halte, hop (exclamations)

Tableau I.1 : Les symboles de l'alphabet phonétique international utilisés en français.

I.2.2.2 Phonétique articulatoire

Il est intéressant de regrouper les sons de parole en classes phonétiques, en fonction de leur mode articulatoire. On distingue généralement :

- ✓ les voyelles,
- ✓ les semi-consonnes (semi-voyelles),
- ✓ les consonnes.

a. Les voyelles

Les voyelles [i, e, ɛ, *a*, a, ɔ, o, y, u, œ, …] diffèrent de tous les autres sons par le degré d'ouverture du conduit vocal. Si le conduit vocal est suffisamment ouvert pour que l'air poussé par les poumons le traverse sans obstacle, il y a production d'une voyelle. Le rôle de la bouche se réduit alors à une modification du timbre vocalique ().

Les voyelles sont *orales* ou *nasales* selon que la cavité nasale n'est pas ou est mise en parallèle à la cavité buccale,

- ✓ **Voyelles orales** : **i**dée, **é**mu, mod**è**le, **a**larme, p**â**te, c**o**rps, b**eau**, él**u**, l**ou**p, d**eu**x, p**eu**r, p**e**tite.
- ✓ **Voyelles nasales** : mat**in**, t**em**ps, b**on**, br**un**.

b. Les consonnes

Les consonnes sont produites lorsqu'un rétrécissement apparaît dans l'appareil phonatoire.

Les cordes vocales peuvent vibrer ou laisser passer librement l'air (sons voisés et non voisés)

Les consonnes sont *fricatives* si le rétrécissement est partiel ou *occlusives* (plosives) si une occlusion totale apparaît dans l'appareil phonatoire, causant une augmentation de la pression et un relâchement brutal de celle-ci lors de l'ouverture.

- ✓ **Fricatives non-voisées** : **ch**anter, **s**oupe, **f**acile.
- ✓ **Fricatives voisées** : **j**ouer, **z**éro, **v**élo.
- ✓ **Occlusives non-voisées : p**a**p**a, **t**apis, **c**arte.
- ✓ **Occlusives voisées** : **b**é**b**é, **d**ébut, **g**auche.
- ✓ **Liquides** : **l**apin, **r**ayon.
- ✓ **Nasales** : **m**aman, **n**ord, gro**gn**er.

c. Les semi-consonnes

Les semi-consonnes, quant à elles, combinent certaines caractéristiques des voyelles et des consonnes. Comme les voyelles, leur position centrale est assez ouverte, mais le relâchement soudain de cette position produit une friction qui est typique des consonnes.

I.3 Perception de la parole

L'audition est le fruit d'un mécanisme complexe assuré principalement par les deux oreilles pour permettre la perception binaurale stéréophonie et les voies centrales avec notamment un rétrocontrôle permanent du cerveau.

L'oreille humaine est un système d'analyse du son étonnant et complexe. Elle est capable de détecter des sons sur une large plage d'intensités et de fréquences. Allant de 20 Hz (son grave) à environ 16 à 20 KHz (son aigu).

Pour qu'on puisse entendre, plusieurs transformations se produisent dans l'oreille.

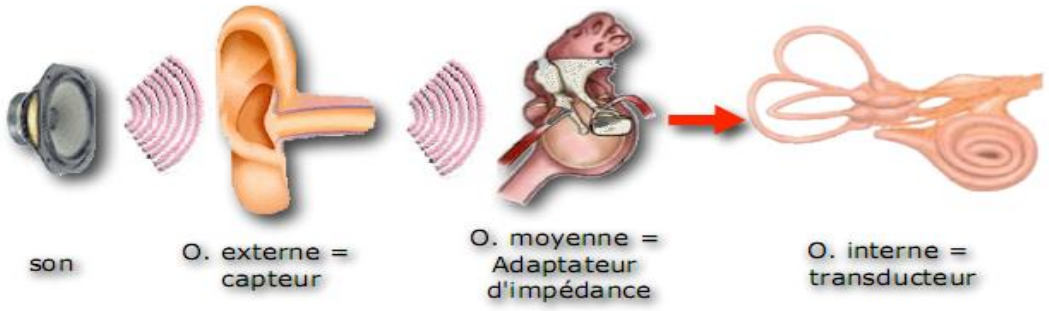

Figure I.7 : Propagation de l'onde sonore dans l'oreille humaine.

I.3.1 Organes de l'audition

L'appareil auditif comprend *l'oreille externe*, *l'oreille moyenne* et *l'oreille interne* (fig. I.8)

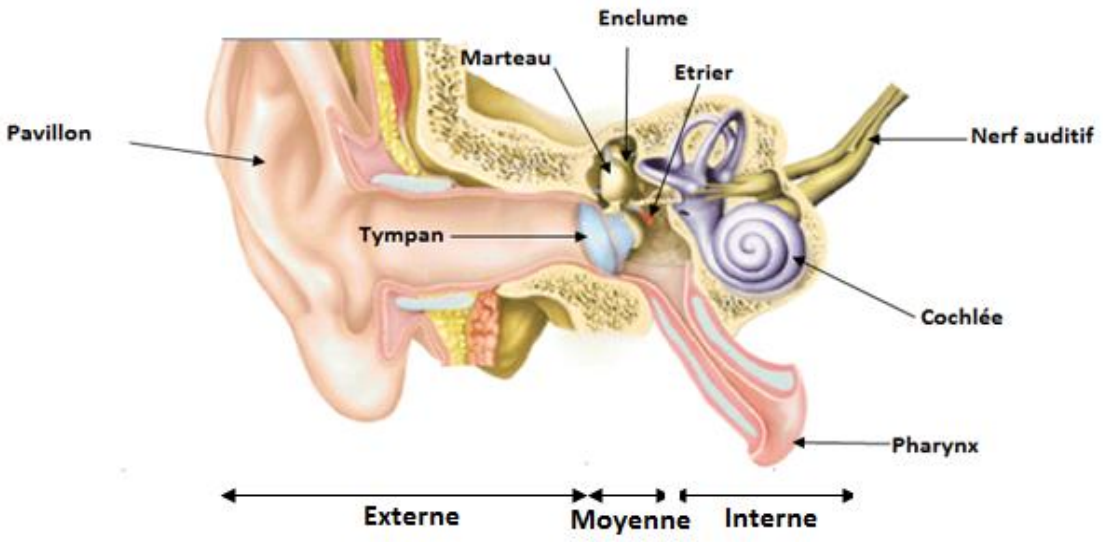

Figure I.8 : Composition anatomique de l'oreille.

I.3.1.1 L'oreille externe (OE)

L'oreille externe est constituée du pavillon et du conduit auditif externe. Le pavillon, capte les ondes acoustiques, il les transmet vers le conduit auditif externe.

Le pavillon, par sa géométrie, permet d'avoir une audition pourvue de directivité. C'est une sorte d'*antenne acoustique*, qui nous permet de localiser un évènement sonore, un bruit.

Le pavillon amplifie de quelques décibels les fréquences voisines de 5 kHz, le conduit auditif externe amplifie d'une dizaine de décibels celles autour de 2,5 et 4 kHz. L'effet total du corps (épaules, tête) et de l'oreille externe engendre globalement une amplification qui va de 5 à 20 décibels entre 2 et 7 kHz.

I.3.1.2 L'oreille moyenne (OM)

Elle est située dans une cavité osseuse du crâne (toujours en milieu aérien) côtoyant l'oreille interne, et sa pression relative est équilibrée par rapport à la face externe du tympan grâce à la trompe d'Eustache communicant avec le nasopharynx.

Composée du tympan, de la chaîne d'osselets, son rôle est de transmettre l'information sonore en provenance de l'oreille externe pour l'acheminer vers l'oreille interne, tout en accomplissant l'adaptation d'impédance nécessitée par le milieu liquide de cette dernière ; Au total, la pression au niveau de la fenêtre ovale est ~ 24 fois plus grande qu'au niveau du tympan, ce qui représente un gain de 27,5 dB. Ce gain est essentiel pour l'adaptation d'impédance entre les milieux aérien (oreille moyenne) et liquide (oreille interne), sans laquelle 99% de l'énergie serait réfléchie au niveau de l'interface [22].

I.3.1.3 L'oreille interne (OI)

C'est dans l'oreille interne que l'énergie mécanique est transformée en énergie bioélectrique, c'est-à-dire en potentiels d'action nerveux.

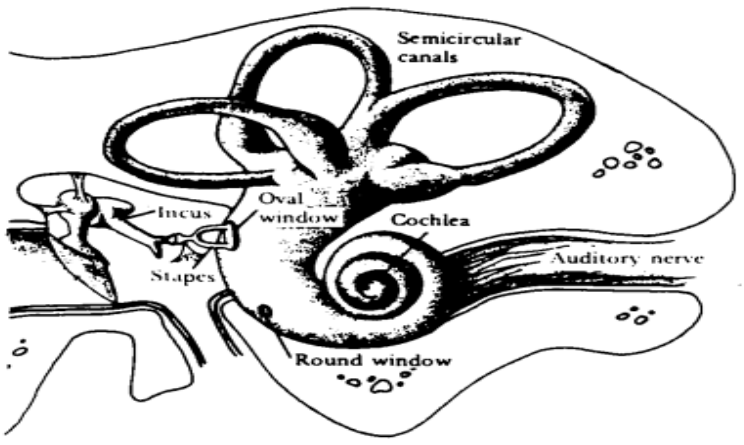

Figure I.9 : vue extérieure de l'oreille interne.

L'oreille interne se compose de :

- ✓ **L'appareil vestibulaire**, comprenant les trois canaux semi-circulaires visibles sur la figure I.9, qui joue un rôle important pour l'équilibre mais n'intervient pas dans l'audition.
- ✓ **La cochlée**, qui a globalement la forme d'un canal en colimaçon, d'une longueur déroulée de 2,5 à 3 centimètres, divisé en deux dans sa longueur par une lame osseuse à laquelle s'attachent deux membranes : la membrane basilaire et la membrane tectorielle.

I.3.2 Le champ audible

Les champs de l'audition, de la musique et de la parole sont représentés sur la figure ci-dessous, dans le plan harmonique.

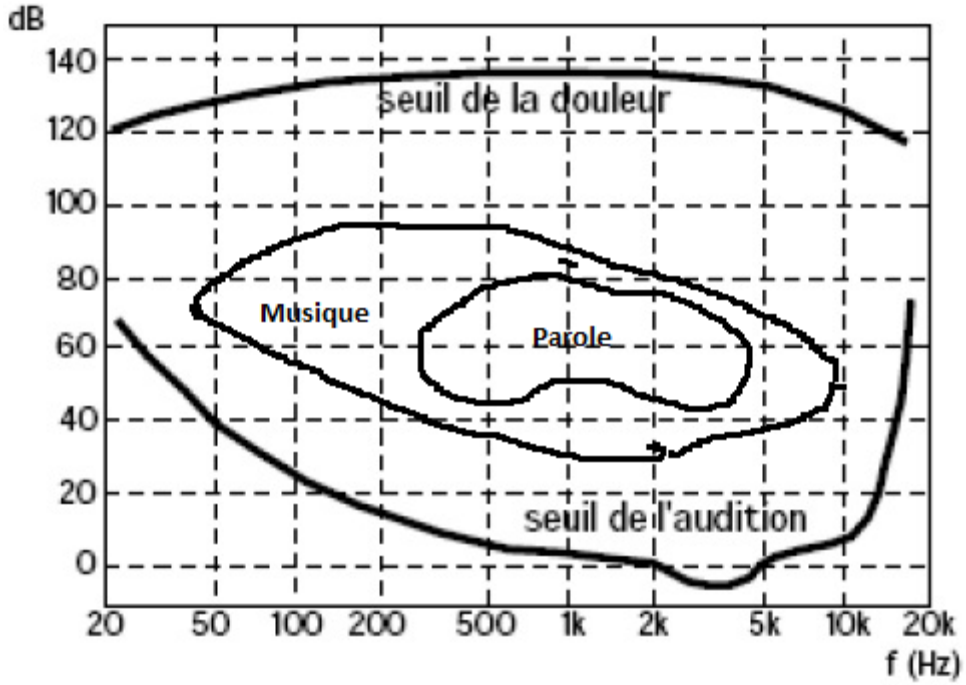

Figure I.10 : champ audible, champs de la musique et de la parole.

Le champ audible est délimité inférieurement par les seuils auditifs, supérieurement par les seuils maximums de confort ou les seuils de douleur. Entre ces deux limites, on voit qu'aux fréquences moyennes la dynamique de l'oreille est de près de 120 dB ; cela signifie que nous sommes capables d'entendre des sons dans un rapport de puissances de 1 à 10^{12} (soit 1000 milliards) !

En fréquences, le champ audible s'étend environ de 20 à 20 000 Hz. En pratique ces limites, surtout vers les hautes fréquences, ne sont valables que pour des sujets jeunes et en bonne santé. Avec l'âge, notre sensibilité auditive décroît, particulièrement dans les aigus [3].

Le champ de la musique s'étend de 50 à 10 000 Hz environ, celui de la parole est plus restreint : l'essentiel de l'énergie est entre 200 à 5000 Hz, et la restriction à la bande [300, 3400] Hz qui est celle du téléphone altère peu l'intelligibilité pour un sujet qui entend normalement.

I.4 Conclusion

Ce chapitre met en évidence les caractéristiques complexes des phénomènes de production et de perception de la parole, en illustrant les deux processus et les organes qui participent dans chacun d'eux ainsi que leurs rôles et les transformations subites par le signal de parole, qu'on doit prendre en compte lors de la conception des systèmes de reconnaissance.

Tous les systèmes de RAP sont conçus en faisant une analogie avec les systèmes de production et de perception de la parole chez l'être humain.

CHAPITRE II

Reconnaissance Automatique de la Parole

Introduction

Le but de la Reconnaissance automatique de la parole (RAP) consiste à extraire l'information lexicale contenue dans un signal de parole.

Dans ce chapitre nous allons présenter les méthodes d'analyse du signal pour une paramétrisation efficiente, les différentes caractéristiques d'un système de reconnaissance de la parole, la structure générale de ce dernier, et enfin les approches de reconnaissance en insistant sur les plus utilisées actuellement.

Les systèmes classiques de RAP sont composés essentiellement de cinq modules (figure II.1) :

- ✓ La paramétrisation du signal, qui doit permettre de ne garder que les informations pertinentes de ce dernier ;
- ✓ Les modèles acoustiques, qui doivent représenter au mieux les unités acoustiques choisies (phonèmes, diphones, mots. . .) ;
- ✓ Les modèles linguistiques, qui doivent être une représentation la plus vraisemblable possible du langage ;
- ✓ Le dictionnaire, qui doit contenir l'ensemble des mots que l'on souhaite pouvoir reconnaître ;
- ✓ Le système de reconnaissance lui même.

Ces différentes composantes d'un système de RAP, bien que toutes nécessaires pour la reconnaissance de parole continue, sont relativement indépendantes les unes des autres.

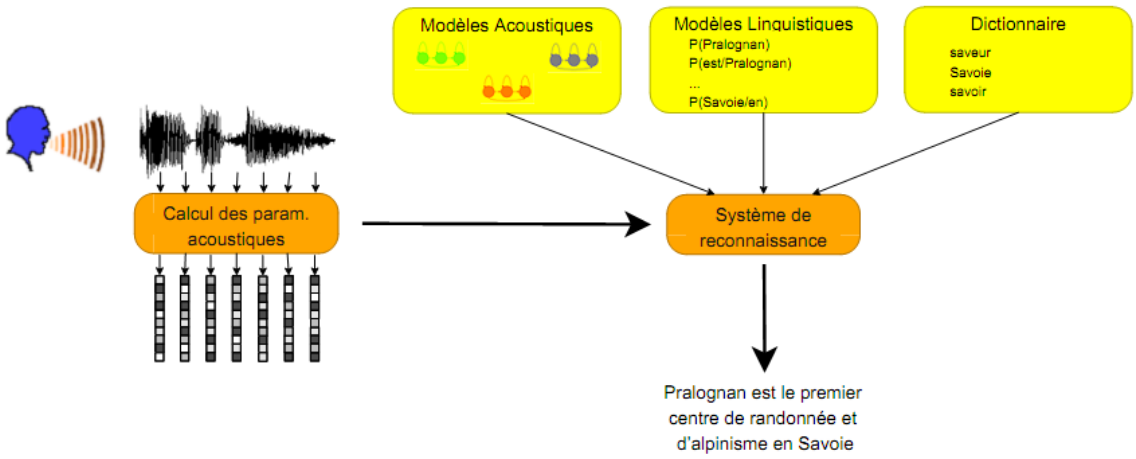

Figure II.1 : Schéma de principe d'un système de reconnaissance automatique de la parole.

II.1 Espace de représentation

Le signal de parole présente de la redondance et contient des informations jugées superflues pour la reconnaissance automatique de la parole, ce qui justifie la recherche d'une représentation plus compacte.

II.1.1 Représentations non paramétriques

Un son se définit classiquement au moyen de son amplitude, de sa durée, et de son timbre. La parole qui est un son particulièrement complexe, n'échappe pas à cette définition.

Le traitement du signal vocal a pour but de fournir une représentation moins redondante de la parole que celle obtenu par codage de l'onde temporelle tout en permettant une extraction précise des paramètres significatifs tels que la fréquence du fondamental, les fréquences des formants, …etc. les applications en sont tout naturellement la reconnaissance, le codage et la synthèse.

Le signal de la parole peut être analysé dans le domaine temporel ou dans le domaine spectral par des méthodes non paramétriques, sans faire l'hypothèse d'un modèle pour rendre compte du signal observé. Les représentations les plus souvent retenues sont l'énergie du signal et les sorties d'un banc de filtres numériques.

II.1.1.1 Chaine de prétraitement

Le calcul de la représentation du signal est réalisé par une chaine d'analyse presque complètement numérique en suivant les étapes suivantes : échantillonnage, préaccentuation, puis analyse à court terme sur des trames successives du signal.

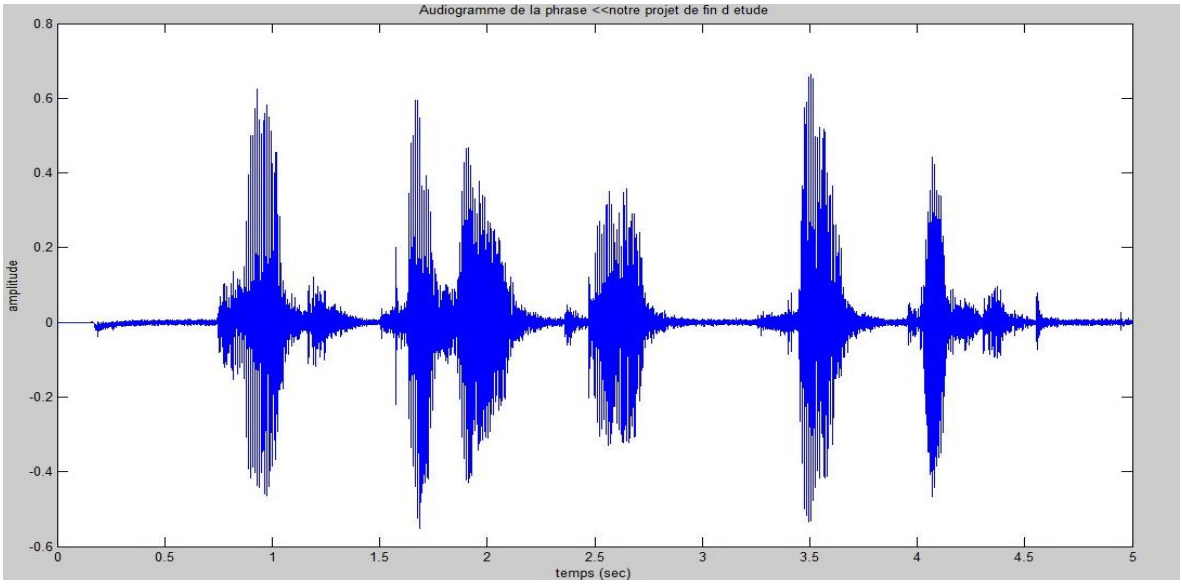

Figure II.2 : Audiogramme de la phrase « notre projet de fin d'étude »

L'échantillonnage transforme le signal à temps continu $s(t)$ en signal à temps discret $s(nT_e)$ défini aux instants d'échantillonnage, multiples entiers de la période d'échantillonnage T_e ; celle-ci est elle-même l'inverse de la fréquence d'échantillonnage f_e. Pour ce qui concerne le signal vocal, le choix de f_e résulte d'un compromis.

On estime que le signal garde une qualité suffisante lorsque son spectre est limité à 3,4 kHz et l'on choisit $f_e = 16\,kHz$ (seulement 8 kHz pour le signal de ligne téléphonique) pour satisfaire raisonnablement le théorème de Shannon (suivant le théorème de Shannon, les signaux doivent être échantillonnés à une fréquence d'échantillonnage supérieur ou égale à deux fois leur plus haute composante fréquentielle.).

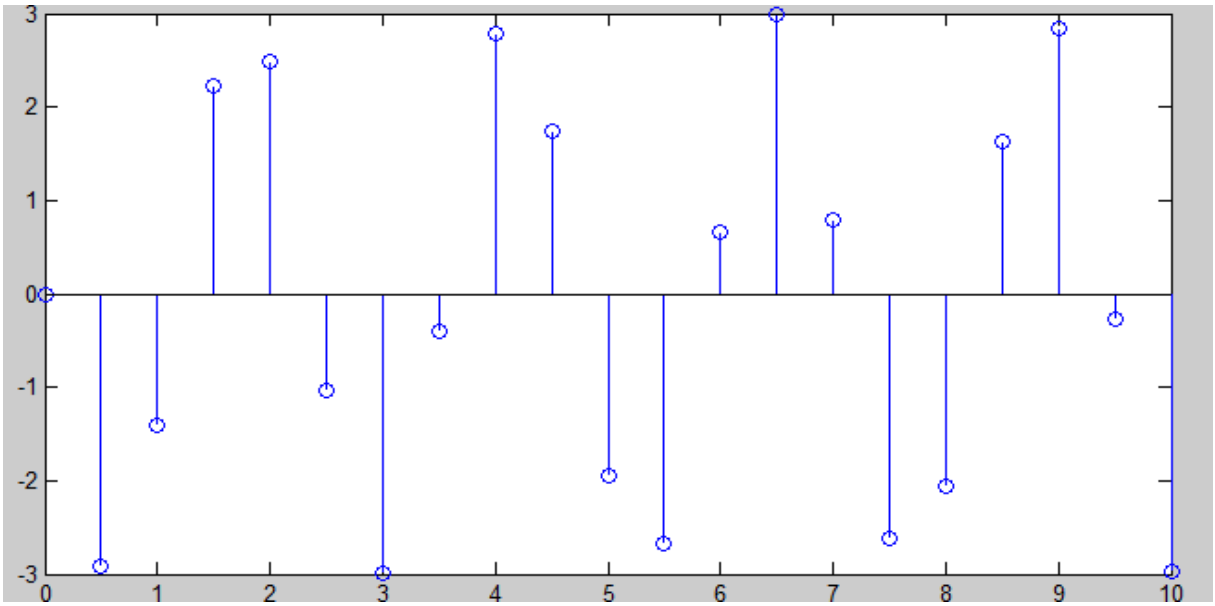

Figure II.3 : Représentation d'un signal échantillonné.

Le signal échantillonné est ensuite pré-accentué pour relevé les hautes fréquences qui sont moins énergétiques que les basses fréquences ; la préaccentuation s'_n de l'échantillon s_n à l'instant n est calculé pour une valeur de α compris entre 0,9 et 1 comme :

$$s'_n = s_n - \alpha . s_{n-1} \tag{2.1}$$

La plupart des représentations font l'hypothèse d'une stationnarité du signal, ce qui n'est pas valide pour le cas de la parole. Une hypothèse à court terme est donc réalisée sur une fenêtre glissante, chaque trame couvrant de 10 à 50 ms sur laquelle le signal est supposé quasi-stationnaire, pour un pas d'analyse entre deux trames successives de l'ordre de la centi-seconde. Le découpage de signal en trames produit des discontinuités aux frontières des trames, qui se manifestent par des lobes secondaires dans le spectre ; ces effets parasites sont réduits en appliquant aux échantillons $\{s'_n\}_{n=0.....N-1}$ de la trame une fenêtre de pondération $\{w_n\}_{n=0....N-1}$ comme par exemple la fenêtre de Hamming [6] :

$$s''_n = w_n . s'_n \tag{2.2}$$

Avec :

$$w_n = 0.54 - 0.46\cos\left(2\pi \frac{n}{N-1}\right), \quad 0 \leq n \leq \mathrm{N} - 1 \tag{2.3}$$

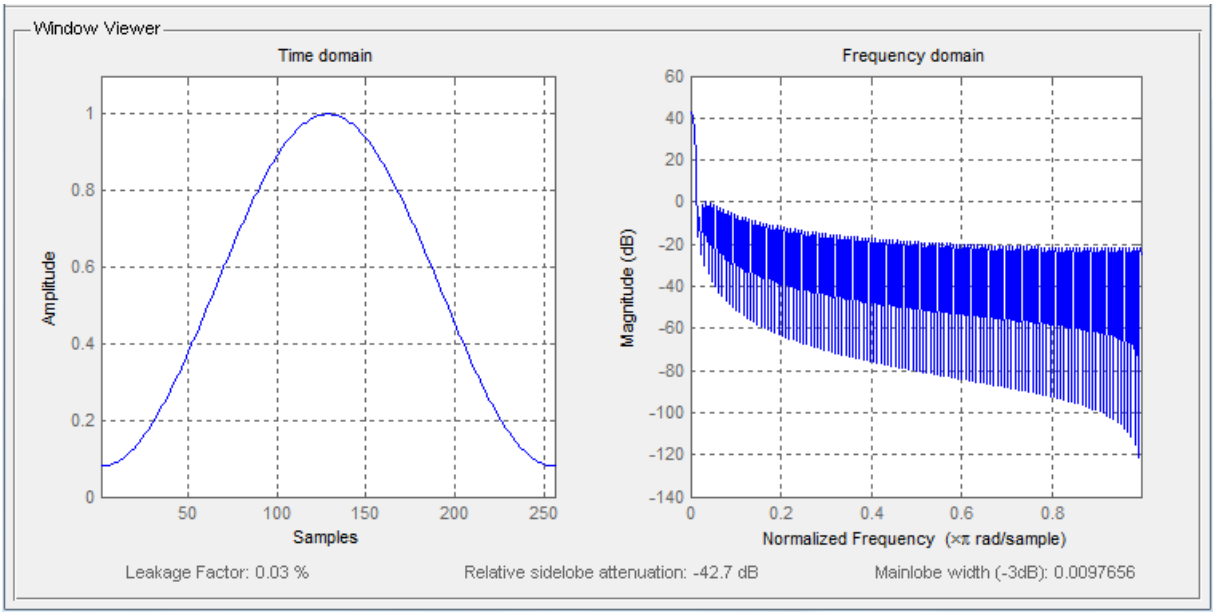

Figure II.4 : fenêtre de Hamming dans le domaine temporel et fréquentiel.

Par la suite, les échantillons prétraités s''_n seront notés simplement s'_n. L'analyse de chaque trame peut être réalisée directement dans le domaine temporel, par exemple pour calculer l'énergie, ou plus couramment dans le domaine fréquentiel.

II.1.1.2 Analyse temporelle

Dans la représentation temporelle du signal on peut extraire des paramètres tels que l'énergie et la fréquence fondamentale.

a. Energie du signal

L'énergie du signal est un indice qui peut par exemple contribuer à la détection du voisement d'un segment de parole. L'énergie totale E_0 est calculée directement dans le domaine temporel sur une trame de signal $\{s_n\}$ $0 \leq n \leq N-1$ comme :

$$E_0 = \sum_{n=0}^{N-1} {s_n}^2 \tag{2.4}$$

L'énergie ainsi obtenue est sensible au niveau d'enregistrement ; on choisit en général de la normalisée, et d'exprimer sa valeur en décibels par rapport à un niveau de référence.

b. Fréquence fondamentale

La parole est obtenue à partir de la vibration des cordes vocales. Cette vibration a eu lieu à une fréquence fondamentale F_0 et elle est caractérisée par la présence de nombreuses harmoniques. Ces harmoniques sont filtrées par la cavité buccale, ce qui permet d'en extraire les sons voisés.

Une analyse d'un signal de parole n'est pas complète tant qu'on n'a pas mesuré l'évolution temporelle de la fréquence fondamentale ou *pitch*. La figure II.5 donne l'évolution temporelle de la fréquence fondamentale de la phrase *"les techniques de traitement de la parole"*. On constate qu'à l'intérieur des zones voisées, la fréquence fondamentale évolue lentement dans le temps. Elle s'étend approximativement de 70 Hz à 250 Hz chez les hommes, de 150 Hz à 400 Hz chez les femmes, et de 200 Hz à 600 Hz chez les enfants.

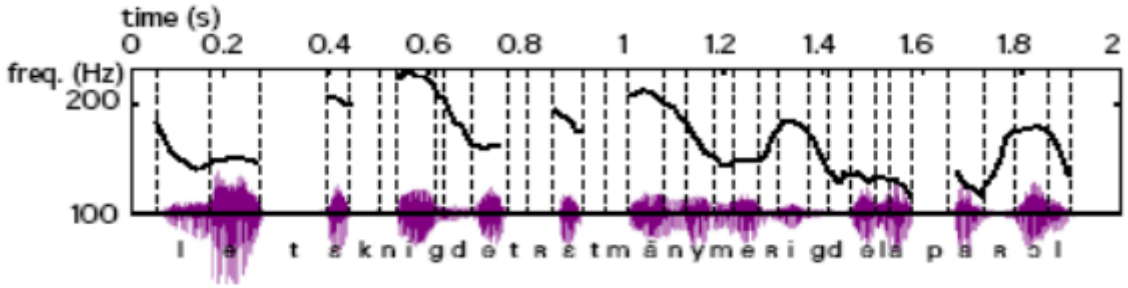

Figure II.5: Evolution de la fréquence de vibration des cordes vocales : la fréquence est donnée sur une échelle logarithmique ; les sons voisés sont associés à une échelle logarithmique ; les sons non-voisés sont associés à une fréquence nulle [7].

L'information prosodique est dominée par la variation de F_0, fréquence du fondamentale : ce paramètre est donc important pour le décodage acoustico-phonétique DAP.

L'estimation de F_0, et la décision de voisement/non-voisement est un problème délicat qui tient principalement aux raisons suivantes :

- L'excitation globale n'est pas rigoureusement périodique.
- Il y a une interaction entre l'excitation et le conduit vocal : la périodicité du signal est en fait causée de manière coopérative par l'excitation quasi périodique et le premier formant à bande étroite.
- La segmentation des débuts et fins de voisement est difficile.
- La dynamique de variation de F_0 est importante.
- La source peut être atténuée dans certain type de transmission comme le téléphone (bande 300-3400 Hz).
- De plus la langueur de l'intervalle de mesure de F_0 est difficile à choisir car il doit être suffisamment court pour que F_0 puisse être considérée constante et suffisamment long pour qu'elle soit mesurable (au moins une période). Dans les zones de transition c'est difficile de satisfaire ces deux conditions à la fois ce qui provoque des erreurs de mesure [8].

D'autres paramètres peuvent être calculés dans le domaine temporel, comme les coefficients d'auto-corrélation.

II.1.1.3 Analyse spectrale

La production de la parole rend souhaitable une analyse du signal dans le domaine spectral pour la reconnaissance.

a. Spectrogramme

Il est souvent intéressant de représenter l'évolution temporelle du spectre à court terme d'un signal, sous la forme d'un *spectrogramme***.** L'amplitude du spectre y apparaît sous la forme de niveaux de gris dans un diagramme en deux dimensions temps-fréquence. On parle de spectrogramme à *large bande* ou à *bande étroite* selon la durée de la fenêtre de pondération.

Les spectrogrammes à bande large sont obtenus avec des fenêtres de pondération de faible durée (typiquement 10 ms); ils mettent en évidence l'enveloppe spectrale du signal, et permettent par conséquent de visualiser l'évolution temporelle des formants. Les périodes voisées y apparaissent sous la forme de bandes verticales plus sombres.

Les spectrogrammes à bande étroite sont moins utilisés. Ils mettent plutôt la structure fine du spectre en évidence: les harmoniques du signal dans les zones voisées y apparaissent sous la forme de bandes horizontales.

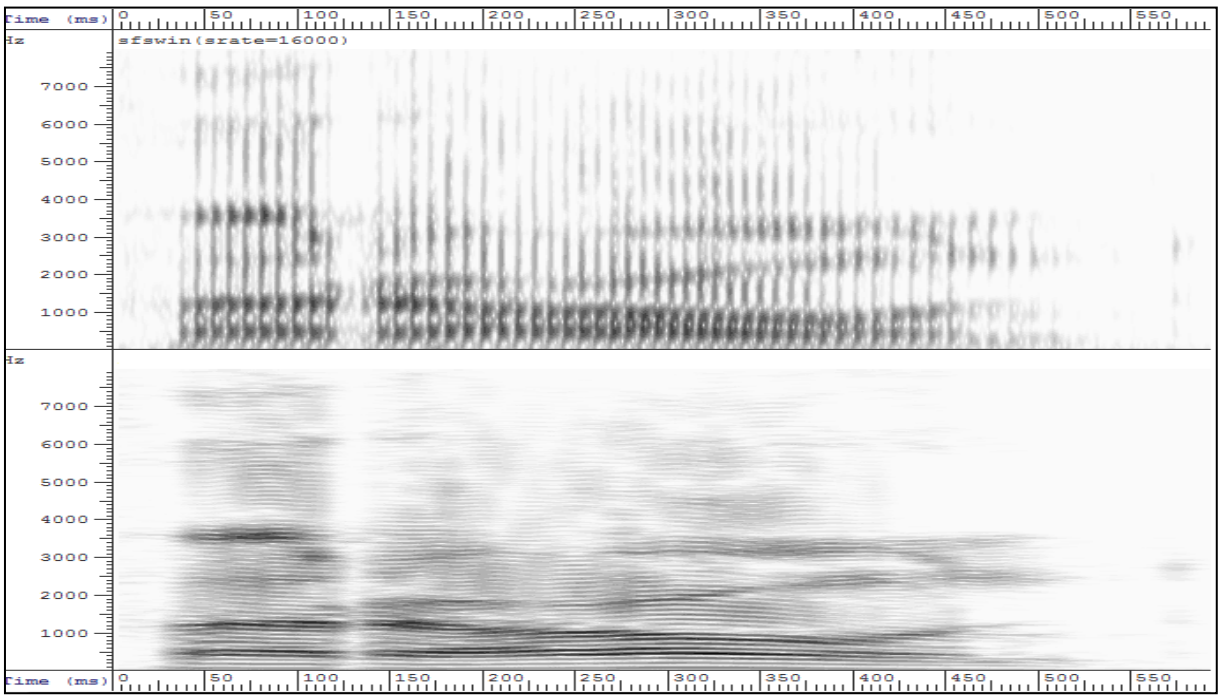

Figure II.6: Spectrogramme large bande en haut et bande étroite en bas.

b. La transformée de Fourier

C'est une généralisation de la décomposition en série de Fourier à tous les signaux déterministes. Elle permet d'obtenir une représentation en fréquence (représentation spectrale) de ces signaux. Elle exprime la répartition fréquentielle de l'amplitude, de la phase et de l'énergie (ou de la puissance) des signaux considérés.

Soit *s(t)* un signal déterministe. Sa transformée de Fourier est une fonction, généralement complexe, de la variable *f* et définie par :

$$S(f) = TF[s(t)] = \int_{-\infty}^{+\infty} s(t) e^{-j2\pi ft}\, dt \qquad (2.5)$$

b.1. La transformés de Fourier discrète TFD

Considérons une suite finie de N échantillons {s(n)}={s(0), s(1),…, s(N-1)}. On définit sa transformée de Fourier Discrète comme la suite {S(k)} :

$$S(k) = \sum_{n=0}^{N-1} s(n)\, W_N^{-nk} \qquad (k = 0 \ldots N-1) \qquad (2.6)$$

Avec :

$$W_N = e^{j\frac{2\pi}{N}} \qquad (2.7)$$

b.2. La transformée rapide de Fourier FFT

En 1965, Cooley et Tukey proposèrent une méthode qui permet de réduire considérablement le temps de calcul de la TFD d'une suite dont le nombre d'échantillons N est décomposable en facteurs (typiquement, une puissance de 2).

La FFT radix 2 avec entrelacement dans le temps :

Cette méthode, qui exige une séquence dont la longueur est une puissance de 2 ($N = 2^M$), a rendu envisageable le calcul de TFD de plusieurs milliers de points. Le nom de radix 2 provient du fait que l'on ramène le calcul d'une TFD de *N* points à un certain nombre de calculs de TFD de 2 points. L'appellation entrelacement dans le temps est liée à la décomposition de la suite {s(n)} en suites plus courtes.

Soit $\{S(k)\}$ la TFD d'une suite $\{s(n)\}$ de longueur $N = 2^M$:

$$S(k) = \sum_{n=0}^{N-1} s(n)\, W^{-nk} \qquad (k = 0 \dots N-1) \tag{2.8}$$

Soient les deux suites *a(n)* et *b(n)* de longueur N/2 et leurs TFD *A(k)* et *B(k)* :

$$a(n) = s(2n) \qquad a(n) \Leftrightarrow A(k) \tag{2.9}$$

$$b(n) = s(2n+1) \qquad b(n) \Leftrightarrow B(k) \tag{2.10}$$

On montre facilement que les *S(k)* peuvent être calculés à partir des *A(k)* et *B(k)* :

$$\begin{aligned} S(k) &= \sum_{n=0}^{\frac{N}{2}-1} s(2n)\, W_n^{-2nk} + \sum_{n=0}^{\frac{N}{2}-1} s(2n+1)\, W_n^{-(2n+1)k} \\ &= A(k) + W^{-k} B(k) \end{aligned} \tag{2.11}$$

$$\begin{aligned} S\left(\frac{N}{2}+k\right) &= \sum_{n=0}^{\frac{N}{2}-1} s(2n)\, W_n^{-2n\left(k+\frac{n}{2}\right)} + \sum_{n=0}^{\frac{N}{2}-1} s(2n+1)\, W_n^{-(2n+1)\left(k+\frac{N}{2}\right)} \\ &= A(k) - W^{-k} B(k) \end{aligned} \tag{2.12}$$

II.1.2 Représentation paramétrique

Les méthodes précédentes ne font pas d'hypothèses sur la nature du signal analysé. Si l'on dispose d'un modèle adapté au signal de la parole, l'estimation des paramètres du modèle doit permettre une meilleure caractérisation du signal.

II.1.2.1 Les coefficients de prédiction linéaire LPC

Le principe fondamental de la prédiction linéaire est qu'un échantillon donné peut être prédit à partir d'une combinaison linéaire des échantillons finis qui le précèdent [9]. Un seul jeu de coefficients du prédicteur est déterminé en minimisant les différences entre les échantillons actuels et ceux prédits. La technique de prédiction linéaire est basée sur le modèle de production de la parole [20].

Ainsi, chaque échantillon de parole $s(n)$ est constitué par une combinaison linéaire des p échantillons passés. Le prédicteur est défini comme un système dont la sortie est:

$$\tilde{s}(n) = \sum_{k=1}^{p} a_k s(n-k) \tag{2.13}$$

L'erreur de prédiction est donnée par :

$$e(n) = s(n) - \tilde{s}(n) = s(n) - \sum_{k=1}^{p} a_k s(n-k) \tag{2.14}$$

On cherche à trouver un ensemble de coefficients a_k de façon à minimiser l'erreur de prédiction $e(n)$ dans un certain intervalle.

La moyenne de l'erreur est donnée :

$$E = \sum_n e^2(n) = \sum_n \left[s(n) - \sum_{k=1}^{p} a_k s(n-k)\right]^2 \tag{2.15}$$

$$\frac{\partial E}{\partial a_i} = 0 \qquad \text{pour } i = 1, \dots, p.$$

Alors :

$$\frac{\partial E}{\partial a_i} = -2 \sum_n \left\{\left[s(n) - \sum_{k=1}^{p} a_k s(n-k)\right] s(n-i)\right\} = 0 \tag{2.16}$$

Cette dernière équation nous conduit à écrire :

$$\sum_n s(n)s(n-i) = \sum_n \sum_{k=1}^{p} a_k s(n-k)s(n-i) \tag{2.17}$$

On déduit : $\phi(i,k) = \sum_n s(n-k)s(n-i)$

Alors : $\sum_{k=1}^{p} a_k \phi(i,k) = \phi(i,0), \quad i = 1, \dots, p.$ (2.18)

Cet ensemble de p équations à p inconnus peut être résolu d'une manière efficace pour les coefficients de prédiction inconnus $\{a_k\}$.

On suppose que le segment de parole est nul en dehors de l'intervalle $0 < n < L_a - 1$, ou L_a est la longueur de la fenêtre d'analyse LPC. Ceci est équivalent à multiplier le signal parole d'entrée par une fenêtre de longueur finie.

$e(n)$ est non nulle uniquement sur l'intervalle $0 < n < L_a + p - 1$.

Ainsi

$$\phi(i,k) = \sum_{n=0}^{L_a+p-1} s(n-i)s(n-k) \qquad i = 1, \dots, p \qquad (2.19)$$

$$k = 0, \dots, p.$$

On pose $m = (n-i)$,

$$\phi(i,k) = \sum_{m=0}^{L_a-1-(p+i)} s(m)s(m+i-k)\,. \qquad (2.20)$$

Donc, $\phi(i,k)$ est l'autocorrélation de $s(m)$ évaluée sur $(i-k)$.

D'où $$\phi(i,k) = R(i-k)$$

Donc : $$\sum_{k=1}^{p} a_k R(|i-k|) = R(i)$$

On obtient :

$$\begin{pmatrix} R(0) & R(1) & R(2) & \cdots & \cdots & R(p-1) \\ R(1) & R(0) & R(1) & \cdots & \cdots & R(p-2) \\ R(2) & R(1) & R(0) & \cdots & \cdots & R(p-3) \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ R(p-1) & R(p-2) & R(p-3) & \cdots & \cdots & R(0) \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_p \end{pmatrix} = \begin{pmatrix} R(1) \\ R(2) \\ R(3) \\ \vdots \\ R(p) \end{pmatrix} \qquad (2.21)$$

La matrice de dimension $p \times p$, des valeurs d'autocorrelations est une matrice de Toeplitz symétrique, tous les éléments d'une diagonale donnée sont égaux. Cette propriété peut être exploitée pour obtenir un algorithme efficace de résolution du système d'équations.

La solution la plus efficace est une méthode itérative connue sous le nom de l'algorithme de Wiener Levinson Durbin [10] :

$$E_0 = R_0 = \sigma_s^2$$

Pour *i* allons de *1* à *p*

$$k_i = -\left[R_i + \sum_{j=1}^{i-1} a_j^{(i-1)} R_{i-j}\right] / E_{i-1}$$

$$a_i^{(i)} = k_i$$

$$a_j^{(i)} = a_j^{(i-1)} + k_i a_{i-j}^{(i-1)}, \quad \forall 1 \le j \le i-1$$

$$E_i = (1 - k_i^2) E_{i-1}$$

$$a_j = a_j^{(p)}, \quad \forall 1 \le j \le p.$$

II.1.2.2 Les coefficients cepstraux de prédiction linéaire LPCC

Les coefficients cepstraux peuvent être calculés à partir de la sortie d'un banc de filtres ou à partir des coefficients de prédiction linéaire, ainsi les coefficients LPCC (*Linear Prediction Cepstral Coefficients*) sont dérivés directement des coefficients LPC.

Les coefficients cepstraux c_k sont obtenus :

$$c_k = -a_k - \sum_{i=1}^{k-1} \left(1 - \frac{i}{k}\right) a_i c_{k-i}\,, \quad k > 0 \tag{2.22}$$

II.1.2.3 Les coefficients MFCC (Mel Frequency Cepstral Coefficients)

Les coefficients cepstraux issus d'une analyse par Transformée de Fourier, caractérisent bien la forme du spectre et permettent de séparer l'influence de la source glottique de celle du conduit vocal.

Le cepstre du signal de parole est défini comme étant la Transformée de Fourier Inverse du logarithme de la densité spectrale de puissance. Pour ce signal, la source d'excitation glottique est convoluée avec la réponse impulsionnelle du conduit vocal [10].

$$s(t) = e(t) * h(t) \tag{2.23}$$

Où $s(t)$ est le signal de parole, $e(t)$ est la source d'excitation glottique et $h(t)$ est la réponse impulsionnelle du conduit vocal.

L'application du logarithme sur le module de la Transformée de Fourier dans l'équation (2.23) donne :

$$\log|S(f)| = \log|E(f)| + \log|H(f)| \tag{2.24}$$

Par une transformée de Fourier inverse on obtient :

$$s'(cef) = e'(cef) + h'(cef) \tag{2.25}$$

La dimension du nouveau domaine est homogène à un temps et s'appel la *quéfrence* (*cef*), le nouveau domaine s'appelle donc le domaine *quéfrentiel*. Un filtrage dans ce domaine s'appelle *liftrage* [11].

Ce domaine est intéressant pour faire la séparation des contributions du conduit vocal et de la source d'excitation dans le signal de parole. En effet, si les contributions relevant du conduit vocal et les contributions de la source d'excitation évoluent avec des vitesses différentes dans le temps, alors il est possible de les séparer par l'application d'un simple fenêtrage dans le domaine quéfrentiel (liftrage passe-bas) pour le conduit vocal.

Les coefficients cepstraux les plus répandus sont les MFCC (Mel Frequency Cepstral Coefficients). Ils présentent l'avantage d'être faiblement corrélés entre eux, et qu'on peut donc approximer leur matrice de covariance par une matrice diagonale.

Pour simuler le fonctionnement du système auditif humain, les fréquences centrales du banc de filtres sont réparties uniformément sur une échelle perceptive. Plus la fréquence centrale d'un filtre est élevée, plus sa bande passante est large. Cela permet d'augmenter la résolution dans les basses fréquences, zone qui contient le plus d'informations utiles dans le signal de parole. Les échelles perceptives les plus utilisées sont l'échelle Mel et l'échelle Bark [10].

- Echelle Mel $$Mel(f) = 2595 \log\left(1 + \frac{f}{700}\right) \quad (2.26)$$
- Echelle Bark $$Bark(f) = 6 \operatorname{arcsinh}\left(\frac{f}{1000}\right) \quad (2.27)$$

f représente la fréquence (Hz).

La procédure de calcul des coefficients MFCC est illustrée dans la figure II.7.

Figure II.7 : Calcul des coefficients MFCC.

Soit un signal discret $s(n)$ avec $0 \leq n \leq N-1, N$ est le nombre d'échantillons d'une fenêtre d'analyse, F_e est la fréquence d'échantillonnage, la Transformée de Fourier Discrète à court terme $S(k)$ est obtenue avec la formule :

$$S(k) = \sum_{n=0}^{N-1} s(n) \exp\left(\frac{-j2\pi nk}{N}\right), \ 0 \leq k \leq N-1 \quad (2.28)$$

Le spectre du signal est filtré par un banc de filtres triangulaires, dont les bandes passantes sont de même largeur dans le domaine des fréquences Mel. Les points de frontières B_m des filtres en échelle de fréquence Mel sont calculés à partir de la formule :

$$B_m = B_b + m\frac{B_h - B_b}{M+1}, \quad 0 \leq m \leq M+1 \quad (2.29)$$

M : Le nombre de filtres.

B_h : La fréquence la plus haute du signal.

B_b : La fréquence la plus basse du signal.

Dans le domaine fréquentiel, et d'après (2.29), les points f_m discrets correspondants sont calculés par la formule :

$$f_m = B^{-1}\left(B_b + m\frac{B_h - B_b}{M+1}\right) \tag{2.30}$$

Où $B^{-1}(x)$ désigne la fréquence correspondante à la fréquence x sur l'échelle Mel,

$$B^{-1}(x) = 700\left(10^{\frac{x}{2595}} - 1\right) \tag{2.31}$$

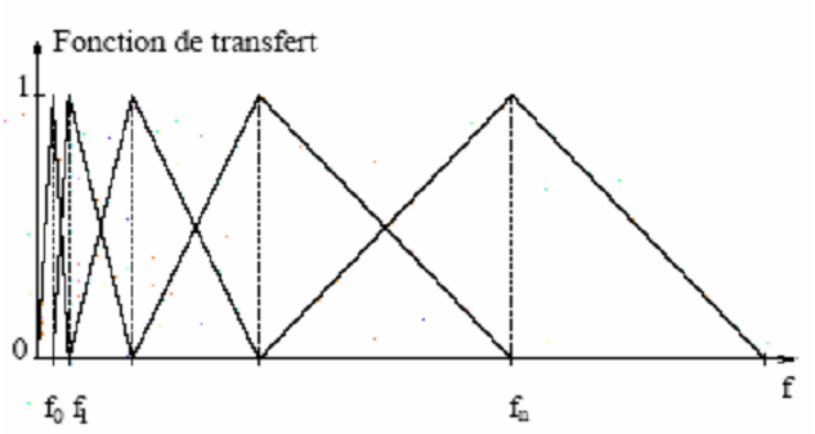

Figure II.8 : Banc de filtre sur l'échelle linéaire.

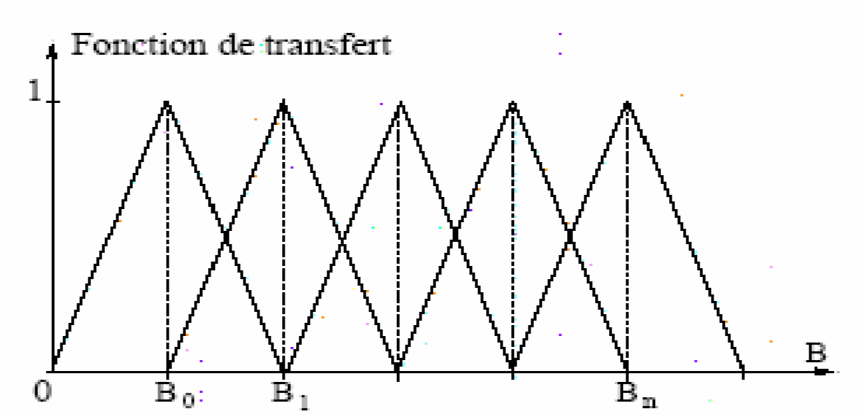

Figure II.9 : Banc de filtre sur l'échelle Mel.

Les coefficients cepstraux peuvent être calculés directement à partir du logarithme des énergies E_i issues d'un banc de M filtres par la transformée en cosinus discrète inverse définie par :

$$c_k = \sum_{i=1}^{M} \log E_i \cos\left[\frac{\pi k}{M}\left(i - \frac{1}{2}\right)\right], \qquad 1 \le k \le d \tag{2.32}$$

et qui permet d'obtenir des coefficients peu corrélés.

Le coefficient c_0 qui est la somme des énergies n'est pas utilisé ; il est éventuellement remplacé par le logarithme de l'énergie totale E calculée dans le domaine temporel et normalisée.

II.1.2.4 Les coefficients LFCC (Linear Frequency Cepstral Coefficients)

Aux coefficients MFCC s'ajoute un autre type de paramètres, les LFCC (Linear Frequency Cepstral Coefficients) qui sont calculés de la même manière que les MFCC, mais avec la seule différence est que les fréquences des filtres sont uniformément réparties sur l'échelle linéaire des fréquences, et non pas sur une échelle perceptive de type Mel [19].

II.1.2.5 Les coefficients différentiels

Pour prendre en compte la dynamique temporelle du signal de parole, on utilise en plus des paramètres cités précédemment, des coefficients différentiels du premier ordre et du second ordre issus des coefficients cepstraux ou de l'énergie. Soit le coefficient cepstral d'indice k de la trame t, alors le coefficient différentiel $\Delta c_k(t)$ correspondant est calculé sur $2n_\Delta + 1$ trames par :

$$\Delta c_k(t) = \frac{\sum_{i=-n_\Delta}^{n_\Delta} i c_k(t+i)}{\sum_{i=-n_\Delta}^{n_\Delta} i^2} \tag{2.33}$$

La dérivée première de l'énergie ΔE est calculée de la même façon par :

$$\Delta E(t) = \frac{\sum_{i=-n_\Delta}^{n_\Delta} i E(t+i)}{\sum_{i=-n_\Delta}^{n_\Delta} i^2} \tag{2.34}$$

Les coefficients différentiels du second ordre peuvent aussi contribuer à l'amélioration des systèmes de reconnaissance. Les coefficients $\Delta\Delta c_k$ et $\Delta\Delta E$ sont calculés par régression linéaire des coefficients Δc_k et ΔE respectivement, et sur $n_{\Delta\Delta}$(typiquement $n_\Delta = n_{\Delta\Delta} = 2$).

La figure II.10 schématise un module d'extraction des paramètres MFCC et leurs dérivés de premier et second ordre.

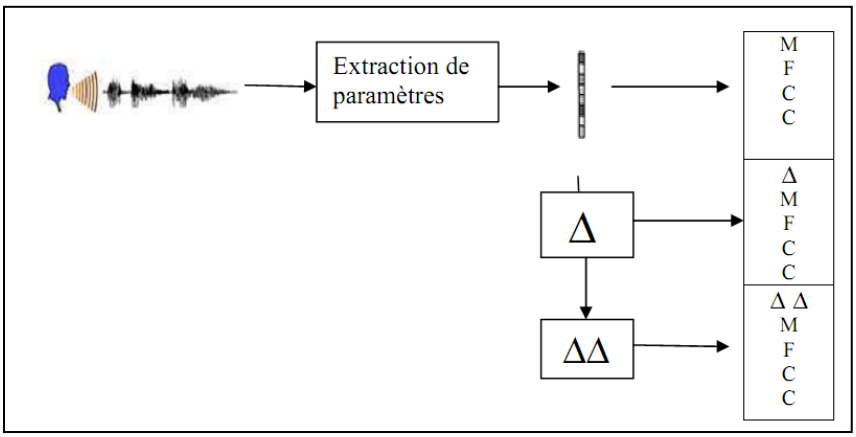

Figure II.10 : module d'extraction des paramètres MFCC et leurs dérivés de premier et second ordre.

II.1.3 Réduction de l'espace de représentation

L'utilisation de la totalité des composantes des vecteurs acoustiques est coûteuse en temps de calcul et des ressources CPU et mémoire. En classant les coefficients acoustiques selon un critère particulier, il est possible de ne considérer que certains coefficients.

Des analyses sont proposées pour réduire la dimension de l'espace des paramètres, comme l'analyse en composantes principales (ACP) et l'analyse linéaire discriminante (ALD).

II.1.3.1 Analyse en Composantes Principales (ACP)

L'ACP, notamment, utilisée dans le domaine de la parole, a pour but, partant d'un espace initial de représentation, de se projeter dans un sous-espace, de dimension inférieure, dans lequel les données seront représentées de manière compacte et dont les axes sont décorrélés. Les axes (orthogonaux entre eux) du nouvel espace sont déterminés de telle manière qu'ils maximisent la variabilité des données. La variabilité suivant l'axe *i* est supérieure à celle de l'axe *i + 1* de part leur orthogonalité. Ce principe est illustré par la figure II.11 pour un espace à deux dimensions.

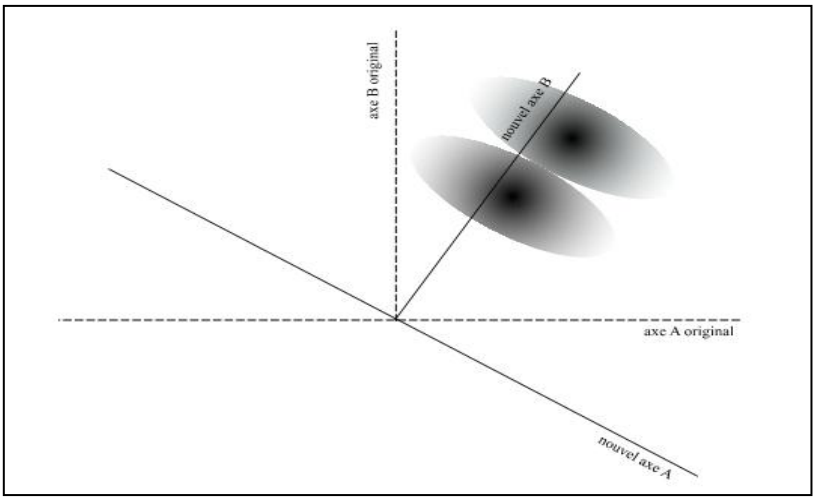

Figure II.11 : Exemple d'une Analyse en Composantes Principales d'un espace à deux dimensions.

Les axes principaux sont déterminés à partir de Σ, la matrice de covariance des données.

$$\Sigma = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)(x_i - \mu)' \qquad (2.35)$$

Où *N* correspond au nombre de vecteurs acoustiques (x_i) disponibles et μ est le vecteur moyenne estimé grâce à :

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i \quad (2.36)$$

Les directions maximisant la variabilité sont déterminées grâce aux valeurs propres (λ) et aux vecteurs propres associés (v_i). La rotation est donc définie par la matrice R :

$$R = [v_1 v_2 \dots v_n] \quad (2.37)$$

A ce stade, le nouvel espace de représentation est de la même taille que l'espace initial. Ce nouvel espace permet juste de décorréler statistiquement les axes entre eux. Afin de réduire la taille du vecteur acoustique, seuls les axes comportant le maximum d'informations (déterminés par les plus grandes valeurs propres) sont conservés.

L'inconvénient principal de l'ACP est qu'elle est focalisée sur la recherche d'axes maximisant la variance des données sans tenir compte de la capacité discriminante des données.

II.1.3.2 Analyse Linéaire Discriminante (ALD)

L'objectif principal de l'ALD (initialement présentée, pour le contexte de la RAP, par Hunt [13]), contrairement à l'ACP, est de séparer l'espace de manière à diminuer la variance intra-classe tout en augmentant la variance interclasse : le choix des classes est donc un paramètre important à déterminer. Ceci impose également de connaître la classe à laquelle appartient chaque vecteur de paramètres.

La figure II.12 illustre le fonctionnement de l'analyse discriminante avec des données en deux dimensions. On note que les deux distributions sont très peu discriminées, que ce soit suivant l'axe original A ou le B. Le premier axe obtenu avec l'ALD permet, lui, une classification nettement plus précise en discriminant bien les deux classes.

La nouvelle base est obtenue grâce aux vecteurs propres du produit : $\Sigma_{ec} * \Sigma_{ic}^{-1}$. Avec Σ_{ec} qui correspond à la variance entre-classe et Σ_{ic} la variance intra-classe.

La variance intra-classe, Σ_{ic}, que l'on cherche à diminuer, est estimée comme la somme pondérée de la matrice de covariance de toutes les classes :

$$\Sigma_{ic} = \frac{1}{N}\sum_{j=1}^{J} N_j \Sigma^j \quad (2.38)$$

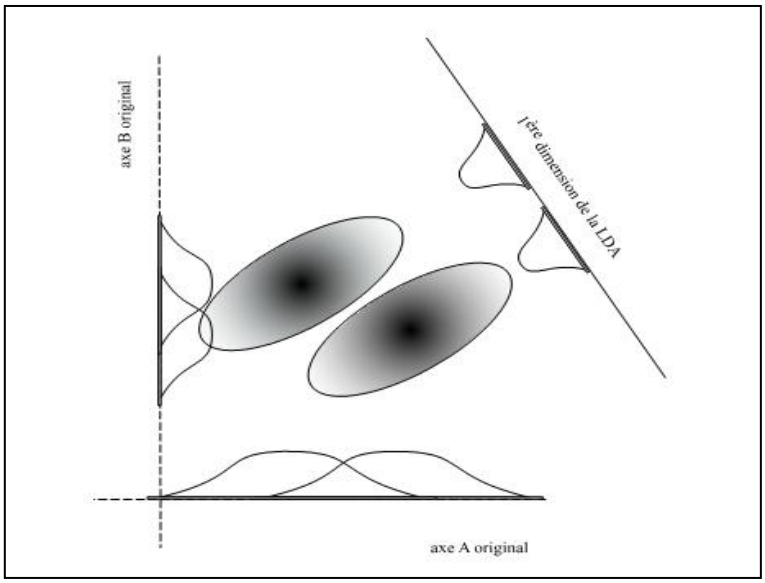

Figure II.12 : Exemple d'une Analyse Linéaire Discriminante d'un espace à deux dimensions.

Où *J* est le nombre de classes, N_j le nombre de vecteurs associés à la classe *j*, *N* le nombre total de vecteurs et, enfin, Σ^j l'estimation de la matrice de covariance de la classe *j* définie par :

$$\Sigma^j = \frac{1}{N_j}\sum_{i=1}^{N_j}(x_i^j - \mu^j)(x_i^j - \mu^j)' \tag{2.39}$$

x_i^j correspond au $i^{ème}$ vecteur de la classe *j*. μ^j est la moyenne estimée de la classe *j* :

$$\mu^j = \frac{1}{N_j}\sum_{i=1}^{N_j} x_i \tag{2.40}$$

La variance entre-classe, Σ_{ec}, représente la variance à augmenter pour discriminer au mieux les différentes classes. Elle est estimée par :

$$\Sigma_{ec} = \frac{1}{N}\sum_{j=1}^{J} N_j(\mu^j - \mu)(\mu^j - \mu)' \tag{2.41}$$

$$= \Sigma - \Sigma_{ic}$$

μ est définie par l'équation (2.36).

De la même manière que pour l'ACP, les valeurs propres de $\Sigma_{ec} * \Sigma_{ic}^{-1}$ sont ordonnées pour le choix des axes (définis par les vecteurs associés aux valeurs propres).

II.2 Approches pour la modélisation acoustique

On distingue usuellement en reconnaissance de la parole l'approche analytique et l'approche globale. La première approche cherche à traiter la parole continue en décomposant le problème, le plus souvent en procédant à un décodage acoustico-phonétique exploité par des modules de niveau linguistique. La seconde consiste à identifier globalement un mot ou une phrase en les comparants avec des références enregistrées. La distinction entre globale et analytique a perdu sa pertinence avec l'introduction des méthodes statistiques à base de modèle de Markov pour la reconnaissance de la parole continue et le traitement de grands vocabulaires ; il s'agit de méthodes globales qui peuvent exploiter de unités sub-lexicales [6].

II.2.1 Approche analytique

L'approche analytique cherche à résoudre le problème de la parole continue en isolant des unités acoustiques courtes comme les phonèmes, les di-phonèmes ou les syllabes. Un exemple classique de cette approche est l'analyse par traits : des indices acoustiques sont calculés à partir du signal de parole ; ils permettent de faire des hypothèses locales sur certains traits phonétiques, comme le voisement, la nasalisation, le lieu d'articulation ou le degré d'ouverture du conduit vocal. En fonction de ces traits, le signal acoustique est segmenté et une identification phonétique des segments est réalisée. Le décodage acoustico-phonétique ainsi obtenu est exploité par des modules d'ordre linguistique. Les niveaux lexical, syntaxique ou sémantique utilisent des sources de connaissance spécialisées et sont organisés avec le module acoustique dans des architectures montantes ou descendantes [14].

Les systèmes analytiques, conçu avec des objectifs ambitieux, sont restés au stade expérimental. Leur faiblesse provient d'un processus de décision trop précoce, à savoir une segmentation préalable à l'identification ou une identification phonétique sans prise en compte des niveaux linguistiques. Les méthodes globales, développées pour la reconnaissance de mots isolés, ne font pas d'hypothèses sur la structure phonétique des mots, ce qui évite une erreur pénalisante au début du traitement.

II.2.2 Approche globale

Les méthodes globales identifient un mot ou une phrase en les considérants comme des unités élémentaires et en les comparants avec des références enregistrées. Leur essor en reconnaissance de la parole est dû à l'exploitation de critères de comparaison performants, comme l'alignement temporel dynamique des formes acoustiques, et à leur application à des représentations adaptées du signal, qu'il s'agisse de l'analyse spectrale ou de la prédiction linéaire.

Disposant d'une représentation du signal de parole, la reconnaissance de mots isolés est un problème classique de reconnaissance des formes. L'ensemble des n_m mots du vocabulaire est noté $E_m = \{m_k\}_{1 \le k \le n_m}$ et chaque mot m_k est représenté par une ou plusieurs formes acoustiques de références R_{m_k}, par exemple les paramètres spectraux calculés de manière périodique sur le signal. Une forme de teste observée O, qui est la suite des spectres d'un mot inconnu, est comparée à chacune des références. Le mot inconnu est identifié au mot de référence $\tilde{m}$ dont il est le plus proche au sens d'une certaine distance D :

$$\tilde{m} = arg \min_{m \in E_m} D(O, R_m) \tag{2.42}$$

Le calcul de la distance nécessite la mise en correspondance d'une forme de référence et la forme inconnue. Or, la durée d'un même mot est variable d'une prononciation à l'autre, et de plus les déformations ne sont pas linéaires en fonction du temps. La distance D est donc calculée sur l'alignement temporel qui rapproche le mieux les deux formes. Mais une recherche exhaustive de toutes les déformations possibles est exclue en raison de l'explosion combinatoire.

L'alignement temporel dynamique (*Dynamic Time Warping* ou DTW) résout efficacement ce problème. La méthode est efficace pour la reconnaissance mono-locuteur à petit vocabulaire et en mots isolés. Des extensions ont été proposées pour la reconnaissance indépendante du locuteur ou la reconnaissance de mots enchainés [15]. Cependant, l'approche statistique propose un formalisme plus général et permet la reconnaissance de grands vocabulaires en parole continue de manière plus efficace que par DTW en intégrant la modélisation des niveaux linguistiques.

II.2.3 Approche statistique

C'est une formalisation statistique simple, qui est aujourd'hui classique pour décomposer le problème de la reconnaissance de la parole continue. Soit O une suite d'observations acoustiques, et M une suite de mots prononcés. Connaissant les observations O, on cherche la suite de mots $\tilde{M}$ la plus probable parmi toutes les suites possibles E_M, soit :

$$\tilde{M} = arg \max_{M \in E_M} P(M/O) \tag{2.43}$$

En utilisant la règle de Bayes, il est possible d'écrire la probabilité qu'une suite de mots correspond aux observations acoustiques comme :

$$P(M/O) = \frac{P(O/M).P(M)}{P(O)} \tag{2.44}$$

Puisque $P(O)$ ne dépend pas de M, l'équation (2.43) est équivalente à :

$$\tilde{M} = arg \max_{M \in E_M} P(O/M).P(M) \tag{2.45}$$

Où

- $P(O/M)$, la probabilité des observations O, étant donnée la suite de mots M, est estimée par une modélisation acoustique ;
- $P(M)$, la probabilité *a priori* de la suite de mots M, est estimée par un modèle linguistique.

L'approche statistique permet ainsi d'intégrer les niveaux acoustiques et linguistiques dans un seul processus de décision. Ces niveaux sont classiquement représentés par des modèles de Markov cachés (*Hidden Markov Models* ou HMM). Les unités acoustiques modélisées peuvent être des mots comme dans l'approche globale, ou des unités plus courtes telles que le phonème comme dans l'approche analytique. La modélisation markovienne est plus générale que l'alignement temporel dynamique et tient compte non seulement de la non linéarité temporelle du processus mais aussi de la variabilité acoustique de la production de la parole [6].

II.3 Conclusion

Dans ce chapitre on a illustré les différentes méthodes d'analyse du signal de la parole qui sont classées en deux grandes classes, les méthodes d'analyse non paramétriques et les méthodes paramétriques.

Pour la première classe, l'analyse peut se faire dans le domaine temporel ou fréquentiel pour extraire des indices acoustiques tells que l'énergie, la fréquence fondamentale, les fréquences des formants.

Les méthodes paramétriques, contrairement aux précédentes, tiennent compte du processus de phonation. Les coefficients issus de ces méthodes tells que les MFCC et LPC sont ceux utilisés en reconnaissance de la parole car ils nous offrent une meilleure représentions du signal ; et pour mieux manipuler ces derniers on procède souvent à une réduction de l'espace de représentation par des méthodes telles que l'analyse en composante principale (ACP) et l'analyse linéaire discriminante (ALD) qui nous permettent d'avoir des espaces à dimension réduite et des coefficients moins corrélés entre eux.

La dernière section de ce chapitre présente les principes des différentes approches utilisées en RAP (l'approche analytique, globale et statistique). Dans notre travail on a opté pour l'approche statistique, ce choix est justifié par la très grande liberté laissée dans le choix des unités lexicales, et par le processus de décision qui fait intervenir simultanément les niveaux acoustiques et linguistiques.

CHAPITRE III

Modélisation à base des HMM

Introduction

Depuis leur introduction dans le traitement de la parole, les modèles de Markov cachés (HMM) ont pris une importance considérable, au point que la quasi-totalité des systèmes de RAP utilisent cette modélisation. Les modèles de Markov cachés supposent que le phénomène modélisé est un processus aléatoire et inobservable qui se manifeste par des émissions elles-mêmes aléatoires. Ces deux niveaux donnent à l'approche markovienne une flexibilité qui est séduisante pour modéliser un phénomène aussi complexe que la production de la parole.

De nombreuses présentations théoriques des HMM existent dans la littérature ; nous reprenons en partie les notations de L.Rabiner [16].

III.1 Définitions

Un modèle de Markov λ est un automate probabiliste constitué de N états. Un processus aléatoire se déplace d'état en état à chaque instant, et on note q_t le numéro de l'état atteint par le processus à l'instant t. L'état réel q_t du processus n'est pas directement observable « on dit qu'il est caché » mais le processus émet après chaque changement d'état un symbole discret o_t qui appartient à un alphabet fini de n_v symboles $V = \{v_k\}_{1\leq k\leq n_v}$. La probabilité de passer de l'état i à l'état j à l'instant t et d'émettre le symbole v_k ne dépend ni du temps, ni des états aux instants précédents. Un modèle de Markov caché HMM est alors défini par [6] :

- $Q = \{q_i\}_{1\leq i\leq N}$ l'ensemble des N états, en sachant que le processus part de l'état initial q_1 à l'instant $t = 0$ et arrive à l'état final q_N à l'instant $t = T$
- $A = \{a_{ij}\}_{1\leq i,j\leq N}$ l'ensemble des probabilités de transition entre les états i et j :

$$a_{ij} = P(q_t = j/q_{t-1} = i) \tag{3.1}$$

- $\pi = \{\pi_i\}$ le vecteur des probabilités initiales :

$$\pi_i = P(q_1 = i), \quad 1 \leq i \leq N \tag{3.2}$$

- $B = \{b_j(k)\}_{1\leq j\leq N, 1\leq k\leq n_v}$ l'ensemble des probabilités d'émission du symbole v_k lors de l'arrivée dans l'état j, avec :

$$b_j(k) = P(o_t = v_k/q_t = j) \tag{3.3}$$

III.2 Modélisation de la parole par un HMM

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

III.2.1 Principe de la modélisation

Un modèle HMM va modéliser un signal de parole d'une telle façon que chaque segment supposé stationnaire ou pseudo-stationnaire de signal va correspondre à un état dans le modèle HMM. Chaque état du 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).

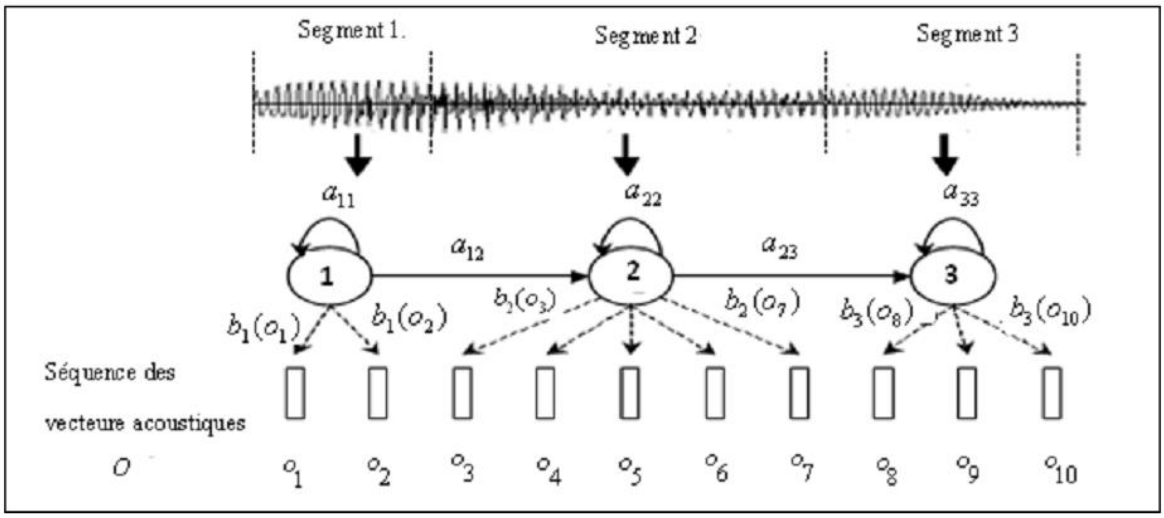

Figure III.1 : Un exemple de HMM à trois états modélisant un signal contenant 10 vecteurs acoustiques.

III.2.2 Topologie des HMMs utilisés pour la parole

La parole est un phénomène dont la dimension temporelle ne peut être ignorée. Les HMMs utilisés pour la représentée 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 :

$$i > j \quad => \quad a_{ij} = 0 \quad 2 \le i \le N,\ 1 \le j \le N-1$$
(3.4)

$$\pi_i = P(q_1 = i) = \begin{cases} 1 & pour \quad i = 1 \\ 0 & pour \quad 1 < i \le N \end{cases} \tag{3.5}$$

Dans ce cadre, R. Bakis a proposé un modèle type pour représenter un mot qui permet le bouclage sur l'état courant et le passage à l'état suivant (figure III.2). Le nombre d'états du modèle est normalement proportionnel à la durée moyenne du mot.

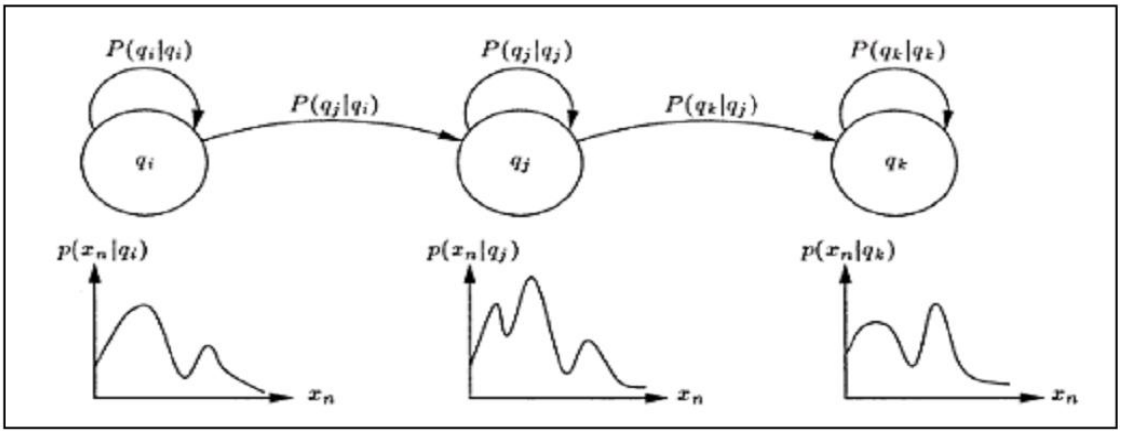

Figure III.2 : Exemple d'un HMM avec une topologie de type Bakis à 3 états.

III.2.3 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.2.3.1 Observations discrètes

L'espace de représentation du signal de parole est généralement un espace multidimensionnel continu E, et les vecteurs de coefficients calculés sur une trame de signal sont des points de cet espace. Il est possible de modéliser ces observations par des modèles de Markov à émissions discrètes au moyen de la quantification vectorielle (QV).

La QV permet le passage de l'espace E vers un espace discret, en partitionnant cet espace et en choisissant un représentant pour chaque classe. L'ensemble des représentants constitue

un dictionnaire de M prototypes noté $V = \{v_k\}_{1 \le k \le M}$. Chaque vecteur $O \in E$ est quantifié par le prototype du dictionnaire dont il est le plus proche au sens d'une distance $d(O, \hat{O})$ définie dans l'espace E :

$$O \xrightarrow{QV} \hat{O} = v_{r'} \qquad avec \qquad r' = \arg\min_{1 \le r \le M} d(O, \hat{O}) \tag{3.6}$$

Divers algorithmes ont été proposés pour la réalisation du dictionnaire des prototypes, basés le plus souvent sur une classification hiérarchique descendante ou sur les nuées dynamiques [17]. Cependant, la QV introduit des distorsions, et il est souvent préférable de travailler directement dans l'espace continu E.

III.2.3.2 Observations continues

Le principe de l'émission de symboles discrets peut se généraliser au cas continu. Les probabilités d'émission discrètes $b_j(k)$ sont alors remplacées par des densités de probabilité continues dans l'espace de représentation. Cette solution évite les distorsions introduites par la QV, mais pose le problème du choix des densités de probabilité et de la robustesse de leur estimation. L'utilisation d'une combinaison linéaire de gaussiennes dans l'espace R^d est fréquente :

$$b_j(o) = \sum_{k=1}^{M} c_{jk} N\left(o, \mu_{jk}, \Sigma_{\text{jk}}\right) = \sum_{k=1}^{M} c_{jk} b_{jk}(o) \qquad 1 \le j \le N \tag{3.7}$$

Avec M est le nombre de gaussiennes à l'état j. $N\left(o, \mu_{jk}, \Sigma_{\text{jk}}\right)$ est la distribution gaussienne du $k^{ème}$ mélange de l'état j. Cette distribution est définie par le vecteur moyen μ_{jk} et la matrice de covariance Σ_{jk} à l'état j. c_{jk} est le coefficient de pondération du $k^{ème}$ mélange qui satisfait la contrainte stochastique :

$$\sum_{k=1}^{M} c_{jk} = 1 \quad avec \quad c_{jk} > 0,\ 1 \le j \le N,\ 1 \le k \le M \tag{3.8}$$

Nous rappelons que la densité de probabilité d'une loi normale de moyenne μ et de matrice de covariance Σ en dimension d est :

$$N(o, \mu, \Sigma) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} \exp -\frac{1}{2} (o - \mu)^t \Sigma^{-1} (o - \mu) \tag{3.9}$$

L'hypothèse qui est souvent faite d'une indépendance entre les d dimensions de l'espace autorise l'utilisation de matrices de covariance diagonales, cela limite le nombre de paramètres à estimer et simplifie les calculs. D'autres types de densités de probabilité sont possibles, comme la sortie d'un réseau de neurones dans le cas de systèmes de reconnaissance hybride [17].

III.3 Un système de RAP à base des HMM

A partir d'une suite d'observations *O* supposées émises par un modèle, différents problèmes peuvent être posés :

- L'évaluation de la probabilité que la suite des observations ait été émise par un modèle. Lorsque plusieurs modèles existent, cette évaluation permet le choix du modèle le plus probable.
- La recherche de la séquence cachée d'états la plus probable d'un modèle ayant produit les observations.
- L'apprentissage des paramètres d'un modèle. A partir d'un modèle initial et d'observations supposées émises par ce modèle, on cherche les probabilités de transition et d'émission maximisant la vraisemblance des observations.

Nous allons commencer par présenter la résolution de ces problèmes dans le cadre simplifié de la reconnaissance de mots isolés dans laquelle chaque modèle HMM représente un mot de vocabulaire. La reconnaissance de parole continue n'est qu'une généralisation de ces résolutions et elle va être abordée dans le paragraphe III.4.

III.3.1 Reconnaissance d'un modèle

Connaissant une suite d'observation O, la règle de décision Bayésienne désigne le modèle qui les a émis. Il est nécessaire pour cela de calculer la probabilité d'émission de la suite des observations par chaque modèle. L'évaluation de cette probabilité est compliquée par le fait que le chemin parcouru n'est pas connu, mais un algorithme récursif efficace peut être utilisé ; de plus, certaines variables introduites ici servent pour l'apprentissage des modèles.

III.3.1.1 Décision bayésienne

Supposons que les observations $O = (o_1 \dots o_T)$ sont les trames acoustiques d'un mot inconnu. On cherche à trouver le mot $\tilde{m}$ qui a été prononcé parmi l'ensemble E_m des mots du vocabulaire, soit d'après la règle de décision bayésienne :

$$\tilde{m} = arg \max_{m \in E_m} P(m/O) = arg \max_{m \in E_m} P(O/m)P(m) \tag{3.10}$$

Chaque mot m est modélisé par une machine λ_m, d'où l'hypothèse suivante :

$$P(O/m) = P(O/\lambda_m) \tag{3.11}$$

L'équation (3.10) peut alors être reformulée :

$$\tilde{m} = arg\max_{m \in E_m} P(O/\lambda_m)P(m) \quad (3.12)$$

Sa résolution nécessite l'estimation de la probabilité d'émission des observations $P(O/\lambda_m)$ pour chacune des machines modélisant un mot.

III.3.1.2 Probabilité d'émission des observations

La probabilité que la suite d'observations $O = (o_1 \dots o_T)$ soit émise par la machine λ n'est pas directement calculable, car le chemin emprunté est *à priori* inconnu. Mais elle peut être reformulée comme la somme des probabilités conjointes de l'observation O et du chemin Q pour l'ensemble E_Q de tous les chemins possible de longueur T :

$$P(O/\lambda) = \sum_{Q \in E_Q} P(O, Q/\lambda) \quad (3.13)$$

D'après les définitions du modèle, la probabilité de partir à l' instant $t = 0$ de l'état d'indice $q_0 = 1$ et de suivre le chemin $Q = (q_1 \dots q_T)$ est le produit des probabilités de transition sur le chemin :

$$P(Q/\lambda) = \prod_{t=1}^{T} a_{q_{t-1}q_t} \quad (3.14)$$

Et la probabilité d'avoir émis les observations O en suivant ce chemin est :

$$P(O/Q\,, \lambda) = \prod_{t=1}^{T} b_{q_t}(o_t) \quad (3.15)$$

D'où la probabilité conjointe du chemin et des observations :

$$P(O, Q/\lambda) = P(Q/\lambda).P(O/Q\,, \lambda) = \prod_{t=1}^{T} a_{q_{t-1}q_t} b_{q_t}(o_t) \quad (3.16)$$

La probabilité d'émission des observations est finalement :

$$P(O/\lambda) = \sum_{Q \in E_Q} \prod_{t=1}^{T} a_{q_{t-1}q_t} b_{q_t}(o_t) \quad (3.17)$$

Avec une machine à N états, le nombre de chemins possible est de l'ordre de N^T, et l'ensemble des chemins E_Q devient très rapidement impossible à décrire. Il existe heureusement un algorithme rapide, dit « Forward-Backward » (directe-rétrograde), qui permet de calculer récursivement cette quantité en maitrisant l'explosion combinatoire.

III.3.1.2.1 Estimation directe

Une variable intermédiaire est introduite pour le calcul de la probabilité d'émission. La variable directe $\alpha_t(i)$ est definie comme la probabilté que les observations jusqu'à l'instant t aient été émises par le modèle λ à N états, et que l'état à cet instant soit l'état d'indice i :

$$\alpha_t(i) = P(o_1 \dots o_t, q_t = i/\lambda) \tag{3.18}$$

Alors une récurrence sur le temps en parallèle pour tous les états permet d'obtenir $P(O/\lambda)$, en suivant les étapes suivantes :

- ✓ Départ du processus dans l'état initial :

$$\alpha_0(i) = \begin{cases} 1, & i = 1 \\ 0, & 1 < i < N \end{cases}$$

- ✓ Récurrence sur le temps et sur les états pour calculer les valeurs de $\alpha_t(j)$:

Pour t allant de 1 à T,

Pour j allant de 1 à N,

$$\alpha_t(j) = \sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij} b_j(o_t)$$

Fin

Fin

- ✓ Arrivée du processus dans l'état final :

$$P(O/\lambda) = \alpha_T(N)$$

Cette méthode exploite une structure en treillis combinant les états et les instants afin de partager des valeurs intermédiaires entre plusieurs chemins.

III.3.1.2.2 Estimation rétrograde

L'estimation directe est suffisante pour obtenir la probabilité d'émission recherchée. Cependant une estimation rétrograde dans le temps est aussi possible, avec la probabilité $\beta_t(i)$ que les observations après l'instant t soient émises en partant de l'état d'indice i :

$$\beta_t(i) = P(o_{t+1} \dots o_T/q_t = i, \lambda) \tag{3.19}$$

Et la récurrence est très similaire à la précédente :

- ✓ Initialisation : $\begin{cases} \beta_T(i) = 0, & 1 \leq i < N \\ \beta_T(N) = 1 & i = N \end{cases}$

- ✓ Récurrence

Pour t allant de T à 1 :

$$\beta_{t-1}(i) = \sum_{j=1}^{N} a_{ij} b_j(o_t) \beta_t(j), \quad 1 \leq i \leq N$$

Fin

- ✓ Terminaison : $P(O/\lambda) = \beta_0(1)$

Les variables directes et rétrogrades sont utilisées conjointement lors de l'apprentissage des modèles.

III.3.2 Recherche des états cachés

La procédure d'estimation directe ou rétrograde fournit la probabilité d'émission des observations cumulée sur toutes les séquences d'états possibles, sans choisir un chemin particulier. L'algorithme de Viterbi cherche la séquence d'états cachés la plus probable et calcule la probabilité d'émission le long de ce chemin. La probabilité ainsi estimée néglige les chemins moins probables.

III.3.2.1 Chemin optimal

Au vu des observations O émises par le modèle λ, la séquence d'état Q la plus probable ayant pu émettre ces observations est donnée par :

$$\tilde{Q} = arg \max_{Q \in E_Q} P(Q/O, \lambda) \tag{3.20}$$

La règle Bayes nous permet d'écrire après simplification du terme constant :

$$\tilde{Q} = arg \max_{Q \in E_Q} P(O, Q/\lambda) \tag{3.21}$$

Et donc, en reprenant l'équation (3.16) :

$$P(O, \tilde{Q}/\lambda) = \max_{Q \in E_Q} \prod_{t=1}^{T} a_{q_{t-1} q_t} b_{q_t}(o_t) \tag{3.22}$$

La recherche du maximum remplace ici la somme de l'équation (3.17). L'algorithme de Viterbi, qui est comme l'algorithme de DTW une application de la programmation dynamique, résout ce problème de manière récursive.

III.3.2.2 L'algorithme de Viterbi

La variable $\Phi(t,i)$ est définie comme la probabilité maximale que les observations observées jusqu'à l'instant t aient été émises par le modèle λ en suivant un chemin qui arrive à l'état d'indice i :

$$\Phi(t,i) = \max_{q_1 \dots q_{t-1}} P(o_1 \dots o_t, q_1 \dots q_{t-1} q_t = i/\lambda) \tag{3.23}$$

Alors une récurrence similaire à celle suivie pour le calcul de la probabilité d'émission s'applique, à laquelle s'ajoute la mémorisation du meilleur chemin :

- ✓ Le processus est initialement dans l'état d'indice 1 :

$$\Phi(0,i) = \begin{cases} 1, & i = 1 \\ 0, & 1 < i \leq N \end{cases}$$

- ✓ Récurrence sur les valeurs de $\Phi(t,i)$:

  Pour t allant de 1 à T,

  Pour j allant de 1 à N,

$$\Phi(t,i) = \max_{1 \leq i \leq N} \Phi(t-1,i) a_{ij} \mathrm{b_j}(\mathrm{o_t})$$

  et l'on conserve la mémoire du meilleur état précédent :

$$m(t,j) = arg \max_{1 \leq i \leq N} \Phi(t-1,i) a_{ij} \mathrm{b_j}(\mathrm{o_t})$$

  Fin

  Fin

- ✓ Arrivée du processus dans l'état final :

$$P(O, \tilde{Q}/\lambda) = \Phi(T,N)$$

  et construction du meilleur chemin $\tilde{Q} = (\tilde{q}_1 \dots \tilde{q}_T)$ avec :

$$\tilde{q}_T = m(T,N)$$

  puis pour t allant de $T-1$ à 1 :

$$\tilde{q}_{t-1} = m(t, \tilde{q}_t)$$

La probabilité d'émission sur le meilleur chemin peut être utilisée pour la reconnaissance comme une approximation de la probabilité d'émission par le modèle ; mais cette méthode de résolution est sous-optimale puisqu'elle néglige les chemins de plus faibles probabilités :

$$P(O, \tilde{Q}/\lambda) < P(O/\lambda) \tag{3.24}$$

La segmentation du signal fournie par l'algorithme de Viterbi sert principalement à l'initialisation des modèles à l'apprentissage et à la reconnaissance de la parole continue.

III.3.3 Apprentissage d'un modèle

La reconnaissance d'un mot prononcé est rendu possible par l'évaluation de la probabilité d'émission des observations par tous les modèles de mots. Cela suppose l'existence d'un modèle au moins pour chaque mot, et l'apprentissage des paramètres de ces modèles. Ces paramètres sont les probabilités de transition entre états et les probabilités d'émission associées aux états, car la topologie du modèle, à savoir le nombre d'états des modèles, les transitions autorisées entre ces états et l'alphabet des symboles émis, sont supposés fixées *a priori*. Ainsi connaissant une suite d'observations émises par un modèle, il est possible de modifier les paramètres du modèle de manière à rendre plus probable l'émission des observations par le modèle. Il s'agit d'une estimation sur le critère du maximum de vraisemblance (*Maximum Likelihood Estimation* ou MLE) qui est réalisée par l'algorithme de Baum-Welch.

III.3.3.1 Maximum de vraisemblance

L'estimation par maximum de vraisemblance (Maximum Likelihood Estimation ou MLE) consiste à choisir les paramètres du modèle λ afin de rendre maximum la probabilité d'émission des observations O par le modèle :

$$\tilde{\lambda} = arg\max_{\lambda} P(O/\lambda) \tag{3.25}$$

Une résolution analytique directe n'est pas possible, mais les formules de Baum-Welch permettent une ré-estimation itérative des paramètres a_{ij} et $b_j(k)$ du modèle en appliquant ce critère [6]. A la suite de la ré-estimation des paramètres du modèle $\tilde{\lambda}_n$, le nouveau modèle $\tilde{\lambda}_{n+1}$ vérifie.

$$P\left(O/\tilde{\lambda}_{n+1}\right) \geq P(O/\tilde{\lambda}_n) \tag{3.26}$$

La convergence vers un optimum local est démontée, mais les valeurs initiales des paramètres *A* et *B* sont cruciales pour assurer une convergence correcte et rapide le plus près possible du maximum global. L'algorithme de Viterbi réalisant le décodage peut servir à l'initialisation des modèles.

III.3.3.2 Ré-estimation du modèle

La ré-estimation des paramètres du modèle λ est basé sur le comptage du nombre moyen de transition observée entre les états *i* et *j*. La probabilité $\omega_t(i,j)$ de suivre cette transition à l'instant *t* peut s'exprimer au moyen des variables directes et rétrogrades.

$$\omega_t(i,j) = P(q_{t-1} = i, q_t = j/O, \lambda) = \frac{\alpha_{t-1}(i) a_{ij} b_j(o_t) \beta_t(j)}{P(O/\lambda)} \tag{3.27}$$

Le nombre moyen de transition entre *i* et *j* est donc :

$$\gamma_{ij} = \sum_{t=1}^{T} \omega_t(i,j) \tag{3.28}$$

Et la probabilité de transition est ré-estimée par :

$$\tilde{a}_{ij} = \frac{\gamma_{ij}}{\sum_{k=1}^{N} \gamma_{ik}} = \frac{nombre\ de\ transitions\ de\ l'état\ i\ vers\ létat\ j}{nombre\ de\ transitions\ depuis\ l'état\ i} \tag{3.29}$$

Il est important de remarquer que la structure du modèle est conservée au cours de la ré-estimation ; les transitions initialement interdites entre deux états le restent :

$$a_{ij} = 0 \ => \tilde{a}_{ij} = 0 \tag{3.30}$$

L'estimation de la probabilité d'émission associée à l'état nécessite le décompte des observations correspondant à chaque catégorie de symbole :

$$\tilde{b}_j(k) = \frac{1}{\gamma_j} \sum_{\substack{t=1 \\ o_t = v_k}}^{T} \omega_t(j) = \frac{nombre\ d'observations\ du\ symbole\ v_k\ dans\ l'état\ j}{nombre\ de\ passages\ par\ l'état j} \tag{3.31}$$

$$1 \leq k \leq n_V$$

Avec :

$$\omega_t(j) = P(q_t = j/O, \lambda) = \frac{\alpha_t(j)\beta_t(j)}{P(O/\lambda)} \quad \text{et} \quad \gamma_j = \sum_{t=1}^{T} \omega_t(j) \tag{4.32}$$

La ré-estimation des probabilités d'émission est différente pour des modèles continus.

Nous détaillons le cas de densité de probabilité continues représentées par une gaussienne multidimensionnelle, mais ces formules peuvent être généralisées au cas de multi-gaussienne. Le vecteur de moyenne et la matrice de covariance de la densité de probabilité associée à l'état *i* sont recalculés comme :

$$\tilde{\mu}_i = \frac{1}{\gamma_i} \sum_{t=1}^{T} \omega_t(j) o_t \tag{3.33}$$

Et :

$$\tilde{\Sigma}_i = \frac{1}{\gamma_i} \sum_{t=1}^{T} \{\omega_t(i)(o_t - \mu_i)\ (o_t - \mu_i)^t\} \tag{3.34}$$

III.4 Reconnaissance de la parole continue

Le processus d'interprétation d'une phrase est en général décrit comme une succession d'étapes depuis le niveau acoustique jusqu'au niveau sémantique. Dans ce schéma général, le niveau de décodage acoustico-phonétique (DAP) constitue une étape importante et une difficulté majeure dans la conception d'un système de reconnaissance [17]. Le DAP concerne l'ensemble des processus de transformation du signal acoustique continu en une description linguistique discrète sous forme d'unités telle que phonème, diphonèmes, mots, … etc.

III.4.1 Modèle acoustique

Dans notre système, on a opté pour que l'unité acoustique la plus petite soit le mot, les modèles des phrases sont alors construits par concaténation des modèles de mots.

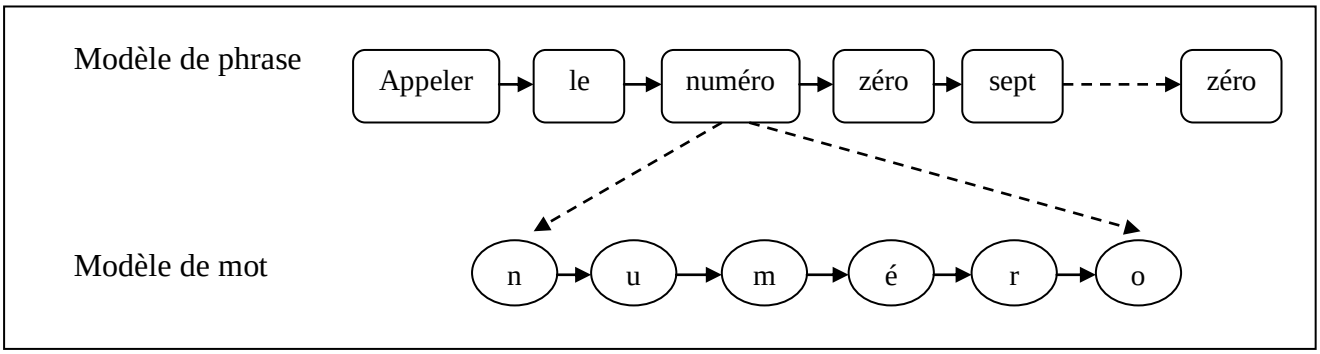

Figure III.3 : Modélisation d'une phrase à partir de modèles de mots.

III.4.2 Modèle de langage

Une reconnaissance acoustique même parfaite ne suffit pas pour obtenir une transcription correcte de la phrase. Il est donc indispensable d'introduire dans les systèmes de RAP des contraintes du langage.

Le modèle de langage à pour objectif de capturer les contraintes du langage naturel afin de guider le décodage acoustique. Ces contraintes peuvent prendre différentes formes : grammaire à syntaxe fixe, modèle probabiliste.

III.4.2.1 Modèle de langage probabiliste

Les modèles de langage probabilistes ont pour objet d'attribuer une probabilité à une séquence de mots. De manière générale, la probabilité de la séquence de mots W de taille k est exprimée comme le produit des probabilités conditionnelles d'un mot sachant tous les mots précédents :

$$P(W) = P(w_1)\prod_{i=2}^{k} P(w_i/w_{i-1}, \dots, w_1) = P(w_1)\prod_{i=2}^{k} P(w_i/h_i) \qquad (3.35)$$

Où h_i est l'historique de longueur k du mot w_i : $h_i = w_{i-1}, \dots, w_1$

Le modèle de type n-grammes est le modèle probabiliste le plus généralement utilisé. Pour ce genre de modèle, l'historique d'un mot est représenté par les n − 1 mots qui le précèdent.

Dans la pratique, la valeur de n ne dépasse pas 3 : on parle alors de modèle trigrammes (uni-gramme pour n = 1, bi-gramme pour n = 2).

Soit $f(w_1 \dots w_n)$ la fréquence de la suite de mots $w_1 \dots w_n$ dans un corpus d'apprentissage, un modèle n-gramme estime la probabilité d'un mot w_i conditionné par son historique $h_i = w_1, \dots, w_{i-1}$ comme :

$$P(w_i/h_i) = \frac{f(h_i,w_i)}{f(h_i)} \qquad si \qquad f(h_i) > 0 \qquad (3.36)$$

III.4.3 Décodage de la parole continue

En reconnaissance de la parole continue, la suite de mots recherchée est celle qui maximise l'équation :

$$\widetilde{W} = arg\max_{M \in E_M} P(O/W)P(W) \qquad (3.37)$$

Pour résoudre cette équation, il n'est pas possible de construire un modèle pour chacune des phrases pouvant être prononcées puis de comparer tous ces modèles avec la phrase à identifier. L'alternative consiste à construire un modèle unique (un réseau de modèles) pouvant émettre toutes les phrases syntaxiquement correctes du langage. Le décodage de la phrase prononcée est déduit du meilleur chemin dans ce modèle obtenu par une variante de l'algorithme de Viterbi qui est l'algorithme du passage de jeton [6].

III.4.3.1 Réseau de modèles

La reconnaissance de la parole continue utilise un seul réseau de modèles correspondant aux mots du vocabulaire. Dans notre cas, le modèle composite (réseau de modèles) est constitué des modèles de mots mis en série ou en parallèle, avec un bouclage de l'état final des modèles vers l'état initial de n'importe quel autre modèle pour permettre l'enchaînement de plusieurs mots (Figure III.4). En pratique le réseau de modèles contient des états qui n'émettent pas d'observations, mais servent uniquement à simplifier la représentation des transitions d'un modèle à un autre (par exemple l'état initial et l'état final du réseau de la figure III.4).

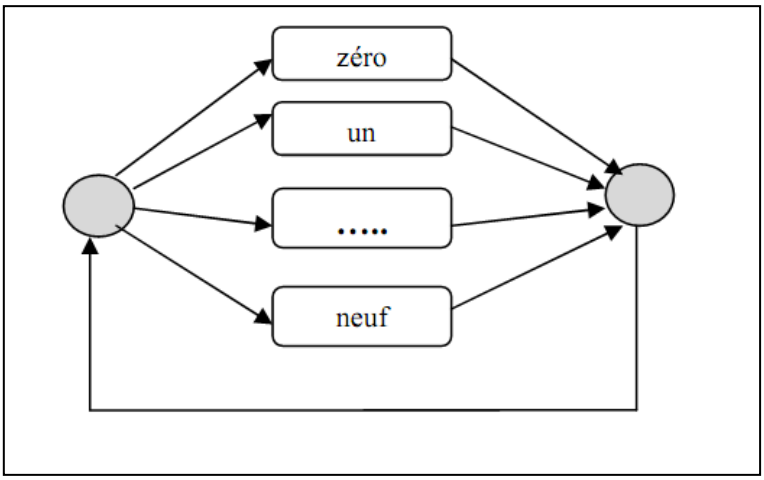

Figure III.4 : Exemple de réseau de modèles autorisant l'émission d'une suite quelconque de chiffres.

La suite d'états optimale trouvée par l'algorithme de Viterbi dans ce réseau fournit un décodage de la phrase en mots (ou en phonèmes). Cependant, l'algorithme recherche la meilleure suite d'états dans le réseau et non pas la meilleure suite de modèles.

III.4.3.2 Algorithme du passage de jeton

Des variantes de l'algorithme de Viterbi ont été proposées pour le décodage de la parole continue. L'algorithme du "Token Passing" ou "passage de jeton" est le plus utilisé [18]. Il s'applique sur un réseau de modèles sub-lexicaux.

Le jeton est un objet placé dans un état qui contient la probabilité du meilleur chemin arrivant dans cet état à l'instant courant, de plus, il conserve la mémoire de ses déplacements :

- ✓ Un jeton de valeur nulle est placé dans tous les états initiaux autorisés ;
- ✓ A chaque instant :
  - Le jeton de chaque état est propagé vers tous les états connectés et il est augmenté du coût de la transition et de l'émission :

$$\log(a_{ij}) + \log(b_j(o_t))$$

  - Dans chaque état, le meilleur jeton arrivant est conservé ;
- ✓ A la fin, le meilleur jeton parmi tous les états est récupéré.

Le décodage de la phrase en mots ou en phonèmes se déduit de la suite des modèles ayant émis les observations sur le chemin optimal.

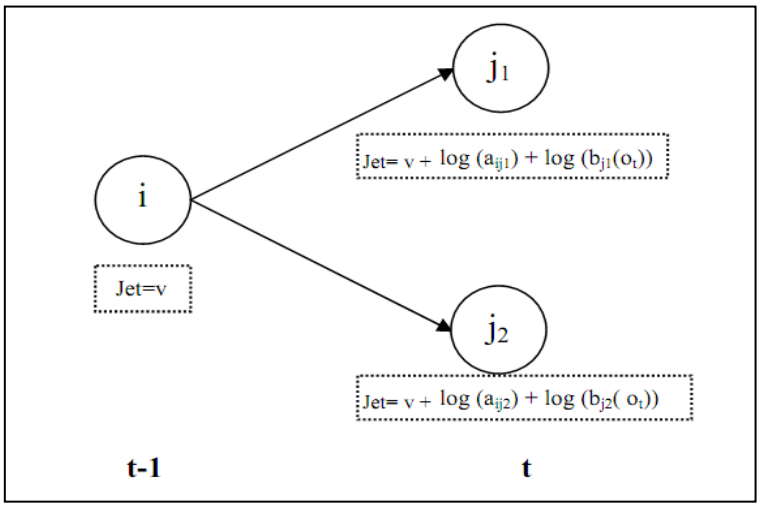

Figure III.5 : Algorithme de base du modèle de propagation de jeton.

III.5 Conclusion

Dans ce chapitre nous avons présenté le concept des modèles de Markov cachés (HMM), leur formalisme ainsi que leur emploi dans le domaine de la reconnaissance de la parole.

L'utilisation des HMM pour la reconnaissance de la parole pose en général trois problèmes à résoudre : l'évaluation de la probabilité d'émission des observations, l'apprentissage et le décodage ; ainsi, nous avons étudié les différents algorithmes utilisés pour pallier à ces problèmes pour les deux modes de reconnaissance : mots isolés et parole continue.

CHAPITRE IV

Evaluation expérimentale

Introduction

Ce chapitre présente le contexte expérimental et l'évaluation de notre travail. Notre objectif est de développer un système de reconnaissance fondé sur les modèles de Markov cachés. Notre système est évalué sur une base de données constituée de 18 mots.

Ce travail consiste donc à la réalisation de deux systèmes l'un pour la reconnaissance de mots isolés et l'autre pour la reconnaissance de la parole continue.

- Le système de reconnaissance de mots isolés est construit par deux outils : MATLAB et ensuite la plate-forme HTK avec une paramétrisation du signal par les coefficients cepstraux de type MFCC et leurs coefficients différentiels.
- Le système de reconnaissance de la parole continue construit sous la plate-forme HTK avec une paramétrisation du signal par les coefficients cepstraux MFCC ensuite par les coefficients de prédiction linéaire LPC.

IV.1 Présentation de la base des données

Dans le cadre de ce travail, on a constitué une base de données comportant dix-huit mots prononcés par 25 locuteurs différents (21 hommes et 5 femmes) pour l'apprentissage des modèles (450 séquences), et par 5 locuteurs pour le test (90 séquences).

Ces séquences audio ont été enregistrées via un microphone intégré d'un téléphone mobile SAMSUNG GT-B3410 dans des conditions réelles en format *.mp3*, ensuite converties en format *.wav* en utilisant le logiciel *Format-Factory* pour pouvoir les manipuler par la suite.

IV.2 Reconnaissance de mots isolés

Pour la reconnaissance de mots isolés, on a utilisé deux outils différents, MATLAB et la plate-forme HTK, pour ensuite évaluer les résultats obtenus et choisir le meilleur outil pour la reconnaissance de la parole continue, car les résultats de cette dernière dépendent étroitement de la reconnaissance de mots isolés et de l'opération de décodage de la parole continue.

IV.2.1 Dans le premier cas (avec Matlab)

IV.2.1.1 Description de l'application

Dans ce travail, on a utilisé la boite à outils de MATLAB « *HMMToolbox* » qui facilite la construction des modèles HMM.

Notre application est constituée de deux fonctions principales, *apprentissage.m* : chargée de la construction des modèles de mots (18 modèles donc 18 fonctions d'apprentissage) et *test.m* : pour la reconnaissance.

a. L'apprentissage

Pour chaque mot, on a développé une fonction d'apprentissage (*apprentissage0.m, apprentissage1.m* …).

Ces fonctions sont organisées comme suit :

- ✓ Chargement des fichiers sonores sur MATLAB avec la fonction *wavread.*
- ✓ Paramétrisation du signal de parole avec la fonction *mfcc*.
- ✓ Initialisation des paramètres du modèle.
- ✓ Ré-estimation des paramètres du modèle sur le critère du maximum de vraisemblance,
- ✓ Enregistrement du modèle dans un fichier *data.mat* qui contient les paramètres suivants :

  - Prior : qui représente la matrice π, (les probabilités initiales du modèle) ;
  - Transmat : qui représente la matrice A, (les probabilités de transition du modèle).
  - Mixmat et Sigma : qui représentent la matrice B, (les probabilités d'émission des observations)

b. Test

Pour tester notre système on commence par charger le fichier sonore à reconnaitre (format *.wav*), qui sera une entrée de la fonction *Test.m* avec tous les fichiers résultants de la phase d'apprentissage (*data0, data1,……data17*), cette fonction cherche le meilleur modèle qui correspond au mot à reconnaitre en utilisant l'algorithme du maximum de vraisemblance.

Les résultats de calcul des probabilités peuvent être affichés sous forme d'un graphe qui donne les valeurs du logarithme de la probabilité pour chaque modèle.

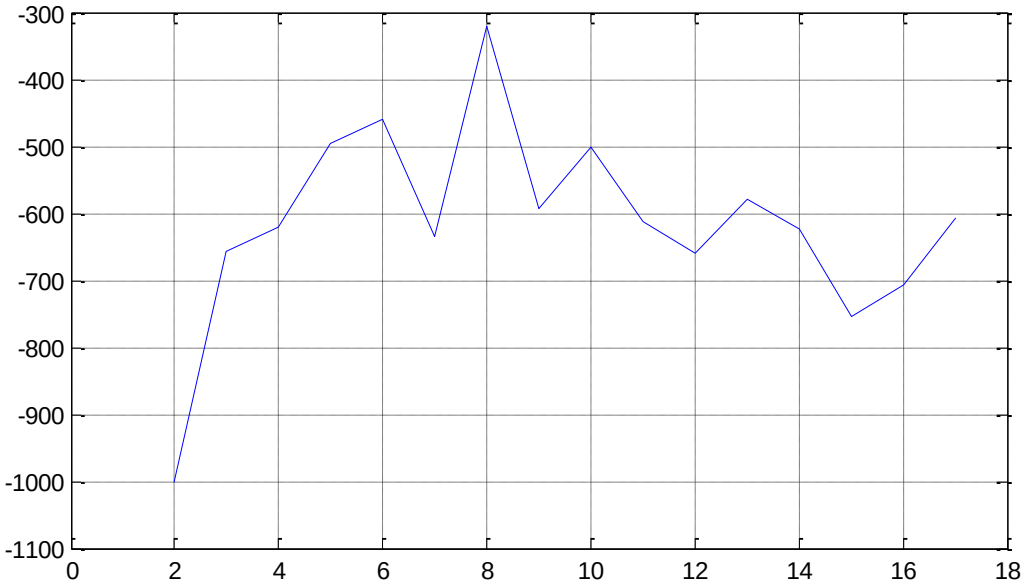

Figure IV.1 : graph représentatif des valeurs de logarithme des probabilités des modèles pour un mot de test « sept ».

Chaque chiffre sur l'axe des abscisses représente un mot (Tableau IV.1).

Indice	Mot	Indice	Mot
1	Zéro	10	Neuf
2	Un	11	Appeler
3	Deux	12	Hamroun
4	Trois	13	Joindre
5	Quatre	14	Le
6	Cinq	15	Monsieur
7	Six	16	Numéro
8	Sept	17	Hadjloum
9	Huit	18	contacter

Tableau IV.1 : Indices des mots de la base de données.

IV.2.1.2 Description de l'interface

Pour mieux présenter notre travail et rendre facile l'utilisation du système de reconnaissance de mots isolés, nous avons réalisé une interface graphique sous MATLAB 7.8. Comme le montre la figure IV.2, cette interface comprend principalement deux volets :

- un volet pour enregistré un mot en temps réel ou apporter un fichier sonore enregistré antérieurement, pour lire le son à reconnaitre, pour démarrer le programme de reconnaissance et pour quitter l'application ;
- un deuxième volet, pour afficher l'audiogramme du mot inconnu ainsi que le résultat du test.

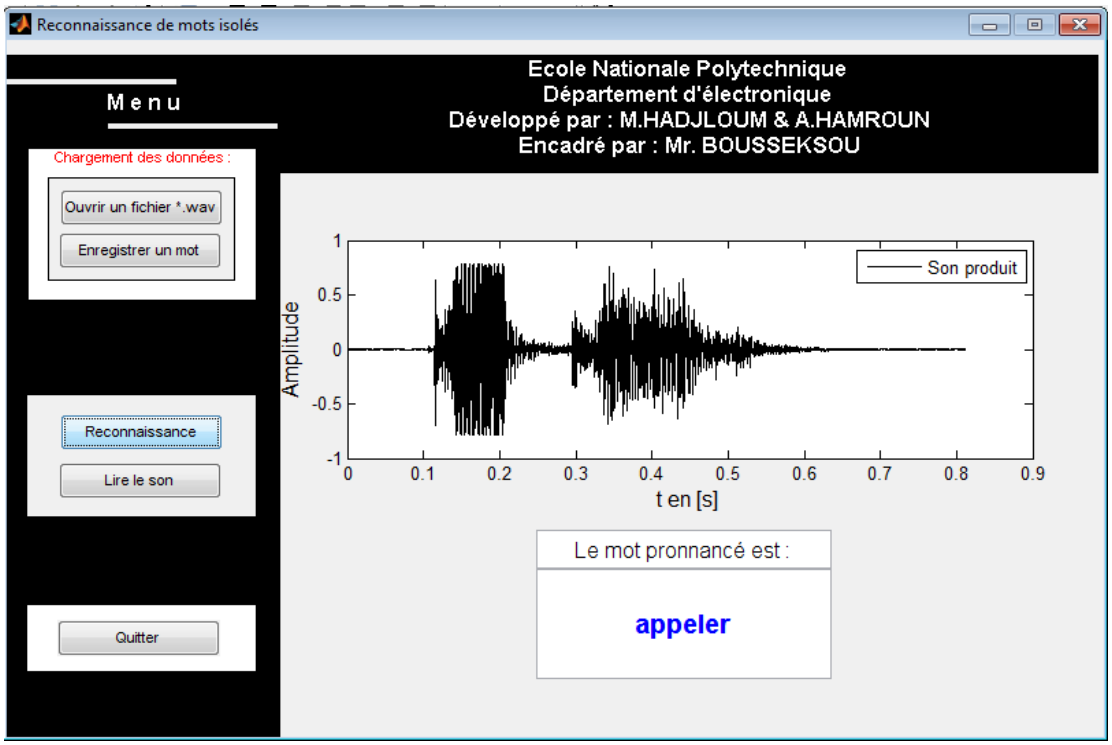

Figure IV.2 : Interface graphique du système de reconnaissance de mots isolés.

IV.2.1.3 Taux de reconnaissance

Nous présentons les taux de reconnaissance obtenu pour cinq séquences de test pour les différents mots du vocabulaire utilisé :

Mot à reconnaitre	Nombre de mots reconnus	Taux de reconnaissance(%)
Zéro	1/5	20%
Un	1/5	20%
Deux	5/5	100%
Trois	4/5	80%
Quatre	5/5	100%
Cinq	5/5	100%
Six	5/5	100%
Sept	5/5	100%
Huit	5/5	100%
Neuf	4/5	80%
Appeler	4/5	80%

Contacter	1/5	20%
Joindre	4/5	80%
Le	5/5	100%
Numéro	*4/5*	*80%*
Monsieur	*5/5*	*100%*
Hadjloum	*5/5*	*100%*
Hamroun	*1/5*	*20%*
Taux de reconnaissance total : 76,67%		

Tableau IV.2 : Taux de reconnaissance obtenu avec MATLAB.

IV.2.2 Dans le second cas (avec HTK)

IV.2.2.1 Plate-forme HTK

La plate-forme HTK (*Hidden Markov Model Toolkit*, ou « boite à outils de modèles de Markov cachés ») a été développée à l'Université de Cambridge par S.J.Young et son équipe. Elle est constituée d'un ensemble d'outils logiciels qui permettent de construire des systèmes de reconnaissance de la parole continue à base de modèles de Markov cachés [18]. HTK est remarquable par la très grande liberté de choix laissée 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 des cas. Les résultats sont enfin évalués par alignement dynamique avec la chaîne phonétique ou lexicale de référence.

L'ensemble de ces outils est écrit en langage C, et la documentation détaille leur utilisation et les principes de leur implémentation, ce qui rend l'outil HTK largement répandu dans le monde de la recherche.

IV.2.2.2 Présentation d'HTK

Notre système de reconnaissance à base de HTK est structuré comme suit :

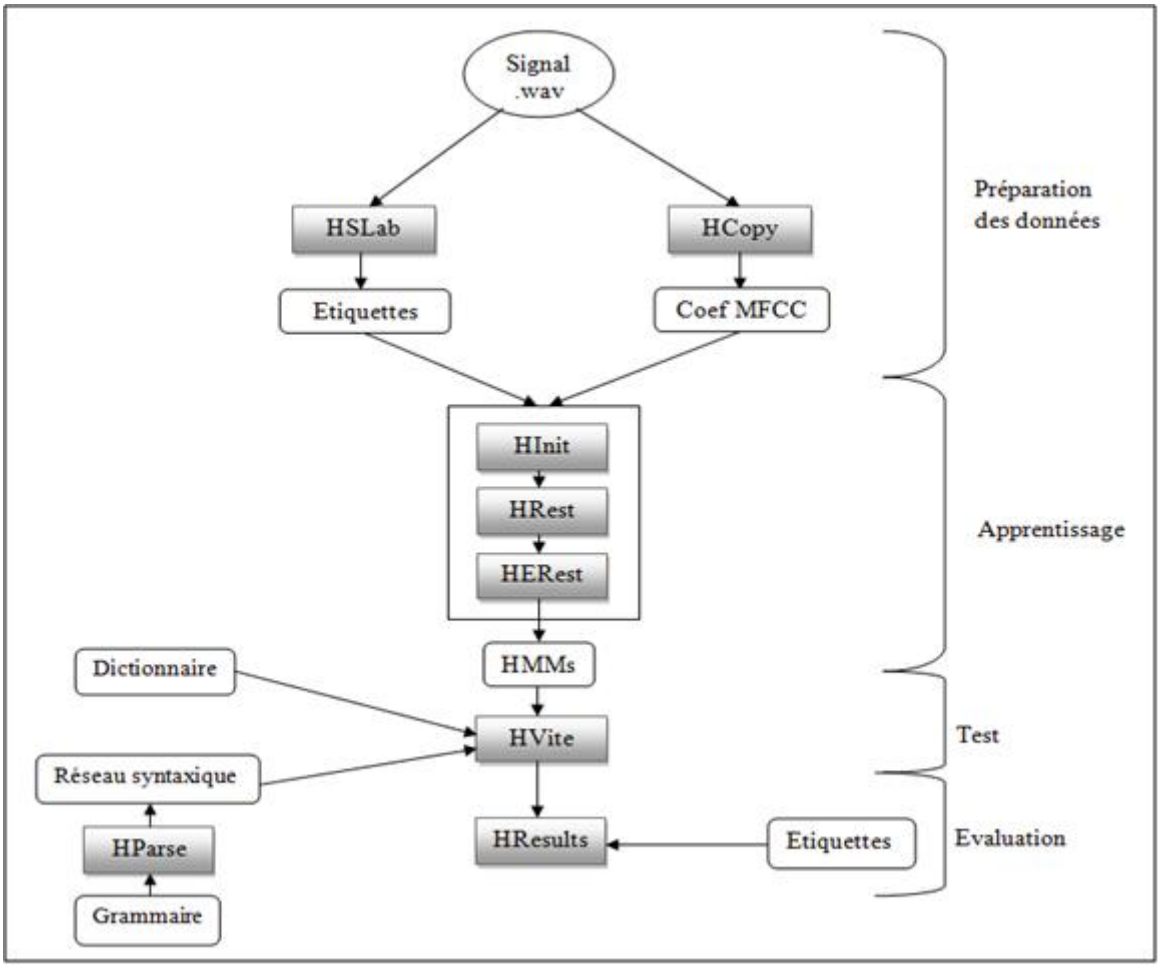

Figure IV.3 : structure de notre système de reconnaissance avec HTK.

- Signal : Enregistrements.
- Dictionnaire : Il contient les transcriptions des mots utilisés.
- Étiquettes: représente les transcriptions des fichiers sons.
- Grammaire : définit le modèle de langage.
- Réseau syntaxique : il contient toutes les séquences possibles constituées à partir des mots du dictionnaire suivant les règles syntaxiques définit dans Grammaire.
- Ceof MFCC : Les coefficients MFCC issus de la paramétrisation.

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 données dans le tableau suivant :

Outils	Rôle
HSLab	Affichage du signal et des étiquettes.
HCopy	Calcul des paramètres des fichiers sons.
HInit	Initialisation des modèles HMM.
HRest	Ré-estimation des HMMs (Baum-Welch).
HERest	Ré-estimation des HMMs en continu (Baum-Welch).
HParse	Génération du réseau syntaxique.
HVite	Décodage de la parole continue (Viterbi).
HResults	Résultats du décodage (alignement dynamique entre les fichiers de résultats et de références).

Tableau IV.3 : Outils de base de HTK.

IV.2.2.3 Utilisation de HTK.

On va détailler maintenant les étapes d'utilisation de HTK comme elles sont indiquées dans la figure IV.3.

IV.2.2.3.1 Préparation des données

Avant l'apprentissage des modèles, il est nécessaire de préparer les données d'apprentissage en calculant les paramètres du signal et en étiquetant les mots d'apprentissage :

a. Paramétrisation du signal

La représentation du signal est obtenue avec l'outil **HCopy**, qui calcul les vecteurs acoustiques du signal (MFCC, LPC, …). Ainsi que leurs coefficients différentiels du premier et du second ordre.

Dans notre système on a utilisé 12 coefficients MFCC + l'énergie + leurs dérivées premières et secondes, ce qui donne un total de 39 coefficients (12 MFCC + E + 12 ΔMFCC + ΔE + 12ΔΔMFCC + ΔΔE = 39).

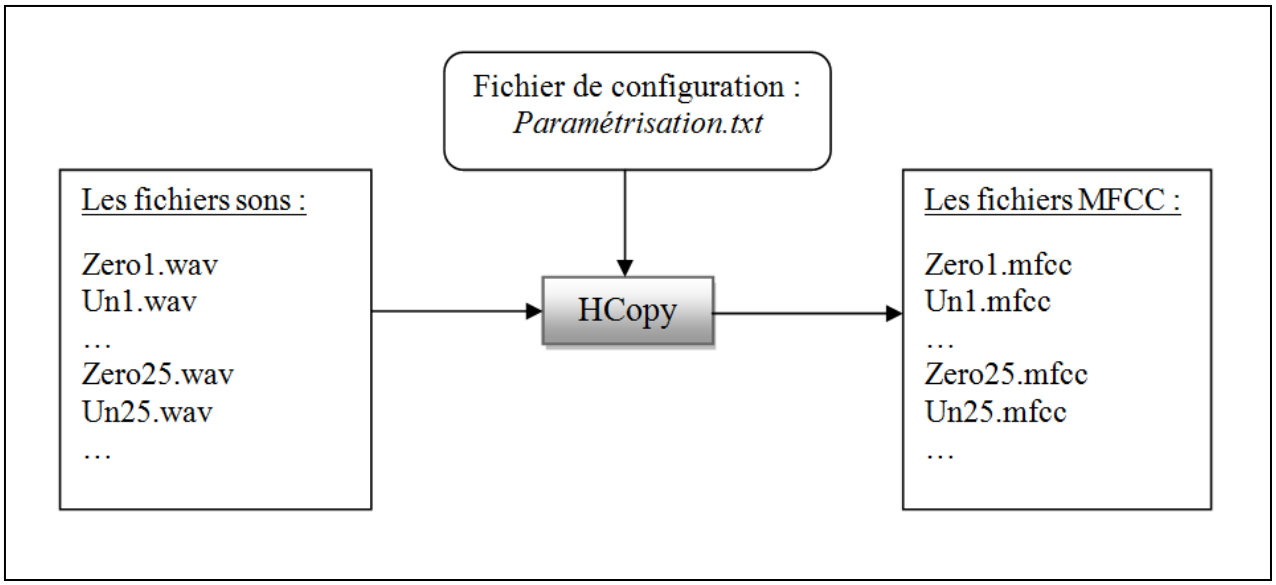

Figure IV.4 : représentation acoustique du signal.

Paramétrisation.txt : contient les paramètres de calcul des coefficients cepstraux MFCC.

b. L'étiquetage de la base d'apprentissage

Le but de cette étape est de délimiter chaque entité lexicale et de lui attribuer un symbole, ceci sera fait manuellement avec l'outil **HSLab**.

Dans notre cas, la base de données est constituée d'un ensemble de mots, l'étiquetage reviens à donner le mot lexical correspondant à chaque signal de la base.

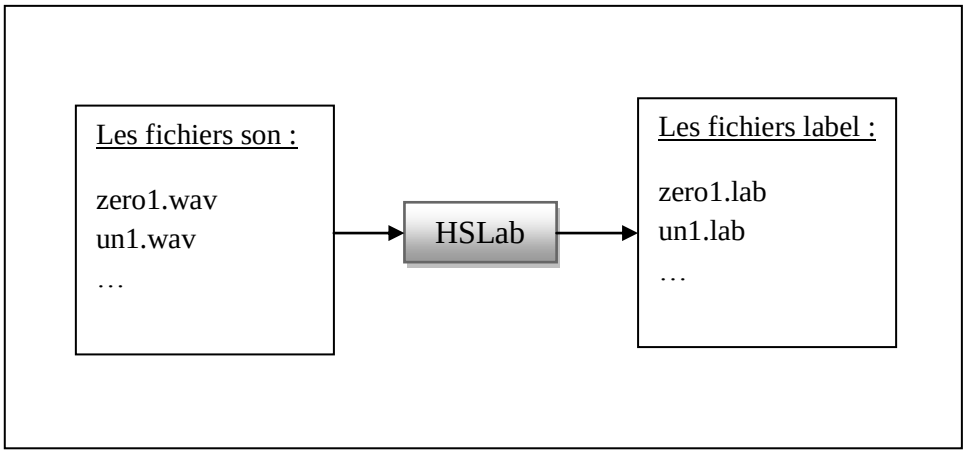

Figure IV.5 : étiquetage du signal acoustique.

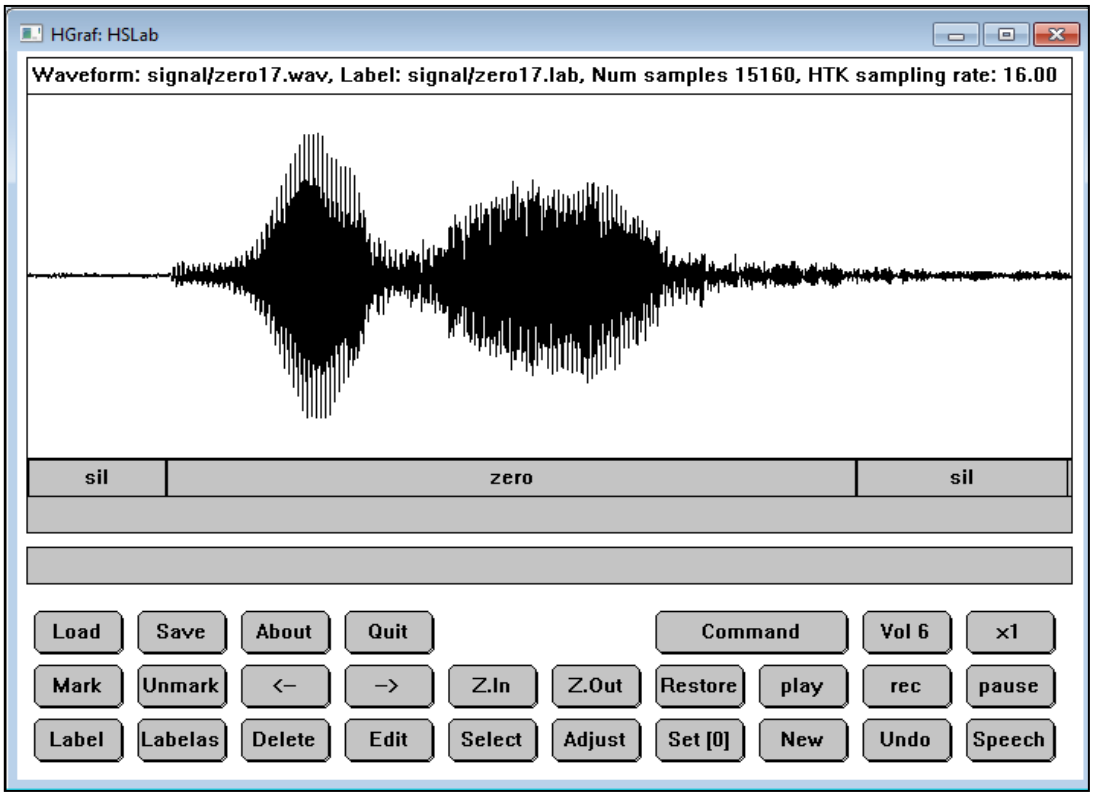

Figure IV.6 : l'étiquetage d'un fichier son « zero.wav ».

c. Topologie des modèles

Pour chaque unité acoustique, il faut définir un modèle prototype contenant la topologie choisie, à savoir :

- Le nombre d'états du modèle.
- Les transitions possibles entre les états.
- Le type de loi de probabilité associée à chaque état.

L'état initial et l'état final ont la particularité de ne pas émettre d'observation, mais de servir uniquement à la connexion des modèles en parole continue.

Les probabilités d'émission associées aux états sont décrites par une combinaison linéaire de gaussiennes, caractérisées par leur moyenne et leur matrice de covariance dans l'espace des paramètres. La matrice de covariance est théoriquement symétrique, mais peut être choisie diagonale si l'on suppose l'indépendance entre les composantes des vecteurs de paramètres.

IV.2.2.3.2 Apprentissage

L'apprentissage des modèles de Markov est une étape essentielle dans la construction d'un système de la RAP.

L'apprentissage consiste à estimer les paramètres des modèles de Markov : les probabilités de transition, les densités d'observation associées aux états, c'est à dire les vecteurs de moyennes et les matrices de covariances d'un ensemble de gaussiennes, ainsi que les pondérations permettant d'établir des mélanges à partir de ces gaussiennes.

L'apprentissage des modèles HMM nécessite trois étapes décrites comme suit :

a. Créer le prototype initial

L'initialisation de ce prototype est indépendante de tout corpus d'entraînement, elle sert juste à définir la topologie du prototype initial et non à l'initialisation de ses paramètres [18]. Nous avons opté pour les paramètres initiaux suivant :

- Initialisation des vecteurs moyennes μ_k par des zéros. Ceci en supposant que les observations acoustiques sont des variables aléatoires centrées réduites.
- Initialisation des matrices de covariance Σ_k par des matrices diagonales unitaires. Ceci en supposant la décorrélation entre tous les paramètres MFCC d'une même observation.
- Initialisation équiprobable des poids d'une loi de probabilité gaussienne :

$$c_k = 1/nombre\ de\ gaussiennes.$$

b. Initialisation de l'apprentissage

Pour chacun des prototypes initiaux (modélisant un mot), l'outil **HInit** initialise les probabilités d'émission des états du modèle au moyen d'une procédure itérative basée sur l'algorithme de Viterbi (figure IV.7). Cette phase aide à répartir d'une façon optimale les trames d'un vecteur acoustique sur l'ensemble des états du modèle correspondant.

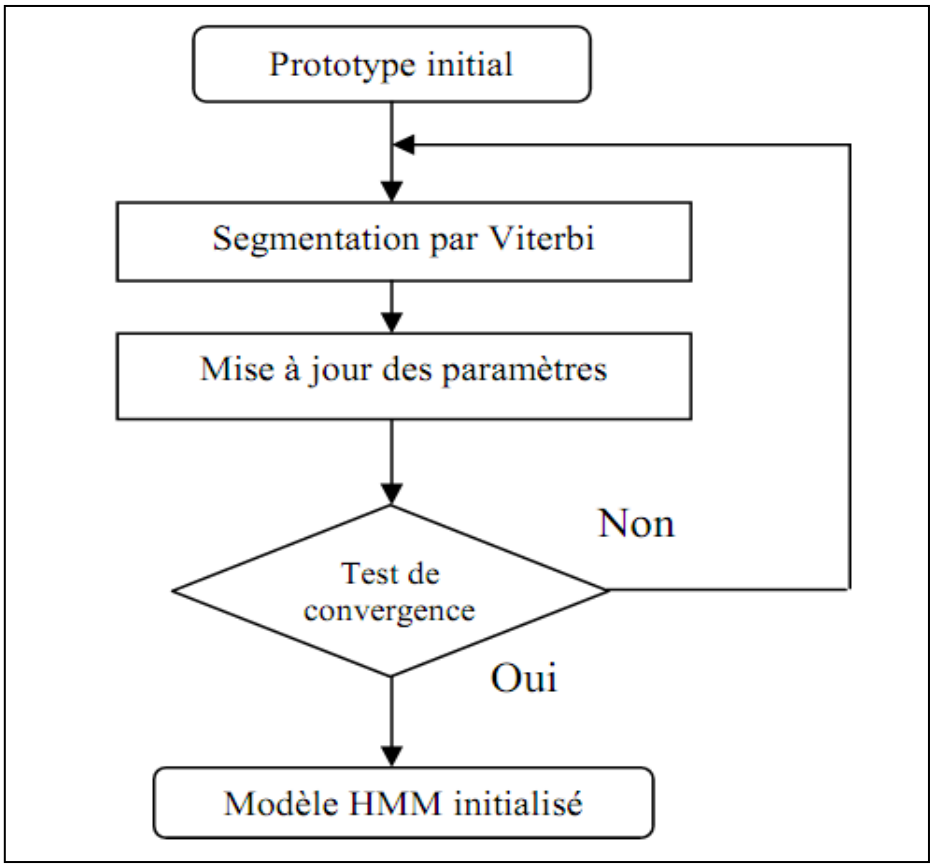

Figure IV.7 : Initialisation d'un modèle HMM avec Viterbi.

c. Estimation des paramètres de l'apprentissage

L'estimation des paramètres d'un modèle est effectuée avec l'outil **HRest**, qui applique l'algorithme optimal de Baum-Welch jusqu'à la convergence et ré-estime les probabilités d'émission et de transition.

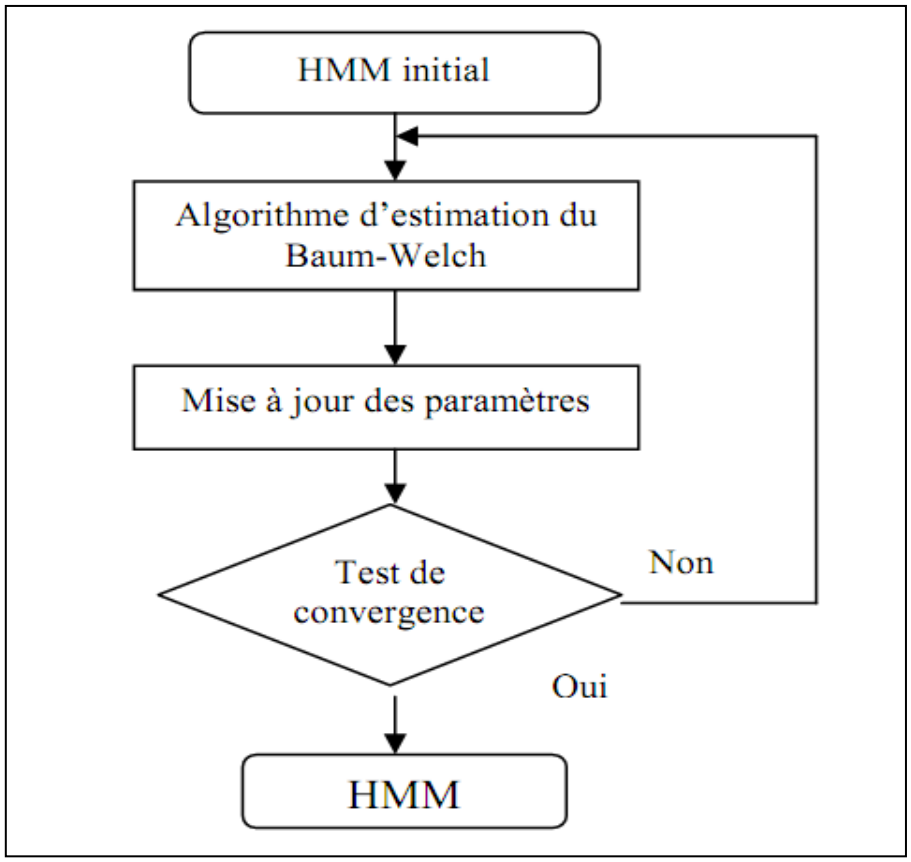

Figure IV.8 : Estimation des paramètres d'un modèle HMM avec l'algorithme de Baum-Welch.

L'utilisation de ces deux outils d'apprentissage peut être résumée par le schéma suivant :

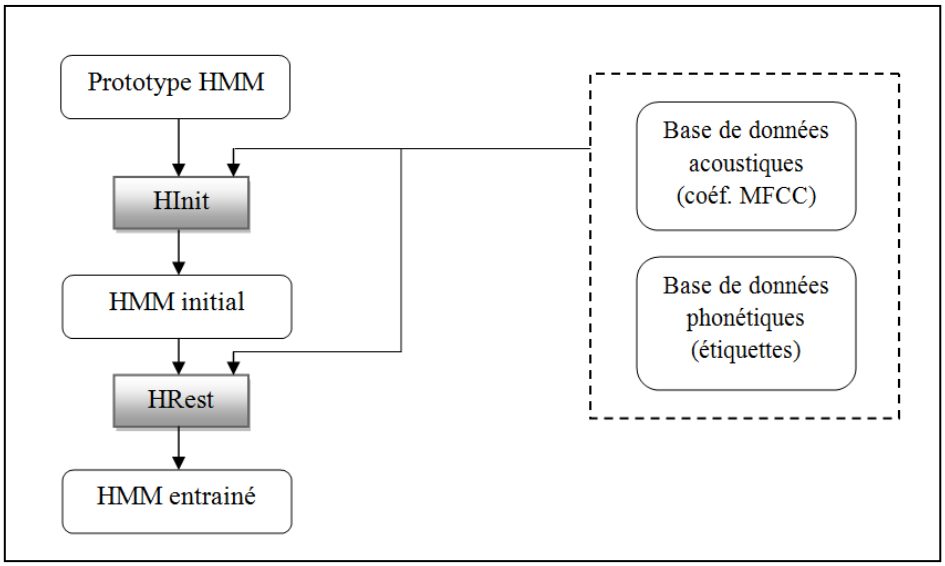

Figure IV.9 : Apprentissage des HMM avec HTK.

IV.2.2.3.3 Reconnaissance

Le module de décodage de la parole continue, **HVite**, utilise l'algorithme de Viterbi pour trouver la séquence d'états la plus probable correspondant aux paramètres observés dans un modèle composite, et en déduire les unités acoustiques correspondantes. Le modèle composite autorise la succession des modèles acoustiques en fonction d'un réseau syntaxe choisi par le concepteur du système.

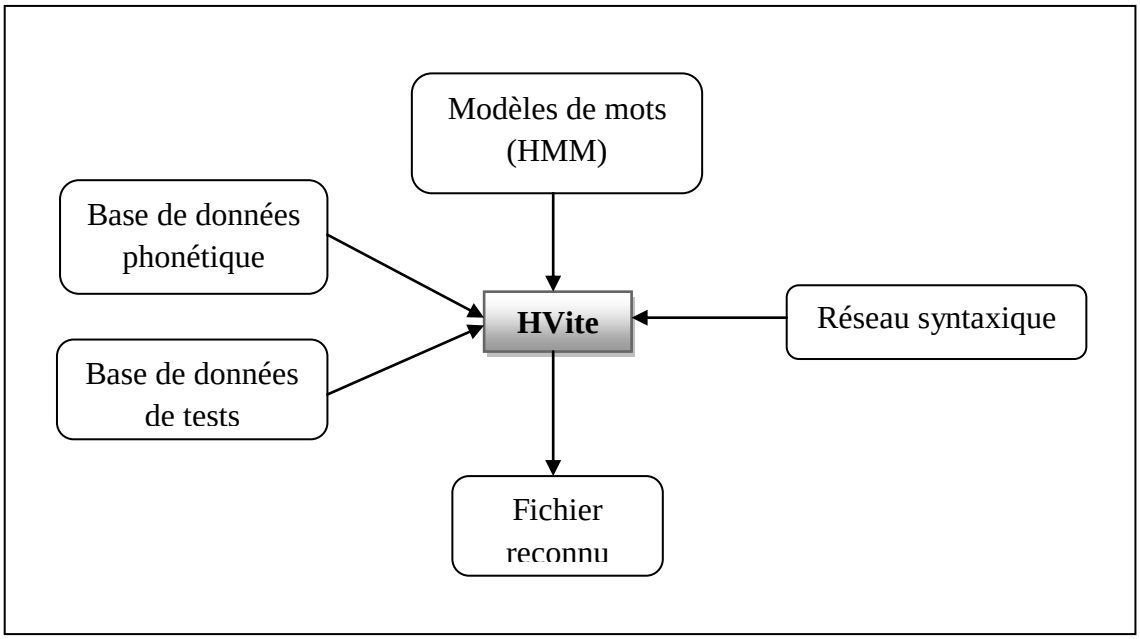

Figure IV.10 : La reconnaissance.

IV.2.2.3.4 Evaluation des résultats

Le résultat du décodage est comparé aux étiquettes de référence par un alignement dynamique réalisé par l'outil **HResults**, afin de compter les étiquettes identifiées, omises, substituées par une autre, et insérées, et de calculer le taux de reconnaissance.

Les performances des systèmes de RAP sont évaluées par le pourcentage de reconnaissance de mots (**%Corr**) et le pourcentage de reconnaissance global (**%Acc**).

Le pourcentage de reconnaissance des mots correspond à l'équation suivante :

$$\%corr = \frac{N - O - S}{N} \times 100$$

Le pourcentage global de reconnaissance en tenant compte des insertions, c'est celui qui indique les performances réelles du système :

$$\%Acc = \frac{N-O-S-I}{N} \times 100$$

Avec :

N : Le nombre total d'unités.

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

S : Le nombre de substitutions (le nombre d'unités pour lesquelles le système à commis une erreur).

I : Le nombre d'insertions (le nombre d'unités admises comme reconnues alors qu'aucun mot n'a été prononcé).

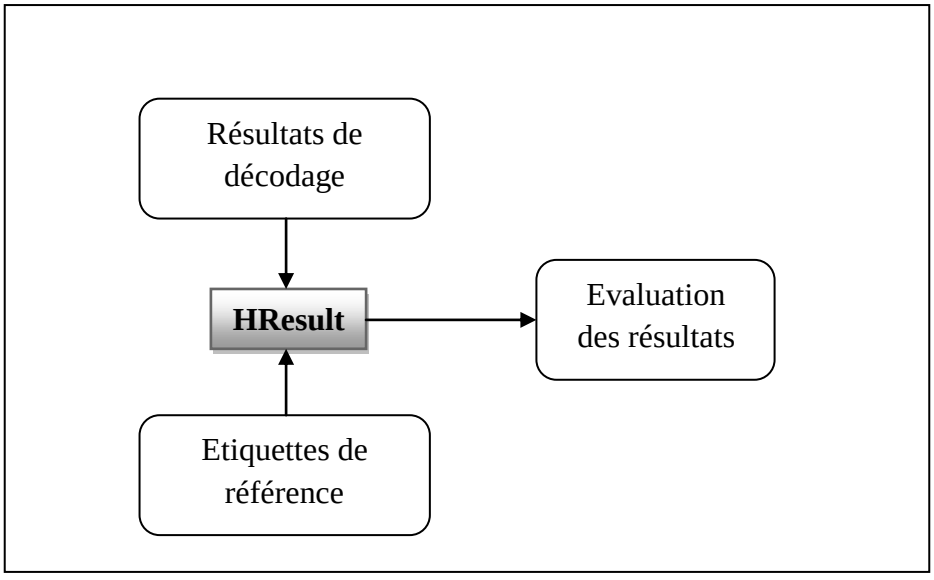

Figure IV.11 : Evaluation des résultats.

En exécutant la commande **HResult** sur chacun des mots de test on a eu les résultats suivants :

Mots à reconnaitre	Taux de reconnaissance(%)
Zéro	100%
Un	93.33%
Deux	100%
Trois	100%
Quatre	73.33%
Cinq	93.33%
Six	100%
Sept	93.33%
Huit	100%
Neuf	100%
Appeler	80.00%
Contacter	100%
Joindre	100%
Le	93.33%
Numéro	100%
Monsieur	100%
Hadjloum	100%
Hamroun	86.67%
Taux de reconnaissance total : 95.13	

Tableau IV .4 : Taux de reconnaissance obtenu avec HTK.

IV.2.3 Interprétation des résultats

En comparant les résultats trouvés par les deux outils, on constate que le HTK donne de meilleurs résultats relativement à MATLAB, alors qu'on a utilisé les mêmes algorithmes dans les deux cas. Cela est dû aux fonctions utilisées dans chacun des outils, Le HTK est implémenté avec le langage C, où tout est programmé par des experts de domaine de reconnaissance, alors que dans MATLAB on a utilisé plusieurs fonctions prédéfinies qui sont mal implémentées pour une application de RAP, surtout les fonctions associées au calcul des paramètres MFCC, où tout est presque prédéfini.

En partant des résultats obtenus pour la reconnaissance de mots isolés, on a décidé de continuer la partie suivante qui est la reconnaissance de la parole continue, avec l'outil qui nous a donné les meilleurs résultats, qui est le HTK.

IV.3 Reconnaissance de la parole continue

Pour la reconnaissance de la parole continue, en plus des étapes de la reconnaissance de mots isolés, on doit procéder à un décodage acoustico-phonétique qui transforme le signal acoustique continu en une description linguistique discrète sous forme d'unités tel que le mot.

On doit aussi définir un modèle de langage. Dans notre cas, on à utilisé un modèle de langage de type grammaire syntaxique qui impose des règles à suivre dans la construction des phrases, à partir des mots du dictionnaire.

La figure ci-dessous illustre la topologie du modèle de langage utilisé dans notre système de reconnaissance de la parole continue.

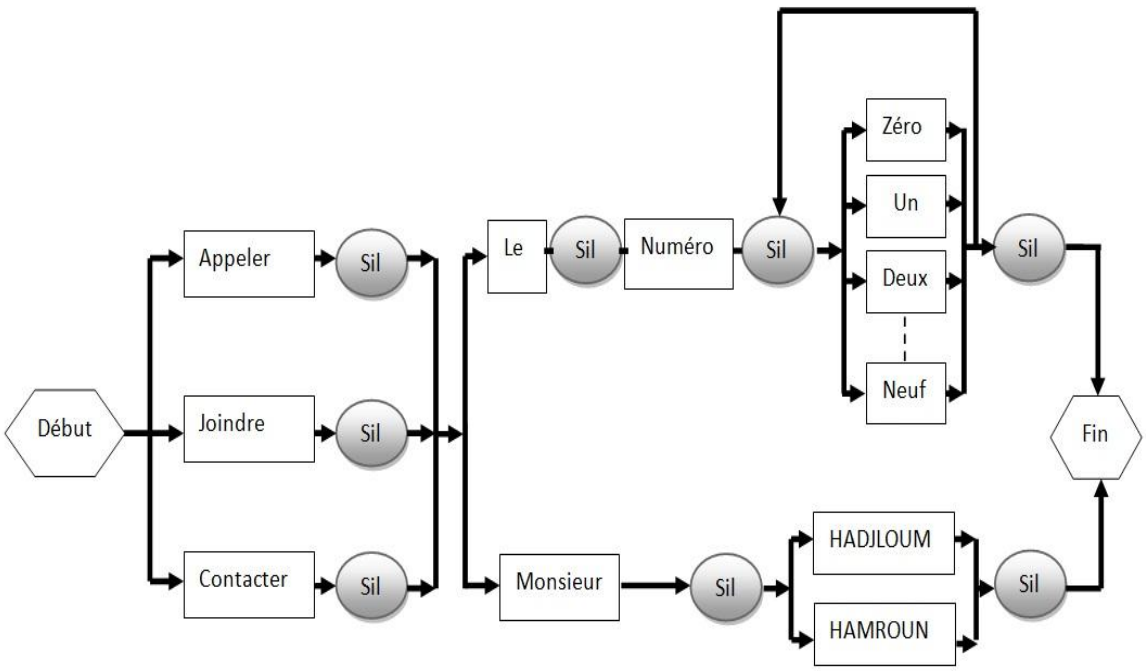

Figure IV.12 : Le modèle de langage.

Nos tests ont été effectués sur des phrases qui sont constituées des mots du vocabulaire utilisé (18 mots).

IV.3.1 Evaluation des résultats

Pour évaluer la fiabilité de notre système de reconnaissance de parole continue nous avons effectué 8 tests pour les deux méthodes d'analyse MFCC et LPC.

Les phrases de test :

1. *« Joindre monsieur Hamroun »*
2. *« contacter le numéro 0553552652 »*
3. *« appeler le numéro 0798494283 »*
4. *« contacter monsieur Hadjloum »*
5. *« joindre le numéro 0661593539 »*
6. *« joindre monsieur Hadjloum »*
7. *« appeler le numéro 0553552652 »*
8. *« contacter monsieur Hamroun »*

IV.3.1.1 Pour les coefficients MFCC

Les résultats de test obtenus avec la commande **HResults** sont représentés sur la figure suivante :

```
HVite -T 1 -A -w listes/modeledemots.txt -d hrest/ -l resultat/hinit/ -
S listes/mfcc.ar.lst listes/dictionnaire.txt listes/listemodeles.txt
File: mfcc/1.mfcc
SENT-START sil joindre sil monsieur sil hadjloum sil  ==  [2280 frames]
-40.7006 [Ac=-92797.4 LM=0.0] (Act=272.6)
File: mfcc/2.mfcc
SENT-START sil appeler sil le sil numero sil zero sil cinq sil cinq sil
trois sil cinq sil cinq sil deux sil six sil cinq sil deux sil SENT-END
==  [15783 frames] -35.4878 [Ac=-560103.9 LM=0.0] (Act=274.6)
File: mfcc/3.mfcc
SENT-START sil contacter sil le sil numero sil zero sil sept sil neuf
sil huit sil quatre sil neuf sil quatre sil deux sil huit sil trois sil
SENT-END  ==  [13205 frames] -36.5572 [Ac=-482738.5 LM=0.0] (Act=274.6)
File: mfcc/4.mfcc
SENT-START sil contacter monsieur hadjloum sil SENT-END  ==  [1690
frames] -44.9966 [Ac=-76044.2 LM=0.0] (Act=271.7)
File: mfcc/5.mfcc
SENT-START sil joindre le sil numero quatre zero sil six sil six sil
quatre sil cinq sil neuf sil trois sil cinq sil quatre sil SENT-END  ==
[9266 frames] -41.2112 [Ac=-381862.8 LM=0.0] (Act=274.4)
File: mfcc/6.mfcc
SENT-START sil joindre sil monsieur sil hadjloum sil SENT-END ==  [2870
frames] -39.0822 [Ac=-112166.0 LM=0.0] (Act=273.1)
File: mfcc/7.mfcc
SENT-START sil joindre sil le sil numero sil zero sil cinq sil cinq sil
trois sil cinq sil cinq sil deux sil six sil cinq sil deux sil SENT-END
==  [15783 frames] -38.3684 [Ac=-605568.0 LM=0.0] (Act=274.6)
File: mfcc/8.mfcc
SENT-START sil contacter le sil numero six six deux huit sil neuf
quatre cinq deux quatre deux sil SENT-END  ==  [3241 frames] -39.5942
[Ac=-128324.9 LM=0.0] (Act=273.3)
```

Figure IV.13 : Résultats des phrases de test.

Le taux de reconnaissance obtenu par phrase est représenté ci-dessous :

```
====================== HTK Results Analysis =======================
  Date: Sun Jun 03 11:47:05 2012
  Ref : labels/
  Rec : resultat/hrest/1.rec
------------------------ Overall Results --------------------------
WORD: %Corr=85.71, Acc=71.43 [H=6, D=0, S=1, I=1, N=7]
===================================================================
====================== HTK Results Analysis =======================
  Date: Sun Jun 03 11:47:05 2012
  Ref : labels/
  Rec : resultat/hrest/2.rec
------------------------ Overall Results --------------------------
WORD: %Corr=100.00, Acc=96.30 [H=27, D=0, S=0, I=1, N=27]
===================================================================
====================== HTK Results Analysis =======================
  Date: Sun Jun 03 11:47:05 2012
  Ref : labels/
  Rec : resultat/hrest/3.rec
------------------------ Overall Results --------------------------
WORD: %Corr=96.30, Acc=92.59 [H=26, D=0, S=1, I=1, N=27]
===================================================================
====================== HTK Results Analysis =======================
  Date: Sun Jun 03 11:47:05 2012
  Ref : labels/testH
  Rec : resultat/hrest/4.rec
------------------------ Overall Results --------------------------
WORD: %Corr=100.00, Acc=80.00 [H=5, D=0, S=0, I=1, N=5]
===================================================================

====================== HTK Results Analysis =======================
  Date: Sun Jun 03 11:47:05 2012
  Ref : labels/testH
  Rec : resultat/hrest/5.rec
------------------------ Overall Results --------------------------
WORD: %Corr=85.71, Acc=85.71 [H=18, D=0, S=3, I=0, N=21]
===================================================================
====================== HTK Results Analysis =======================
  Date: Sun Jun 03 11:47:05 2012
  Ref : labels/
  Rec : resultat/hrest/6.rec
------------------------ Overall Results --------------------------
WORD: %Corr=100.00, Acc=85.71 [H=7, D=0, S=0, I=1, N=7]
===================================================================
====================== HTK Results Analysis =======================
  Date: Sun Jun 03 11:47:05 2012
  Ref : labels/
  Rec : resultat/hrest/7.rec
------------------------ Overall Results --------------------------
WORD: %Corr=100.00, Acc=96.30 [H=27, D=0, S=0, I=1, N=27]
===================================================================
====================== HTK Results Analysis =======================
  Date: Sun Jun 03 11:47:05 2012
  Ref : labels/
  Rec : resultat/hrest/8.rec
------------------------ Overall Results --------------------------
WORD: %Corr=71.43, Acc=-85.71 [H=5, D=0, S=2, I=11, N=7]
===================================================================
```

Figure IV.14 : Taux de reconnaissance pour les phrases avec les coefficients MFCC.

%Corr : représente le taux de reconnaissance des mots dans la phrase.
%Acc : représente le taux de reconnaissance de la phrase.

Le taux de reconnaissance total pour toutes les phrases de test est de 72.66 % et pour les mots 93.75 % :

```
====================== HTK Results Analysis =======================
  Date: Sun Jun 03 13:18:07 2012
  Ref : labels/
  Rec : resultat/hrest/1.rec
      : resultat/hrest/2.rec
      : resultat/hrest/3.rec
      : resultat/hrest/4.rec
      : resultat/hrest/5.rec
       : resultat/hrest/6.rec
       : resultat/hrest/7.rec
       : resultat/hrest/8.rec
---------------------- Overall Results --------------------------
WORD: %Corr=93.75, Acc=72.66 [H=120, D=0, S=8, I=27, N=128]
===================================================================
```

Figure IV.15 : Taux de reconnaissance total avec les coefficients MFCC.

IV.3.1.2 Pour les coefficients LPC

Le taux de reconnaissance obtenu par phrase est représenté ci-dessous :

```
====================== HTK Results Analysis =======================
  Date: Sun Jun 03 16:54:19 2012
  Ref : labels/
  Rec : resultat/hrest/1.rec
---------------------- Overall Results --------------------------
WORD: %Corr=71.43, Acc=-142.86 [H=5, D=0, S=2, I=15, N=7]
===================================================================
====================== HTK Results Analysis =======================
  Date: Sun Jun 03 16:54:19 2012
  Ref : labels/
  Rec : resultat/hrest/2.rec
---------------------- Overall Results --------------------------
WORD: %Corr=66.67, Acc=62.96 [H=18, D=0, S=9, I=1, N=27]
===================================================================
====================== HTK Results Analysis =======================
  Date: Sun Jun 03 16:54:19 2012
  Ref : labels/
  Rec : resultat/hrest/3.rec
---------------------- Overall Results --------------------------
WORD: %Corr=77.78, Acc=74.07 [H=21, D=0, S=6, I=1, N=27]
===================================================================
```

```
====================== HTK Results Analysis =======================
  Date: Sun Jun 03 16:54:19 2012
  Ref : labels/testH
  Rec : resultat/hrest/4.rec
------------------------ Overall Results --------------------------
WORD: %Corr=100.00, Acc=80.00 [H=5, D=0, S=0, I=1, N=5]
===================================================================
====================== HTK Results Analysis =======================
  Date: Sun Jun 03 16:54:19 2012
  Ref : labels/testH
  Rec : resultat/hrest/5.rec
------------------------ Overall Results --------------------------
WORD: %Corr=66.67, Acc=47.62 [H=14, D=0, S=7, I=4, N=21]
===================================================================
====================== HTK Results Analysis =======================
  Date: Sun Jun 03 16:54:19 2012
  Ref : labels/
  Rec : resultat/hrest/6.rec
------------------------ Overall Results --------------------------
WORD: %Corr=71.43, Acc=-114.29 [H=5, D=0, S=2, I=13, N=7]
===================================================================
====================== HTK Results Analysis =======================
  Date: Sun Jun 03 16:54:19 2012
  Ref : labels/
  Rec : resultat/hrest/7.rec
------------------------ Overall Results --------------------------
WORD: %Corr=51.85, Acc=51.85 [H=14, D=2, S=11, I=0, N=27]
===================================================================
====================== HTK Results Analysis =======================
  Date: Sun Jun 03 16:54:20 2012
  Ref : labels/
  Rec : resultat/hrest/8.rec
------------------------ Overall Results --------------------------
WORD: %Corr=71.43, Acc=-142.86 [H=5, D=0, S=2, I=15, N=7]
===================================================================
```

Figure IV.16 : Taux de reconnaissance obtenus pour les phrases avec les coefficients LPC.

Le taux de reconnaissance total pour toutes les phrases de test est de 28.91 % et pour les mots 67.97 % :

```
====================== HTK Results Analysis =======================
  Date: Sun Jun 03 17:04:05 2012
  Ref : labels/
  Rec : resultat/hrest/1.rec
      : resultat/hrest/2.rec
      : resultat/hrest/3.rec
      : resultat/hrest/4.rec
      : resultat/hrest/5.rec
       : resultat/hrest/6.rec
       : resultat/hrest/7.rec
       : resultat/hrest/8.rec

------------------------ Overall Results --------------------------
WORD: %Corr=67.97, Acc=28.91 [H=87, D=2, S=39, I=50, N=128]
===================================================================
```

Figure IV.17 : Taux de reconnaissance total avec les coefficients LPC.

IV.3.2 Interprétation des résultats

Les résultats obtenus avec les coefficients MFCC sont largement meilleurs que ceux obtenus avec les LPC, cela est dû à plusieurs facteurs :

- Les MFCC sont des coefficients robustes moins sensibles au bruit relativement aux LPC.
- Les MFCC sont issus d'un banc de filtres répartis sur une échelle perceptuelle ce qui simule bien le fonctionnement du système auditif humain, donc permet d'avoir une meilleur résolution dans les basses fréquences, zone qui contient le plus d'information dans le signal de parole.
- Les méthodes perceptuelles permettent d'avoir un nombre de coefficients réduit et peu corrélés entres eux.

IV.4 Conclusion

Notre travail expérimental est constitué de deux grandes parties : Reconnaissance de mots isolés et reconnaissance de la parole continue.

- Reconnaissance de mot isolé :
  Cette partie a été réalisée sous deux outils différents, MATLAB et HTK, et les résultats obtenus pour un corpus constitué de 18 mots sont 76,67% et 95.13 % sur MATLAB et HTK respectivement.
- Reconnaissance de la parole continue :
  Suite aux résultats obtenus pour les mots isolés, nous avons décidé de faire cette partie avec HTK, en utilisant deux types de coefficients de paramétrisation, MFCC et LPC, et les résultats obtenus pour 8 phrases de testes constituées par des mots du corpus d'apprentissage sont beaucoup mieux avec les coefficients MFCC (72.66%) que avec les LPC (28.91%).

Conclusion et perspectives

Dans ce travail, le but était de construire un système de reconnaissance de la parole continue robuste aux variabilités acoustiques. La performance d'un système de reconnaissance de la parole est étroitement liée à la qualité de la modélisation acoustique des données utilisées.

Au cours de ce travail nous avons réalisé une base de données constituée de 18 mots d'un corpus orienté vers une application téléphonique, ces derniers seront utilisés pour former des phrases, suivant des règles définit par un modèle linguistique.

Pour mettre en œuvre notre système, nous avons choisi une modélisation statistique par l'utilisation des modèles HMM.

La première partie de ce travail est consacrée à la reconnaissance de mots isolés, cette partie est réalisée sous deux outils différents, MATLAB et HTK, et la deuxième partie pour la reconnaissance de la parole continue sous HTK, dans cette partie nous avons évalué les performances des paramètres acoustiques MFCC et LPC.

Les résultats obtenus nous ont permis de conclure que :

- La plate-forme HTK est plus performante que MATLAB suite aux résultats obtenus pour la reconnaissance de mots isolés.
- La méthode d'analyse avec les coefficients MFCC est la plus adéquate à la RAP.

Malgré les résultats remarquables des taux de reconnaissance obtenus avec les paramètres MFCC, le travail effectué dans ce mémoire reste à notre avis perfectible.

Comme perspectives à ce travail nous proposons :

- D'utiliser une nouvelle paramétrisation acoustique du signal de parole basée sur la fusion des paramètres MFCC et de nouveaux paramètres acoustiques dits auxiliaires qui sont le pitch, l'énergie et les fréquences des trois premiers formants, dans les mêmes vecteurs acoustiques, pour améliorer la robustesse du système.
- La mise au point d'une base de données plus riche à enregistrer en conditions réelles, construite avec des phrases phonétiquement équilibrées, enregistrées par une centaine de locuteurs, pour ensuite procéder à une segmentation en phonèmes dépendant du contexte qui seront modélisés par des HMM.

Bibliographie

[1] M.Kunt, 1980, Traitement numérique des signaux, Presses polytechniques romandes, Lausanne.

[2] Alain Goyé, Pérception Auditive 2002, TelecomParis.

[3] Dan Gnansia, Modèle auditif en temps réel, Université UPMC.

[4] René Boite et Murat Kunt, Traitement de la parole, 1987.

[5] Lawrence Rabiner, Biing-Hwang Juang, « Fundamentals of speech recognition », édition 1993.

[6] C. Barras, Reconnaissance de la parole continue : adaptation au locuteur et contrôle temporel dans les modèles de Markov cachés, thèse de doctorat de l'université Paris VI, 1996.

[7] G.Almouzni, 2011, Traitement de la parole, EISTI.

[8] J.P.Haton, J.M.Pierrel, G.Perennou, J.Caelen, J.L.Gauvain, 1991, Reconnaissance automatique de la parole, DUNOD.

[9] J.Makhoul, Linear prediction: A tutorial review, Proc, IEEE, vol. 63, pp. 561-580.

[10] M.Bouchamekh, Identification du locuteur indépendante du contexte, mémoire de Magister, ENP.

[11] J.M.Pierrel, 1982, utilisation de contraintes linguistiques en compréhension automatique de la parole continue : Le système MYRTILLE II.

[12] K. Bouchefra, 1995, Contribution à la reconnaissance automatique de la parole continue : Etude et réalisation d'un système de reconnaissance acoustico-phonétique » mémoir Magister, électronique ENP 1995.

[13] Hunt M.-J. et Lefebvre C., 1989, A comparison of several acoustic representations for speech recognition with degraded and undegraded speech, dans Proceedings of International Conference on Acoustics Speech and Signal Processing (ICASSP'1989), tome 1, Glasgow, UK.

[14] J-P Haton, A Bonneau, D Fohr, Y Laprie,Y Gong, J-M Pierrel, Décodage acoutico-phonétique : problèmes et éléments de solution.

[15] C. S. Myers and L. R. Rabiner, 1981, a comparative study of several dynamic time-warping algorithms for connected word recognition.

[16] LAWRENCE R. RABINER, 1989, A tutorial on Hidden Markov Models and selected applications in speech recognition.

[17] Amrous Anissa Imene, 2009, Coopération de connaissances dans les modèles de Markov cachés pour la reconnaissance automatique de la parole, Mémoire Magister USTHB.

[18] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, P. Woodland, 2006 , « The HTK Book (for HTK Version 3.4) », Cambridge University.

[19] H.TAKHEDMIT & N.AIT SAADI, 2005, « Identification du locuteur en mode indépendant du texte », Projet de fin d'étude à l'ENP, Dép. d'Electronique.

[20] M. Dekker, 2003, Speech Processing, a Dynamic and Optimization-Oriented Approach, Series: Signal Processing and Communications Series. ISBN-13:

9780824740405.

[21] Calliope, 1989, La parole et son traitement automatique, Edition Masson, Paris.

[22] www.wikipédia.com , Audition.

Annexe

Listes des figures de l'annexe

Listes des tableaux de l'annexe

L'outil HTK

Nous décrivons dans cette partie les détails techniques de la mise en œuvre d'un système de reconnaissance de la parole sous l'outil HTK. L'implémentation de ce système est réalisée par un programme écrit en langage script *Perl* sous le système d'exploitation Windows.

Organisation d'un espace de travail

On crée la hiérarchie de répertoires suivante :

- signal/ : Emmagasine les fichiers sons.
- labels/ : Emmagasine les étiquettes des fichiers sons.
- config/ : Emmagasine les fichiers de configuration.
- listes/ : Emmagasine les listes utilisées dans l'analyse et l'apprentissage.
- mfcc/ : Emmagasine les fichiers des coefficients mfcc calculés.
- gabarits/ : Emmagasine le HMM initial de chaque mot.
- hinit/ : Emmagasine le HMM de chaque mot estimé avec l'algorithme de Viterbi.
- hrest/ : Emmagasine le HMM de chaque mot ré-estimé avec l'algorithme de Baum-Welch.
- resultat/ : Emmagasine les fichiers résultants de la reconnaissance.

1. Préparation des données :

L'étape de préparation de données consiste à fournir les données acoustiques et lexicales nécessaires pour l'étape d'entraînement et l'étape de test.

1.1. Paramètres acoustiques :

Pour modéliser le signal de la parole on a utilisé les coefficients MFCC et les coefficients LPC, On a utilisé dans chaque type 12 coefficients en plus de l'énergie, ainsi que leurs dérivées premières et secondes ce qui fait 39 coefficients.

Les fichiers de paramètres sont calculés par l'outil **HCopy** qui prend comme entrée deux fichiers.

- Le premier fichier ***hcopyliste.txt*** indique la liste des fichiers sonores à paramétriser et l'emplacement souhaité pour stocker les résultats.
- Le deuxième fichier ***parametrisation.txt*** sert à mentionner la configuration acoustique souhaitée.

```
/*Les fichiers sons (.wav)   /* les fichier mfcc

signal/zero1.wav             mfcc/zero1.mfcc
signal/un1.wav               mfcc/un1.mfcc
signal/deux1.wav             mfcc/deux1.mfcc
signal/trois1.wav            mfcc/trois1.mfcc
signal/quatre1.wav           mfcc/quatre1.mfcc
signal/cinq1.wav             mfcc/cinq1.mfcc
signal/six1.wav              mfcc/six1.mfcc
signal/sept1.wav             mfcc/sept1.mfcc
signal/huit1.wav             mfcc/huit1.mfcc
signal/neuf1.wav             mfcc/neuf1.mfcc
signal/appeler1.wav          mfcc/appeler1.mfcc
signal/contacter1.wav        mfcc/contacter1.mfcc
signal/joindre1.wav          mfcc/joindre1.mfcc
signal/monsieur1.wav         mfcc/monsieur1.mfcc
signal/le1.wav               mfcc/le1.mfcc
signal/numero1.wav           mfcc/numero1.mfcc
signal/hamroun1.wav          mfcc/hamroun1.mfcc
signal/hadjloum1.wav         mfcc/hadjloum1.mfcc
```

Figure 1 : Exemple d'un fichier ***hcopyliste.txt***

```
SOURCEFORMAT=WAVE        #  le format des fichiers sources
TARGETKIND=MFCC_E_D_A    #  le type de coefficients à utiliser
WINDOWSSIZE=250000.0     #  25 ms = la longueur de la trame
TARGETRATE=10000.0       #  10 ms = la période des trames
NUMCEPS=12               #  nombre de coefficients MFCC
USEHAMMING=T             #  utilisation de la fenêtre
                            de pondération Hamming
PREEMCOEF=0.97           #  le coefficient de préaccentuation
NUMCHANS=26              #  le nombre des bancs de filtre MEL
CEPLIFTER=22             #  la longueur des bancs de filtres MEL
```

Figure 2 : Exemple d'un fichier ***paramatrisation.txt***

La commande qui permet d'obtenir les fichiers MFCC est la suivante :

```
Hcopy -T 1 -C Config/parametrisation.conf –S Listes/hcopyliste.txt
```

On peut afficher les coefficients MFCC obtenus dans un fichier *listeMFCC.txt* à l'aide de la commande suivante :

```
HList mfcc/appeler3.mfcc|more > listeMFCC.txt
```

```
------------------------------------ Samples: 0->-1 ------------------------------
0:  -9.853 -11.197  -7.512 -13.396  -1.849  -7.244  -2.164  -5.212   5.539   3.174
     9.253  -4.644  55.329  -0.012   0.050   0.192   0.171  -0.043   0.217   0.309
     0.061  -0.284  -0.183   0.218  -0.237  -0.012   0.007   0.013   0.076   0.090
    -0.022  -0.017  -0.026  -0.043  -0.100  -0.148  -0.089  -0.066   0.004
1:  -9.881 -11.072  -7.166 -13.144  -1.952  -6.728  -1.356  -4.944   5.021   3.082
     10.087 -4.906  55.286  -0.004   0.077   0.351   0.349  -0.088   0.225   0.316
    -0.004  -0.485  -0.462   0.118  -0.412  -0.005   0.012   0.016   0.096   0.127
    -0.037  -0.071  -0.102  -0.086  -0.148  -0.222  -0.199  -0.043   0.006
2:  -9.897 -11.009  -6.724 -12.667  -2.013  -6.415  -1.024  -5.038   4.381   2.303
     9.925  -5.695  55.292   0.017   0.100   0.493   0.530  -0.129   0.127   0.175
    -0.119  -0.683  -0.786  -0.176  -0.478   0.005   0.010   0.016   0.076   0.120
    -0.050  -0.123  -0.172  -0.113  -0.168  -0.213  -0.265   0.049   0.002
3:  -9.850 -10.904  -6.151 -12.017  -2.205  -6.534  -1.154  -5.320   3.693   1.301
     9.504  -6.176  55.320   0.033   0.107   0.523   0.626  -0.185  -0.092  -0.133
    -0.280  -0.824  -0.990  -0.581  -0.329   0.011  -0.004   0.007   0.005   0.051
```

Figure 3 : Exemple des coefficients MFCC.

Et pour changer le type de coefficient il sufi de changer dans le fichier ***paramatrisation.txt*** l'expression `MFCC_E_D_A` par `LPC_E_D_A` ou autres paramètres.

Indice	Signification
_A	Coefficients d'accélération (seconde dérivé)
_D	Coefficients de vitesse (première dérivé)
_E	Logarithme de l'énergie
_N	Valeur absolue de Log de l'énergie
_T	Troisième dérivé
_0	Coefficient cepstral C_0

Tableau 1 : Indices des paramètres et leurs significations.

1.2. Étiquetage des fichiers sons :

Le but de l'étiquetage est de délimiter chaque entité lexicale. Ceci sera fait manuellement avec la commande **HSLab** dont la syntaxe est :

```
HSlab –F WAVE –L labels/fichier.lab Signal/fichierson.wav
```

2. Description des modèles de Markov : définir les prototypes HMMs

A chaque entité lexicale (mot du vocabulaire) on va créer un fichier Gabarit (model) représentant l'entité lexical dans la pratique.

Définir le modèle prototype revient à définir manuellement la structure des modèles HMMs à utiliser sous une syntaxe propre à HTK. Le modèle prototype utilisé dans notre travail est un modèle gauche-droite à trois états émetteurs avec une gaussienne à matrice de covariance diagonale de dimension 39 (12 coefficients MFCCs et l'énergie plus leurs dérivées premières et secondes) définit comme suit :

```
<BeginHMM>
<VecSize> 39
<MFCC_0_D_A>
<NumStates> 5
<State> 2
<Mean> 39
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> 39
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1
<State> 3
<Mean> 39
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> 39
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1
<State> 4
<Mean> 39
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> 39
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1
<TransP> 5
0.0 1.0 0.0 0.0 0.0
0.0 0.6 0.4 0.0 0.0
0.0 0.0 0.6 0.4 0.0
0.0 0.0 0.0 0.7 0.3
0.0 0.0 0.0 0.0 0.0
<EndHMM>
```

Figure 4 : Exemple d'un modèle prototype.

<NumStates> : nombre d'état du modèle

<VecSize> : nombre de composantes du vecteur spectral (12 MFCC + 1 énergie + 13 dérivés premières + 13 dérivés secondes)

<state > : qui contient les lois d'émission <mean> et <variance>

<TransP> : contient les probabilités des transitions entre les états du model (délimitées par <TransP> et <EndHMM>)

3. Apprentissage

Après l'initialisation des modèles acoustiques, l'étape suivante consiste à les entrainer en utilisant la base acoustique et la base phonétique.

3.1. Apprentissage avec l'algorithme Viterbi (HInit)

Chaque modèle doit être entrainé : les moyennes, les variances et les probabilités de transitions entre états sont ré estimées jusqu'à ce qu'un seuil de convergence ou qu'un nombre maximum d'itérations soit atteint. Ceci est fait par l'algorithme de Viterbi.

On commence d'abord par création de deux fichiers ***config/hinit.txt*** et ***listes/mfcc.lst***

```
#Fichier de configuration pour l'initialisation des

#modèles de Markov par l'algorithme de Viterbi

TARGETKIND = MFCC_0_D_A
```

Figure 5 : Contenu d'un fichier ***hinit.txt.***

```
mfcc/zero1.mfcc
mfcc/un1.mfcc
mfcc/deux1.mfcc
mfcc/trois1.mfcc
mfcc/quatre1.mfcc
mfcc/cinq1.mfcc
mfcc/six1.mfcc
mfcc/sept1.mfcc
mfcc/huit1.mfcc
mfcc/neuf1.mfcc
mfcc/appeler1.mfcc
mfcc/contacter1.mfcc
mfcc/joindre1.mfcc
mfcc/monsieur1.mfcc
mfcc/le1.mfcc
mfcc/numero1.mfcc
mfcc/hamroun1.mfcc
mfcc/hadjloum1.mfcc
        ⋮
        ⋮
```

Figure 6 : Contenu d'un fichier ***mfcc.lst.***

La syntaxe de la commande HInit :

```
HInit -C config/hinit.txt -A -o appeler -l appeler -L labels/
-i 30 -T 1 -m 2 gabarits/appeler -S listes/mfcc.lst
```

Pour chaque mot du vocabulaire on exécute la même ligne de commande, toutes ces lignes sont mises dans un fichier script nommé **HInit.bat**

```
HInit -C config/hinit.txt -A -o zero -l zero -L labels/ -i 30 -T 1 -m 2
gabarits/zero -S listes/mfcc.lst
HInit -C config/hinit.txt -A -o un -l un -L labels/ -i 30 -T 1 -m 2
gabarits/un -S listes/mfcc.lst
HInit -C config/hinit.txt -A -o deux -l deux -L labels/ -i 30 -T 1 -m 2
gabarits/deux -S listes/mfcc.lst
HInit -C config/hinit.txt -A -o trois -l trois -L labels/ -i 30 -T 1 -m
2 gabarits/trois -S listes/mfcc.lst
HInit -C config/hinit.txt -A -o quatre -l quatre -L labels/ -i 30 -T 1
-m 2 gabarits/quatre -S listes/mfcc.lst
HInit -C config/hinit.txt -A -o cinq -l cinq -L labels/ -i 30 -T 1 -m 2
gabarits/cinq -S listes/mfcc.lst
HInit -C config/hinit.txt -A -o six -l six -L labels/ -i 30 -T 1 -m 2
gabarits/six -S listes/mfcc.lst
HInit -C config/hinit.txt -A -o sept -l sept -L labels/ -i 30 -T 1 -m 2
gabarits/sept -S listes/mfcc.lst
HInit -C config/hinit.txt -A -o huit -l huit -L labels/ -i 30 -T 1 -m 2
gabarits/huit -S listes/mfcc.lst
HInit -C config/hinit.txt -A -o neuf -l neuf -L labels/ -i 30 -T 1 -m 2
gabarits/neuf -S listes/mfcc.lst
HInit -C config/hinit.txt -A -o appeler -l appeler -L labels/ -i 30 -T
1 -m 2 gabarits/appeler -S listes/mfcc.lst
HInit -C config/hinit.txt -A -o contacter -l contacter -L labels/ -i 30
-T 1 -m 2 gabarits/contacter -S listes/mfcc.lst
HInit -C config/hinit.txt -A -o joindre -l joindre -L labels/ -i 30 -T
1 -m 2 gabarits/joindre -S listes/mfcc.lst
HInit -C config/hinit.txt -A -o le -l le -L labels/ -i 30 -T 1 -m 2
gabarits/le -S listes/mfcc.lst
HInit -C config/hinit.txt -A -o numero -l numero -L labels/ -i 30 -T 1
-m 2 gabarits/numero -S listes/mfcc.lst
HInit -C config/hinit.txt -A -o monsieur -l monsieur -L labels/ -i 30 -
T 1 -m 2 gabarits/monsieur -S listes/mfcc.lst
HInit -C config/hinit.txt -A -o hamroun -l hamroun -L labels/ -i 30 -T
1 -m 2 gabarits/hamroun -S listes/mfcc.lst
HInit -C config/hinit.txt -A -o hadjloum -l hadjloum -L labels/ -i 30 -
T 1 -m 2 gabarits/hadjloum -S listes/mfcc.lst
HInit -C config/hinit.txt -A -o sil -l sil -L labels/ -i 30 -T 1 -m 2
gabarits/sil -S listes/mfcc.lst
```

Figure 7 : Contenu du fichier script HInit.bat.

Pour chaque ligne de commande dans le fichier script **HInit.bat** on aura un nouveau fichier qui sera créé dont le contenu est le suivant :

```
~o
<STREAMINFO> 1 39
<VECSIZE> 39<NULLD><MFCC_D_A_0><DIAGC>
~h "appeler"
<BEGINHMM>
<NUMSTATES> 5
<STATE> 2
<MEAN> 39
 -1.732802e+000 -1.181614e+001 -9.816505e+000 -2.045082e+001 -
4.287344e+000 -3.383931e+000 -1.538001e+000 -8.151063e+000 -
3.556501e+000 -6.377594e+000 9.844422e-001 -6.442388e+000 6.886425e+001
2.061237e-003 2.952124e-002 5.081705e-002 5.609526e-002 -3.115970e-002
-3.766382e-002 -3.223830e-002 4.910277e-003 2.648256e-002 -1.079687e-
002 1.833437e-002 1.864489e-002 -7.290933e-003 -8.372002e-004 -
5.244153e-004 3.298007e-003 2.789897e-003 2.180701e-004 -2.661208e-003
-1.519953e-003 -1.357060e-003 1.227583e-003 -1.238543e-004 3.480991e-
003 2.865069e-003 7.145580e-004
<VARIANCE> 39
 2.363630e+001 4.579180e+001 3.419992e+001 5.319261e+001 2.449642e+001
4.139298e+001 3.858567e+001 3.319038e+001 3.490162e+001 2.925568e+001
3.793683e+001 4.128314e+001 4.060265e+001 9.635400e-002 1.670488e-001
1.848241e-001 3.011754e-001 2.716977e-001 3.745350e-001 3.186164e-001
4.541562e-001 3.507645e-001 3.366036e-001 3.185650e-001 3.316636e-001
9.918816e-002 1.000000e-002 1.412639e-002 1.619316e-002 3.188354e-002
2.821518e-002 4.291806e-002 3.378295e-002 4.970991e-002 3.413881e-002
3.295778e-002 3.102792e-002 3.139162e-002 1.000000e-002
<GCONST> 5.243670e+001
<STATE> 3
<MEAN> 39
 -3.299233e+000 -1.034282e+001 9.274348e+000 -1.418876e+001 -
1.294330e+001 -1.789176e+001 -1.450006e+001 -8.546938e+000 -
5.389149e+000 -6.284163e+000 5.865854e+000 -3.192379e+000 7.273571e+001
-9.469452e-003 1.236156e-002 1.853902e-002 -1.548759e-002 -1.764142e-
002 -3.872519e-002 -6.043246e-003 -1.509473e-002 -3.564721e-002 -
1.654216e-003 3.030550e-003 3.844985e-003 -1.946589e-002 -7.395988e-004
1.299346e-003 -3.260131e-003 -1.627930e-003 2.266659e-003 4.559717e-003
3.923875e-003 1.050179e-003 3.422099e-004 4.037903e-004 -2.906989e-003
-8.772525e-004 -2.362581e-003
<VARIANCE> 39
 1.442633e+001 1.844429e+001 5.407204e+001 1.023092e+002 4.891072e+001
3.407600e+001 6.315559e+001 5.572584e+001 2.328495e+001 4.292968e+001
3.021903e+001 2.968725e+001 9.497493e+000 3.916818e-002 9.604666e-002
1.388682e-001 1.948542e-001 2.342010e-001 2.617077e-001 2.790399e-001
2.634808e-001 2.361955e-001 2.896339e-001 2.743956e-001 2.290335e-001
3.809873e-002 1.000000e-002 1.000000e-002 1.196338e-002 1.861572e-002
2.202675e-002 2.842564e-002 2.989336e-002 2.907500e-002 2.738356e-002
3.000226e-002 2.536967e-002 2.310595e-002 1.000000e-002
<GCONST> 4.291458e+001
<STATE> 4
<MEAN> 39
 -4.678326e+000 -6.977304e+000 -1.252134e-001 -1.515621e+001 -
9.461864e+000 -1.403540e+001 -6.489711e+000 -1.055979e+001 -
3.561410e+000 -5.907343e+000 3.680595e+000 -1.663229e+000 6.226024e+001
```

```
-1.861301e-002 -4.407093e-003 -4.141203e-002 7.281237e-003 4.851337e-
002 4.736140e-002 1.670582e-002 1.961367e-002 3.505298e-002 -2.219470e-
003 -3.239943e-002 2.191959e-003 -3.479389e-002 -2.991139e-004 -
5.436804e-004 1.835582e-006 1.872124e-005 -7.838555e-004 -2.301347e-003
-2.837890e-003 -1.434136e-003 -2.327763e-003 1.144071e-003 4.171875e-
003 1.952790e-003 4.658367e-004
<VARIANCE> 39
 9.997120e+000 8.002876e+000 1.661961e+001 2.745561e+001 3.213916e+001
2.329008e+001 3.296492e+001 3.008710e+001 2.370621e+001 2.587943e+001
1.615950e+001 1.417958e+001 5.057935e+000 3.390260e-002 4.961784e-002
1.113367e-001 1.479108e-001 2.012962e-001 2.526064e-001 3.597533e-001
3.192262e-001 3.339782e-001 3.219400e-001 3.054864e-001 2.649196e-001
1.288752e-002 1.000000e-002 1.000000e-002 1.000000e-002 1.161017e-002
1.682235e-002 2.105004e-002 2.959420e-002 2.809834e-002 3.005190e-002
2.566830e-002 2.584465e-002 2.307177e-002 1.000000e-002
<GCONST> 3.193491e+001
<TRANSP> 5
 0.000000e+000 1.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
 0.000000e+000 9.953146e-001 4.685408e-003 0.000000e+000 0.000000e+000
 0.000000e+000 0.000000e+000 9.952766e-001 4.723348e-003 0.000000e+000
 0.000000e+000 0.000000e+000 0.000000e+000 9.934457e-001 6.554308e-003
 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
<ENDHMM>
```

Figure 8 : Exemple d'un modèle HMM estimé avec l'algorithme Viterbi.

3.2. Apprentissage avec l'algorithme de Baum Welch (HRest)

Les modèles résultants de l'étape précédente seront ré-estimés de façon indépendante avec l'algorithme de Baum Welch en utilisant la commande **HRest,** cette commande est appliquée sur chacun des modèles.

On doit d'abord créer un fichier ***config/hrest.txt***

```
#Fichier de configuration pour la ré-estimation des modèles de Markov
#par l'algorithme de Baum Welch

 TARGETKIND = MFCC_0_D_A
```

*Figure 9 : Contenu du fichier **hrest.txt.***

La syntaxe de la commande **HRest** :

```
HRest -C config/hrest.txt -A -l appeler -M hrest/ -L labels/ -
i 20 -T 1 -m 2 hinit/appeler -S listes/mfcc.lst.
```

Pour chaque mot du vocabulaire on exécute la même ligne de commande, toutes ces lignes sont mises dans un fichier script nommé **HRest.bat**.

```
HRest -C config/hrest.txt -A -l zero -M hrest/ -L labels/ -i 40 -T 1 -m
2 zero -S listes/mfcc.lst
HRest -C config/hrest.txt -A -l un -M hrest/ -L labels/ -i 40 -T 1 -m 2
un -S listes/mfcc.lst
HRest -C config/hrest.txt -A -l deux -M hrest/ -L labels/ -i 40 -T 1 -m
2 deux -S listes/mfcc.lst
HRest -C config/hrest.txt -A -l trois -M hrest/ -L labels/ -i 40 -T 1 -
m 2 trois -S listes/mfcc.lst
HRest -C config/hrest.txt -A -l quatre -M hrest/ -L labels/ -i 40 -T 1
-m 2 quatre -S listes/mfcc.lst
HRest -C config/hrest.txt -A -l cinq -M hrest/ -L labels/ -i 40 -T 1 -m
2 cinq -S listes/mfcc.lst
HRest -C config/hrest.txt -A -l six -M hrest/ -L labels/ -i 40 -T 1 -m
2 six -S listes/mfcc.lst
HRest -C config/hrest.txt -A -l sept -M hrest/ -L labels/ -i 40 -T 1 -m
2 sept -S listes/mfcc.lst
HRest -C config/hrest.txt -A -l huit -M hrest/ -L labels/ -i 40 -T 1 -m
2 huit -S listes/mfcc.lst
HRest -C config/hrest.txt -A -l neuf -M hrest/ -L labels/ -i 40 -T 1 -m
2 neuf -S listes/mfcc.lst
HRest -C config/hrest.txt -A -l appeler -M hrest/ -L labels/ -i 40 -T 1
-m 2 appeler -S listes/mfcc.lst
HRest -C config/hrest.txt -A -l contacter -M hrest/ -L labels/ -i 40 -T
1 -m 2 contacter -S listes/mfcc.lst
HRest -C config/hrest.txt -A -l joindre -M hrest/ -L labels/ -i 40 -T 1
-m 2 joindre -S listes/mfcc.lst
HRest -C config/hrest.txt -A -l monsieur -M hrest/ -L labels/ -i 40 -T
1 -m 2 monsieur -S listes/mfcc.lst
HRest -C config/hrest.txt -A -l le -M hrest/ -L labels/ -i 40 -T 1 -m 2
le -S listes/mfcc.lst
HRest -C config/hrest.txt -A -l numero -M hrest/ -L labels/ -i 40 -T 1
-m 2 numero -S listes/mfcc.lst
HRest -C config/hrest.txt -A -l hamroun -M hrest/ -L labels/ -i 40 -T 1
-m 2 hamroun -S listes/mfcc.lst
HRest -C config/hrest.txt -A -l hadjloum -M hrest/ -L labels/ -i 40 -T
1 -m 2 hadjloum -S listes/mfcc.lst
HRest -C config/hrest.txt -A -l sil -M hrest/ -L labels/ -i 40 -T 1 -m
2 sil -S listes/mfcc.lst
```

*Figure10 : Contenu du fichier script **HRest.bat**.*

Pour chaque ligne de commande dans ce fichier, un modèle ré-estimé sera créé dans le répertoire ***hrest***.

```
~o
<STREAMINFO> 1 39
<VECSIZE> 39<NULLD><MFCC_D_A_0><DIAGC>
~h "appeler"
<BEGINHMM>
<NUMSTATES> 5
<STATE> 2
<MEAN> 39
 -1.601450e+000 -1.375824e+001 -9.386985e+000 -1.978215e+001 -
6.354636e+000 -5.653336e+000 -1.715753e+000 -9.863820e+000 -
1.156782e+000 -4.826396e+000 -4.291885e-001 -5.623782e+000
7.313920e+001 3.498766e-003 2.411531e-002 6.228152e-002 4.369111e-002 -
2.819116e-002 -5.512110e-002 -3.375026e-002 6.843538e-005 2.561934e-003
-1.875808e-002 6.442603e-003 1.870411e-002 -1.216795e-002 -1.054222e-
004 -8.676595e-004 6.612976e-004 7.091962e-005 -9.235650e-004 -
3.995738e-003 -3.384039e-003 -2.655447e-003 -1.478308e-003 -1.576491e-
003 4.183956e-004 9.487689e-005 -5.144259e-004
<VARIANCE> 39
 1.860672e+001 3.419173e+001 3.038787e+001 4.654590e+001 3.162519e+001
5.975270e+001 3.713322e+001 3.804234e+001 3.572962e+001 5.580003e+001
3.642212e+001 3.066439e+001 7.290754e+001 9.884768e-002 1.484611e-001
1.967780e-001 2.918081e-001 3.095984e-001 4.098380e-001 3.550462e-001
4.444291e-001 3.617191e-001 3.571665e-001 3.375826e-001 3.046301e-001
1.152960e-001 5.951102e-003 1.288700e-002 1.751398e-002 2.820080e-002
3.029211e-002 4.137107e-002 3.529570e-002 4.337888e-002 3.766581e-002
3.586591e-002 3.171665e-002 2.897377e-002 3.337902e-003
<GCONST> 5.206429e+001
<STATE> 3
<MEAN> 39
 -1.316653e+000 -1.253949e+001 9.812311e+000 -1.195927e+001 -
1.657534e+001 -1.922692e+001 -1.198572e+001 -8.466376e+000 -
5.512595e+000 -6.236686e+000 2.520418e+000 -2.157949e+000 7.561269e+001
5.442245e-004 1.277082e-002 2.563134e-002 1.404465e-003 -3.702433e-002
-1.662679e-002 1.418753e-004 -1.053626e-002 -2.219049e-002 1.627536e-
002 2.106927e-002 1.353302e-002 -1.555678e-002 -5.799731e-004
1.564807e-003 -2.826308e-003 1.200523e-004 2.095751e-003 2.666032e-003
3.387540e-003 1.069781e-003 1.120144e-003 8.065065e-004 8.277658e-005
1.158294e-003 -2.357507e-003
<VARIANCE> 39
 1.048140e+001 1.205076e+001 5.757897e+001 5.616758e+001 5.943454e+001
6.395554e+001 5.219715e+001 6.181977e+001 5.390060e+001 6.212344e+001
6.689722e+001 3.529959e+001 1.958454e+001 3.177018e-002 8.510653e-002
1.223849e-001 1.556168e-001 2.056046e-001 2.454749e-001 2.773903e-001
2.714450e-001 2.509159e-001 2.932756e-001 2.572709e-001 2.111491e-001
3.181905e-002 2.751658e-003 8.481376e-003 1.156590e-002 1.638972e-002
2.031305e-002 2.498588e-002 2.836183e-002 2.850294e-002 2.755288e-002
2.884063e-002 2.500466e-002 2.193651e-002 1.409516e-003
<GCONST> 4.020822e+001
<STATE> 4
<MEAN> 39
 -1.947714e+000 -8.557197e+000 1.436459e+000 -1.491578e+001 -
1.473596e+001 -1.811414e+001 -8.754389e+000 -1.474002e+001 -
4.578990e+000 -5.305638e+000 3.741263e+000 -1.789996e-001 6.724818e+001
-1.362159e-002 4.459932e-003 -4.531585e-002 -1.005998e-002 3.954646e-
002 1.683782e-002 9.660539e-003 1.059152e-002 1.513962e-002 -1.767118e-
003 -2.544611e-003 -7.231834e-003 -5.791822e-002 9.801694e-005 -
9.472367e-004 1.665681e-003 -9.557726e-004 -1.812176e-003 -1.205887e-
```

```
003 -8.374074e-004 1.058297e-003 -1.977643e-003 5.270273e-005
3.568858e-005 4.816013e-004 1.321431e-003
<VARIANCE> 39
 1.058899e+001 1.389742e+001 3.600969e+001 3.531015e+001 4.802316e+001
3.776861e+001 2.833572e+001 3.154842e+001 2.212526e+001 2.672835e+001
2.303193e+001 2.494123e+001 4.899735e+001 3.459323e-002 7.024219e-002
1.277754e-001 1.750976e-001 2.405518e-001 3.083746e-001 3.321275e-001
3.496141e-001 3.249561e-001 3.457129e-001 3.205292e-001 3.066695e-001
1.804290e-002 3.076787e-003 6.598868e-003 1.182455e-002 1.763533e-002
2.398208e-002 3.017689e-002 3.261076e-002 3.506978e-002 3.314646e-002
3.297461e-002 3.027147e-002 2.683181e-002 9.750000e-004
<GCONST> 3.749548e+001
<TRANSP> 5
 0.000000e+000 1.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
 0.000000e+000 9.952090e-001 4.791036e-003 0.000000e+000 0.000000e+000
 0.000000e+000 0.000000e+000 9.948912e-001 5.108862e-003 0.000000e+000
 0.000000e+000 0.000000e+000 0.000000e+000 9.936153e-001 6.384722e-003
 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
<ENDHMM>
```

Figure 11 : Exemple d'un modèle HMM ré-estimé avec l'algorithme Baum Welch.

4. Reconnaissance :

La reconnaissance se fait avec l'algorithme de Viterbi, implémenté sous HTK avec l'outil **HVite** , cet outil nécessite l'utilisation de différentes connaissances : les modèles HMMs (déjà entraînés), le réseau syntaxique des différents chemins à suivre, le modèle du langage et le dictionnaire.

4.1. Le modèle de langage

L'utilisation d'un modèle de langage permet d'améliorer la qualité de la reconnaissance, la grammaire de notre langage est définie dans le fichier **listes/modeleLangage.txt** :

```
$chif=zero|un|deux|trois|quatre|cinq|six|sept|huit|neuf;
$verbe=appeler|contacter|joindre;
$propre=hamroun|hadjloum;
$sil=sil;
$num=[$sil] $chif [$sil] $chif [$sil] $chif [$sil] $chif [$sil] $chif
[$sil] $chif [$sil] $chif [$sil] $chif [$sil] $chif [$sil] $chif;
(SENT-START
 ([$sil] $verbe [$sil] monsieur [$sil] $propre [$sil])|
 ([$sil] $verbe [$sil] le [$sil] numero  $num [$sil])
SENT-END)
```

Figure 12 : Modèle de langage utilisé pour la reconnaissance de phrase.

4.2. Le réseau syntaxique

Le réseau syntaxique est un fichier qui sert à déterminer les différents chemins possibles dans le processus de reconnaissance. Il est constitué d'un ensemble de nœuds et d'arcs, chaque nœud représente un mot de vocabulaire et chaque arc représente une transition possible entre deux mots.

En appliquant la commande **HParse** à cette grammaire (modèle de langage), on génère le réseau syntaxique (modeleDeMots)

La syntaxe de la commande **HParse :**

```
HParse -T 1 listes/modelelangage.txt listes/modeledemots.txt
```

```
VERSION=1.0

N=147  L=355

I=0    W=SENT-END
I=1    W=!NULL
I=2    W=sil
I=3    W=neuf
I=4    W=!NULL
I=5    W=huit
I=6    W=sept
I=7    W=six
I=8    W=cinq
I=9    W=quatre
I=10   W=trois
…
…

J=0     S=2    E=0
J=1     S=4    E=0
J=2     S=0    E=1
J=3     S=132  E=1
J=4     S=4    E=2
J=5     S=14   E=3
J=6     S=16   E=3
J=11    S=8    E=4
…
…
```

Figure 13 : Réseau syntaxique.

4.3. Le décodage

Après le modèle de langage et la construction de réseau syntaxique, il ne reste qu'a exécuté la commande **HVite** pour effectuer la reconnaissance proprement dite.

On doit d'abord créer deux fichiers ***mfccTest.lst*** et ***listemodeles.txt***

- ***listes/mfccTest.lst*** : contient le chemin de chaque fichier MFCC correspondant à une phrase (ou un mot) inconnue à reconnaître.
- ***listes/listemodeles.txt*** : contient les mots du langage.

```
mfcc/joinlenum06H.mfcc
mfcc/conmonshadjH2.mfcc
mfcc/applenum052.mfcc
mfcc/conmonsham2.mfcc
mfcc/joinmonshad2.mfcc
mfcc/appmonshadjb.mfcc
mfcc/joinmonshamb.mfcc
mfcc/applenum07b.mfcc
mfcc/conlenum05b.mfcc
mfcc/joinlenum05b.mfcc
```

Figure 14 : Fichier ***mfccTest.lst***

```
Sil
Zero
Un
Deux
Trois
Quatre
Cinq
six
sept
huit
neuf
appeler
contacter
joindre
monsieur
le
numero
hamroun
hadjloum
```

Figure 15 : Fichier ***listemodeles.txt***

La syntaxe de la commande **HVite :**

```
HVite -T 1 -A -w listes/modeledemots.txt -d hrest/ -l
resultat/hrest/ -S listes/mfcc.ar.lst listes/dictionnaire.txt
listes/listemodeles.txt
```

Le résultat est un fichier « .rec » pour chaque fichier à reconnaître.

```
0.0000000 620000  sil -2471.170898
620000 10400000   joindre -34045.300781
10400000 10720000 sil -1113.744385
10720000 19370000 monsieur -31420.802734
19370000 21360000 sil -6874.365723
21360000 25290000 hadjloum -16830.714844
25290000 28700000 sil -12241.989258
```

*Figure 16 : Fichier **joinmonshadj.rec***

4.5. Evaluation des résultats

L'évaluation des résultats revient à comparer les résultats obtenus après l'étape de décodage avec des résultats de référence. L'outil **HResuls** permet de réaliser cette comparaison par alignement dynamique entre le décodage obtenu et la source de référence.

La syntaxe de la commande **HResults** :

```
HResults -T 1 -L labels/ listes/listemodeles.txt
resultat/hrest/joinlenum05b.rec
```

Pour exécuter cette commande sur chacune des phrases de test on crée un fichier script dans lequel on met toutes les lignes de commandes de tous les tests.

```
HResults -T 1 -L labels/ listes/listemodeles.txt
resultat/hrest/joinlenum05b.rec
HResults -T 1 -L labels/ listes/listemodeles.txt
resultat/hrest/applenum07b.rec
HResults -T 1 -L labels/ listes/listemodeles.txt
resultat/hrest/joinmonshamb.rec HResults -T 1 -L labels/testH
listes/listemodeles.txt resultat/hrest/joinmonshad2.rec
HResults -T 1 -L labels/testH listes/listemodeles.txt
resultat/hrest/applenum052.rec
HResults -T 1 -L labels/ listes/listemodeles.txt
resultat/hrest/conmonsham2.rec
HResults -T 1 -L labels/ listes/listemodeles.txt
resultat/hrest/joinlenum06H.rec
HResults -T 1 -L labels/ listes/listemodeles.txt
resultat/hrest/conmonshadjH2.rec
PAUSE
```

Figure 17 : Fichier script HResults1.bat.

Après exécution de ce script on aura le taux de reconnaissance pour chaque phrase de test séparément.

```
====================== HTK Results Analysis =======================
Date: Tue Jun 05 12:23:43 2012
Ref : labels/
Rec : resultat/hrest/joinmonshad2.rec
------------------------ Overall Results --------------------------
WORD: %Corr=100.00, Acc=85.71 [H=7, D=0, S=0, I=1, N=7]
===================================================================
```

Figure 18 : Résultat d'évaluation pour une phrase de test « Joindre monsieur hadjloum ».

Pour évaluer le taux de reconnaissance global on crée un fichier script **HResults2.bat**

```
#Évaluation des résultats obtenus pour les modèles de Baum Welch)

HResults -T 1 -L labels/ listes/listemodeles.txt
resultat/hrest/conlenum05b.rec resultat/hrest/applenum07b.rec
resultat/hrest/joinmonshamb.rec resultat/hrest/joinmonshad2.rec
resultat/hrest/applenum052.rec resultat/hrest/conmonsham2.rec
resultat/hrest/joinlenum06H.rec resultat/hrest/conmonshadjH2.rec
PAUSE
```

Figure 19 : Fichier script HResults2.bat.

En exécutant le script on aura le taux de reconnaissance global :

```
====================== HTK Results Analysis =======================
  Date: Tue Jun 05 12:33:10 2012
  Ref : labels/
  Rec : resultat/hrest/conlenum05b.rec
      : resultat/hrest/applenum07b.rec
      : resultat/hrest/joinmonshamb.rec
      : resultat/hrest/joinmonshad2.rec
      : resultat/hrest/applenum052.rec
------------------------ Overall Results --------------------------
WORD: %Corr=93.75, Acc=72.66 [H=120, D=0, S=8, I=27, N=128]
===================================================================
```

Figure 20 : Taux de reconnaissance global.