

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

Université Dr. YAHIA FARES de Médéa

Faculté des Sciences et de la Technologie

Département Génie électrique et d'Informatique

Projet de Fin d'Etudes

Pour obtenir le diplôme de MASTER

Option : Systèmes Intelligents et Communicants

THEME : Reconnaissance audiovisuelle de la parole par les Modèles de Markov Cachés

Présentée par FERARSA AMEL

Encadré par : Monsieur. DJAHID RAGGAI

Soutenu le 4 juillet 2012 devant le jury composé de

President: Mr.MAZARI AHMED CHERIF

Examinateurs: Mr. MAAZOUZ MOHAMED

Mr. ALIMOHAD ABDENOUR

Remerciements

En premier lieu, je tiens à remercier comme il se doit les membres de jury qui m'ont fait l'honneur d'accepter d'évaluer ce travail.

Plus particulièrement, je tiens à remercier chaleureusement mon encadreur, je remercie Monsieur **Djahid Raggai** pour son soutien, son intérêt constant vis-à-vis de mon travail et sa disponibilité. Je suis honorée d'avoir fait mon projet sous sa direction. Je tiens à lui exprimer le réel plaisir que fut pour moi cette collaboration.

Je remercie le chef département de génie électrique et d'informatique Monsieur le docteur Mounir Bouhedda et Monsieur El Said Billal.

À ce stade des remerciements, une question se pose : Comment remercier plus particulièrement certains sans risquer d'en oublier d'autres ? Quel dilemme, dois-je commencer une énumération non-exhaustive ? Je me lance.

Je remercie donc ma voisine et mon amie Amina,razika,khadija, Hassina, Amina, Amina, Faiza, Yasmina, Rima, Thoraya et toutes mes amies de classe.

J'en viens enfin à la partie familiale. Je remercie toute ma famille, mes sœurs et mes frères qui m'ont toujours soutenue. Plus particulièrement, je remercie ma mère qui m'a toujours encouragée dans mes études, qui a toujours exprimé sa fierté à mon égard. Ce soutien permanent et cette croyance en moi m'a toujours poussée de l'avant, merci pour tout.

Dédicace

Je dédie ce travail :

A ma très chère mère qui m'a éclairée et qui m'a encouragé et soutenue toute au long de mes études (que dieu la guérir).

A mon chère père (que dieu ait son âme).

A mes sœurs : Laounia, Assia, Aicha et Rekia.

A mes frères : Menad et Yousef.

ملخص

في هذه المذكرة نقدم نظام التعرف السمعي البصري على الكلام بدمج المعلومات الصوتية والبصرية.هذا النظام يستعمل نماذج ماركوف المخفية المستمرة كمحرك للتعرف السمعي البصري. المعلومة البصرية المقترنة بالمعلومة الصوتية تستند على شكل وحركة الشفتين أثناء الكلام. تجارب التعرف على الكلمة المنفردة أثبتت أن دمج المعلومة الصوتية بالبصرية يرفع من فعالية هذا النظام في بيئة حقيقية (صاخبة جدا)

Abstract

In this thesis, we present a system of audiovisual speech recognition (AVSR) combining the acoustic data and visual data. The recognition system uses audiovisual recognition engine as the continuous Hidden Markov Models (Continuous Hidden Markov Model, CHMM). The visual information used in conjunction with the acoustic data is based on the shape and lip movements during speech. Experiments for isolated word recognition show that the combined use of acoustic modality and visual modality increases the performance of ASR system in a real environment (very noisy).

Résumé

Dans ce mémoire, nous présentons un système de reconnaissance audiovisuelle de la parole (RAVP) combinant les données acoustiques et les données visuelles. Ce système de reconnaissance audiovisuelle utilise comme moteur de reconnaissance les Modèles de Markov Cachés continus (Continuous Hidden Markov Model, CHMM). L'information visuelle utilisée conjointement avec les données acoustiques est basée sur la forme et les mouvements des lèvres lors de l'élocution. Les expériences pour la reconnaissance des mots isolés indiquent que l'utilisation conjointe de la modalité acoustique et de la modalité visuelle augmente la performance du système de RAP en milieu réel (fortement bruité).

Glossaire

Signal acoustique	Signal analogique représentatif de l'onde sonore
Signal stationnaire	c'est-à-dire ses propriétés statistiques caractérisées par des espérances mathématiques sont indépendantes du temps.
Voix chuchotée	Voix base en soufflant
ARPAbet	code développé par Advanced Research Projects Agency (ARPA), il représente chaque phonème avec une séquence distincte de caractères ASCII.
Théorème de Shannon	Fréquence d'échantillonnage supérieure à 2 fois la fréquence maximale du signal
audiogramme	Courbe représentant la variation de la fréquence du signal parole en fonction du temps
Formant	Endroit qui représente le maximum d'énergie sonore
Harmoniques	Les multiples de la fréquence fondamentale F_0
Echelle de Mel	est une échelle de fréquences basée sur la perception humaine. Elle se mesure en Mels

Sigles et Abréviations (par apparition)

V	Voiced
U	Unvoiced
S	Silence
TFCT	Transformée de Fourier discrète à Court Terme
LPC	Lineair Prédiction Coefficients
TFD	Transformée de Fourier Discrète
DCT	Discrete Cosine Transform
MFCC	Mel Frequency Cepstral Coefficients
MMC	Modèles de Markov Cachés
RAP	Reconnaissance Automatique de la Parole
DTW	Dynamic Time Warping
WER	Word Error Rate
WRR	Word Recognition Rate
CMC	Chaine de Markov Caché
CUED	Cambridge University Engineering Departement
HMM	Hidden Markov Model
ADN	Acide Désoxyribonucléique

Table des matières

Introduction générale……...1

Chapitre I : Généralités……4

I-2 Production de la parole…….....5

I-3 Caractéristiques phonétiques…….....7

I 3-1 Phonème…….7

I-4 Caractéristiques acoustiques……...9

I-5 Analyse et codage de la parole…….....12

I-5-1 Analyse spectrale……..12

I-5-2 Bancs de filtres passe bandes……14

I-5-3 Prédiction linéaire……15

I-5-4 Analyse cepstrale…….....16

I-6 Conclusion…….19

Chapitre II Les techniques de la reconnaissance de la parole……...20

II-1 Introduction…….21

II-2 Définition……21

II-3 Contraintes…….....22

II-4 Les types de taches de reconnaissance……...23

II-5 Les approches utilisées en RAP…….....23

II-5-1 l'approche acoustique phonétique…….....23

II-5-2 l'approche par reconnaissance de formes……...24

II-5-3 la programmation dynamique(DTW)…….....24

II-5-4 l'approche d'intelligence artificielle……...25

II-6 Performance des systèmes de reconnaissance de la parole……25

II-7 Conclusion…….....26

Chapitre III les Modèles de Markov Cachés……...27

III-1 Introduction……...27

III-2 Principes généraux des MMC……...27

III-3 Reconnaissance des mots isolés……28

III-4 Spécification de la probabilité d'émission……32

III-5 Probabilités Forward/Backword…….....33

III-6 Réestimation des paramétres(algorithme de Baum-Welch)…….....34

III-7 Reconnaissance et décodage de l'algorithme de Viterbi…….....35

III-8 Conclusion…….....36

Chapitre IV Résultat…….

IV-1 Introduction……...37

IV-2 Description de la base de données utilisée…….....37

IV-3 Description de l'outil de reconnaissance (HTK)……...37

IV-4 Prétraitement…….....38

IV-5Choix de la topologie des modèles (prototypes)…….....40

IV-6 Initialisation des modèles…….....43

IV-7 Apprentissage…….....44

IV-8 Reconnaissance…….....47

IV-9 Tests de robustesse et interprétations……..49

IV-9-1 Reconnaissance en l'absence de bruit……...49

IV-9-2 reconnaissance en milieu bruité…….51

IV-10 Conclusion……...53

Conclusion générale……...54

Annexe…….....55

Table des Figures

Figure (I-1) : l'appareil phonatoire…….6

Figure (I-2) : exemple de segmentation d'un signal en segments : Silence (s), Voisé (v) et Fricatif (u)……...9

Figure (I-3) : enregistrement numérique d'un signal acoustique……...10

Figure (I-4) : audiogramme des signaux représentant les deux mots : « Parenthèse » (En haut) et « effacer » (en bas)…….....10

Figure (I-5) : exemple de son voisé (haut) et non-voisé (bas)…….....11

Figure (I-6) : du signal vers son spectre lissé…….....13

Figure (I-7) : spectrogramme d'un son voisé…….....14

Figure (I-8): spectrogramme d'un son non voisé…….....14

Figure (I-9) : analyse par bancs de filtres…….....15

Figure (I-10) : détermination de la fréquence fondamentale…….....17

Figure (I-11) : chaine de calcul des coefficients MFCC…….18

Figure (II-1) : modèle de base de la reconnaissance de la parole……...21

Figure (II-2) : créneaux de la reconnaissance vocale…….....22

Figure (III-1) : exemple de modèle générateur de Marko……..29

Figure (III-2) : Utilisation des MMC dans la reconnaissance…….....32

Figure (IV-1) : fonctionnement de HCOPY……38

Figure (IV-2) : topologie choisie pour les MMC……....40

Figure (IV-3) : fonctionnement de (hinit)……...43

Figure (IV-4) : fonctionnement de (hrest)……...44

Figure (IV-5) : le réseau de connexion entre mot……46

Figure (IV-6) : illustration de tout le processus du codage à la reconnaissance……49

Figure (IV-7) : Processus de reconnaissance hybride…….50

Figure (IV-8) : taux d’erreur (WER) pour la reconnaissance audio, vidéo et Audiovisuelle Pour différents niveau de bruit……....52

Liste des Tableaux

Tableau (I-1) : les phonèmes en anglais……..8

Tableau (II-1) : les contraintes des systèmes de reconnaissance……22

Tableau (IV-1) : résultats audio et vidéo en l'absence de bruit……...50

Tableau (IV-2) : résultats obtenus pour différents niveaux de bruit……...52

Introduction Générale

L'usage de la parole, définie comme l'utilisation qu'un sujet fait de sa langue, semble être au cœur même de la nature humaine. Nos pensées se formulent en mots. La langue apparaît comme un système cohérent de signes linguistiques, vocaux, graphiques ou gestuels, attachés en propre à différentes communautés humaines, qui permet la communication entre les individus. La parole est douée d'un aspect foncièrement relationnel ; elle ne se conçoit pas indépendamment des conditions d'écoute. La parole ne se limite donc pas à l'utilisation de mots : elle met en jeu des signes linguistiques en même temps que des éléments spectaculaires (des ensembles d'éléments qui s'offrent au regard).

La reconnaissance automatique de la parole est un domaine de la science ayant toujours eu un grand attrait auprès des chercheurs comme auprès du grand public. En effet, qui n'a jamais rêvé de pouvoir parler avec une machine ou, du moins, piloter un appareil ou un ordinateur par la voix. Ne plus avoir à se lever pour allumer ou éteindre tel ou tel appareil électrique, ne plus avoir à taper pendant des heures sur un clavier pour rédiger un rapport (par exemple). L'homme étant par nature paresseux, une telle technologie a toujours suscité chez lui une part d'envie et d'intérêt, ce que peu d'autres technologies ont réussi à faire.

En dépit des progrès récents en reconnaissance automatique de la parole, les avantages de la reconnaissance automatique de la parole sont tellement importants que l'on trouve déjà sur le marché des dispositifs d'utilisation limitée, mais néanmoins efficaces. Citons certaines applications qui ont déjà vu le jour : saisie vocale de données, chambre d'hôpital avec possibilités de commandes vocales pour le malade, commande vocale de robot, commande vocale d'une montre portable ou téléphones portables, etc

La mise au point d'un système de reconnaissance automatique de la parole rencontre des difficultés comme : les variations intra-locuteur et les variations inter-

locuteur. De telles variations sont dues à la complexité physique de l'appareil vocal qui présente une soixantaine de muscles pouvant participer à la phonation.

Les variations intra-locuteur concerne un même locuteur, qui tout en prononçant un même mot peut parler de différentes façons d'une fois a l'autre.

Les variations inter-locuteur sont dues aux dimensions et proportions de l'appareil vocal variables selon l'âge, le sexe et la taille, etc

Ces systèmes actuels restent sensibles à l'environnement et au bruit et au style d'élocution. Plusieurs solutions ont été proposées pour accroître la robustesse des systèmes. Cependant, ces techniques ne sont pas efficaces pour un signal fortement dégradé. Une autre approche du problème consiste à utiliser d'autres sources d'information que le signal lui même. En particulier, les informations visuelles, comme le mouvement des lèvres, semblent présenter une bonne piste dans la mesure où ces informations sont fortement corrélées au signal de parole et moins sensibles aux bruits et à l'environnement. Il a également été montré [15] que les auditeurs humains s'appuient sur l'information visuelle pour reconnaître la parole.

Le présent travail s'inscrit justement dans cette perspective, c'est à dire l'utilisation d'une modalité supplémentaire (modalité visuelle représentant la forme des lèvres) pour améliorer les performances du système de reconnaissance.

Organisation du document

Le présent document n'a nullement la prétention d'être un ouvrage de traitement de la parole. De ce fait, seules les notions de bases nécessaires à l'exposition de notre travail seront exposées. Notre mémoire sera donc organisé de la manière suivante :

Le **premier chapitre** expose brièvement les notions générales de base sur la parole notamment : le mécanisme de production de la parole, ses caractéristiques phonétiques et acoustiques, les méthodes utilisées pour l'analyse et le codage de la parole. On terminera par une conclusion

Le **deuxième chapitre** expose les principales approches utilisées dans le domaine de la reconnaissance. Nous essayons également dans ce chapitre de tracer sommairement une chronologie illustrant le développement de ce domaine à travers les années. Il importe de noter ici que nous avons volontairement séparé ce chapitre du précédent pour deux raisons importantes : la première est d'alléger le premier chapitre rendant ainsi sa lecture plus agréable ; la deuxième est de permettre de mettre plus d'accent sur les techniques de reconnaissance. Ce chapitre sera clôturé par une conclusion.

Notre approche adoptée étant basée sur les **Modèles de Markov Cachés(MMC)**, le **troisième chapitre** leur sera entièrement dédié. L'aspect théorique des **MMC** ainsi que leur utilisation dans le domaine de la reconnaissance seront abordés. Ce chapitre sera clôturé par une conclusion.

Le **quatrième chapitre** expose en détail les travaux de reconnaissance réalisés dans des ambiances acoustiques à différent niveaux de bruis afin de mettre en évidence l'apport positif, en terme de performance de reconnaissance, de l'approche adoptée. Ce chapitre sera clôturé par une conclusion.

Enfin nous terminerons ce mémoire par une **conclusion générale** récapitulative de ce qui a été fait dans ce travail et de ce qui pourra être fait en perspective.

Chapitre I : Généralités

I-1 Introduction

La **parole** est l'un des moyens les plus naturels par le quel des personnes communiquent. La **reconnaissance de la parole** a pour objet la transformation du signal acoustique en une séquence de mots qui, idéalement, correspond à la phrase prononcée par un locuteur.

L'extraction des caractéristiques du signal sonore est délicate du fait de la propriété temporelle du signal. Cet aspect temporel du signal sonore est crucial pour la compréhension de celui-ci. Classiquement dans le domaine de l'analyse de la parole, notamment en ce qui concerne la **reconnaissance de la parole**, il est supposé que le signal de la **parole** peut être considéré comme stationnaire pour des durées de l'ordre de 20 à 30 millisecondes. De ce fait, un signal continu de parole est généralement segmenté en des unités temporelles de 20 à 30ms, durées dans lesquelles le signal est quasi stationnaire. Des caractéristiques du signal sont ensuite extraites de chaque unité temporelle. Mais la **parole** ne se réduit pas uniquement à du son transmis entre la bouche d'un locuteur et l'oreille de celui qui le reçoit.

La chaîne de production de la **parole** est un système complexe mettant En œuvre un ensemble d'articulateurs dont certains sont invisibles car placés à l'intérieur du conduit vocal, et d'autres visibles tels ceux engendrant les mouvements faciaux, principalement les lèvres. Des gestes de la main, peuvent aussi venir en appoint du mouvement des lèvres. La **parole** est donc multimodale et met en éveil les sens du système de perception de la **parole** qui sait recruter non seulement l'audition, mais aussi la vision. Bref dans ce premier chapitre nous tenterons de montrer quelle est la place de la phonation et de la vision dans la perception de la **parole**.

I-2 Production de la parole

La figure (I-1) montre l'appareil phonétique [1] :

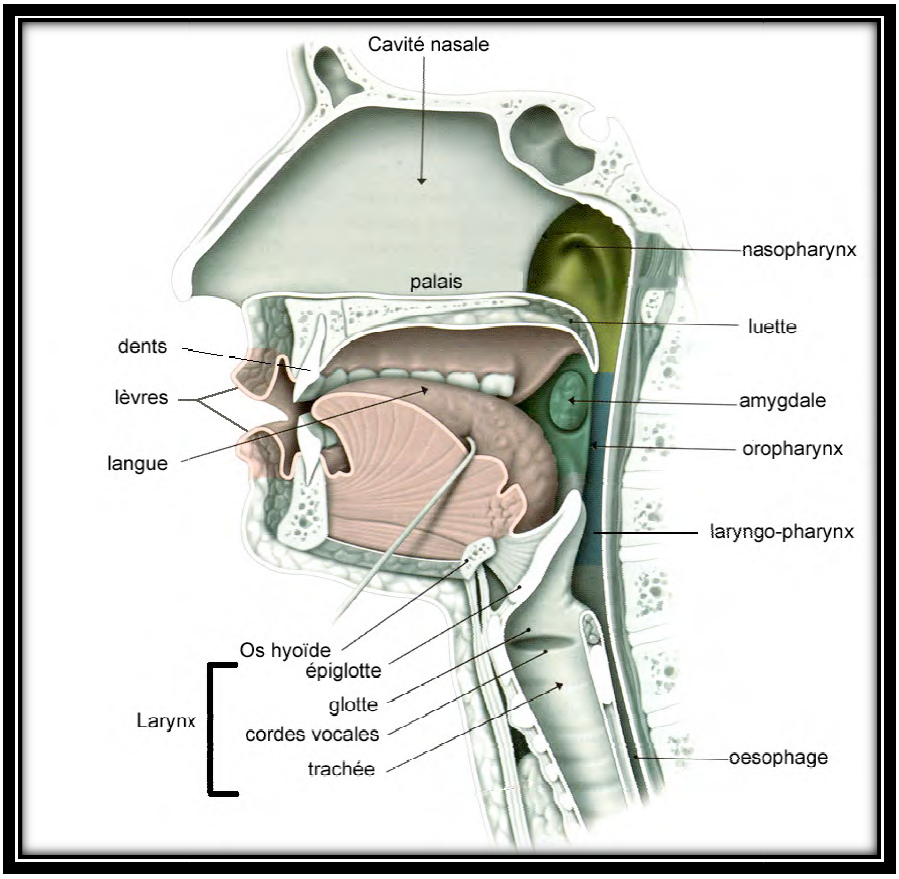


Figure (I-1) : l'appareil phonatoire

Les parties du corps impliquées dans la production de la **parole** sont :

- **Les poumons** et la cage thoracique, qui sont la source de l'air qui stimulera le système.
- **La trachée**, qui conduit l'air des poumons aux cordes vocales.
- **Les cordes vocales**, qui vibrent et modulent le flot d'air.
- **Le canal vocal**, qui raffine la production du son.

La **parole** peut être décrite comme le résultat de l'action volontaire et cordonnée d'un certain nombre de muscles. 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 à la production de sons, en poussant de l'air à travers 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. 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 progressivement, déterminer une ouverture triangulaire appelée **glotte**. L'air y passe librement pendant la respiration et la voix chuchotée, ainsi que pendant la phonation des sons **non-voisés** [2].

Les sons **voisés** résultent au contraire d'une vibration périodique des cordes vocales [3]. Le larynx est d'abord complètement fermé, ce qui accroit la pression en amont des cordes vocales, et les force à s'ouvrir, ce qui fait tomber la pression, et permet aux cordes vocales de se refermer ; des impulsions périodiques de pression sont ainsi appliquées au conduit vocal, composé des cavités pharyngienne et buccale ou nasale pour la plupart des sons.

I-3 Caractéristiques phonétiques

I -3-1 Phonème

La plupart des langues naturelles sont composées à partir de sons distincts, les **phonèmes** [3]. Un **phonème** est la plus petite unité présente dans la **parole**. Le nombre de **phonèmes** est toujours très limité (normalement inférieur à cinquante) et ça dépend de chaque langue, que l'on peut séparer en trois grandes catégories principales :

- **les voyelles** sont produites par les vibrations des cordes vocales. Ces sons sont souvent considérés comme quasi-périodiques. Elles peuvent être **nasales** ou **orales** selon que l'air passe par la cavité nasale ou la cavité buccale.
- **les consonnes** sont quand à elles produites par occlusion totale (**consonnes occlusives**) ou partielle (**consones fricatives**) du conduit vocal. Elles peuvent être **non voisées** : il n'y a alors pas de vibration des cordes vocales et le son est essentiellement produit par un bruit (bruit de friction, d'explosion ou de relâchement) ou au contraire **voisées**.

- **Les semi-voyelles** quant à elles, combinent certaines caractéristiques des **voyelles** et des **consonnes** [3].

IPA Phoneme	ARPAbet	Example	IPA Phoneme	ARPAbet	Example
/i/	IY	beet	/ŋ/	NX	sing
/I/	IH	bit	/p/	P	pat
/ɚ/	AXR	butter	/t/	T	ten
/ɛ/, /e/	EH	bet	/k/	K	kit
/æ/	AE	bat	/b/	B	bet
/a/	AA	Bob	/d/	D	debt
/ʌ/	AH	but	/g/	G	get
/ɔ/	AO	bought	/h/	HH	hat
/o/	OW	boat	/f/	F	fat
/ʊ/	UH	book	/θ/	TH	thing
/u/	UW	boot	/s/	S	sat
/ə/	AX	about	/ʃ/, /sh/, /š/	SH	shut
/ɨ/	IX	roses	/v/	V	vat
/ɝ/	ER	bird	/ð/	DH	that
/e^y/	EY	bait	/z/	Z	zoo
/a^w/	AW	down	/ʒ/, /zh/, /ž/	ZH	azure
/a^y/	AY	buy	/tʃ/, /č/	CH	church

Tableau (I-1) : les phonèmes en anglais [3]

En analysant l'onde ou le spectre du signal, on tentera de retrouver les propriétés du son (vibration des cordes vocales, périodicité, etc.).Une tâche simple est d'analyser l'onde d'un signal pour classifier les régions selon trois types de sons (figure(I.2)) [3] :

- **Voisé (V)** : le canal vocal est ouvert, le signal est quasi périodique.
- **Fricatif (U)** : le canal vocal est fermé en un point, le signal est plein de turbulences.
- **Silence (S)** : il n'y a aucun bruit, ou encore seulement le bruit de fond.

Les sections voisées sont faciles à identifier de par leur forme périodique.

Les sections fricatives sont plus difficiles à identifier avec précision en raison

De leur similarité avec le bruit de fond. Ce faisant, il sera très difficile de déte-

rminer avec grande précision le début et la fin d'un mot.

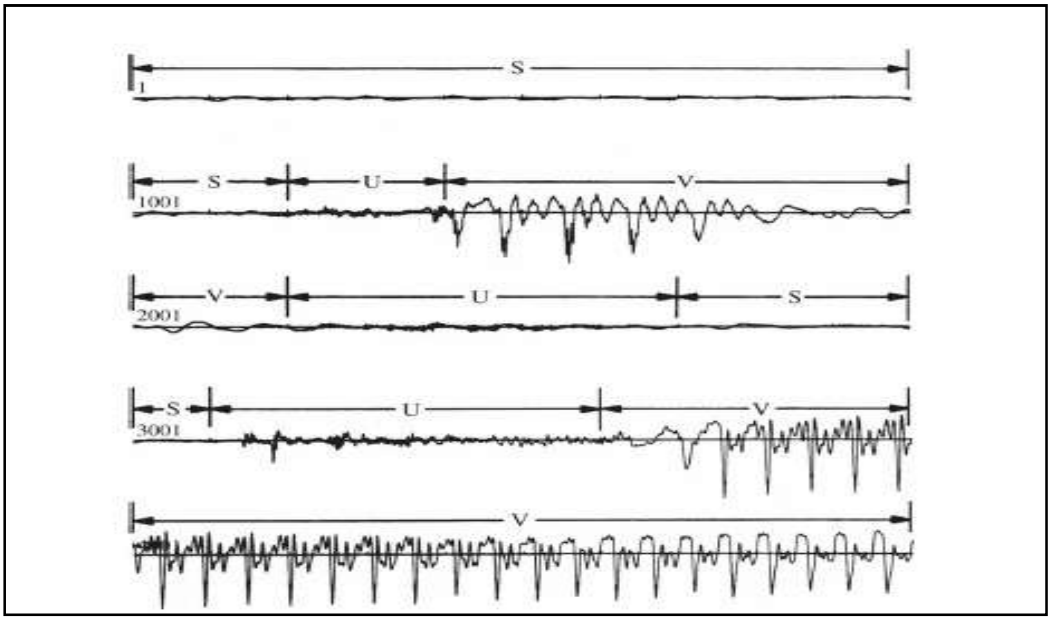


Figure (I-2) : exemple de segmentation d'un signal en segments

I -4 Caractéristiques acoustiques [4]

La **parole** apparait physiquement comme une variation de la pression d'air causé et émise par le système articulatoire. En le transformant dans un premier temps en signal électrique grâce au transducteur approprié : le microphone.de nos jours, le signal électrique résultent est le plus souvent numérisé.

L'opération de **numérisation**, schématisée à la figure (I.3) requiert successivement : un filtrage de garde (filtre anti-repliement), un **échantillonnage**, et une **quantification**.

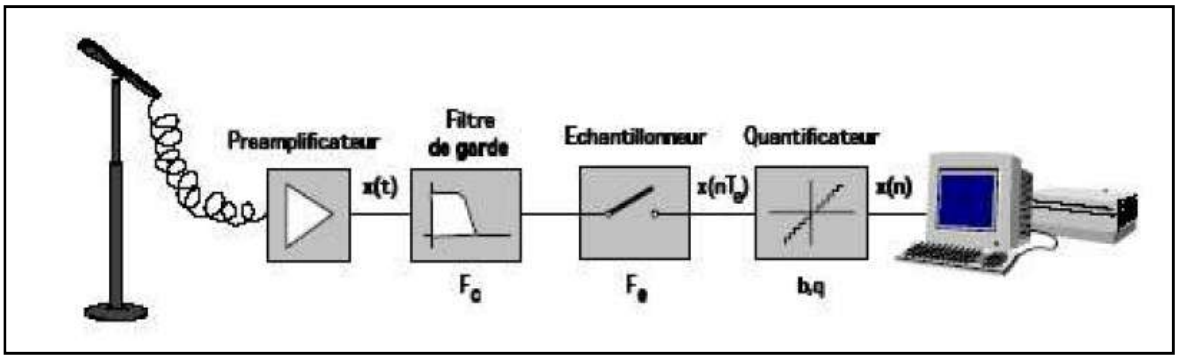


Figure (I-3) : enregistrement numérique d'un signal acoustique [4]

L'échantillonnage transforme le signal à temps continu **x(t)** en signal à temps discret **x(nte)** défini aux instants d'échantillonnage, multiple entiers de la période d'échantillonnage**(te)** ; pour ce qui concerne le signal vocal, son spectre peut s'étendre jusqu'à 12 khz. Il faut donc en principe choisir une fréquence **(fe)** égale à 24 khz au moins pour satisfaire raisonnablement au théorème de Shannon.

La **quantification** ne retient qu'un nombre fini $\mathbf{2^b}$ de valeurs (**b** étant le nombre de bits de quantification), espacées du pas de quantification **q**, le signal numérique résultant est noté **x(n).**

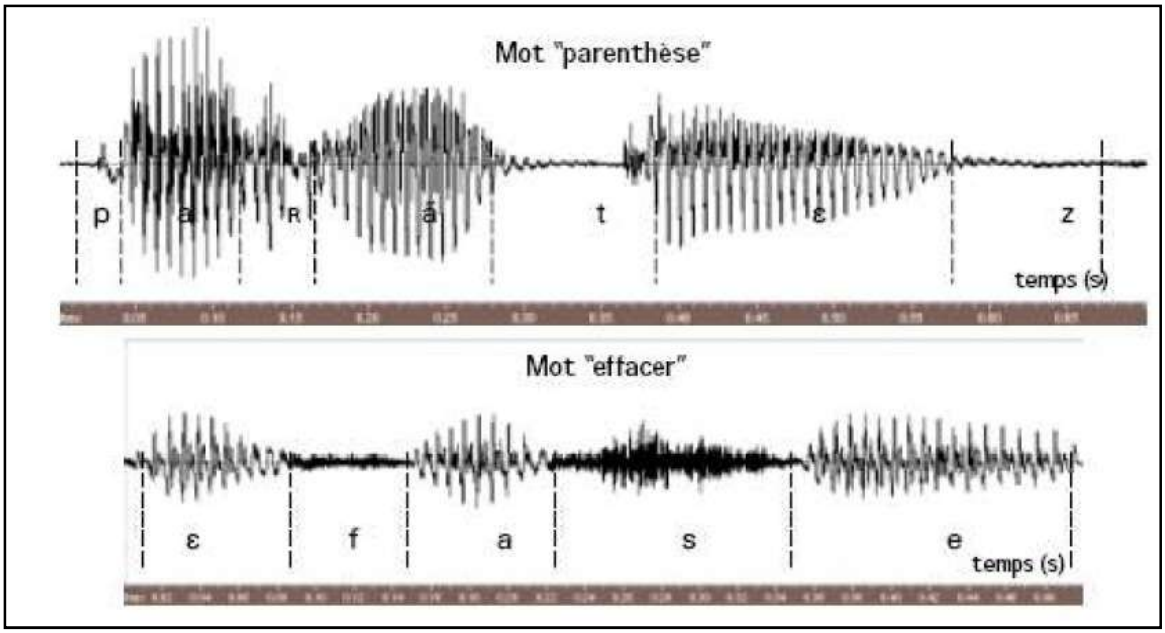


Figure (I-4) : audiogramme des signaux représentant les deux mots :

« Parenthèse »(en haut) et « effacer » (en bas) [4]

La figure (I.4) représente l'évolution temporelle, ou **audiogramme**, du signal vocal pour les mots « parenthèse », et « effacer » [4]. On y constate une alternance de zones assez périodiques et de zones bruitées, appelées zones **voisées** et **non-voisées**.

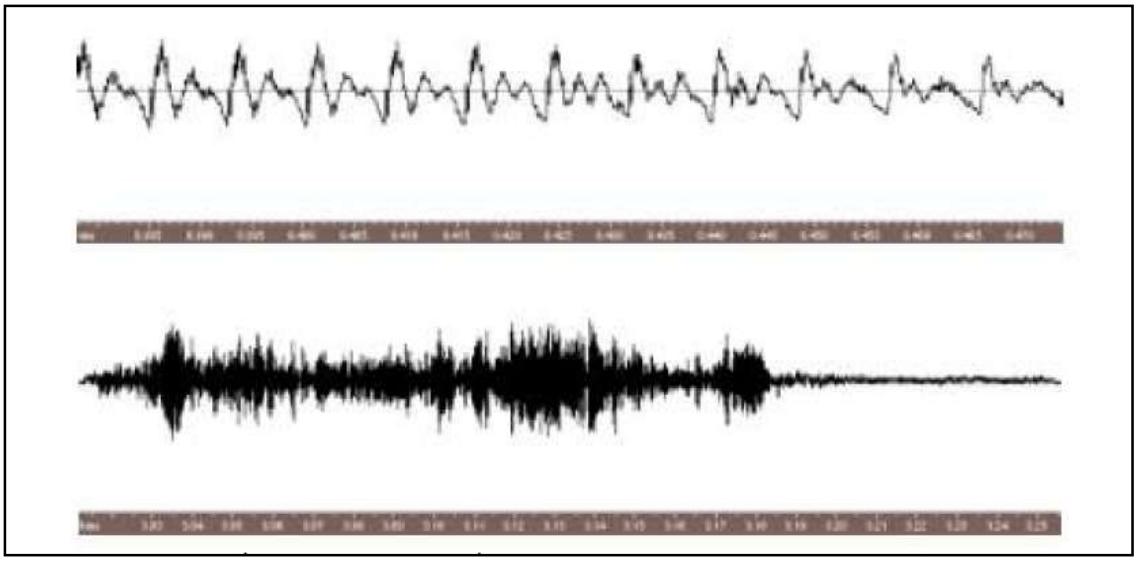


Figure (I-5) : exemple de son voisé (haut) et non-voisé (bas) [4]

La figure (I.5) donne une représentation plus fine de tranches de signaux **voisés** et **non-voisés [**4]. L'évolution temporelle ne fournit cependant pas directement les **traits acoustiques** (transformations fréquentielles) du signal. Mais quels sont ces traits ?

Le premier trait est la **fréquence fondamentale F_0** fréquence de vibration de cordes vocales : ce mouvement vibratoire correspond à une succession de cycles d'ouverture et de fermeture de la glotte. Le nombre de ces cycles par seconde correspond à la **fréquence fondamentale F_0**, elle dépend essentiellement :

- De l'âge et du sexe du locuteur
- Du locuteur

Ses valeurs standards [5] :

- 100 à 150 Hz pour l'homme adulte
- 140 à 240 Hz pour la femme adulte

Et elle peut présenter des variations considérables chez un même locuteur :

- selon le type de phrase prononcée
- selon l'état émotif et l'attitude du locuteur

Ses variations au court du temps définissent le **pitch** qui constitue la perception de la hauteur (où les sons s'ordonnent de grave à aigu). Seuls les sons quasi périodiques (**voisés**) engendrent une sensation de hauteur tonale bien définie [6].

Le deuxième trait est le **spectre fréquentiel** dont dépend principalement le **timbre** de la voix. Ce spectre résulte du filtrage dynamique du signal Glottique (impulsions, bruit, ou combinaison des deux) par le conduit vocal. Le **timbre** est une caractéristique permettant d'identifier une personne à la simple écoute de sa voix [6].

Le dernier trait acoustique est **l'énergie** correspond à l'intensité sonore qui est liée à la pression de l'air en amont du larynx [6].

I -5 Analyse et codage de la parole

Le signal de **parole** est un processus aléatoire non-stationnaire à long terme, mais il est considéré comme stationnaire dans des fenêtres temporelles d'analyse de l'ordre de 20 à 30ms. Cette propriété de stationnarité à court terme permet donc une analyse et modélisation progressive du signal de **parole** accompagnée, bien sur, d'un chevauchement de fenêtres pour permettre une continuité temporelle des caractéristiques de l'analyse et du modèle [6].

I -5-1 Analyse spectrale

- **Transformée de Fourier Discrète à Court Terme**

La **Transformée de Fourier à Court Terme** est obtenue en extrayant de l'audiogramme une trame d'une trentaine de (ms), en pondérant ces échantillons par une fenêtre de pondération (souvent une fenêtre de **Hamming**)

La fenêtre de **Hamming** est donnée par la formule suivante [7]:

$$h(n) = \begin{Bmatrix} 0.54 - 0.46\ cos\left(2\pi \frac{n}{N-1}\right)\ si\ 0 \le n \le N-1 \\ 0\ sinon \end{Bmatrix} \qquad (1-1)$$

et en effectuant une **Transformée de Fourier** sur ces échantillons, Cette analyse simplifie le signal **parole** et modélise le fait que l'oreille humaine réalise une sorte d'analyse spectrale en décomposant le signal **parole** temporel en une suite de spectres de fréquence(figure (I-6)) [4].

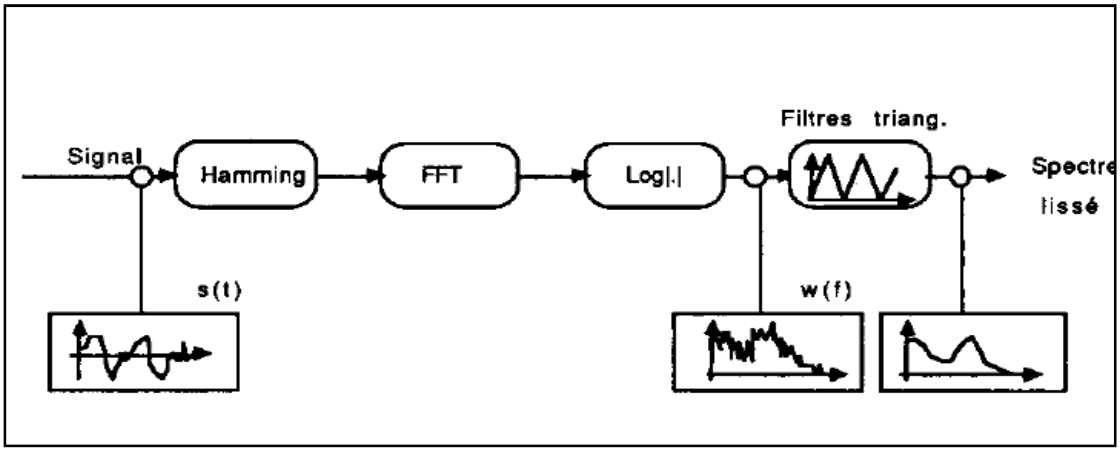


Figure (I-6) : du signal vers son spectre lissé [7]

Soit x[n] un signal numérique sur l'intervalle [0, (N-1)Te]. Sa **Transformée de Fourier Discrète à Court Terme** sera [8]:

$$x(f) = \sum_{n=0}^{N-1} h(\eta - n)x[nTe]\, e^{-j2\pi fn} \qquad (1-2)$$

Où h (η) est un signal fenêtre(**Hamming**) de longueur η.

- **Spectrogramme**

La **Transformé de Fourier à Court Terme** permet d'obtenir une représentation temporelle et fréquentielle simultanée pour un signal : **le spectrogramme**. L'amplitude du spectre y apparait sous la forme de niveaux de gris [8].

- Une fenêtre plus large permettra d'identifier **les sons voisés**. Ceux-ci Seront caractérisés par des rayures verticales.

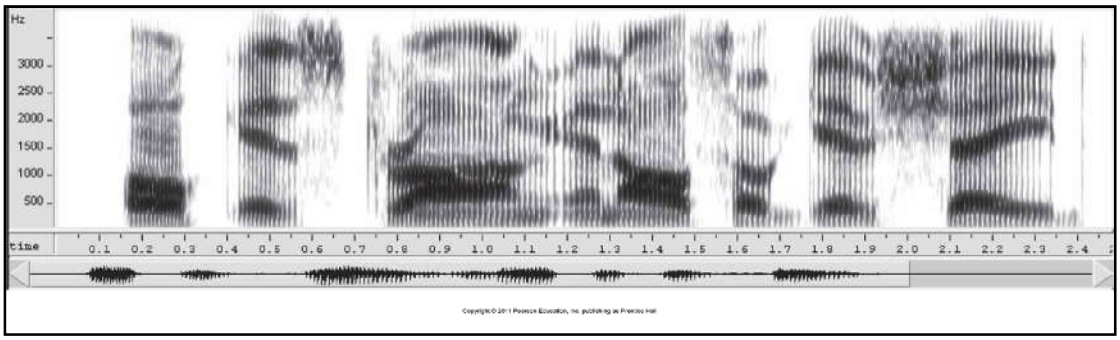


Figure (I-7) : spectrogramme d'un son voisé

- Une fenêtre plus courte permettra d'identifier **les sons fricatifs**. Ceux-ci Seront caractérisés par des régions remplies en raison d la nature aléatoire de Leur onde.

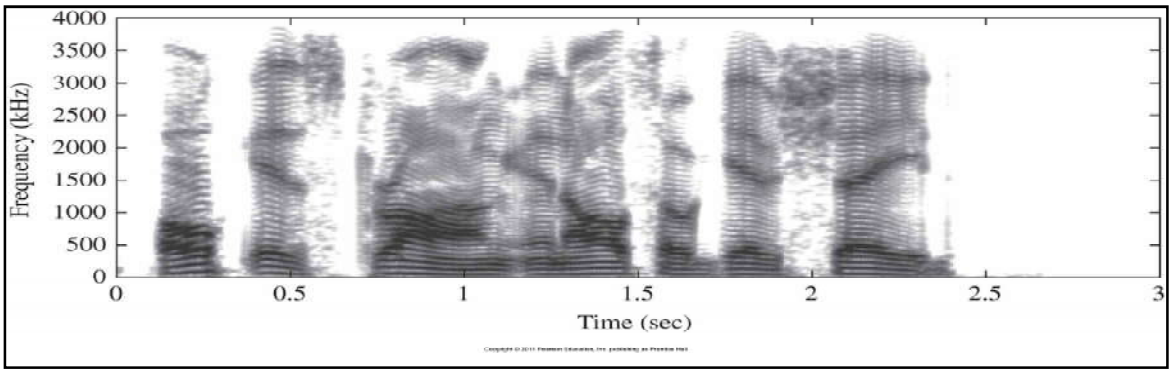


Figure (I-8): spectrogramme d'un son non voisé

Les figures (I.7) et (I.8) [4] illustrent la **Transformée de Fourier** d'une tranche **voisée** et d'une tranche **non-voisée** respectivement. Les parties **voisées** du signal apparaissent sous la forme de successions de pics spectraux marqués et des creux correspond aux résonnances et aux antis résonnances du conduit vocal et sont appelés **formants** et **anti formants**, par contre, le spectre d'un signal non voisé ne présente aucune structure particulière. La forme générale de ces spectres, appelée **enveloppe spectrale**.

I -5-2 Bancs de filtres passe bandes

Le signal électrique obtenu à la sortie du microphone (après avoir été amplifié et pré accentué) passe à travers un banc de filtres passe-bande. Le signal à la sortie de ces filtres passe ensuite a travers un circuit détecteur

d'enveloppe, On obtient, ainsi, une approximation du module de la **TFCT** à court terme dans différentes bandes de filtres.

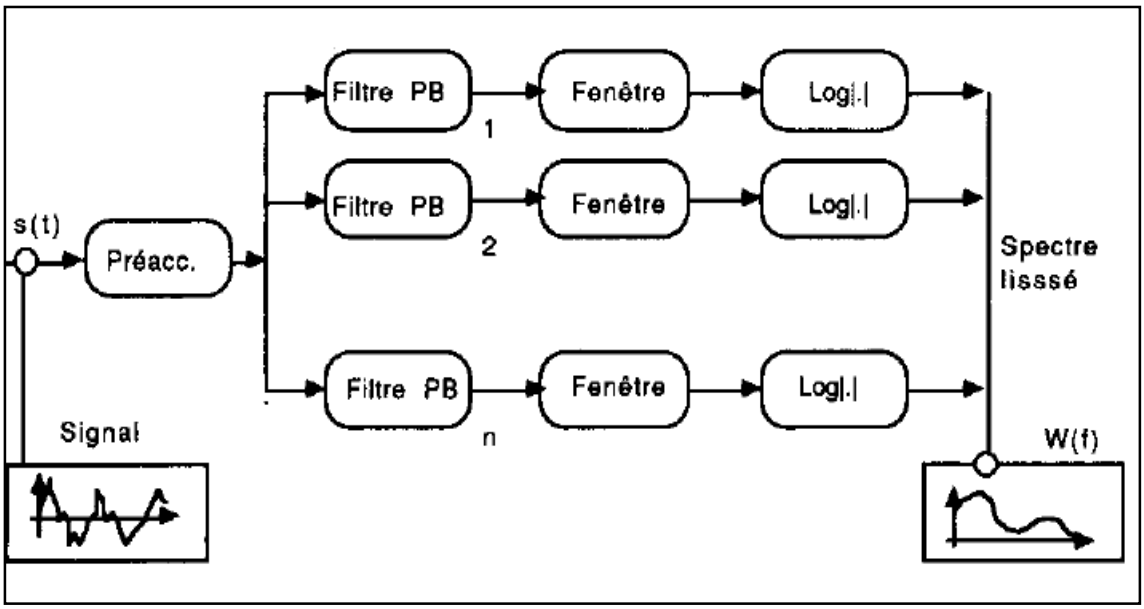


Figure (I-9) : analyse par bancs de filtres [5]

I-5-3 Prédiction linéaire

Dans l'analyse par prédiction linéaire **LPC** [6], le conduit vocal est modélisé par une fonction de transfert qui suit un modèle autorégressif. Cette analyse est fortement utilisée dans le codage de parole dans le but de réduire la redondance du signal vocal, ou pour extraire des paramètres pertinents pour la reconnaissance de la **parole**. L'estimation des coefficients de la fonction de transfert du conduit vocal est faite en supposant connaitre le signal d'excitation. Pour les sons **non voisés**, le signal d'excitation est un bruit blanc de moyenne nulle et de variance unité. Pour les **sons voisés**, cette excitation est une suite d'impulsions d'amplitude unité. La fonction de transfert du conduit vocal dans le domaine Z est donnée par :

$$H(z) = \frac{S(z)}{U(z)} = \frac{G}{1-A(z)} \qquad (1\text{-}3)$$

Où $A(z) = \sum_{k=1}^{p} a_k z^{-k}$ est le prédicateur linéaire, $a_k$sont les coefficients de prédiction, S(z) est le signal de **parole** produit en sortie, U(z) est le signal d'excitation et G est un gain. Le signal de **parole** s(n) à la sortie du modèle est donc représenté par la somme d'une combinaison linéaire des échantillons précédents et de la fonction d'excitation, tel que :

$$S(n) = \sum_{k=1}^{p} a_k s(n-k) + Gu(n) \quad (1\text{-}4)$$

Il permet de représenter la **parole** juste par ses paramètres pertinents, sans redondance. Signalons également que les coefficients sont choisis de façon à minimiser l'erreur quadratique de prédiction sur chaque segment de la fenêtre d'analyse.

I-5-4 Analyse cepstrale

Les éléments d'un vecteur spectral (contenant les spectres d'énergie des fréquences) ne sont pas indépendants et il existe par la suite une redondance d'information. Le cepstre est une manière d'obtenir une version non-corrélée du vecteur spectral corrélé.

Le **cepstre** est aussi une interprétation du modèle de production de la **parole**. On part de l'hypothèse que la suite S constituant un signal vocal est le résultat de la convolution du signal de la source par le filtre correspondant au conduit vocal :

$$s = u * h \quad (1\text{-}5)$$

Où s, u et h le signal de parole, le signal d'excitation (source) et la fonction de transfert du conduit vocal (filtre), Le but du cepstre est de séparer ces deux contributions (source et filtre) par application de la déconvolution à travers une **Transformée en Cosinus Discret**.

$$TFD(s) = UH \quad (1\text{-}6)$$

Le logarithme de l'amplitude transforme le produit de la (**TFD)** en somme. On obtient alors :

$$\text{Log } |S(\nu)| = \log |U| + \log |H| \qquad (1\text{-}7)$$

Par transformation en cosinus discret (**DCT**), on obtient le cepstre.L'expression du cepstre réel est donc :

$$C = \text{DCT } (\log (\text{TFD}(s))). \qquad (1\text{-}8)$$

Les coefficients cepstraux permettent la séparation des effets du conduit vocal et de la source périodique. En effet, la contribution du conduit vocal est représentée par l'enveloppe du spectre de fréquence et affecte en conséquence les premiers coefficients cepstraux. Les caractères plus fins du spectre de fréquence, qui représentent la source périodique, sont décrits par les coefficients cepstraux d'ordre supérieur. Il est possible de définir la **fréquence fondamentale** de la source u en détectant les pics périodiques (harmoniques) au-delà d'un certain nombre N de coefficients (Voire figure (I.10)). Cependant, déterminer la fréquence fondamentale d'un signal de parole reste encore un problème difficile. En effet, les algorithmes classiques manquent de robustesse quand le bruit est présent, quand la **fréquence fondamentale** change rapidement ou quand la valeur de celle-ci n'est pas assez élevée [6].

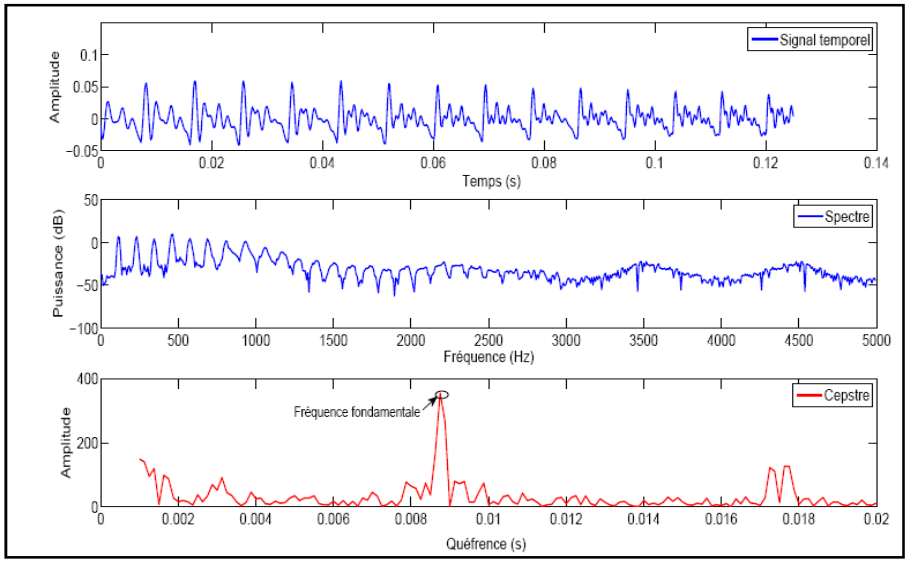


Figure (I-10) : détermination de la fréquence fondamentale

Le schéma de la figure(I.12) illustre le calcul des coefficients cepstraux.

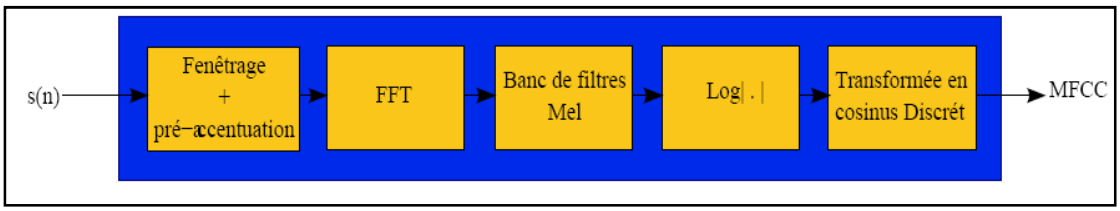


Figure (I-11) : chaine de calcul des coefficients MFCC

L'analyse cepstrale est basée sur un calcul de coefficients dits coefficients cepstraux de **Mel**, soit en abrégé **MFCC** [6]. Le calcul est en effet basé sur une échelle de **Mel**. Cette échelle se rapproche de la perception fréquentielle de l'oreille. L'idée est de moyenner le spectre dans des bandes de fréquences correspondant grossièrement au filtrage effectué par la membrane basilaire. L'´echelle **Mel** est approchée par un banc de 15 à 24 filtres triangulaires espacées linéairement jusqu''a 1 KHz, puis espacés logarithmiquement jusqu'aux fréquences maximales. Elle a été conçue de telle façon que 1000 Hz correspondent à 1000mels.

La formule de conversion Hertz en **Mel** [9] la plus utilisée est la suivante :

$$\text{Mel} = 2595 \log_{10}(1+\frac{fhz}{700}) \qquad (1\text{-}9)$$

La procédure de calcul pas à pas des **MFCC** est la suivante :

- Découpage en trame (stationnarité), chevauchement (éviter les transitions brusques de trame en trame).
- Préaccentuation (pour donner plus d'´energie et renforcer la contribution des hautes fréquences) avec un filtre passe-haut de la forme $1-0.9z^{-1}$ puis fenêtrage de Hamming (pour la continuité aux bords).
- Calcul de la **TFD** sur chaque trame.
- Filtrage par un banc de filtres triangulaires répartis le long de l'échelle de **Mel**.
- Calcul du logarithme du module de l'´energie en sortie du banc de filtres.

- Application de la **Transformée en Cosinus Discrète** inverse (joue le rôle d'une **TFD** inverse).
- Seuls les premiers coefficients sont conservés (12 dans notre travail) [6].

I-6 Conclusion

Ont été présentées dans ce chapitre quelques notions générales de base sur la **parole** ; qui serviront de plateforme pour l'exposition du deuxième chapitre, ce dernier sera consacré aux techniques de reconnaissance de la parole. L'accent sera notamment mis sur les **Modèles de Markov Cachés** (**MMC**).

Chapitre II : les techniques de la reconnaissance de la parole

II-1 Introduction

Ce chapitre est dédié à la reconnaissance automatique de la **parole**. Nous exposerons les principales approches utilisés dans ce domaine ainsi qu'une brève chronologie du développement de ce domaine à travers les années.

II-2 Définition

La reconnaissance automatique de la parole (**RAP**) consiste en la conversion d'un signal vocal en un message caractérisé par une suite de symboles ou de mots et ce en utilisant un algorithme implémenté sur machine. Ce processus est illustré dans la figure (II-1).

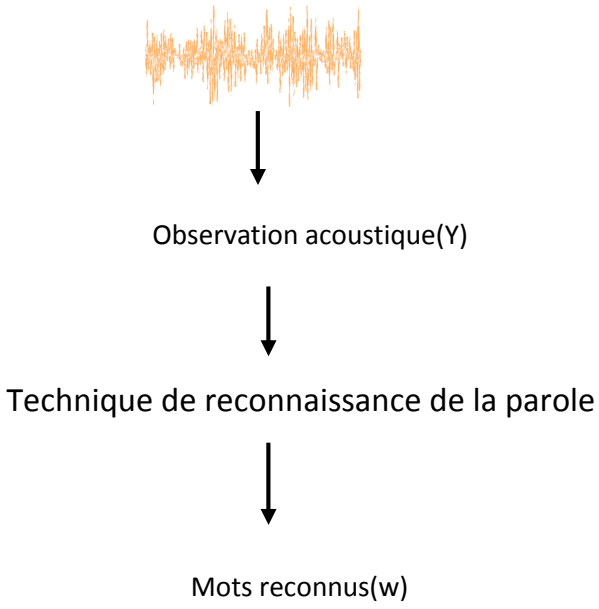


Figure (II-1) : modèle de base de la reconnaissance de la parole

La figure (II-2) illustre les créneaux possibles dans la reconnaissance vocale.

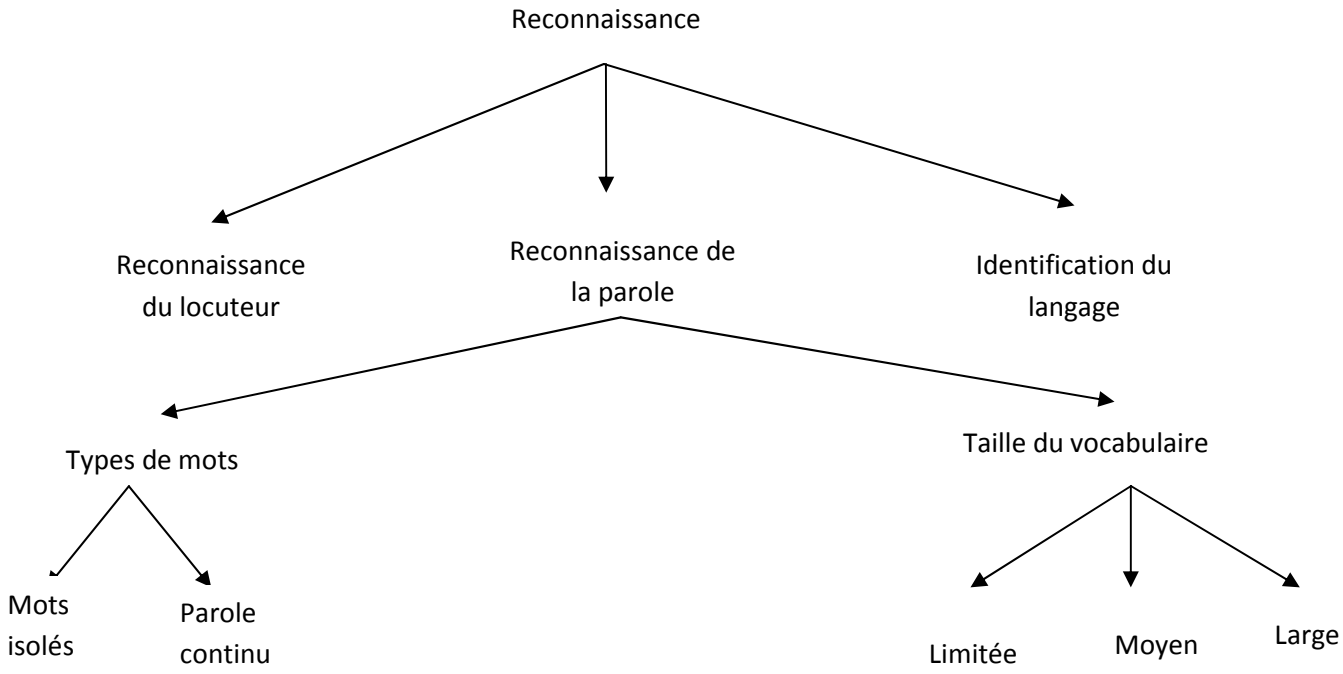


Figure (II-2) : créneaux de la reconnaissance vocale

II-3 Contraintes

Les principales contraintes dont dépend la performance d'un système de reconnaissance sont présentées dans le tableau (II-1)

Environnement	Type de bruit, rapport signal sur bruit, les conditions de travail
transducteur	Microphone, téléphone
Canal	L'amplitude de la bande, distorsion
Locuteur	Sexe, âge, état physique et psychique
Styles de parole	Mots isolés ou continus, vitesse (lente, normale, rapide)
Vocabulaire	Caractéristiques des données d'apprentissage disponibles

Tableau (II-1) : les contraintes des systèmes de reconnaissance

II-4 Les types de taches de reconnaissance

Elles sont citées par ordre de difficulté :

o **Reconnaissance de mots isolés :** chaque prononciation doit être suivie et précédée d'un silence. Ce type de système peut être dans l'un des deux états : écoute, non écoute. Le locuteur doit attendre entre deux prononciations ; le temps que le système effectue le traitement de la prononciation précédente.

o **Reconnaissance de mots connectés :** similaire au cas précédent avec la différence que le locuteur est autorisé à prononcer les différents mots avec un minimum de repos entre les prononciations.

o **Reconnaissance de parole continue :** Le locuteur peut parler d'une manière presque naturelle (dictée automatique).

o **Reconnaissance de parole spontanée :** Le locuteur peut même prononcer des mots concaténé (combinaison de deux mots). C'est le niveau le plus difficile à atteindre.

II-5 Les approches utilisées en RAP [3] [10]

Parmi les plus connues citons :

- Approche acoustique phonétique
- Approche par reconnaissance de formes
- Approche par intelligence artificielle
-

II-5-1 L'approche acoustique phonétique :

C'est la base de l'approche phonétique acoustique [10], qui postule qu'il existe des unités phonétiques distinctives finies (**phonèmes**) dans la langue parlée et que ces unités se caractérisent généralement par un ensemble de propriétés acoustiques qui se manifestent dans le signal de parole au fil du temps. Il est supposé dans l'approche acoustique-phonétique que les règles régissant la variabilité sont simples et peuvent être facilement appris par une machine.

- La première étape dans l'approche acoustique phonétique est une analy-

se spectrale de la parole combinée avec une fonctionnalité de détection qui convertit les mesures spectrales à un ensemble de caractéristiques qui décrivent les larges propriétés acoustiques des différentes unités phonétiques.

- La deuxième étape est une phase de segmentation et d'étiquetage dans l-aquelle le signal est segmenté en régions acoustiques stables suivies par une ou plusieurs étiquettes phonétiques à chaque région segmentée.
- La dernière étape de cette approche tente de déterminer un mot valide (ou une chaîne de mots) à partir de l'étiquette phonétique. L'approche acoustiq-ue phonétique n'a pas été largement utilisée dans la plupart des applications commerciales [10].

II-5-2 L'approche par reconnaissance de formes

Cette approche [3] comporte deux étapes essentielles, l'apprentissage et la comparaison. À partir des données d'apprentissage on obtient une repré-ésentation des mots par des modèles statistiques. On effectue ensuite une co-mparaison entres les signaux de paroles inconnus avec les modèles apprentis lors de la phase d'apprentissage afin de déterminer les mots prononcés. Cette approche est devenue la méthode prédominante de la reconnaissance de la **parole** dans les six dernières décennies.

II-5-3 La programmation Dynamique (DTW)

La **DTW** ou déformation temporelle dynamique est un algorithme pe-rmettant de mesurer la similarité entre deux séquences qui peuvent varier dans le temps ou la vitesse. **DTW** est un procédé permettant de rechercher une cor-respondance optimale entre deux séquences données avec certaines restrictrict-ions. **DTW** est un algorithme particulièrement adapté aux séquences avec un manque d'informations. Le procédé d'optimisation utilise la programmation dynamique, d'où le nom [10].

II-5-4 L'approche d'intelligence artificielle

L'approche par **Intelligence Artificielle** est une approche hybride combinant l'approche acoustique phonétique et l'approche par reconnaissance des formes. En cela, elle exploite les idées et les concepts de la phonétique acoustique et des méthodes de reconnaissance des modèles. La démarche d'un phonéticien, expert en lecture de spectrogrammes de **parole** consiste à décoder des spectrogrammes [11]. L'expertise se fait en deux parties :

- une analyse visuelle permettant d'extraire du spectrogramme les indices Acoustiques pertinents.
- un raisonnement contextuel à l'aide des indices détectés pour en déduire Les phonèmes prononcés.

Ces deux expertises sont intimement mêlées lors du décodage ; l'expert extrait d'abord visuellement quelques caractéristiques pertinentes puis il ébauche un raisonnement pour obtenir une première liste de phonèmes candidats. Ensuite il va chercher d'autres indices visuels pour confirmer ou infirmer ses premières hypothèses. Les réseaux de neurones peuvent être classés dans cette catégorie d'approche.

II-6 Performance des systèmes de reconnaissance de la parole

La performance des systèmes de reconnaissance vocale est habituellement spécifiée en termes de précision et de vitesse. La précision peut être généralement exprimée en taux d'erreur de mots (**WER**) tandis que la vitesse est mesurée avec le facteur de temps réel. Le **WER** est donné par :

$$WER = \frac{S + I}{N} \qquad (2-1)$$

Où :

S : est le nombre de substitutions (mots incorrectement reconnus).

D : est le nombre d'effacement (mots omis).

I : est le nombre d'insertions (mots ajoutés).

N : est le nombre des mots dans la référence.

Lors de la déclaration de la performance d'un système reconnaissance de la parole, parfois le taux de reconnaissance des mots (**WRR**) est utilisé au lieu:

$$WRR = 1 - WER = \frac{H-I}{N} \qquad (2-2)$$

Ou H est la N-(S + D), le nombre de mots correctement reconnus.

II-7 Conclusion :

Ont été exposées dans ce chapitre les principales approches utilisées dans la reconnaissance de la parole, la plus fameuse d'entre elles, la modélisation Markovienne, qui constitue le cadre de notre travail sera exposée dans le prochain chapitre.

Chapitre III : les Modèles de Markov Cachés

III-1 Introduction

Dans ce chapitre nous exposons les aspects théoriques et pratiques des Modèles de Markov Cachés.

Un modèle de Markov caché (**MMC**) ou chaîne de Markov cachée (**CMC**) connu sous le sigle **HMM** pour **Hidden Markov Model** est un modèle statistique dans lequel le système modélisé est supposé être un processus Markovien avec des paramètres inconnus. Le défi est donc de déterminer ces paramètres à partir d'autres paramètres observables. Puis, les paramètres extraits peuvent ainsi être employés pour réaliser une analyse telle que la reconnaissance acoustique ou visuelle.

III-2 Principes généraux des MMC

Les systèmes de reconnaissance automatique de la parole (**RAP**) assument généralement que le signal vocal est une réalisation d'un certain message codé sous la forme d'une séquence d'un ou de plusieurs symboles (mots). Ayant donc une prononciation donnée, il s'agit de faire le passage du signal vers le message (séquence de mots) correspondant.

Le rôle du système de reconnaissance sera donc de faire une correspondance (mapping) entre la séquence de vecteurs de paramètres et la séquence de symboles (mots) sous-jacente.

Deux facteurs rendent la tache de reconnaissance très difficile :

- La correspondance entre mots et signal n'est pas directe (un à un) à partir du moment où plusieurs séquence de mot peuvent générer le même signal **parole**. En plus il y a une grande variabilité dans la réalisation de la prononciation à cause des variabilités : interlocuteur, intra locuteur et de l'environnement.
- Les frontières entre les mots ne peuvent pas être déterminées explicitement à partir du signal vocal et donc il n'est pas possible de considérer le signal comme étant la concaténation de formes statiques.

Le second problème de la non connaissance des frontières des mots est contourné en réduisant la tache du système à la **reconnaissance de mots isolés**. Ceci implique que le signal vocal correspond à un seul symbole (mot) choisi dans un **vocabulaire** donné. Le vocabulaire représente l'ensemble des mots que le système est capable de reconnaitre.

C'est précisément dans ce cadre que notre travail se place.

III-3 Reconnaissance des mots isolés [9]

Considérons le signal issu d'un mot prononcé par un locuteur donné. Comme expliqué dans le premier chapitre, ce signal subira certains prétraitements (subdivision en fenêtres ou bien trames de 10 à 30 (ms), puis calcul de coefficients cepstraux pour chaque trame). Notre mot (O) sera finalement présenté sous forme d'une suite de vecteurs acoustiques ou bien vecteurs de paramètres (feature vectors) ou encore vecteurs d'observations :

$$O = O_1, O_2 \ \ldots\ldots, O_t, \ldots\ldots, O_T \qquad (3-1)$$

Avec O_t le vecteur de paramètre observé à l'instant t et O_T le dernier vecteur de paramètres (le mot contient donc T vecteurs). Dans ce qui suit, O sera donc appelée séquence d'observations.

Le nombre de coefficients extraits par trame de signal (par vecteur) sera noté n.

Le problème de reconnaissance de mots isolés revient à calculer :

$$arg\ \max_i\{P(W_i/O)\} \qquad (3-2)$$

Où W_i représente le i^{ème} mot du vocabulaire

Cette probabilité n'est pas calculable directement, mais en utilisant la formule de Bayes on obtient :

$$P(W_i/O) = P(O/W_i)P(W_i)/P(O) \qquad (3-3)$$

Donc pour un ensemble donné de probabilités à priori P(W_i), le mot prononcé le plus probable dépend uniquement de la vraisemblance P(O/W_i) .

Ayant la dimensionnalité de la séquence d'observation O, l'estimation directe de la probabilité conditionnelle jointe P ($O_1,O_2 \ldots O_T/W_i$) à partir d'exemples de mot prononcé n'est pas envisageable. Cependant si on dispose d'un modèle

paramétrique de production du mot, tel qu'un **MMC**, l'estimation à partir des données (exemples de mots) est possible puisque l'estimation de P (O/W_i) est remplacée par l'estimation des paramètres du **MMC**.

Dans la reconnaissance par **MMC** on considère que la séquence d'observations *O* correspondant à chaque mot prononcé est générée par un **MMC**. Un **Modèle de Markov** est une machine à nombre fini d'états qui change d'état une fois chaque unité de temps. La présence dans chaque état (j) est accompagnée de l'émission par cet état d'un vecteur d'observation O_t selon une densité de probabilité $b_j(O_t)$.

La transition d'un état (i) vers un état (j) est, elle aussi, probabiliste et est donnée par la probabilité discrète a_{ij}.

La figure (III-1) montre un exemple de **MMC** à 6 états. Dans cet exemple la séquence d'observations ($O=O_1O_2O_3O_4O_5O_6$) est générée en passant par les séquence d'états X=1,2,2,3,4,4,5,6 les états 1 et 6, représentant respectivement l'état initial et final, sont des états non émetteurs.

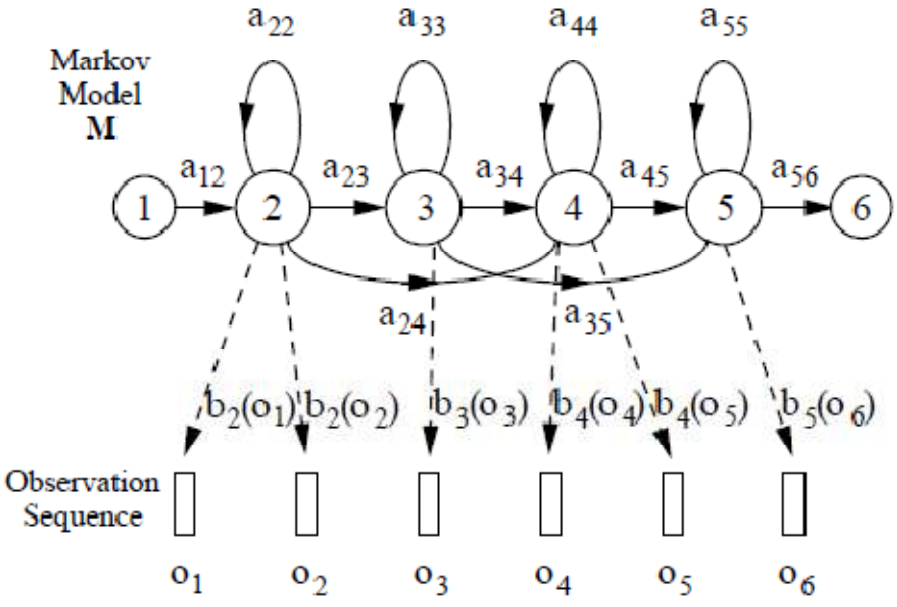


Figure (III-1) : exemple de modèle générateur de Markov

La probabilité jointe de génération de la séquence d'observations $O=O_1O_2O_3O_4O_5O_6$ par le modèle de notre exemple en passant par la séquence d'états X=1,2,2,3,4,4,5,6 est donnée par :

$$P(O,X/M) = a_{12}\, b_2(O_1)a_{22}b_2(O_2)a_{23}b_3(O_3)a_{34}b_4(O_4)a_{44}b_4(O_5)a_{45}b_5(O_6)a_{56} \qquad (3-4)$$

Il est facile de remarquer qu'il y a plusieurs manières de transiter entre les états (et donc plusieurs chemins) pour générer notre séquence d'observations *O*. En pratique, seule la séquence d'observation *O* est connue, la séquence d'états X sous-jacente est cachées d'où le nom de **Chaine de Markov Cachée**.

Etant donné que X est inconnue, la vraisemblance est calculée en sommant sur toutes les séquences d'états possibles.

$$X = X(1), X(2), \dots\dots . X(T)$$

C'est-à-dire :

$$P(O/M) = \sum_X a_{X(0)X(1)} \prod_{t=1}^{T} b_{X(t)}(O_t) a_{X(t)X(t+1)} \qquad (3-5)$$

Avec les contraintes : *X*(0) état initial et *X*(T+1) état final du modèle.

Une alternative à l'équation (3-6) peut être obtenue en considérant la séquence X (le chemin) la plus probable, soit donc :

$$\hat{P}(O/M) = \max_X \{a_{X(0)X(1)} \prod_{t=1}^{T} b_{X(t)}(O_t) a_{X(t)X(t+1)}\} \qquad (3-6)$$

Des méthodes itératives permettent un calcul efficace de (3-5) et (3-6)

Comme on l'a signalé lorsqu'on a évoqué l'équation (3-2), la résolution de cette équation équivaut à la résolution du problème de reconnaissance.

Etant donné un ensemble de modèles Mi correspondant aux mots W_i

L'équation (3-2) est résolue par l'utilisation de (3-3) et en assumant que

$$P(O/W_i) = P(O/M_i) \qquad (3-7)$$

Tout cela suppose que les paramètres du **MMC** $\{ a_{ij} \}$ et $\{ b_j(O_t) \}$ sont connus. C'est là que réside toute la puissance et l'élégance de la modélisation Markovienne.

En disposant d'un ensemble d'exemples d'apprentissage correspondant à un modèle particulier, les paramètres de ce modèle peuvent être déterminés automatiquement par une procédure efficace de ré estimation de paramètres. Ainsi donc, si l'on collecte un nombre suffisants d'exemples représentatifs de notre mot, on pourra donc construire un **MMC** correspondant qui modélisera implicitement les sources de variabilité inhérente au signal vocal réel.

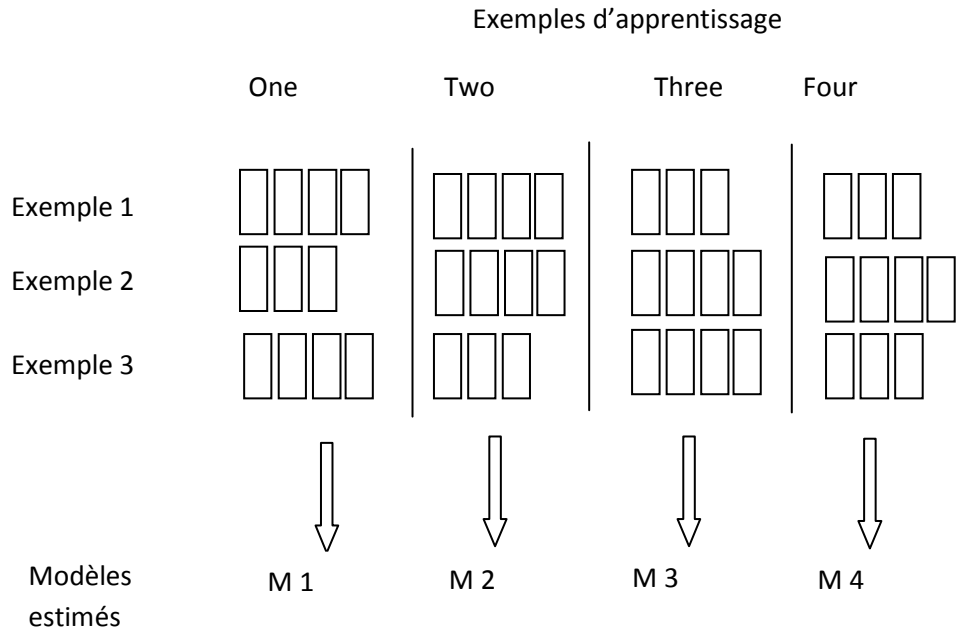
La figure (III-2) récapitule l'utilisation des **MMC** pour la reconnaissance de la parole. Nous allons nous placer dans le cadre de notre travail.

Le vocabulaire de notre système est composé de quatre(4) mots isolés représentant les chiffres (**one, two, three et four**) prononcés en anglais.

Etant donné que chaque mot est représenté par un **MMC** on doit disposer de quatre **MMC** (un **MMC** pour chaque mot)

- Premièrement chaque **MMC** est entrainé par un ensemble d'exemples du mot lui correspondant. On obtiendra à la fin de cette phase les modèles entrainés (M1, M2, M3 et M4)
- Deuxièmement ayant un mot inconnu, on calcule la vraisemblance de génération de ce mot par tous les **MMC**. On choisira le **MMC** ayant donné le maximum de vraisemblance ;

A : apprentissage



B : reconnaissance

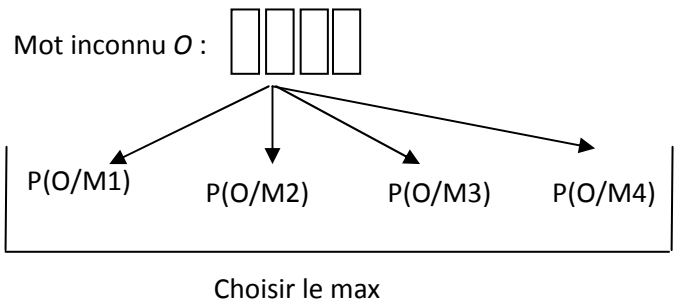


Figure (III-2) : Utilisation des MMC dans la reconnaissance

III-4 Spécification de la probabilité d'émission [9][12]

Selon que la loi de probabilité d'émission $b_x(O_t)$ spécifique à un état (x) est continue ou discrète on parle de **Modèles de Markov** Continus ou Discrets respectivement.

Les **MMC** continus on montrés des performances meilleurs que les **MMC** discrets [9],[12]; . Nous utiliserons donc des **MMC** continus avec des lois d'émission guaussiennes :

$$b_x(O_t) = N(O_t;\ \mu_x, \sum_x) \qquad (3-8)$$

Où N est une gaussienne multivariable à moyenne μ et à matrice de covariance $\sum$. Cette matrice est presque toujours considérée comme diagonale [12] hypothèse simplificatrice permettant de réduire considérablement le nombre de paramètres à estimer.

On aura donc :

$$b_x(O) = \frac{1}{\sqrt{|\sum_x|(2\pi)^n}} e\left\{-\frac{1}{2}(O-\mu_x)'\ {\sum_x}^{-1}(O-\mu_x)\right\} \qquad (3-9)$$

Où n est la dimension de O

III-5 Probabilités Forward/Backword [12]

Si l'on veut calculer la probabilité de générer la séquence d'observation $O=O_1O_2O_3\ldots O_T$ à l'aide du M**,** l'approche directe consiste à calculer, la probabilité de génération pour chaque chemin possible et faire la somme de ces Probabilités$(3-5)$.

Avec (X(1),…,X(T)) l'ensemble des séquences d'états possibles.une procédure bien Plus efficace existe pour réaliser ce calcul : l'algorithme Forward backward.

Considérons la variable forward $\alpha_t(x)$ définie

$$p(o_1o_2 \ldots \ldots o_t, x(t) = x|M) \qquad (3-10)$$

Qui exprime la probabilité d'avoir générer la séquence $O_1O_2O_3\ldots O_t$ en partant de l'état X(1) et d'être arrivé sur l'état X à l'instant t. cette variable peut être calculée de manière inductive :

1. Initialisation. à t =1 et pour 1≤ X≤N on a :

$$\alpha_X(1) = a_{1X}b_X(o1) \qquad (3-11)$$

2. Induction pour t=2,…, T on a:

$$\alpha_X(t) = \sum_{i=2}^{N-1} (a_i(t-1)\ a_{iX}) b_X(o_t) \qquad (3-12)$$

Pour 1<i<N et comme condition finale on a :

$$P(O/M) = \alpha_N(T) = \sum_{i=2}^{N-1} \alpha_i(T) a_{iN} \qquad (3-13)$$

La probabilité Backward $\beta_X(t)$ pour 1<x<N et T>t≥1 est calculé par la récursivité Backword

1. Induction pour t=2,…, T on a:

$$\beta_X(t) = \sum_{i=2}^{N-1} a_{xi}\, b_i(o_{t+1}) \beta_i(t+1) \qquad (3-14)$$

Avec $\beta_X(T) = a_{XN}$

Pour 1<X<N et comme condition finale on a :

$$\beta_1(1) = \sum_{i=2}^{N-1} a_{1i}\, b_i(o_1) \beta_i(1) \qquad (3-15)$$

III-6 Ré estimation des paramètres (algorithme de Baum-Welch) [12]

Les formules de ré estimation des paramètres d'un MMC entrainé par un nombre d'exemples égal à (R) sont données par :

$$\hat{a}_{ij} = \frac{\sum_{r=1}^{R} \frac{1}{P_r} \sum_{t=1}^{T_r-1} \alpha_i^r(t) a_{ij}\, b_j(o_{t+1}^r)\, \beta_j^r(t+1)}{\sum_{r=1}^{R} \frac{1}{P_r} \sum_{t=1}^{T_r} \alpha_i^r(t) \beta_i^r(t)} \qquad (3-16)$$

$$\hat{a}_{1j} = \frac{1}{R} \sum_{r=1}^{R} \frac{1}{P_r}\, \alpha_j^r(1) \beta_j^r(1) \qquad (3-17)$$

$$\hat{a}_{iN} = \frac{\sum_{r=1}^{R} \frac{1}{P_r} \alpha_i^r(T) \beta_i^r(T)}{\sum_{r=1}^{R} \frac{1}{P_r} \sum_{t=1}^{T} \alpha_i^r(t) \beta_i^r(t)} \qquad (3-18)$$

Pour les gaussiennes simples, il est plus efficace dans ce cas d'utiliser

$$l_j^r(t) = \frac{1}{P_r} \alpha_j(t) \beta_j(t) \text{ ou } p_r = p(O|M) \qquad (3-19)$$

Les formules de ré-estimation peuvent maintenant être exprimées en termes de l_j^r(t)

$$\widehat{\mu_j} = \frac{\sum_{r=1}^{R}\sum_{t=1}^{Tr} l_j^r(t) o_t^r}{\sum_{r=1}^{R}\sum_{t=1}^{Tr} l_j^r(t)} \qquad (3-20)$$

$$\widehat{\sum}_j = \frac{\sum_{r=1}^{R}\sum_{t=1}^{Tr} l_j^r (o_t^r - \widehat{\mu}_j)(o_s^t - \widehat{\mu}_j)'}{\sum_{r=1}^{R}\ \sum_{t=1}^{Tr} l_j^r(t)} \qquad (3-21)$$

III-7 Reconnaissance et décodage de l'algorithme de Viterbi[12]

La section précédente a décrit les idées de base qui sous-tendent le paramètre **HMM** ré-estimation à l'aide l'algorithme de **Baum-Welch**, il a été noté que l'algorithme récursif est efficace pour le calcul de la probabilité de Forward

Ainsi, cet algorithme peut également être utilisé pour trouver le modèle qui donne la valeur maximale de P (O|M), et, par conséquent, il pourrait être utilisé pour la reconnaissance. Cependant, il est préférable à la reconnaissance de baser sur la séquence d'états qui a générer la probabilité du maximum de vraisemblance, Cette probabilité est calculée en utilisant essentiellement le même algorithme de Farward sauf que la sommation est remplacée par une opération maximale. Pour un modèle M donné.

Soit φ_x(t) le maximum de vraisemblance de la séquence o_1 à o_t sachant qu'on se trouve dans l'état X au temps t, cette vraisemblance partielle peut être calculée comme suit :

$$\varphi_X(t) = \underbrace{max}_{i} \{\varphi_i(t-1) a_{iX}\}\, b_X(o_t) \qquad (3-22)$$

Avec $$\varphi_x(1) = a_{1X}\, b_X(o_1) \qquad (3-23)$$

Pour 1<X<N, Le maximum de vraisemblance P (O|M) est alors donnée par :

$$\varphi_N(t) = \underbrace{max}_{i} \{\varphi_i(T) a_{iN}\} \qquad (3-24)$$

Le calcul direct des probabilités de vraisemblance conduit à des résultats « **underflow** », par conséquent la récurrence de l'équation $(3-10)$ devient alors :

$$\psi_X(t) = \underbrace{max}_{i} \{\psi_i(t-1) + \log(a_{iX})\} + log(b_X(o_t)) \qquad (3-25)$$

Cet algorithme est dit **l'algorithme de Viterbi**.

III-8 Conclusion

Dans ce chapitre, nous avons exposé les **Modèles de Markov Cachés** et leurs utilisations dans la reconnaissance de la parole. Le chapitre suivant sera dédié à l'application des concepts de ce présent chapitre.

Chapitre IV : Résultats

IV-1 Introduction

Nous exposons dans ce chapitre, avec le maximum de détails, les travaux de reconnaissance réalisés ainsi que leur interprétation. Il s'agira de l'application des notions du troisième chapitre pour faire la reconnaissance audiovisuelle de la parole.

IV-2 Description de la base de données utilisée

Nous utilisons la base de données audiovisuelle **TULIPS** [13]

C'est une petite base de données contenant 12 locuteurs provenant des deux sexes. Chaque locuteur prononce deux fois les quatre chiffres anglais (1 à 4). Au total on dispose donc de 96 prononciations, à chaque prononciation correspond 2 fichiers l'un au format audio (réalisation acoustique du mot) et l'autre contenant les paramètres hauteur et largeur de la bouche.

Les fichiers audio sont stockés en entiers 8 bits avec une fréquence d'échantillonnage de 11127 Hz.

Pour les fichiers vidéo, une image est prise chaque 1/30 secondes, les dimensions de la bouche en pixel (6 coefficients) sont ensuite extraits manuellement de l'image et sont stockés en entiers 8 bits.

Notons que les fichiers de notre base de données seront groupés en deux ensembles : un ensemble utilisé pour l'apprentissage et un ensemble pour le test.

IV-3 Description de l'outil de reconnaissance (HTK)

HTK est le "Hidden Markov Model boite à outils" développé par L'Université de Cambridge, Département de génie (**CUED**) [14]. Cette trousse à outils vise à créer et manipuler des Modèles de Markov Cachés(**HMM**).
HTK est principalement utilisé dans le domaine de la recherche de la reconnaissance vocale, bien qu'il soit tout à fait utilisable dans de nombreuses autres applications telles que la synthèse vocale, la reconnaissance de l'écriture ou la

reconnaissance de séquences **d'ADN**. **HTK** consiste en un ensemble de modules de bibliothèques et d'outils disponibles écrits en langage **C**, ces différents outils facilitent l'analyse vocale, l'apprentissage des **HMMs**, la réalisation de tests et l'analyse des résultats.

La version compilée de **HTK** permet d'exploiter ses fonctionnalités en tapant une simple ligne de commande. Les principaux outils que nous avons utilisés seront évoqués progressivement dans ce chapitre.

IV-4 Prétraitement

Le but de cette phase est de transformer le signal vocal en une suite de vecteurs de paramètres. Cette opération est réalisée par la fonction (**hcopy**) dont le mécanisme de fonctionnement est illustré par la figure (IV -1)

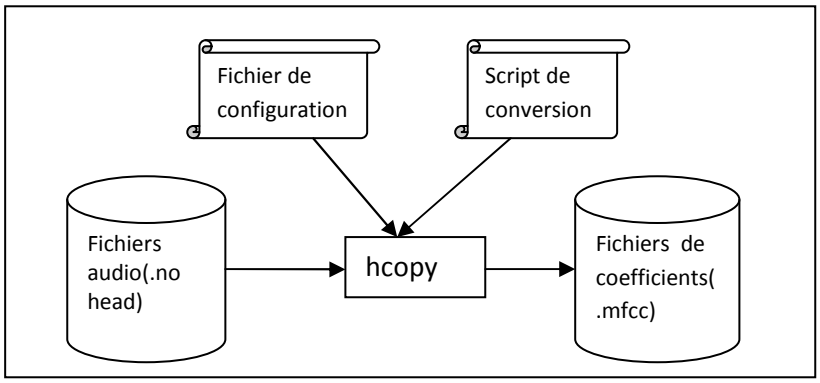


Figure (IV -1) : fonctionnement de HCOPY [15]

hcopy doit être invoquée comme suit :

hcopy -C codconf.txt -S trglstcode.txt

codeconf.txt est un fichier texte de configuration définissant les paramètres de l'extraction des coefficients acoustiques. Son contenu est le suivant :

```
SOURCEKIND = WAVEFORM
SOURCEFORMAT = NOHEAD          #identifie fichier parole source
SOURCERATE = 899               # période du signal en unité de 100 ns
TARGETKIND = MFCC_0_D          # type de coefficients utilisés
WINDOWSIZE = 333333.3          # = 25 ms = longueur  fenêtre
TARGETRATE = 333333.3          # = 10 ms = périodicité trame
NUMCEPS = 12                   # Nombre de coeffs (ici    c1 à c12)
USEHAMMING = T                 #  fenêtrage Hamming
PREEMCOEF = 0.97               # coeff  de Préaccentuation
NUMCHANS = 26                  # Nombre canaux banc de filtre
CEPLIFTER = 22                 # liftering
```

Remarque : le symbole # est utilisé pour introduire des commentaires.

Avec un tel fichier de configuration, l'analyse **MFCC** (Mel Frequency Cepstral Coefficient) est effectuée (préfixe "MFCC" dans l'identifiant TARGETKIND). Pour chaque trame de signal, les coefficients suivants sont extraits:

- Les 12 premiers coefficients **MFCC** [c1, ..., c12] (depuis NUMCEPS = 12).
- Le "nul" **MFCC** coefficient c0 indique que le cepstre 0 est ajouté.
- 13 "coefficients Delta", estimant la dérivée de premier ordre de [c0, c1, ..., c12] (suffixe "_D" Dans TARGETKIND).

Au total, un vecteur de 26 coefficients est extrait de chaque trame de signal.

Le script **trglstcode.txt** est un simple fichier texte spécifiant le nom et la location de chaque fichier signal à traiter et du fichier de coefficients correspondant à générer.

```
ann11     an11.mfcc
ann12     an12.mfcc
……        ……
ann32     an32.mfcc
sin42     sin42.mfcc
```

Remarque : le chemin des fichiers n'étant pas spécifié dans le script, tous les fichiers (audio et **mfcc**) seront dans le répertoire en cours.

Nous disposons à présent de tous les fichiers audio sous forme d'une suite de vecteurs de paramètres (fichier à extension .**mfcc**).

Les fichiers vidéo ne subiront aucun prétraitement, ils sont déjà sous forme de suite de vecteurs de paramètres, chaque vecteur contient 6 coefficients.

L'étape de prétraitement étant achevée, nous allons passer à la modélisation markovienne par **HTK**. Dans ce qui suivra nous expliquerons en détail les différentes étapes nécessaires à l'obtention de Modèles de Markov entrainés (donc prêts à être utilisés en reconnaissance) dans le cas de l'audio. Le cas de la vidéo, étant similaire, il ne sera pas évoqué pour éviter la répétition pouvant rendre la lecture de ce chapitre assez fastidieuse.

IV-5 Choix de la topologie des modèles (prototypes)

Nous devons définir quatre **MMC** (un pour chaque mot à reconnaitre), Soit donc 4 **MMCs** à définir. Nous opterons pour une topologie commune pour les 4 MMC. Ces 4 **MMCs** seront donc caractérisés par :

- Un nombre d'état égal à 6 dont 4 états émetteurs.les transitions possibles sont montrées sur la figure (IV-2). Les probabilités de transitions sont données par la matrice de probabilité de transition.
- Chaque état émetteur est caractérisé par une loi d'émission continue à une seule gaussienne caractérisée par un vecteur moyenne μ (26 composantes) et une matrice de covariance ∑ diagonale.

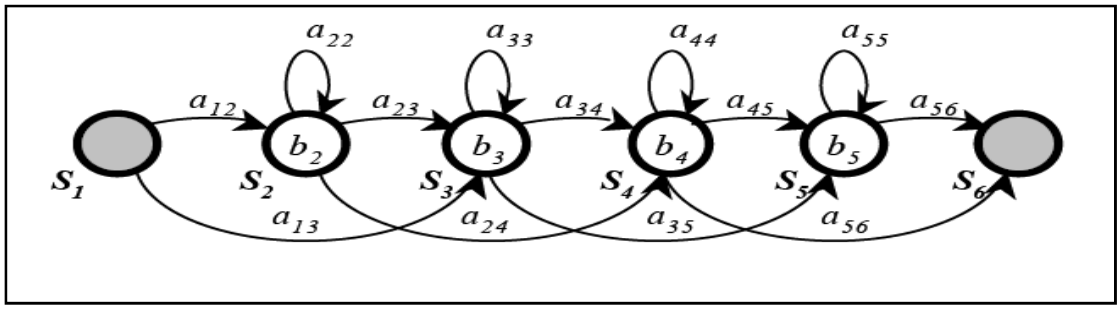


Figure (IV -2) : **topologie choisie pour les MMC [15]**

Remarque : il n'y a pas une loi fixe pour choisir telle ou telle topologie, cependant la topologie la plus communément adoptée est celle de **BAKIS** [12].

En **HTK**, un modèle **HMM** est décrit dans un simple fichier texte de description. Le fichier de description pour le **HMM** est de la forme :

```
~o <VecSize> 26 <MFCC_0_D>
~h "one"
<BeginHMM>
<NumStates> 6
<State> 2
<Mean> 26
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 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> 26
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
<State> 3
<Mean> 26
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 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> 26
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
<State> 4
<Mean> 26
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 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> 26
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
<State> 5
<Mean> 26
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 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> 26
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
<TransP> 6
0.0 0.5 0.5 0.0 0.0 0.0
0.0 0.4 0.3 0.3 0.0 0.0
0.0 0.0 0.4 0.3 0.3 0.0
0.0 0.0 0.0 0.4 0.3 0.3
0.0 0.0 0.0 0.0 0.5 0.5
0.0 0.0 0.0 0.0 0.0 0.0
<EndHMM>
```

`~o <VecSize> 26 <MFCC_0_D>`
Est l'en-tête du fichier, ce qui donne la taille du vecteur de coefficient (26 coef-

ficients ici), et le type du coefficient est ici (MFCC_0_D).

`~h "one"<BeginHMM>(……)<EndHMM>`

renferme la description d'un HMM appelé «one».

`<NumStates> 6`

donne le nombre total d'états dans le HMM, y compris les 2 états non émetteurs (États 1 et 6).

`<State> 2`

introduit la description de la fonction d'observation de l'état 2. Ici, nous avons choisi d'utiliser une fonction d'observation gaussienne unique, avec des matrices diagonales. Une telle fonction est tout à fait décrite par un vecteur moyenne et un vecteur variance (les éléments diagonaux de la matrice d'autocorrélation). Les états 1 et 6 ne sont pas décrits, car ils n'ont pas des fonctions d'observations.

`<Mean> 26`

`0.0 0.0 (...) 0.0` (x 26)

Donne le vecteur moyen de l'observation courante, chaque élément est arbitraire initialisée à 0: le fichier ne donne que le "prototype" de la **HMM** (sa topologie globale). Les valeurs de ces coefficients seront obtenues plus tard après apprentissage.

`<Variance> 26`

1,0 1,0 (...) 1,0 (x 26) donne le vecteur variance de la fonction d'observation courante. Chaque élément est arbitrairement initialisé à 1, les vraies valeurs seront obtenues après apprentissage.

`<TransP> 6`

donne la matrice de transition 6x6 du **HMM**, où aij est la probabilité de transition de l'état i à l'état j. Les valeurs nulles indiquent que les transitions correspondantes ne sont pas autorisés. Les autres valeurs sont arbitrairement initialisées(mais la somme sur chaque ligne de la matrice doit être égale à 1): elles seront plus tard modifiées, au cours du processus d'apprentissage.

a11 a12 a13 a14 a15 a16 *a21 a22 a23 a24 a25 a26* *a31 a32 a33 a34 a35 a36* *a41 a42 a43 a44 a45 a46* *a51 a52 a53 a54 a55 a56* *a61 a62 a63 a64 a65 a66*

Un tel prototype doit être généré pour chaque mot. Dans notre cas, nous avons à écrire un prototype pour chacun des 4 HMM que nous appellerons «one», «two», «three» et «four» avec les en-têtes ~ h "one", ~ de h "two", ~ h "three" et ~ de h "four" dans les 4 fichiers de description).
Les 4 fichiers sont nommés hmm_one,hmm_two,hmm_three et hmm_four.

IV-6 Initialisation des modèles

La fonction (**hinit)** dont le mécanisme de fonctionnement est illustré sur la figure (IV-3) permet d'initialiser les modèles.

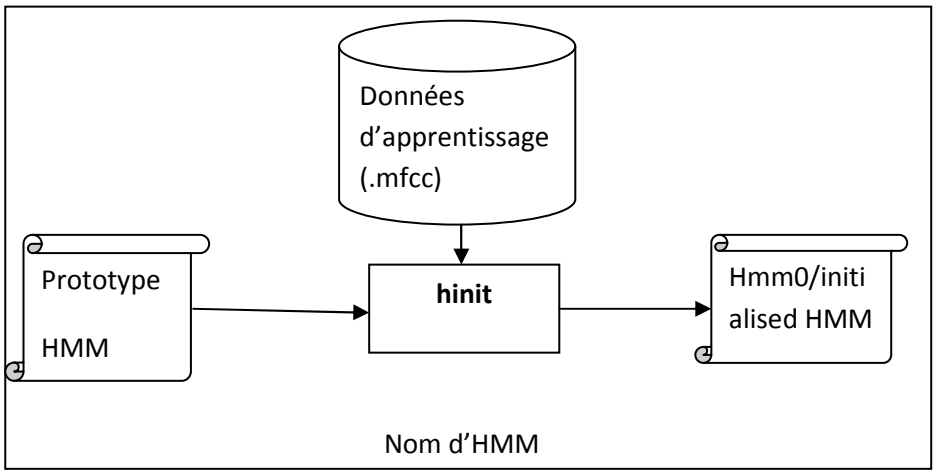


Figure (IV -3) : fonctionnement de (hinit) [15]

Hinit doit être invoquée come suit :

hinit –S trlist1.txt –M model/hmm0 -H model/proto/fichierhmm nomfichierhmm

nomfichierhmm est le nom du hmm à initialiser (ici : one, two, three et four).
fichierhmm est un fichier de description contenant les prototype HMMs (ici : proto/hmm_one,hmm_two,hmm_three,hmm_four).

Trlist(1,2,3,4).txt donne la liste complète des fichiers MFCC formant le corpus d'apprentissage pour les mots (**one**,**two**,**three** et **four**).

model/hmm0 est le nom du sous répertoire **hmm0** contenant le fichier de description résultant après initialisation.

Ce fichier de sortie HMM par **hinit** a le même nom que le prototype d'entrée.

Cette procédure doit être répétée pour chaque modèle (hmm_one,hmm_two, ,hmm_three, hmm_four).

IV-7 Apprentissage

C'est le rôle de la fonction (**hrest**) dont le mécanisme de fonctionnement est illustré sur la figure (IV -4)

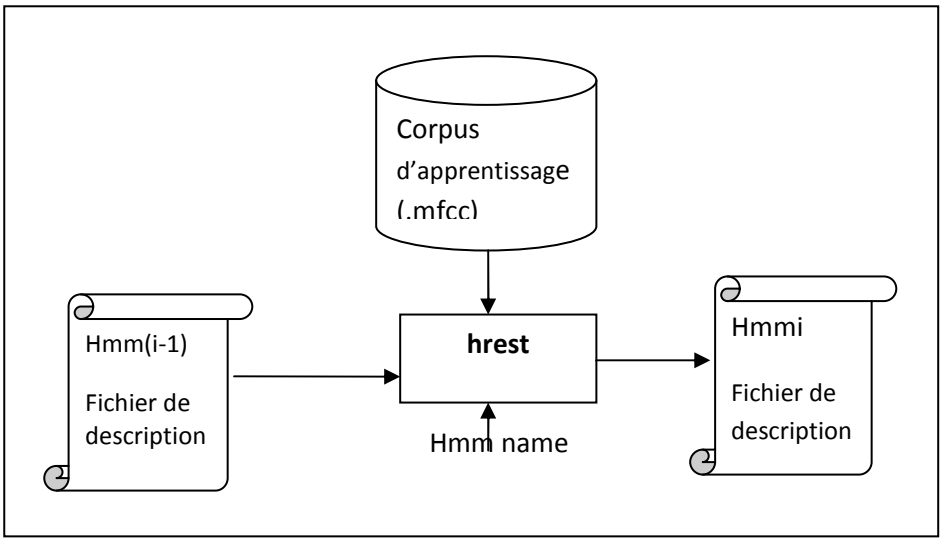


Figure (IV -4) : fonctionnement de (hrest) [15]

La ligne de commande suivante effectuer une itération ré-estimation avec (**hrest):**

hrest –S trlist.txt -M model/hmmi –H model/hmmi-1/fichierhmm nom-fichierhmm

nomfichierhmm est le nom du HMM pour apprendre.

fichierhmm est le fichier de description du HMM appelé nomfichierhmm. Il est stocké dans un répertoire dont le nom indique l'indice de la dernière itération (ici model/hmmi-1 /par exemple).

Modèle / hmmi, le répertoire de sortie, indique l'indice de l'itération courante i.

3 itérations de réestimation sont suffisantes pour obtenir les modèles finaux.
Finalement on dispose des 4 modèles audio entrainés et prêts à être utilisés pour la reconnaissance.
Comme on l'a souligné précédemment, l'obtention des 4 modèles vidéo passe par les mêmes étapes que celles de l'audio.
En fin de compte pour chaque mot on dispose de deux **HMMs** (audio et vidéo), soit au total 8 **HMMs**.

Avant d'utiliser nos modèles de mots, nous devons définir l'architecture de base de notre reconnaissance (la grammaire). En **HTK**, la grammaire est écrite dans un fichier texte, selon certaines règles syntaxiques [9]. Dans notre cas, la grammaire est assez simple.

```
$WORD = one|two|three|four;
($WORD)
```

La variable **WORD** peut être remplacée par one ou two ou three ou four.
Le système doit bien sûr savoir à quel **HMM** correspond chacune des variables de grammaire one, two, three et four. Cette information est stockée dans un fichier texte appelé dictionnaire. Dans une telle tâche simple, la correspondance est simple, et la tâche dictionnaire renferme tout simplement les 4 entrées.

One	one
Two	two
Three	three
Four	four

Les éléments gauche se réfèrent aux noms des variables de grammaire. Les éléments droits aux noms des **HMM** (introduit par ~ h dans les fichiers de description **HMM**).

Le fichier grammaire doit être compilé avec l'outil **hparse** pour obtenir un réseau de connexion reliant les 4 mots.

hparse gram.txt wdnet.slf

gram.txt est le fichier grammaire.

wdnet.slf est le fichier obtenu

A ce stade, notre tâche de reconnaissance vocale est entièrement définie par son réseau figure (IV-5), son dictionnaire, et son jeu de **HMM** (les 4 modèles audio et les 4 modèles vidéo), est prêt à l'emploi.

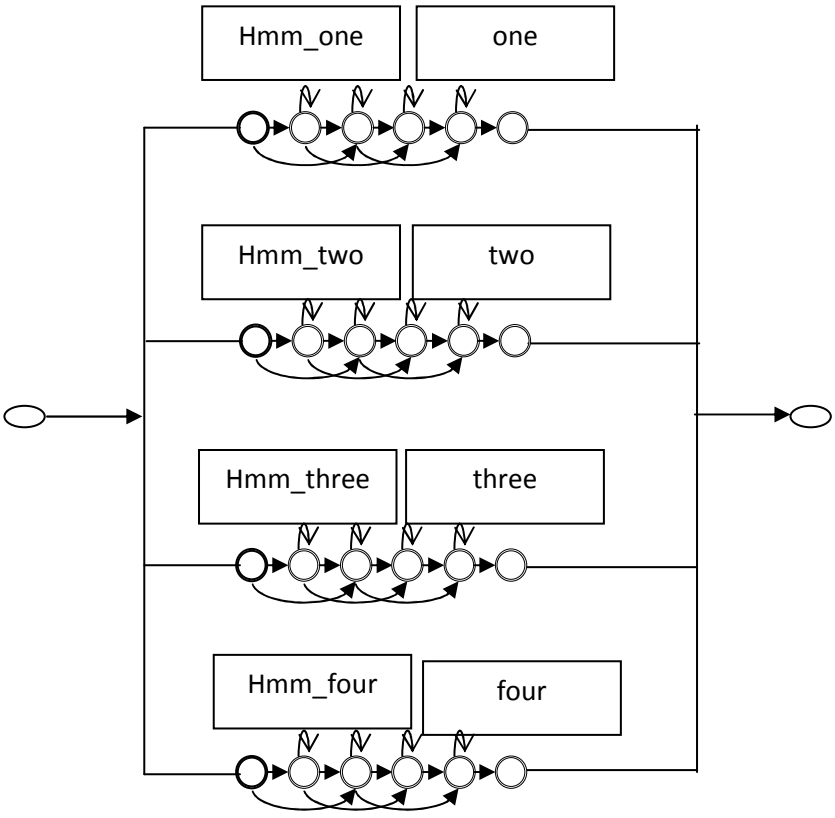


Figure (IV -5) : le réseau de connexion entre mot

IV-8 Reconnaissance

Dans cette étape nous allons tester la capacité de nos modèles à reconnaitre des mots inconnus présentés à l'entrée du système de reconnaissance. Ayant un mot inconnu à l'entrée, la fonction (**hvite**) calculera

sa vraisemblance à partir des 4 modèles puis choisira le modèle donnant le maximum de vraisemblance. Cette fonction est invoquée comme suit :

hvite -H hmmsdef.mmf -i reco1.mlf -wdnet.slf dict.txt listhmm.txt –S

tslist1.txt

Hmmlist.txt énumère les noms des modèles à utiliser (one, two, three et four). Chaque élément est séparé par une nouvelle ligne de l'autre élément; et une nouvelle ligne après le dernier élément.

One

Two

three

four

Tslist1.txt script contenant les noms des fichiers (.mfcc) de test représentant les mots à reconnaitre.

Dict.txt est le dictionnaire.

Wdnet.slf est le fichier de réseau de connexion obtenu.

hmmsdef.mmf contient la définition des **HMMs**. Un tel fichier est simplement obtenu en copiant chaque définition après l'autre dans un fichier unique, sans répéter l'information d'en-tête.

La sortie est stockée dans un fichier (reco.mlf) qui contient le fichier de reconnaissance avec la vraisemblance de reconnaissance pour chaque mot.

Un exemple d'un tel fichier est donné dans le cas de la reconnaissance du mot (one). 12 prononciations du mot (one) ont été présentées au système, parmis les quelles 11 ont été correctement reconnues et une seule (la deuxième) à été Confondue avec le mot (four).

```
#!MLF!#
"audio/an12.rec"
0 2666664 one -466.720520
.
"audio/be12.rec"
0 3333330 four -666.968140
.
"audio/ca12.rec"
0 1999998 one -378.184204
.
"audio/cy12.rec"
0 3999996 one -780.565979
.
"audio/do12.rec"
0 3333330 one -674.765747
.
"audio/ge12.rec"
0 2999997 one -591.658997
.
"audio/is12.rec"
0 2333331 one -395.854401
.
"audio/ja12.rec"
0 3666663 one -702.823364
.
"audio/je12.rec"
0 2333331 one -412.612061
.
"audio/ol12.rec"
0 2666664 one -482.604126
.
"audio/re12.rec"
0 2333331 one -428.220947
.
"audio/si12.rec"
0 2333331 one -432.041687
.
```

Le schéma de la figure (IV-6) illustre tous le processus, du codage jusqu'à la reconnaissance :

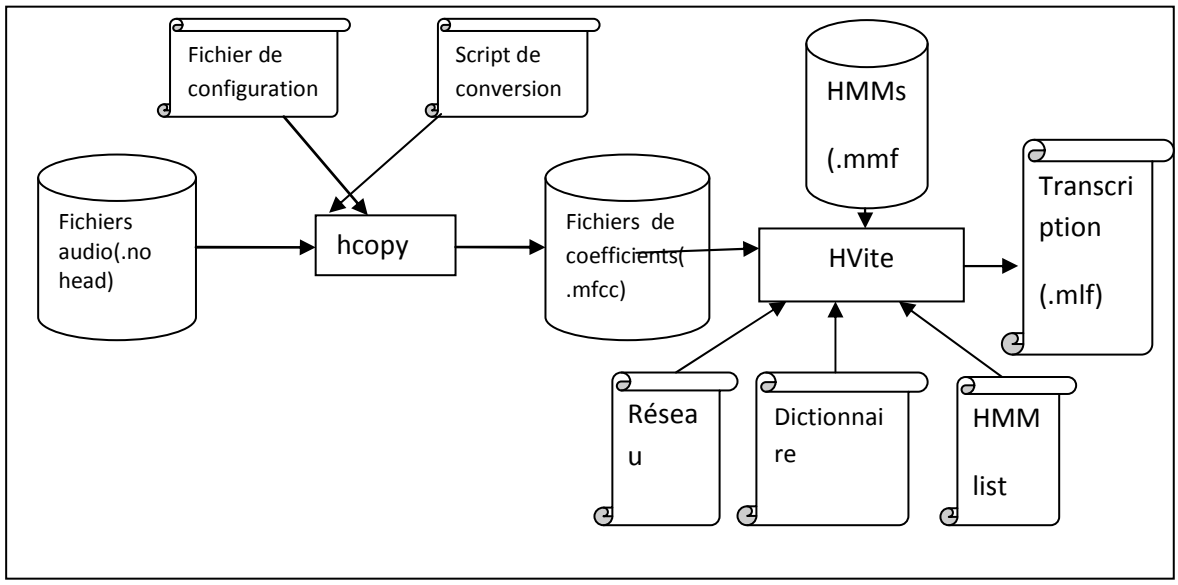


Figure (IV-6) : illustration de tout le processus du codage à la reconnaissance

IV-9 Tests de robustesse et interprétations

Nous avons effectués une série d'expérimentation s'inscrivant dans la chronologie suivante :

- Reconnaissance audio en l'absence de bruit
- Reconnaissance vidéo
- Combinaison audio–vidéo et comparaison avec audio seule et vidéo seule
- Reconnaissance audio en présence de bruit
- Reconnaissance audio-vidéo en présence de bruit et comparaison avec audio en présence de bruit.
- Conclusion

IV-9-1 Reconnaissance en l'absence de bruit

Le tableau suivant illustre les taux d'erreur de reconnaissance obtenus par mot :

Audio seule	Video seule	A-V
4.17 %	10.42 %	2.08 %

Tableau (IV-1) : résultats audio et vidéo en l'absence de bruit

La troisième colonne du tableau précédent est obtenue en exploitant les modèles **MMC** audio et les modèles **MMC** vidéo obtenus précédemment pour former des modèles hybrides ; le processus de reconnaissance (décision) est illustré sur la figure (IV-7) :

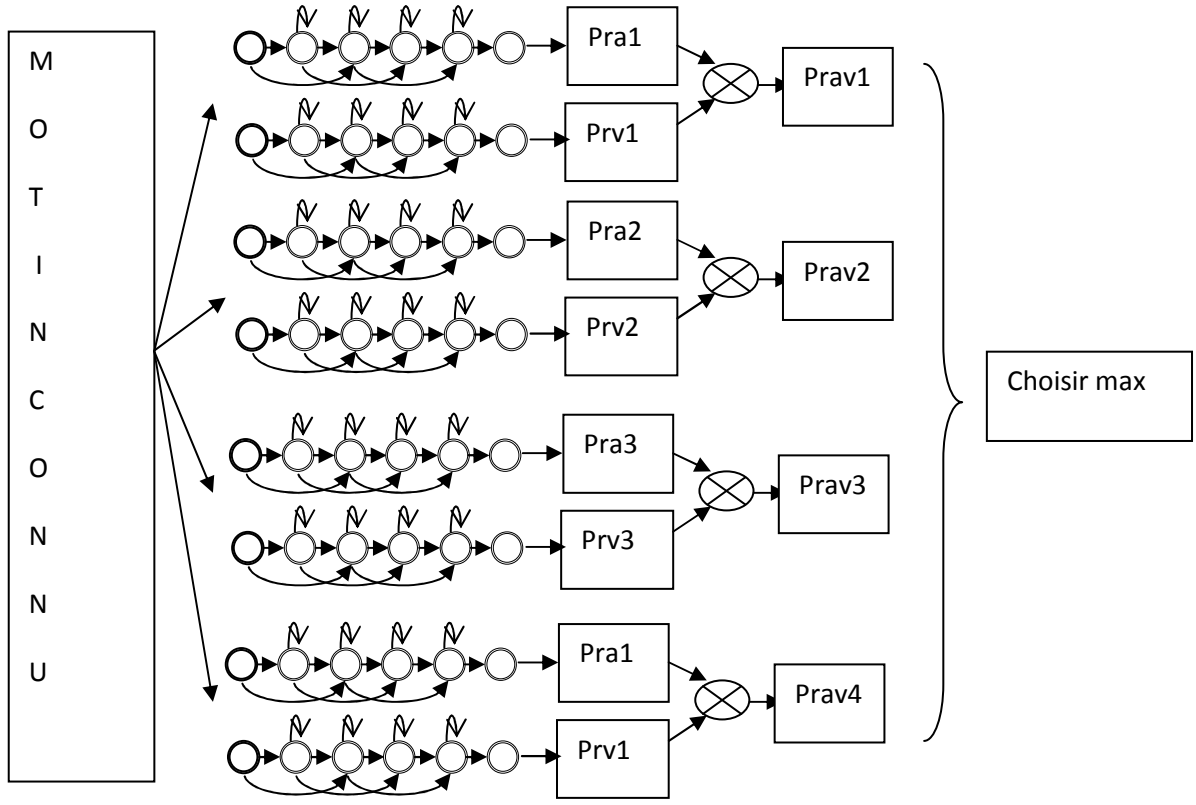


Figure (IV-7) : Processus de reconnaissance hybride

Le mot inconnu à l'entrée du système est composé de deux parties : audio et vidéo. La vraisemblance audio (Pra) est calculée dans tous les modèles audio, on obtient ainsi Pra1,Pra2 ,Pra3 et Pra4. De même, la vraisemblance vidéo est calculée dans tous les modèles vidéo, on obtient ainsi Prv1, Prv2, Prv3 et Prv4. On calcule ensuite le produit des vraisemblances Pra et Prv de chaque modèle ;

on obtient ainsi Prav1, Prav2, Prav3 et Prav4 ; la décision sera faite au profit du max.

Remarques :

- Ce schéma utilisé pour la combinaison des modalité audio et vidéo, qui est en fait une fusion de décision n'est pas le seul, on trouve dans la littérature [15][16][17] plusieurs autres schémas.
- Le mode de fonctionnement de la fonction (hvite), ne permettant pas de réaliser un tel schéma de fusion, nous avons réalisé une routine (en PASCAL) qui réalise le processus de la figure (IV-7). Cette routine est donnée en annexe.

Interprétations :

- La modalité audio donne le meilleur score de reconnaissance, ceci était prévisible, étant donné qu'en l'absence de bruit la modalité audio est prédominante, en ce sens qu'elle apporte suffisamment d'informations pour reconnaitre le message prononcé.
- La combinaison audio vidéo donne le meilleur score par rapport à l'audio seule et la vidéo seule.

IV-9-2 Reconnaissance en milieu bruité

Nous avons effectués des expérimentations en ajoutant différents niveaux de bruit, les résultats sont montrés dans le tableau (IV-2) ; ainsi que sur la figure (IV-7).

Taux d'erreur RSB(db)	Audio seule	Video seule	A-V
10	39.58 %	10.42 %	18.75 %
20	12.5 %	10.42 %	2.08 %
30	4.17 %	10.42 %	2.08 %

Tableau (IV-2) : résultats obtenus pour différents niveaux de bruit

Interprétations :

- La performance de reconnaissance audio baisse dramatiquement lorsque le rapport signal sur bruit (**RSB**) baisse. (le niveau de bruit augmente).
- La performance de reconnaissance vidéo est indépendante du niveau de bruit acoustique.
- Pour les fortes valeurs du **RSB** la reconnaissance audio est meilleure que la reconnaissance vidéo, la reconnaissance audio-visuelle dépasse les deux.
- Pour les faibles valeurs du **RSB** (ambiance fortement bruitée), la reconnaissance vidéo devient meilleure que la reconnaissance audio,

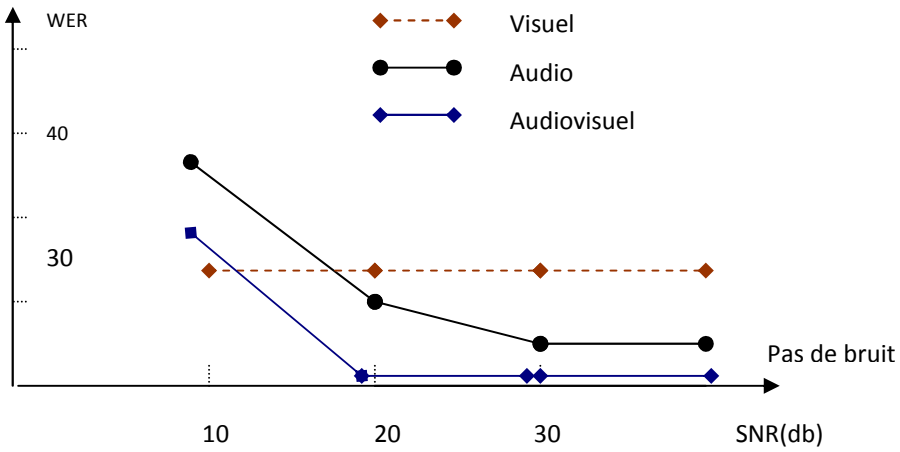


Figure (IV-8) : taux d'erreur (WER) pour la reconnaissance audio, vidéo et audiovisuelle pour différents niveau de bruit

IV-10 Conclusion

Dans ce chapitre, nous avons exposé les résultats de reconnaissance obtenus dans différentes ambiances acoustiques pour mettre en évidence l'apport positif de la modalité visuelle sur les performances du système de reconnaissance.

Conclusion générale

Dans les modèles classiques de reconnaissance des mots parlés, les premières étapes entre l'entrée auditive et les unités phonémiques font appel à des phénomènes de segmentation du signal de parole et de catégorisation des segments selon des traits distinctifs. L'exécution de ces deux processus est complexe du fait de la variabilité importante du signal de parole. Cette variabil-
ité peut émerger aussi bien de facteurs intrinsèques au locuteur que de facteurs extrinsèques de l'environnement. Dans cette thèse, nous avons mentionné à la fois à travers les quatres chapitres la segmentation de la parole arrivant à la modélisation Markovienne et les résultats obtenus avec celle-ci.

En conclusion**,** arrivé à terme de ce travail, il m'est très agréable de montrer ma joie d'avoir abordé un tel domaine. Le caractère multidisciplinaire de ce domaine a élargi les horizons de mes connaissances qui vont du traitement de signal à l'informatique.

Le temps faisant défaut, beaucoup de démarches peuvent être entreprises dans le futur en termes de perspectives à savoir :

- Pondérer les deux modalités audio et vidéo selon le niveau de bruit.
- Adopter d'autres approches de fusion de modalités, telle que la fusion au niveau des vecteurs de paramètres (entrée du système)
- Appuyer le présent système par des techniques de filtrage de bruit.
- Utilisation de base de données en langue arabe.

Bibliographie

[1]Thomas Hueber, reconstitution de la parole par imagerie ultrasonore et vidéo de l'appareil vocal : vers une communication parlée silencieuse, Thèse de doctorat de l'université pierre et marie curie, 2009.

[2]Marc Antonie Rondeau Beauchamp, effet du canal sur la reconnaissance de la parole, Mémoire de l'école de technologie supérieure Montréal, 2010.

[3]Lawrence Rabiner et Biing –Hwang Juang, Fundamentals of speech recognition,

PTR Prentice-Hall,Inc,1993.

[4]Thierry Dutoit, introduction au traitement automatique de la parole, Faculté Polytechnique de Mons TCTS Lab, 2000.

[5]Linares Georges, reconnaissance automatique de la parole, Laboratoire d'Informatique d'Avignon, Université d'Avignon et des Pays de Vaucluse

[6]Asmaa Amehraye, Débruitage perceptuel de la parole, Thèse de doctorat l'Ecole Nationale Supérieure des Télécommunications de Bretagne en habitation conjointe avec l'université de Bretagne sud en cotutelle avec l'université Mohamed –V-agdal de Rabat, 2009.

[7]Jean-Paul Haton, LORIA-INRIA, La reconnaissance automatique de la parole, Université Henri Poincaré, Nancy 1, Institut Universitaire de France, 2005.

[8]Olivier Godin, Chapitre 5 - Analyse de la parole, Université de Sherbrooke, 2011.

[9]Steve Young,Dan Kershaw,Julian Odell,Dave Ollason,Valtcho Valtchev,Phil Woodland, The HTK Book, Microsoft Corporation, 1995.

[10]M.A.Anusuya & S.K.Katti, Speech recognition by machine, International Journal of Computer Science and Information Security,6(3), 2009.

[11]Jean-Paul HATON，Anne BONNEAU, Dominique FOHR, Yves LAPRIE, Yifan

GONG, Jean-Marie PIERREL, Décodage acoustico-phonétique problèmes et

éléments de solution (*),7(4) :293 -313,1989.

[12]L.R.Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition , Proc.IEEE 77(2):257-286,1989.

[13]TULIPS database (Movellan, 1995).

[14]Nathalie CAMELIN**,** thèse de doctorat Stratégies robustes de compréhension de la parole basées sur des méthodes de classification automatique, l'Université d'Avignon et des Pays de Vaucluse, 26 novembre 2007.

[15]Mihaela Gordan, Constantine Kotropoulos, Ioannis Pitas, A Support Vector Machine-Based Dynamic Network for Visual Speech, EURASIP Journal on Applied Signal Processing, 2001

[16]Petar S. Aleksic and Aggelos K. Katsaggelos, comparison of low-and high-level visual features for audio-visual continuous automatic speech recognition, Department of Electrical and Computer Engineering Northwestern University.

[17]Simon Lucey, BEng(Hons), Thése de doctorat Audiovisual Speech Processing, Queensland University of Technology Speech Research, Laboratory School of Electrical & Electronic Systems Engineering,2002.

Annexe

Les fichiers HMMs obtenus après la dernière ré-estimation pour l'audio et le vidéo sont ci-joint :

```
~o
<STREAMINFO> 1 26
<VECSIZE> 26<NULLD><MFCC_D_0>
~h "one"
<BEGINHMM>
<NUMSTATES> 6
<STATE> 2
<MEAN> 26
 -9.884175e+000 -4.548594e-001 -4.813449e+000 -1.077774e+001 -
8.725080e+000 -1.157341e+000 -3.615088e+000 1.785954e+000 2.347450e+000
-1.562874e+000 -4.580260e+000 -4.115699e+000 6.700611e+001 5.518968e-001
-1.297312e+000 -1.956620e+000 -9.064169e-001 1.419877e+000 5.157043e-001
-2.410649e-001 -2.737788e-001 1.332887e-001 1.666204e-001 -9.730530e-001
-1.147162e+000 1.335232e+000
<VARIANCE> 26
 1.179568e+001 5.274962e+000 1.885939e+001 3.542189e+001 1.761927e+001
1.584214e+001 1.601198e+001 1.551633e+001 1.614903e+001 2.976844e+001
2.318448e+001 2.117647e+001 9.577699e+000 3.844016e-001 2.443951e+000
2.522164e+000 2.592661e+000 4.593184e+000 2.686206e+000 2.962819e+000
4.044639e+000 3.105588e+000 5.161752e+000 1.888794e+000 1.927707e+000
7.695045e-001
<GCONST> 9.481419e+001
<STATE> 3
<MEAN> 26
 -7.364307e+000 -8.350857e+000 -1.463408e+001 -1.442325e+001 4.753543e-
001 2.384739e+000 -6.220398e+000 2.410727e-001 2.605572e+000 -4.289921e-
001 -7.138042e+000 -1.041505e+001 7.345649e+001 -3.404526e-001 -
2.994843e+000 -4.803959e-001 2.227496e+000 2.390152e+000 -1.774629e+000
-2.835444e-001 5.703905e-001 -9.488743e-002 -1.137987e+000 -
1.476760e+000 -5.818663e-001 1.466150e+000
<VARIANCE> 26
 1.681933e+000 1.392282e+001 3.370890e+000 5.982977e+001 2.278133e+001
1.489594e+001 2.504415e+001 4.448753e+001 3.401034e+001 4.508703e+001
2.650010e+001 1.496961e+001 1.377157e+001 1.110092e-001 1.150084e+000
3.015830e+000 2.648999e+000 4.064067e+000 2.762103e+000 3.601534e+000
2.942352e+000 4.046005e+000 4.067046e+000 3.318968e+000 2.936172e+000
5.787224e-001
<GCONST> 9.435783e+001
<STATE> 4
<MEAN> 26
 -1.012226e+001 -1.251797e+001 -8.074086e+000 -6.644699e+000 -
2.752881e+000 -7.468369e+000 -2.342417e+000 5.396595e+000 -1.707774e+000
-7.118074e+000 -8.512799e+000 -7.693664e+000 7.404688e+001 -6.609346e-
001 7.173071e-001 2.629710e+000 1.000662e+000 -1.973177e+000 -
1.245808e+000 1.254783e+000 1.846597e-001 -1.943619e+000 -5.154635e-001
1.276979e+000 4.305329e-001 -7.541243e-001
<VARIANCE> 26
 6.191029e+000 1.272377e+001 2.064213e+001 2.104580e+001 2.967783e+001
1.737531e+001 4.256359e+001 1.980206e+001 5.713909e+001 3.357392e+001
2.716236e+001 2.203120e+001 9.387103e+000 1.823670e-001 4.153071e+000
1.405519e+000 2.973483e+000 1.895144e+000 3.737732e+000 5.621407e+000
3.673039e+000 3.447290e+000 4.212267e+000 2.887099e+000 3.591100e+000
1.672089e+000
<GCONST> 9.906653e+001
```

```
<STATE> 5
<MEAN> 26
 -9.581262e+000 -2.700911e+000 -2.157460e+000 -9.248905e+000 -
7.396785e+000 -5.429020e+000 -7.050182e+000 -6.132513e-001 -6.160106e+000
-6.243891e+000 -6.560473e+000 -8.882648e+000 6.917467e+001 -1.847214e-001
1.686936e+000 1.234130e+000 7.214621e-001 -4.247800e-001 -1.557958e-001 -
1.414239e+000 -1.285802e+000 -4.316900e-001 5.285447e-001 3.179951e-001
4.589245e-001 -1.227925e+000
<VARIANCE> 26
 2.479487e+000 6.818613e+000 1.315534e+001 4.320838e+001 2.058522e+001
1.585651e+001 1.938914e+001 7.170470e+000 2.321676e+001 3.624461e+001
1.755556e+001 3.001517e+001 3.654332e+000 1.791027e-001 1.599398e+000
1.292356e+000 2.037343e+000 1.693833e+000 2.094039e+000 1.475670e+000
1.289257e+000 3.407631e+000 3.021615e+000 1.208189e+000 1.052683e+000
5.428450e-001
<GCONST> 8.556493e+001
<TRANSP> 6
 0.000000e+000 9.007951e-001 9.920491e-002 0.000000e+000 0.000000e+000
0.000000e+000
 0.000000e+000 6.492061e-001 3.507940e-001 0.000000e+000 0.000000e+000
0.000000e+000
 0.000000e+000 0.000000e+000 4.822040e-001 5.177960e-001 0.000000e+000
0.000000e+000
 0.000000e+000 0.000000e+000 0.000000e+000 6.126270e-001 2.905223e-001
9.685070e-002
 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 6.255162e-001
3.744838e-001
 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
0.000000e+000
<ENDHMM>
```

```
~o
<STREAMINFO> 1 26
<VECSIZE> 26<NULLD><MFCC_D_0>
~h "two"
<BEGINHMM>
<NUMSTATES> 6
<STATE> 2
<MEAN> 26
 -1.951265e+001 -3.885202e+000 -1.334690e+000 -5.730364e+000 -
3.667167e+000 -1.289273e-001 -2.186209e+000 3.989655e+000 8.868209e-002
1.563345e+000 9.374836e-001 -2.584112e+000 6.743808e+001 2.317339e-001
4.616638e-001 9.806173e-001 -1.010195e+000 -1.519141e+000 -1.513949e+000
-4.622836e-001 -9.442415e-001 -6.689304e-001 5.905052e-002 -2.709552e-001
-9.366862e-001 1.071168e+000
<VARIANCE> 26
 1.113911e+001 9.514844e+000 1.072461e+001 1.181881e+001 9.513068e+000
1.504664e+001 2.690140e+001 2.373553e+001 1.510595e+001 1.945558e+001
1.727243e+001 1.989184e+001 1.158191e+001 2.687447e+000 9.494051e-001
3.322461e+000 3.789034e+000 3.179171e+000 2.462362e+000 2.339116e+000
3.339226e+000 2.603710e+000 2.345632e+000 1.490584e+000 1.990800e+000
1.208081e+000
<GCONST> 9.335439e+001
<STATE> 3
<MEAN> 26
 -1.347960e+001 -2.413619e+000 5.224234e+000 -1.285015e+001 -
1.472238e+001 -1.005683e+001 -6.078808e+000 -2.406040e+000 -2.736776e+000
-1.034129e-001 -3.715119e+000 -6.754046e+000 7.251436e+001 2.217542e+000
-9.877607e-002 7.092066e-001 -1.181319e+000 -2.518719e+000 -3.254606e+000
-1.578968e+000 -1.320187e+000 2.147326e-001 -7.467701e-001 -1.219646e+000
1.393619e-001 8.693613e-001
```

<VARIANCE> 26
 3.975859e+000 8.199358e+000 2.742516e+001 4.489253e+001 1.700632e+001
2.143787e+001 1.986896e+001 3.054677e+001 3.470636e+001 2.900583e+001
2.027130e+001 1.511872e+001 2.064270e+001 8.427376e-001 1.394155e+000
2.015218e+000 2.482750e+000 1.648798e+000 2.764183e+000 2.028080e+000
2.786646e+000 2.165555e+000 2.255221e+000 2.235773e+000 2.423256e+000
7.180876e-001
<GCONST> 9.432340e+001
<STATE> 4
<MEAN> 26
 -1.187375e+001 -4.826375e+000 2.395222e+000 -1.281214e+001 -
1.611743e+001 -1.237422e+001 -7.834713e+000 7.640247e-001 -5.113114e-002
-2.432002e+000 -3.916727e+000 -3.124019e+000 7.215192e+001 8.279144e-001
-1.956341e-001 -1.201449e+000 2.259422e+000 1.664941e+000 -5.982829e-001
-2.204623e+000 1.970157e-001 4.235020e-001 -9.807268e-001 -7.645765e-001
1.104257e+000 -7.417239e-001
<VARIANCE> 26
 3.297271e+000 7.703012e+000 8.483621e+000 3.972175e+001 1.544096e+001
6.055971e+001 2.448886e+001 1.061177e+001 1.200919e+001 3.078072e+001
1.969761e+001 2.976665e+001 1.529588e+001 1.309937e-001 5.378230e-001
1.460386e+000 1.470569e+000 1.803872e+000 9.569821e-001 4.194530e+000
2.408874e+000 3.506692e+000 1.277500e+000 1.824531e+000 5.205470e-001
5.651339e-001
<GCONST> 8.610217e+001
<STATE> 5
<MEAN> 26
 -9.873178e+000 -1.123607e+000 -7.888124e-001 -6.906549e+000 -
8.527912e+000 -1.017822e+001 -1.414942e+001 -4.332423e+000 -3.289208e-001
-1.120326e+000 -4.588202e+000 -2.870134e+000 6.862329e+001 2.237778e-002
5.636700e-001 -1.108657e+000 7.229036e-001 1.768718e+000 1.114917e+000 -
7.066940e-001 -8.569553e-001 -1.117701e-001 -1.083027e-001 7.197208e-001
1.780539e-001 -1.011871e+000
<VARIANCE> 26
 3.226147e+000 1.267137e+001 1.418270e+001 1.322054e+001 1.480081e+001
2.836020e+001 1.792618e+001 3.365947e+001 3.127765e+001 2.566127e+001
1.801053e+001 2.788615e+001 1.099808e+001 4.385289e-001 4.889949e-001
6.584340e-001 2.061999e+000 2.686141e+000 2.636703e+000 3.591142e+000
1.750086e+000 4.599322e+000 5.073196e+000 1.958487e+000 2.516236e+000
5.295407e-001
<GCONST> 9.114706e+001
<TRANSP> 6
 0.000000e+000 1.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
0.000000e+000
 0.000000e+000 7.142983e-001 2.857017e-001 0.000000e+000 0.000000e+000
0.000000e+000
 0.000000e+000 0.000000e+000 5.555795e-001 3.701900e-001 7.423049e-002
0.000000e+000
 0.000000e+000 0.000000e+000 0.000000e+000 4.452515e-001 5.547485e-001
0.000000e+000
 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 6.361254e-001
3.638746e-001
 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
0.000000e+000
<ENDHMM>

```
~o
<STREAMINFO> 1 26
<VECSIZE> 26<NULLD><MFCC_D_0>
~h "three"
<BEGINHMM>
<NUMSTATES> 6
<STATE> 2
<MEAN> 26
 -1.565530e+001 -2.091109e+000 -1.125734e+000 -3.051327e+000 -
3.842551e+000 -5.364350e-001 -2.656621e+000 1.610944e+000 1.597069e-001
5.371401e-001 7.128986e-001 -1.414357e-001 6.360707e+001 9.250318e-001 -
6.151118e-001 -6.156024e-001 6.975553e-002 -1.020318e+000 -1.611979e+000
-1.316014e+000 2.923853e-001 1.067517e-001 6.276735e-002 -4.588202e-001 -
9.135597e-001 1.005528e+000
<VARIANCE> 26
 2.848792e+000 2.883318e+000 6.652005e+000 6.298687e+000 9.615944e+000
8.444209e+000 1.033287e+001 2.151105e+001 1.890789e+001 7.930620e+000
1.602164e+001 1.656197e+001 9.942882e-001 1.165742e+000 7.861037e-001
1.133655e+000 1.274696e+000 2.671579e+000 3.282873e+000 2.709960e+000
3.291133e+000 3.076842e+000 1.615103e+000 3.407232e+000 3.716928e+000
1.022502e+000
<GCONST> 8.281708e+001
<STATE> 3
<MEAN> 26
 -9.397676e+000 -5.465613e+000 -4.520076e+000 -4.241547e+000 -
8.612263e+000 -1.144513e+001 -1.162399e+001 4.743148e+000 4.271320e+000
1.260309e+000 -3.463299e+000 -7.337816e+000 7.046229e+001 1.562827e+000 -
1.147295e+000 -1.006993e-001 -2.775139e-001 -2.927693e+000 -3.464480e+000
-2.713372e-001 2.020875e-001 -6.128667e-001 1.379787e-001 2.761989e-001 -
1.271762e+000 1.735408e+000
<VARIANCE> 26
 1.660004e+001 8.049737e+000 1.066751e+001 2.226658e+001 2.197133e+001
2.712201e+001 3.313220e+001 2.140391e+001 3.305353e+001 3.377639e+001
3.670199e+001 1.192909e+001 1.335009e+001 1.343888e+000 7.599666e-001
2.172714e+000 1.996083e+000 5.778028e+000 1.387627e+000 5.295483e+000
2.366864e+000 5.178859e+000 2.497809e+000 3.548870e+000 2.570898e+000
6.265485e-001
<GCONST> 9.719720e+001
<STATE> 4
<MEAN> 26
 -9.705480e+000 -7.242923e+000 -1.091362e-001 -3.549276e+000 -
1.731609e+001 -1.559536e+001 -4.991346e+000 2.831232e+000 -3.419055e+000
-4.377526e-001 -9.693274e-001 -7.225358e+000 7.204796e+001 -8.230600e-001
4.347934e-001 3.115084e+000 -2.612282e+000 -2.016066e+000 4.208236e-001
6.068296e-001 -2.101361e+000 -1.659336e+000 -4.990323e-001 2.968508e-002
1.378660e-002 7.828676e-001
<VARIANCE> 26
 5.779384e+000 1.083796e+001 3.040736e+001 2.139844e+001 2.962233e+001
2.597064e+001 4.403889e+001 1.674278e+001 3.365986e+001 2.246023e+001
1.841321e+001 1.441214e+001 9.797752e+000 1.707295e+000 9.545355e-001
3.674084e+000 4.323785e+000 2.610349e+000 2.812549e+000 3.269243e+000
2.978035e+000 4.618266e+000 5.018836e+000 1.424042e+000 2.508116e+000
4.421657e-001
<GCONST> 9.726342e+001
<STATE> 5
<MEAN> 26
 -1.365866e+001 -1.929538e+000 6.979832e+000 -1.430730e+001 -
1.412801e+001 -8.639726e+000 -8.399387e+000 -1.295154e+000 -1.125669e+000
-5.853367e-001 -2.376741e+000 -6.879067e+000 7.075026e+001 -9.991376e-001
1.578249e+000 1.536443e+000 -2.449330e+000 1.464374e+000 9.580885e-001 -
1.560057e+000 -8.340742e-001 9.158573e-001 3.322217e-002 -1.447296e-001
1.200484e-001 -1.896640e-001
```

```
<VARIANCE> 26
 6.042296e+000 1.413212e+001 2.776913e+001 3.104017e+001 1.691096e+001
1.547863e+001 1.506751e+001 3.925821e+001 5.685538e+001 3.848262e+001
1.177211e+001 1.746853e+001 1.281000e+001 4.542787e-001 8.475707e-001
2.861305e+000 4.032212e+000 2.117890e+000 9.915286e-001 2.250374e+000
3.899936e+000 2.490685e+000 1.571561e+000 1.413961e+000 1.256075e+000
3.277522e-001
<GCONST> 9.173711e+001
<TRANSP> 6
 0.000000e+000 1.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
0.000000e+000
 0.000000e+000 6.569040e-001 3.430960e-001 0.000000e+000 0.000000e+000
0.000000e+000
 0.000000e+000 0.000000e+000 5.004721e-001 4.995280e-001 0.000000e+000
0.000000e+000
 0.000000e+000 0.000000e+000 0.000000e+000 4.740334e-001 4.821269e-001
4.383966e-002
 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 6.356064e-001
3.643936e-001
 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
0.000000e+000
<ENDHMM>
```

```
~o
<STREAMINFO> 1 26
<VECSIZE> 26<NULLD><MFCC_D_0>
~h "four"
<BEGINHMM>
<NUMSTATES> 6
<STATE> 2
<MEAN> 26
 -1.627881e+001 -3.011568e+000 -1.949039e+000 -3.063527e+000 -
2.336705e+000 -5.302733e-001 -3.600921e+000 1.018318e+000 -1.526362e-001
2.599598e-001 -5.826283e-001 -1.400157e+000 6.398005e+001 1.258838e+000
1.795061e-001 -1.552856e+000 -2.272090e+000 -1.211087e+000 7.133953e-002
3.642862e-002 2.512510e-001 -4.260846e-001 -4.672758e-001 -1.757437e-001 -
4.601864e-001 1.064999e+000
<VARIANCE> 26
 1.749035e+000 4.023590e+000 4.205657e+000 6.303103e+000 7.942510e+000
8.882433e+000 9.674199e+000 1.133843e+001 9.826317e+000 9.065825e+000
1.203673e+001 1.011706e+001 1.362126e+000 1.765607e+000 8.856192e-001
1.947558e+000 4.598486e+000 1.796589e+000 1.060590e+000 1.634676e+000
1.766383e+000 1.334566e+000 1.205915e+000 1.868878e+000 2.246863e+000
8.360748e-001
<GCONST> 7.764438e+001
<STATE> 3
<MEAN> 26
 -7.093392e+000 -2.634126e-001 -1.114525e+001 -1.749482e+001 -
1.049545e+001 1.839145e-001 -1.266364e+000 1.131771e+000 -2.421358e+000 -
3.147420e+000 -3.107265e+000 -4.468002e+000 6.984245e+001 2.610960e+000
6.825780e-001 -2.680496e+000 -4.390880e+000 -1.956242e+000 -2.852800e-001
1.154530e-001 -3.559533e-001 -3.541156e-001 -7.695479e-001 -9.667392e-001
-9.592305e-001 1.525377e+000
<VARIANCE> 26
 2.972254e+000 7.896070e+000 8.842253e+000 8.560227e+000 1.344623e+001
1.328503e+001 9.999383e+000 1.949554e+001 6.615376e+000 1.088920e+001
1.245050e+001 1.827241e+001 5.828697e+000 4.481998e-001 1.056233e+000
8.540531e-001 1.736105e+000 2.541788e+000 1.049151e+000 1.773394e+000
1.839238e+000 1.050639e+000 1.351523e+000 1.686829e+000 1.013314e+000
4.985041e-001
<GCONST> 7.919207e+001
```

<STATE> 4
<MEAN> 26
 -7.094132e+000 -2.961530e+000 -1.304812e+001 -2.017257e+001 -6.790008e+000
7.827770e-001 -3.226109e+000 1.871748e+000 -3.358403e-001 -1.777405e+000 -
3.338901e+000 -5.345194e+000 7.125696e+001 2.866666e-001 -1.574566e+000 -
9.510923e-001 4.022750e-001 5.416180e-001 -1.301810e-001 -7.599322e-001 -
3.062473e-001 1.073932e+000 7.697008e-001 -8.438925e-001 -1.385528e+000
6.487008e-001
<VARIANCE> 26
 2.932144e+000 1.345197e+001 1.049220e+001 1.466341e+001 5.052429e+001
1.956160e+001 1.309370e+001 2.447684e+001 3.181601e+001 1.932623e+001
9.445696e+000 1.910161e+001 8.898386e+000 5.203609e-001 1.374331e+000
1.609220e+000 7.381124e+000 1.901984e+000 2.088541e+000 1.664318e+000
1.546742e+000 3.435996e+000 2.740840e+000 2.325952e+000 2.512599e+000
6.562636e-001
<GCONST> 9.101534e+001
<STATE> 5
<MEAN> 26
 -8.796195e+000 -1.053999e+001 -1.244184e+001 -6.746072e+000 -5.957185e+000
-4.399360e+000 -4.540878e+000 1.846191e+000 3.715687e+000 -1.487987e+000 -
7.813820e+000 -9.207281e+000 7.096523e+001 -4.266554e-001 -1.388512e+000
9.149027e-001 3.186145e+000 -8.248769e-001 -1.870671e+000 1.042896e+000
9.637964e-001 -4.018746e-003 -2.697023e-001 -3.653831e-001 1.671164e-001 -
9.587226e-002
<VARIANCE> 26
 5.793354e+000 1.865162e+001 1.504935e+001 2.807839e+001 1.524545e+001
1.847066e+001 2.852164e+001 2.238066e+001 3.773396e+001 4.038546e+001
2.304316e+001 3.488585e+001 9.038031e+000 3.612860e-001 1.394188e+000
9.094784e-001 2.888305e+000 3.291317e+000 2.751932e+000 2.313128e+000
3.420772e+000 6.951764e+000 4.178690e+000 3.278042e+000 4.234923e+000
3.826282e-001
<GCONST> 9.642402e+001
<TRANSP> 6
 0.000000e+000 1.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
0.000000e+000
 0.000000e+000 6.756741e-001 2.972979e-001 2.702804e-002 0.000000e+000
0.000000e+000
 0.000000e+000 0.000000e+000 4.120785e-001 5.879215e-001 0.000000e+000
0.000000e+000
 0.000000e+000 0.000000e+000 0.000000e+000 6.624963e-001 2.812530e-001
5.625082e-002
 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 6.636962e-001
3.363038e-001
 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
0.000000e+000
<ENDHMM>

```
~o
<STREAMINFO> 1 6
<VECSIZE> 6<NULLD><LPREFC>
~h "one"
<BEGINHMM>
<NUMSTATES> 6
<STATE> 2
<MEAN> 6
 5.261919e-001 2.791538e-001 1.112432e-001 2.641640e-002 9.472515e-002
1.580122e-001
<VARIANCE> 6
 4.272612e-003 1.279189e-003 3.931223e-003 2.562686e-004 7.863412e-004
7.203632e-004
<GCONST> -2.928176e+001
<STATE> 3
<MEAN> 6
 5.241870e-001 3.082307e-001 3.110062e-001 1.105274e-001 5.801334e-002
1.396900e-001
<VARIANCE> 6
 1.341099e-003 5.826447e-004 2.691571e-003 9.564859e-004 2.775982e-004
1.180505e-004
<GCONST> -3.313855e+001
<STATE> 4
<MEAN> 6
 6.011736e-001 3.884170e-001 3.150459e-001 9.346889e-002 1.201110e-001
1.748371e-001
<VARIANCE> 6
 4.286758e-003 8.378816e-004 5.203939e-003 7.263413e-004 3.122827e-004
1.103302e-003
<GCONST> -2.887648e+001
<STATE> 5
<MEAN> 6
 5.983796e-001 3.617101e-001 4.339383e-001 1.405495e-001 8.149461e-002
1.396660e-001
<VARIANCE> 6
 1.445916e-003 2.272955e-003 3.474787e-003 1.284511e-003 6.520772e-004
6.247687e-004
<GCONST> -2.863150e+001
<TRANSP> 6
 0.000000e+000 1.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
0.000000e+000
 0.000000e+000 6.966650e-001 1.632253e-001 1.401097e-001 0.000000e+000
0.000000e+000
 0.000000e+000 0.000000e+000 6.525111e-001 0.000000e+000 3.474889e-001
0.000000e+000
 0.000000e+000 0.000000e+000 0.000000e+000 4.294301e-001 5.705699e-001
0.000000e+000
 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 7.218542e-001
2.781458e-001
 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
0.000000e+000
<ENDHMM>
```

```
~o
<STREAMINFO> 1 6
<VECSIZE> 6<NULLD><LPREFC>
~h "two"
<BEGINHMM>
<NUMSTATES> 6
<STATE> 2
<MEAN> 6
 5.879399e-001 3.299402e-001 3.577384e-001 1.260061e-001 6.496946e-002
1.389647e-001
<VARIANCE> 6
 9.366721e-004 1.152507e-003 2.621015e-003 5.606713e-004 9.234042e-004
3.358932e-004
<GCONST> -3.112846e+001
<STATE> 3
<MEAN> 6
 5.963721e-001 3.849380e-001 2.990098e-001 1.318672e-001 8.208963e-002
1.709812e-001
<VARIANCE> 6
 3.813021e-003 9.643050e-004 1.058530e-003 1.158107e-003 1.905121e-004
3.259784e-004
<GCONST> -3.169249e+001
<STATE> 4
<MEAN> 6
 5.528838e-001 2.932698e-001 2.699898e-001 9.129632e-002 6.588741e-002
1.360861e-001
<VARIANCE> 6
 1.670192e-003 6.264258e-004 1.201546e-003 5.277921e-004 5.970438e-004
1.608081e-004
<GCONST> -3.317281e+001
<STATE> 5
<MEAN> 6
 5.468522e-001 3.043408e-001 1.869351e-001 5.197767e-002 9.126534e-002
1.610978e-001
<VARIANCE> 6
 1.806795e-003 2.797203e-003 2.210600e-003 2.791875e-004 6.026127e-004
1.145586e-003
<GCONST> -2.965227e+001
<TRANSP> 6
 0.000000e+000 6.666662e-001 3.333338e-001 0.000000e+000 0.000000e+000
0.000000e+000
 0.000000e+000 8.000754e-001 0.000000e+000 1.999246e-001 0.000000e+000
0.000000e+000
 0.000000e+000 0.000000e+000 8.703413e-001 0.000000e+000 1.296587e-001
0.000000e+000
 0.000000e+000 0.000000e+000 0.000000e+000 5.778784e-001 4.221216e-001
0.000000e+000
 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 6.151728e-001
3.848272e-001
 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
0.000000e+000
<ENDHMM>
```

```
~o
<STREAMINFO> 1 6
<VECSIZE> 6<NULLD><LPREFC>
~h "three"
<BEGINHMM>
<NUMSTATES> 6
<STATE> 2
<MEAN> 6
 6.070203e-001 3.419144e-001 3.807999e-001 1.236697e-001 7.017113e-002
1.480736e-001
<VARIANCE> 6
 8.368058e-004 1.373102e-003 1.574828e-003 8.076958e-004 7.763845e-004
4.385416e-004
<GCONST> -3.111719e+001
<STATE> 3
<MEAN> 6
 5.667626e-001 2.843804e-001 2.846412e-001 6.315774e-002 6.749388e-002
1.537288e-001
<VARIANCE> 6
 3.723953e-003 1.377147e-003 5.713852e-003 5.141699e-004 7.021169e-004
2.575832e-004
<GCONST> -2.941684e+001
<STATE> 4
<MEAN> 6
 5.421068e-001 2.991324e-001 1.912921e-001 4.935893e-002 9.594698e-002
1.538265e-001
<VARIANCE> 6
 2.207711e-003 2.905767e-003 4.495754e-003 5.072795e-004 8.053027e-004
8.657604e-004
<GCONST> -2.809686e+001
<STATE> 5
<MEAN> 6
 6.001583e-001 3.605994e-001 3.736943e-001 1.276026e-001 8.459947e-002
1.483973e-001
<VARIANCE> 6
 3.592989e-003 1.963386e-003 3.238138e-003 9.086643e-004 8.990837e-004
3.765505e-004
<GCONST> -2.846948e+001
<TRANSP> 6
 0.000000e+000 5.448195e-001 4.551804e-001 0.000000e+000 0.000000e+000
0.000000e+000
 0.000000e+000 7.238023e-001 1.548865e-001 1.213112e-001 0.000000e+000
0.000000e+000
 0.000000e+000 0.000000e+000 6.077678e-001 2.652165e-001 1.270157e-001
0.000000e+000
 0.000000e+000 0.000000e+000 0.000000e+000 7.649663e-001 2.350337e-001
0.000000e+000
 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 5.484757e-001
4.515244e-001
 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
0.000000e+000
<ENDHMM>
```

```
~o
<STREAMINFO> 1 6
<VECSIZE> 6<NULLD><LPREFC>
~h "four"
<BEGINHMM>
<NUMSTATES> 6
<STATE> 2
<MEAN> 6
 5.805441e-001 2.016584e-001 2.651050e-001 4.586024e-002 5.112455e-002
1.046736e-001
<VARIANCE> 6
 2.939508e-003 1.121271e-003 2.003668e-003 2.419711e-004 6.523673e-004
6.025459e-004
<GCONST> -3.088426e+001
<STATE> 3
<MEAN> 6
 5.379782e-001 2.670473e-001 2.068235e-001 5.867174e-002 7.349499e-002
1.348806e-001
<VARIANCE> 6
 1.500141e-003 8.759851e-004 5.218763e-003 3.837875e-004 5.606516e-004
8.443020e-004
<GCONST> -3.019942e+001
<STATE> 4
<MEAN> 6
 5.098748e-001 3.560259e-001 1.705821e-001 7.390711e-002 1.132449e-001
1.688739e-001
<VARIANCE> 6
 1.002583e-003 7.648147e-004 1.230410e-003 3.517471e-004 1.811901e-004
1.421408e-004
<GCONST> -3.518145e+001
<STATE> 5
<MEAN> 6
 5.594254e-001 3.384856e-001 2.873854e-001 9.909055e-002 9.191351e-002
1.474816e-001
<VARIANCE> 6
 1.401326e-003 1.815350e-003 1.565087e-003 8.780803e-004 9.927504e-004
5.070865e-004
<GCONST> -2.985400e+001
<TRANSP> 6
 0.000000e+000 6.596522e-001 3.403479e-001 0.000000e+000 0.000000e+000
0.000000e+000
 0.000000e+000 7.163876e-001 2.836125e-001 0.000000e+000 0.000000e+000
0.000000e+000
 0.000000e+000 0.000000e+000 6.550486e-001 8.628990e-002 2.586615e-001
0.000000e+000
 0.000000e+000 0.000000e+000 0.000000e+000 8.270584e-001 1.729416e-001
0.000000e+000
 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 7.388149e-001
2.611851e-001
 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
0.000000e+000
<ENDHMM>
```

La routine de calcul de vraisemblance(audio+vidéo) :

```
{ calcule la vraissemblance conjointe vid aud}
program viterbi;
uses crt;
type fichier1=file of single;fichier2=file of real;
    nom=string;fichier=file of integer;
    vect1=array[1..26] of single;vect2=array[1..6] of real;
     var f2:fichier2; f11:fichier1;f12:fichier;f3,f4:text;
         mat:array[1..6,1..6] of real;
         mu1,cov1:array[1..6,1..26] of real; mu2,cov2:array[1..6,1..6] of
real;
         nom1,nom2,noma,nomsca,nomscv:nom;
         alpha1,alpha:array[1..6] of real;
         V1:vect1;v2:vect2;vra,vrv,vr:array[1..4,1..12] of
real;max:real;indice,i,j,z:integer;
        vecteura,vecteurv:array[1..12] of nom;lamda:real;
function prob1(v:vect1;k:integer):real;
var i:integer; p:real;
begin
p:=0;
for i:=1 to 26 do
p:=p-(ln(sqrt(6.28*cov1[k,i])))-((sqr(v1[i]-mu1[k,i]))/(2*(cov1[k,i])));
prob1:=p;
end;
function prob2(v:vect2;k:integer):real;
var i:integer; p:real;
begin
p:=0;
for i:=1 to 6 do
p:=p-(ln(sqrt(6.28*cov2[k,i])))-((sqr(v[i]-mu2[k,i]))/(2*(cov2[k,i])));
prob2:=p;
end;
function lhd1(nom1,nom2:nom):real;
var ind,i,i1,j:integer;max:real;
begin
assign(f2,nom2);
reset(f2);
 i:=2;
while i<=5 do {chargement de mu et cov}
begin
for j:=1 to 26 do
read(f2,mu1[i,j]);
for j:=1 to 26 do
 read(f2,cov1[i,j]);
i:=i+1;
end;
for i:=1 to 6 do { chargement de mat}
for j:=1 to 6 do
read(f2,mat[i,j]);
{writeln('affichage de mat:');
for i:=1 to 6 do
begin for j:=1 to 6 do write(mat[i,j]:8);writeln;end;readkey}
;
assign(f11,nom1);
reset(f11);{ouverture du fichier … reconnaitre}
for i:=1 to 6 do alpha1[i]:=0;
for i:=1 to 26 do
read(f11,v1[i]);
for i:=2 to 5 do
if mat[1,i]<>0 then begin alpha1[i]:=ln(mat[1,i])+prob1(v1,i);end;
while not eof(f11) do
```

```
begin
   for i:=1 to 26 do {lecture de l'observation Ot}
    read(f11,v1[i]);
   for i:=2 to 5 do
     begin
       for i1:=2 to 5 do if (alpha1[i1]*mat[i1,i])<>0 then begin
max:=alpha1[i1]+ln(mat[i1,i]);ind:=i1; end;
       for j:=2 to 5 do
       if( (alpha1[j]*mat[j,i] <>0) and (alpha1[j]+ln(mat[j,i]) > max))
then begin max :=alpha1[j]+ln(mat[j,i]);ind:=j;end;
        if (mat[ind,i]<>0) then
alpha[i]:=prob1(v1,i)+alpha1[ind]+ln(mat[ind,i]);
     end;
   for i:=2 to 5 do begin alpha1[i]:=alpha[i];alpha[i]:=0;end;
end;
for i:=1 to 6 do if (alpha1[i]*mat[i,6]<>0) then
max:=alpha1[i]+ln(mat[i,6]);
for i:=2 to 6 do
if (((alpha1[i]*mat[i,6])<>0)and ((alpha1[i]+ln(mat[i,6])) >max)) then
max:=alpha1[i]+ln(mat[i,6]);
lhd1:=max;
close(f2);close(f11);
end;
function lhd2(nom1,nom2:nom):real;
var ind,i,i1,j,y:integer;max:real;x:real;
begin
assign(f2,nom2);
reset(f2);
 i:=2;
while i<=5 do {chargement de mu et cov}
begin
for j:=1 to 6 do
begin read(f2,x);mu2[i,j]:=x;end;
for j:=1 to 6 do
begin read(f2,x);cov2[i,j]:=x;end;
i:=i+1;
end;
for i:=1 to 6 do { chargement de mat}
for j:=1 to 6 do
begin read(f2,x);mat[i,j]:=x ;end;
{writeln('affichage de mat:');
for i:=1 to 6 do
begin for j:=1 to 6 do write(mat[i,j]:8);writeln;end;}

assign(f12,nom1);
reset(f12);{ouverture du fichier … reconnaitre}
for i:=1 to 6 do alpha1[i]:=0;
for i:=1 to 6 do
begin read(f12,y);v2[i]:=y/128;end;
for i:=2 to 5 do
if mat[1,i]<>0 then begin alpha1[i]:=ln(mat[1,i])+prob2(v2,i);end;
while not eof(f12) do
begin
   for i:=1 to 6 do {lecture de l'observation Ot}
    begin read(f12,y);v2[i]:=y/128;end;
   for i:=2 to 5 do
     begin
for i1:=1 to 6 do if (alpha1[i1]*mat[i1,i])<>0 then begin
max:=alpha1[i1]+ln(mat[i1,i]);ind:=i1; end;
       for j:=1 to 5 do
       if( (alpha1[j]*mat[j,i] <>0) and (alpha1[j]+ln(mat[j,i]) > max))
then begin max :=alpha1[j]+ln(mat[j,i]);ind:=j;end;
        if (mat[ind,i]<>0) then
alpha[i]:=prob2(v2,i)+alpha1[ind]+ln(mat[ind,i]);
     end;
```

```
for i:=2 to 5 do begin alpha1[i]:=alpha[i];alpha[i]:=0;end;
end;    for i:=2 to 5 do if (alpha1[i]*mat[i,6]<>0) then
max:=alpha1[i]+ln(mat[i,6]);
for i:=2 to 6 do
if (((alpha1[i]*mat[i,6])<>0)and ((alpha1[i]+ln(mat[i,6])) >max)) then
max:=alpha1[i]+ln(mat[i,6]);
lhd2:=max;
close(f12);close(f2);
end;

begin{progprincipal}
clrscr;
writeln('donner nom du script audio');
readln(nomsca);
writeln('donner nom de script video ');
readln(nomscv);
assign(f3,nomsca);
assign(f4,nomscv);
reset(f3);reset(f4);
for z:=1 to 12 do
readln(f3,vecteura[z]);
for z:=1 to 12 do
readln(f4,vecteurv[z]);

for i:=1 to 12 do
begin
vra[1,i]:= lhd1(vecteura[i],'hmma1');{writeln('vraissemblance=',vr[1,i]);}
vra[2,i]:= lhd1(vecteura[i],'hmma2');{writeln('vraissemblance=',vr[2,i]); }
vra[3,i]:= lhd1(vecteura[i],'hmma3');{writeln('vraissemblance=',vr[3,i]);}
vra[4,i]:= lhd1(vecteura[i],'hmma4');{writeln('vraissemblance=',vr[4,i]);}

{writeln(indice);}
vrv[1,i]:= lhd2(vecteurv[i],'hmmv1');{write(vra[1]);write('
',vrv[1]);writeln;}
vrv[2,i]:= lhd2(vecteurv[i],'hmmv2');{write(vra[2]);write('
',vrv[2]);writeln; }
vrv[3,i]:= lhd2(vecteurv[i],'hmmv3');{write(vra[3]);write('
',vrv[3]);writeln;}
vrv[4,i]:= lhd2(vecteurv[i],'hmmv4');{write(vrv[4]);write('  ',
vra[4]);writeln;  }
end;
writeln('donner lamda:');
readln(lamda);
for i:=1 to 4 do
for j:=1 to 12 do
vr[i,j]:=(lamda*vra[i,j])+((1-lamda)*vrv[i,j]);
for i:=1 to 12 do
begin
max:=vr[1,i];indice:=1;
for j:=2 to 4 do
if vr[j,i]>max then begin indice:= j;max:=vr[j,i];end;
write(indice:8);
end;

readkey;
end.
```