الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي و البحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البليدة
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique

Mémoire de Master II

Mention Électronique
Spécialité Automatique

présenté par

RAMDANI Ridha

Commande vocale d'une plateforme mobile

Proposé par :

Mr Z. IRKI (Maître de conférences B/MCB)

&

Mr A. NEMRA (Maître de conférences A/MCA)

Année Universitaire 2016-2017

Remerciements

Je remercie « DIEU » pour m'avoir guidé et aidé à terminer ce mémoire.

Avec beaucoup de gratitude et de sincérité, je remercie vivement mon promoteur le Docteur Mr Z. IRKI pour sa présence scientifique et humaine ainsi que pour tout le soin qu'il apporte à nous diriger vers des sujets d'actualité.

Je remercie également les Docteurs Mr A. NEMRA et Mr M.Djellab, pour avoir aidé dans ce travail. Qu'ils trouvent ici l'expression de mon profond respect.

Mes remerciements s'adressent au même titre aux professeurs du département d'Électronique de l'université de Blida.

Finalement, je remercie toute personne ayant contribué de près ou de loin à l'accomplissement de ce travail.

ملخص: يصف هذا العمل مجموعة متنوعة من النهج التي من خلالها يتم التعرف الألي على التحدث بالكلمات المنفردة و هذا باستخدام معاملات "ميل" لتردد النطاق (MFCC) وخوارزميات التصنيف و المتمثلة في التشوه الديناميكي في الزمن (DTW) ، تكمية الشعاع (VQ) ، نموذج ماركوف غير المكشوف (HMM) وكذا نموذج الخلائط القوسية (GMM). يتم جمع قاعدة بيانات تجريبية من مجموعة من متكلمين، يتحدث كل منهم 5 كلمات في ظروف واقعيه. يتم استخراج معاملات MFCC من إشارة الكلام للكلمات المنطوقة .يتم استخدام مختلف خوارزميات التصنيف من أجل مقارنة معطيات التمرين و التجريب و الحكم فيما بعد على الكلمة الأكثر مطابقة وأخيرا نقوم باستخدام هذا النظام للسيطرة على نموذج مماثل لروبوت متحرك.

كلمات مفتاحية:
التعرف الألي على الكلام ، التعرف الألي على التحدث بالكلمات المنفردة، معاملات ميل تردد النطاق (MFCC) ، التشوه الديناميكي في الزمن (DTW)، تكمية الشعاع (VQ) ، نموذج الخلائط القوسية (GMM) ، نموذج ماركوف غير المكشوف (HMM).

Résumé :

Ce travail décrit différentes approches de reconnaissance de la parole de mots isolés IWR en utilisant la représentation par coefficients MFCC (Mel-Scale Fréquence Cepstral Coefficients) pour l'extraction des vecteurs des caractéristiques et pour la classification il a été opté pour quatre méthodes à savoir l'alignement temporel dynamique (Dynamic Time Warping DTW) la quantification Vectorielle (VQ), Modèle de Markov caché (HMM), Mélange de Gaussiennes GMM. Une base de données expérimentale est recueillie de plusieurs locuteurs, parlant cinq mots chacun, enregistrés dans des conditions réelles et avec différents types de microphones. Les approches de reconnaissance vocale sont utilisées et testées pour la mesure de similarité entre les séquences d'apprentissage et de test pour décider à la fin quel mot était prononcé. Ce système de RAP est utilisé pour commander une simulation d'une plateforme mobile.

Mots clés :

Reconnaissance automatique de la parole, IWR, coefficients MFCC, l'alignement temporel dynamique (DTW) la quantification Vectorielle (VQ), Modèle de Markov caché (HMM), Mélange de Gaussiennes GMM,

Abstract:

This work describes different approaches of isolated speech recognition IWR by using the Mel-Scale Frequency Cepstral Coefficients (MFCC) for the extraction of the features vectors, for their classification many methods are used to know: Dynamic Time Warping (DTW), Vector quantization (VQ), Hidden Markov Model (HMM) and Gaussian Mixtures Model (GMM). An experimental database of many speakers, speaking five words each, is collected under real conditions of recording. MFCC are extracted from speech signal of spoken words. The classification algorithms are used for measuring similarity between test and train sequences to decide at last which word was pronounced .This ASR system is used to control a simulated mobile robot.

Keywords:

Automatic Speech Recognition, Mel-Scale Frequency Cepstral Coefficients (MFCC), Dynamic Time Warping (DTW), Vector quantization (VQ), Gaussian Mixtures Model (GMM) and Hidden Markov Model (HMM) .

Table des matières :

Liste des Figures :

Liste des tableaux :

Acronyms

ANN	**A**rtificial **N**eural **N**etwork.
DFT	**D**iscrete **F**ourier **T**ransform.
DCT	**D**iscrete **C**osine **T**ransform.
DTW	**D**ynamic **T**ime **W**arping.
FFT	**F**ast **F**ourier **T**ransform.
GMM	**G**aussian **M**ixtures **M**odel.
HFCC	**H**umain **F**acteur **C**epstral **C**oefficient.
HMM	**H**iden **M**arkov **M**odel.
HTK	**H**idden **M**arkov **M**odel **T**oolkit.
IDCT	**I**nverse **D**iscrete **C**osine **T**ransform.
IFFT	**I**nverse **F**ast **F**ourier **T**ransform.
IWR	**I**solated **W**ord **R**ecognition**.**
KNN	**K**-**N**EAREST **N**EIGHBOR.
LPC	**L**inear **P**redictive **C**oding.
LPCC	**L**inear **P**rediction **C**epstral **C**oefficient.
MFCC	**M**el **F**requency **C**epstral **C**oefficient.
MMC	**M**odèle de **M**arkov **C**aché.
ML	**M**aximum **L**ikelihood **E**stimation.
MLI	**M**odulation de **L**argeur d'**I**mpulsion.
PLP	**P**erceptual **L**inear **P**rediction.
PLP-RASTA	**PLP**-**R**elative **S**pectra.
TFCT	**T**ransformée de **F**ourier à **C**ourt **T**erme.
RAP	**R**econnaissance **A**utomatique de la **P**arole.
RTOS	**R**eal **T**ime **O**perating **S**ystem.
RISC	**R**educed **I**nstruction **S**et **C**omputing.
SRAP	**S**ystème de **R**econnaissance **A**utomatique de la **P**arole.
SVM	**S**upport **V**ector **M**achine.
TFR	**T**ransformée de **F**ourier **R**apide.
UART	**U**niversal **A**synchronous **R**eceiver **T**ransmitter.
VAD	**V**oice **A**ctivity **D**etection**.**
VQ	**V**ector **Q**uantization.

INTRODUCTION GENERALE

INTRODUCTION GENERALE

Notre monde qui devient de plus en complexe donne parfois des problèmes qui dépassent la capacité intellectuelle de l'être humain et le rend ainsi incapable de prendre de bonnes décisions. Ces problèmes peuvent se produire dans des systèmes créés par l'humain lui-même comme les systèmes de production. Les problèmes les plus difficiles se posent néanmoins avant tout dans les systèmes naturels, qui ne sont ni créés ni vraiment contrôlables par les humains.

Les Systèmes de Reconnaissance Automatique de la Parole (SRAP) sont aujourd'hui bien connus dans le monde de l'automatique et suscitent l'intérêt d'un public de plus en plus large.

Le travail présenté dans ce mémoire de Master II s'inscrit dans le cadre général de la reconnaissance automatique de la parole pour pouvoir commander une plateforme robot mobile avec un vocabulaire restreint de cinq mots en deux langues (Anglais, Français). Pour qu'on puisse créer un système interactif capable de reconnaître nos paroles avec toute ses complexités, on a besoin d'abord de mieux représenter les informations portées sur un tel signal de parole, ensuite d'effectuer des choix fructueux d'une ou de plusieurs méthodes de classification parmi les grandes familles de ces dernières.

Les approches statistiques et les modèles probabilistes sont très utilisés, de nos jours, dans les systèmes de reconnaissance automatique de la parole. Ces approches, notamment celles basées sur les Modèles d'alignement temporel dynamique DTW, de Mélanges de Gaussiennes (GMM) et les Modèles de Markov Cachés (HMM), ont atteint des performances remarquables avec des vocabulaires de plus en plus importants et une robustesse au bruit et à la variabilité des locuteurs de plus en plus grande.

Globalement, nous avons réparti ce mémoire en quatre chapitres :

Le chapitre I expose en générale les principaux axes de la Reconnaissance Automatique de la Parole, le domaine dont nous nous intéressons essentiellement dans ce mémoire.

Dans le chapitre II, nous allons décrire et éclairer au mieux la composition d'un système de reconnaissance automatique de la parole.

Dans le chapitre III ,qui est le chapitre le plus riche et long ,nous détaillons la première étape commune entre tous les modèles étudiés il s’agit de l’extraction les vecteurs des caractéristiques en utilisant la méthode MFCC puis nous abordons les différents Modèles de classification à savoir DTW , HMM , GMM , VQ ainsi que les étapes qui les composent et leurs principaux algorithmes. .

Finalement dans le chapitre VI, on va présenter l’aspect simulation et discussion des résultats obtenus et conclusion générale et perspectives pour des travaux futures.

Chapitre I

GENERALITE SUR LE TRAITEMENT AUTOMATIQUE DE LA PAROLE

GENERALITE SUR LE TRAITEMENT AUTOMATIQUE DE LA PAROLE

Dans la dernière décennie, avec l'avènement des nouvelles technologies, l'industrie des Télécommunications a connu un progrès considérable, notamment dans le domaine de la Reconnaissance Automatique de la Parole avec l'une de ses applications, la commande vocale, qui fait l'objet de ce mémoire. Ce domaine trouve ses applications dans la téléphonie, dans la commande, dans la reconnaissance du locuteur...etc. La reconnaissance automatique des mots isolés IWR (Isolated Word Recognition) est l'un des principaux axes de la reconnaissance de la parole. Cependant, son problème majeur est de réaliser des systèmes de reconnaissances fiables, rapides, non complexes et capables d'être implémentés.

Le présent chapitre a pour objectif de présenter des notions élémentaires et des termes relatifs à la description de la parole. Nous présentons les appareils auditif et phonatoire de l'être humain. Nous présentons ensuite les problèmes dus à la complexité du signal parole : variabilité, non stationnarité, redondance et coarticulation.

Dans la suite de ce chapitre nous présentons les paramètres les plus utilisés en reconnaissance automatique de la parole, qu'on peut décomposer en deux catégories :

- Ceux qui dépendent de la modélisation du système de production de la parole tels que les paramètres LPC et les paramètres LPCC
- Ceux qui dépendent de la perception de l'oreille humaine tels que les paramètres acoustiques MFCC, PLP et PLP-RASTA.

I.1 Processus humain de production de la parole :

Ce processus est un mécanisme très complexe qui se repose sur une interaction entre les systèmes neurologique et physiologique. La parole commence par une activité neurologique. Après, le cerveau dirige les opérations relatives à la mise en action des organes phonatoires. Le fonctionnement de ces organes est bien, quant à lui, de nature physiologique.

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

Le processus de production de la parole peut être résumé en trois étapes essentielles :

- La génération d’un flux d’air qui va être utilisé pour faire naître une source sonore.
- La génération d’une source sonore sous la forme d’une onde quasi-périodique résultant de la vibration des cordes vocales sous la forme d’un bruit résultant d’une constriction ou d’une occlusion du conduit vocal.
- La mise en place des cavités supra glottiques pour obtenir le son désiré.

Les organes intervenants dans la production de la parole sont (Figure I-1) :

- La soufflerie et le vibrateur.
- Le larynx et ces muscles.
- Les cavités du pharynx.
- Les cordes vocales.
- Les cavités supra glottiques : le conduit vocal, le conduit nasal, la mâchoire, les lèvres et la langue.

Trois groupes d'organes assurent les fonctions essentielles dans l'acte de parole, ou phonation :

- L’appareil respiratoire, (diaphragme, poumons, trachées), soufflerie qui fournit l’énergie et la quantité d'air nécessaire.
- le larynx, organe vibrant, où naît le son.
- Le conduit vocal, formé des cavités résonantes supra-laryngées (pharynx, bouche, nez) où s’effectue l'articulation proprement dite par les changements de forme du tractus vocal. Ces changements résultent surtout des mouvements des lèvres, de la langue, du voile du palais (dont l'abaissement fait intervenir une cavité supplémentaire, les fosses nasales) et de la mâchoire inférieure.

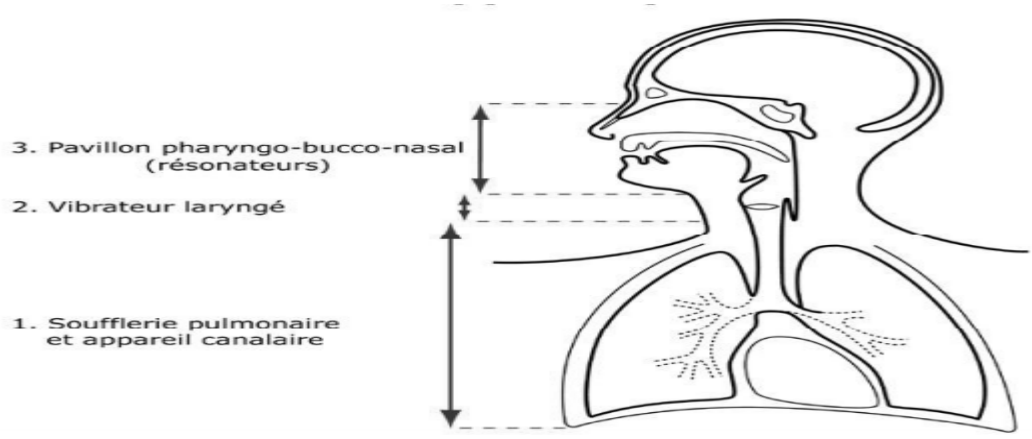

Figure I. 1: Schéma de l’appareil phonatoire.

I.2 **Perception de la parole :**

Dans la pratique, les paramètres utilisés en SRAP dépendent fortement de la perception de l’oreille humaine. Il est nécessaire de comprendre la composition et le fonctionnement du système auditif humain avant de passer à la modélisation de la parole.

La parole est un vecteur de transmission d'information d'une grande complexité. En tant que récepteur de ce vecteur, l'appareil auditif de l'être humain se caractérise par une grande finesse d'analyse de cette complexité et par une grande robustesse à l'environnement. Pour cette raison, de nombreux systèmes de traitement de la parole tentent de reproduire les fonctionnalités de cet appareil. Ce dernier est composé de l'oreille externe, l'oreille moyenne et l'oreille interne. La perception de l'appareil auditif humain à une bande de fréquences qui s'étend entre 800 Hz et 8 KHz et au maximum entre 20Hz et 20KHZ [Boi87].

La Figure II-2 illustre l'appareil auditif, qui est composé de trois parties:

- **Oreille externe**

Le pavillon qui est la grande partie de l'oreille externe, protège l'oreille contre les corps étrangers et permet aussi une localisation du son qui est transmis au tympan à travers le conduit auditif.

- **Oreille moyenne**

L'oreille moyenne est une cavité d'air qui est constituée du tympan et des osselets (le marteau, l'enclume et l'étrier). Ces derniers ont pour rôle de transmettre les vibrations reçus par le tympan au milieu liquide de l'oreille interne. L'oreille moyenne permet aussi de protéger l'oreille interne des sons très forts.

- **Oreille interne**

L'oreille interne est formée d'un milieu liquide. Elle contient la cochlée qui comprend la membrane basilaire. Quand cette dernière reçoit des vibrations, les cellules ciliées, des milliers de cellules, de l'organe de corti situé sur la membrane basilaire déclenchent des influx nerveux au nerf auditif.

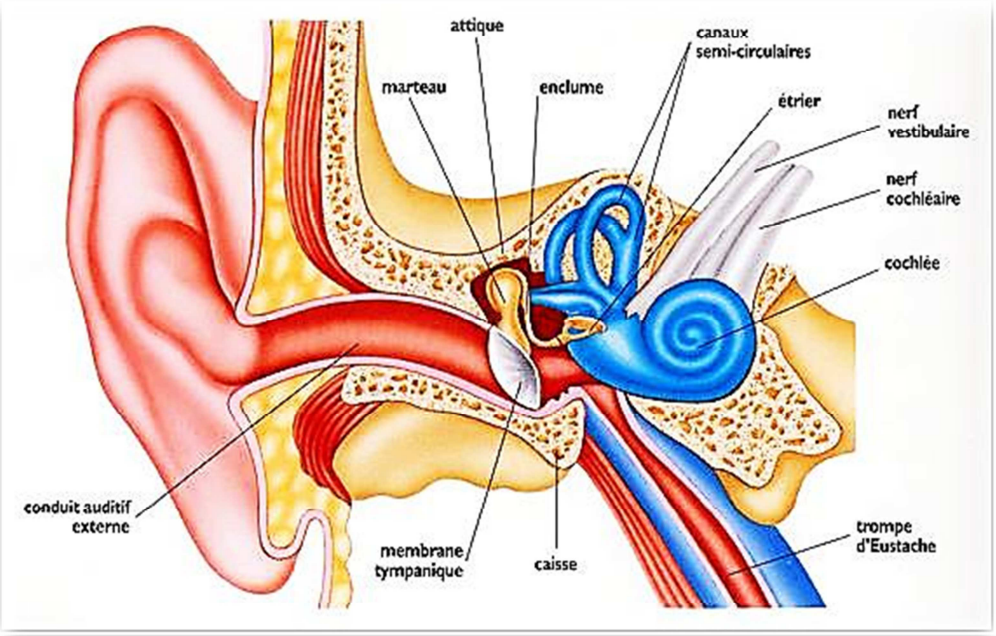

Figure I. 2 Appareil auditif humain.

I.2.1 **Propriétés psycho-acoustiques du système auditif**

L’objectif de la psycho-acoustique est d’étudier les relations quantitatives entre les stimuli acoustiques et les réponses du système auditif de l'être humain.

Les résultats les plus marquants de cette science sont les suivants :

- **Échelle d'intensité**

Le système auditif ne présente pas une sensibilité à l'intensité sonore identique à toutes les fréquences. En effet, des sons d'intensité sonore égale n'auront pas la même sonie (l'intensité perçue) selon qu'ils soient de haute fréquence 10kHz, de basse fréquence 100Hz ou de fréquence moyenne 1kHz. Ainsi, si ces trois sons ont une même intensité de 40dB, les sons de fréquence 100Hz et 10kHz sont plus faiblement perçus que les sons de fréquence 1kHz.

Les courbes d'isotonie représentent les niveaux d'intensité sonore générant une perception auditive d'égale intensité en fonction de la fréquence du son stimulant (Figure. I.3).

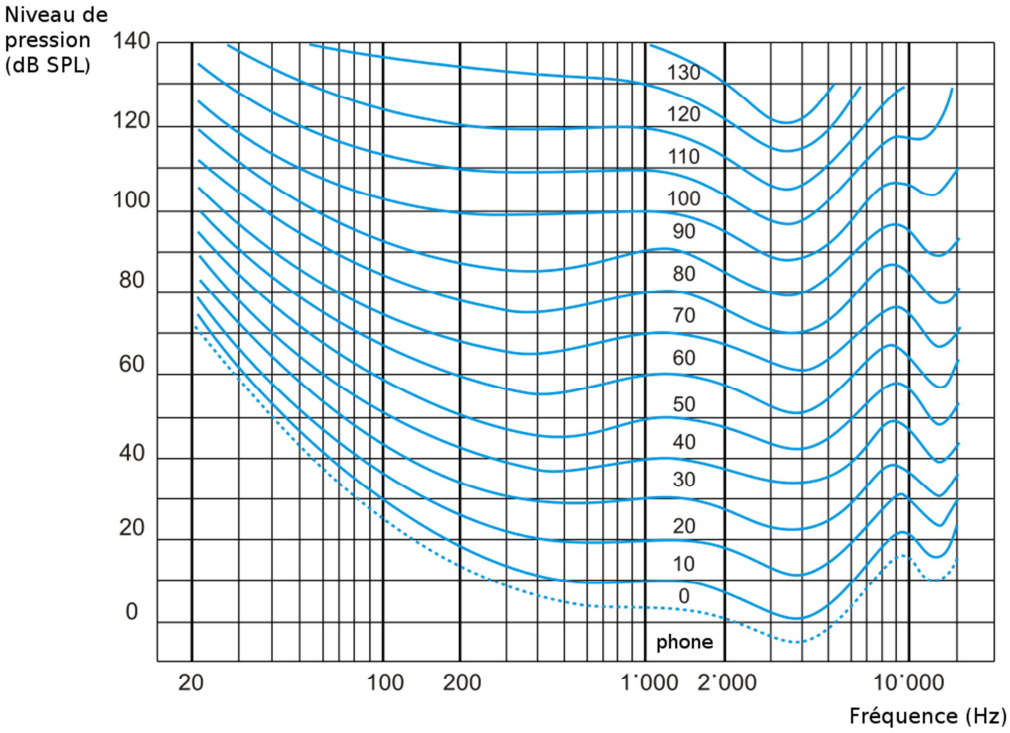

Figure I. 3: Courbes d’isotonie.

- **Échelle d'hauteur :**

La tonie (la hauteur) d'un son est la qualification subjective de sa fréquence. Des études psycho-acoustiques ont en effet montré que la perception humaine du contenu fréquentiel des sons ne suit pas une échelle linéaire mais une échelle fréquentielle de « Mel »[Mud10] . Cette échelle est approximativement linéaire de 20 Hz jusqu'à 1kHz et logarithmique de 1kHz jusqu'à 20kHz

I.3 **Définition de la parole :**

La parole est un signal continu, d’énergie finie, non stationnaire. Sa structure est complexe et variable dans le temps :

- Tantôt périodique (plus exactement pseudopériodique) pour les sons voisés.
- Tantôt aléatoire pour les sons fricatifs.
- Tantôt impulsionnelle dans les phases explosives des sons occlusifs.

L’information portée par le signal de parole peut être analysée par plusieurs façons. On en distingue généralement plusieurs niveaux de description non exclusifs : acoustique, phonétique, phonologique, morphologique, syntaxique, sémantique, et pragmatique.

I.4 **Paramètres du signal parole :**

Le signal vocal est généralement caractérisé par trois paramètres : son énergie, sa fréquence fondamentale et son spectre.

I.4.1 **La fréquence fondamentale :**

Elle représente la fréquence du cycle d'ouverture/fermeture des cordes vocales. Elle varie

- De 80Hz à 200Hz pour une voix masculine,
- De 150Hz à 450Hz pour une voix féminine,
- De 200Hz à 600Hz pour une voix d'enfant.

I.4.2 **L'énergie :**

L'amplitude du signal de la parole varie au cours du temps selon le type du son , son énergie, dans une trame, est donnée par :

$$E = \sum_{n=0}^{N-1} S^2(n) \qquad \text{I. 1}$$

Avec :

S: Un signal de parole.
N: La taille de la trame.
E: Energie du signal.

I.4.3 **Le spectre :**

Le spectre représente l'intensité de la voix selon la fréquence, elle est généralement obtenue par une analyse de Fourier à court terme. La quasi stationnarité du signal de parole permet de mettre en œuvre des méthodes efficaces d'analyse et de modélisation utilisées pour le traitement à court terme du signal vocal sur des fenêtres de durée généralement comprise entre *20ms* et *30ms* appelées trames, avec un recouvrement entre ces fenêtres qui assure la continuité temporelle des caractéristiques de l'analyse. La Transformée de Fourier à Court Terme (*TFCT*) d'un signal échantillonné est par définition la transformée du signal pondéré.

$$\widehat{S}(\mathrm{k}) = \widehat{S}\left(\mathrm{f} = \frac{\mathrm{k}}{\mathrm{N}}\right) = \sum_{\mathrm{n}=0}^{\mathrm{N}-1} \mathrm{S}(\mathrm{n}).\mathrm{W}(\mathrm{n}).\exp\left(-\frac{2\mathrm{J}\pi\mathrm{nk}}{\mathrm{N}}\right) \quad ,0 \leq \mathrm{k} \leq \mathrm{N} \qquad \text{I. 2}$$

Où :

N : la taille de la trame.

$\widehat{S}(k)$: Spectre complexe.

S (n): Segment analysé**.**

W(n) : Fenêtre de temps.

Le spectre de puissance (appelé aussi densité spectrale de puissance de la transformé de Fourier) est donné par :

$$\left|\widehat{S}(k)\right|^2, 0 \leq k \leq \frac{N}{2} \qquad \text{I. 3}$$

I.5 **Traitement de la parole :**

L'enregistrement numérique d'un signal acoustique, requiert successivement : un filtrage de garde, un échantillonnage, une quantification et un codage [Boi00].

I.5.1 **L'échantillonnage :**

L'échantillonnage consiste à prélever les valeurs d'un signal à intervalles définis, en général réguliers. Il produit une suite de valeurs discrètes. Le spectre de la parole peut s'étendre jusque 12 kHz. Il faut donc en principe choisir une fréquence « Fe » égale à 24 kHz au moins pour satisfaire le théorème de Shannon.

I.5.2 **La quantification :**

La quantification consiste à approximer les valeurs réelles des échantillons selon une échelle à plusieurs niveaux appelée échelle de quantification. Le but de la quantification est de réduire la quantité de données sans détériorer les performances de la reconnaissance.

I.5.3 **Le codage :**

Cette étape consiste à établir une représentation binaire des valeurs quantifiées qui rend possible le traitement du signal sur la machine.

I.6 **Modélisation acoustique du signal de parole :**

Tout système de reconnaissance de la parole est divisé en deux étapes, une première partie qui représente la phase d'extraction des paramètres, et une deuxième partie qui est le moteur de reconnaissance. Les performances des systèmes de reconnaissance de la parole dépendent de façon considérable des paramètres acoustiques utilisés.

Un système de paramétrisation du signal a pour rôle de fournir et d'extraire des informations caractéristiques et pertinentes du signal. Il produit ainsi une représentation moins redondante de la parole. Le signal analogique est fourni en entrée et une suite discrète de vecteurs est obtenue en sortie

I.6.1 **Représentation non paramétrique de la parole :**

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

I.6.2 **Représentation paramétrique du signal parole :**

La Figure I.4 représente la décomposition d'un système de paramétrisation acoustique.

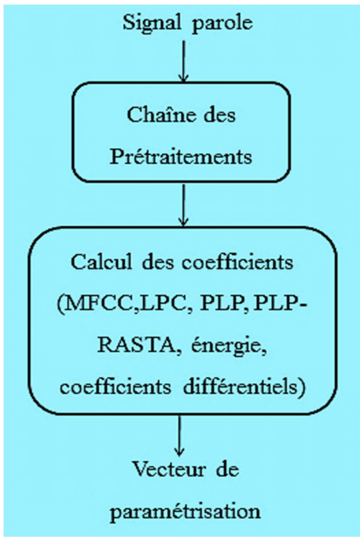

Figure I. 4: Paramétrisation acoustique du signal parole.

Pour résoudre les problèmes liés à la complexité de la parole, il est possible de calculer des coefficients représentatifs du signal traité. Ces coefficients sont calculés à l'intervalle temporel régulier. En simplifiant les choses, le signal de parole est transformé en une série de vecteurs de coefficients.

Ces coefficients doivent représenter au mieux le signal qu'ils sont censés modéliser, et extraire le maximum d'informations utiles pour la reconnaissance.

I.6.2.1 Prétraitement du signal parole

Pour mettre en forme le signal de parole, il est nécessaire d'effectuer quelques opérations avant tout traitement. On procède à une préaccentuation afin de relever les hautes fréquences, qui sont moins énergétiques que les basses fréquences. La préaccentuation de chaque échantillon est calculée par la formule :

$$Y(n) = S(n) - \alpha S(n) \qquad \text{I. 4}$$

Avec α comprise entre 0.9 et 1.

L'étape suivante est de segmenter le signal en trames. Chaque trame est composée de « N » échantillons de parole. En général, « N » est fixé de telle manière que chaque trame corresponde à environs 25 ms de parole.

Enfin on multiplie chaque trame acoustique par une fenêtre de pondération W_n de Hamming afin de réduire les effets de bords (la discontinuité au début et à la fin de chaque trame).
La Figure I.5 illustre l'ensemble des opérations de prétraitement.

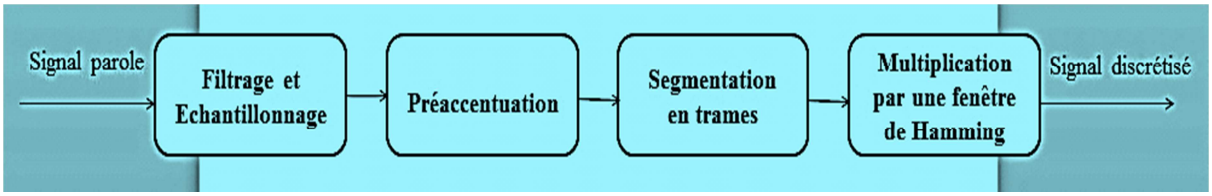

Figure I. 5: Prétraitement du signal parole.

La mise en forme du signal est une étape commune dans la plupart des méthodes d'analyse de la parole, le passage du domaine temporel au fréquentiel est assuré par l'application d'une transformée de Fourier discrète DFT.

I.6.2.2 L'analyse par prédiction linéaire LPC :

La parole peut être considérée comme étant un signal pseudo-stationnaire, ou tout simplement, stationnaire sur de courtes durées allant en général de 15 jusqu' à 30 ms. Sur cette période, il est possible de caractériser le spectre du signal par deux attributs :

- L'enveloppe spectrale.
- La structure fine du spectre.

Le codage par prédiction linéaire se fonde sur la connaissance du modèle fortement simplifié de la production de la parole [Roy90]. Ce modèle peut être décomposé en deux parties : la source active et le conduit passif.

Le codage LPC consiste à estimer le modèle décrivant le conduit, en connaissant le signal d'excitation et il permet ainsi la modélisation du signal $S(n)$ comme suit :

$$S(n) = \sum_{i=1}^{p} a_i S(n-i) + Gu(n) \qquad \text{I. 5}$$

La transformation en Z de $S(n)$ est :

$$S(Z) = \sum_{i=1}^{p} a_i Z^{-i} S(Z) + GU(Z) \qquad \text{I. 6}$$

La fonction de transfert du filtre est :

$$H(Z) = \frac{S(Z)}{GU(Z)} = \frac{1}{1-\sum_{i=1}^{p} a_i Z^{-i}} \qquad \text{I. 7}$$

Les paramètres de ce modèle, à savoir le gain, l'excitation et les coefficients a_i peuvent être estimés par des méthodes d'analyse.

A partir du modèle décrit, une estimation de l'échantillon $S(n)$ peut être calculée par :

$$\hat{S}(n) = \sum_{i=1}^{p} a_i S(n-i) \qquad \text{I. 8}$$

I.6.2.3 Les coefficients LPCC (Linear Prediction Cepstral Coefficients) :

Les paramètres LPCC sont calculés à partir d'une modélisation autorégressive du signal. Soit le modèle autorégressif A (1, $a_{1,}...,a_p$) d'ordre p, estimé sur une trame du signal, les d premiers coefficients cepstraux C_n sont obtenus par :

$$C_n = -a_n + \frac{1}{n}\sum_{i=1}^{n-1}(n-i).a_i.C_{n-1} \qquad 1 \le n \le d \qquad \text{I. 9}$$

Ensuite un filtrage est effectué afin d'assurer la robustesse des coefficients :

$$\forall\, i \in [1, L] \quad W(i) = 1 + \frac{L}{2}\sin\left(\frac{\pi.i}{L}\right) \qquad \text{I. 10}$$

D'où :

$$C_i = \left(1 + \frac{L}{2}\sin\left(\frac{\pi.i}{L}\right)\right) a_i \qquad \text{I. 11}$$

Où L, est le nombre de coefficients.

I.6.2.4 L'analyse MFCC :

Le codage MFCC (Mel Frequency Cepstral Coding) est une technique de codage très utilisée en traitement de la parole. C'est une représentation qu'on retrouve dans des applications très diverses comme la reconnaissance de la parole, du locuteur ou bien de la langue de la locution ou encore dans la discrimination parole/musique.

Le codage MFCC intègre deux notions importantes. La première est la notion de bancs de filtres qui modélisent la membrane basilaire. Ces bancs de filtres sont déployés sur une échelle non linéaire : l'échelle Mel. Cette échelle est issue de connaissances sur la perception humaine.

Pour transformer une fréquence linéaire en une fréquence Mel, on utilise la formule de transformation suivante :

$$\mathrm{Mel(f)} = 2595\log_{10}\left(1+\frac{\mathrm{f}}{700}\right) \qquad \text{I. 12}$$

L'analyse MFCC comporte plusieurs étapes représentées dans la Figure I.6.

Par exemple si on va prendre un exemple numérique; le prétraitement consiste à effectuer sur le signal de parole, échantillonné à 11025 Hz et quantifié sur 16 bits, les opérations suivantes :

- Toutes les 10ms (110 échantillons), une trame acoustique de 25ms (275 échantillons) est extraite du signal.
- La composante continue des échantillons constituant cette trame est enlevée.
- Afin de compenser l'atténuation naturelle du spectre du signal de parole, la séquence des échantillons constituant la trame subit une préaccentuation avec le filtre du premier ordre [Mud10] :

$$\mathrm{H(Z)} = 1 - 0.97\mathrm{Z}^{-1} \qquad \text{I. 13}$$

- Pour atténuer les distorsions spectrales introduites par l'extraction de la trame du signal de parole, on pondère les échantillons de cette trame par la fenêtre de Hamming.

L'analyse MFCC consiste à effectuer sur chacune des trames résultantes du prétraitement les opérations suivantes :

✓ La transformation de Fourier permet de calculer le spectre d'amplitude de la trame.

✓ Pour chacun des 22 filtres triangulaires répartis sur l'échelle des fréquences de Mel, l'énergie du spectre d'amplitude en sortie de ce filtre est calculée. Cette opération donne un vecteur de 22 valeurs énergétiques $\mathrm{E_j}$:

$$\mathrm{E_j} = \sum_{\mathrm{k=0}}^{\mathrm{N-1}} |\mathrm{S(k)}|^2 \mathrm{H_j(k)} \qquad \text{I. 14}$$

✓ Les logarithmes de ces 22 valeurs sont alors transformés en 12 coefficients MFCC par l'inverse de la transformée en cosinus discrète :

$$\mathrm{C_i} = \sqrt{\frac{2}{\mathrm{N}}}\sum_{\mathrm{j=1}}^{\mathrm{N}} \log_{10}\left(\mathrm{E_j}\right)\cos\left(\frac{\pi \mathrm{j}}{\mathrm{N}}(\mathrm{j}+0.5)\right) \qquad \text{I. 15}$$

Où C_i est le i^{eme} coefficient mel-cepstral, E_j est l'énergie du spectre calculée sur la bande passante du j^{eme} filtre, et N et le nombre de filtres (N=22).

- ✓ Afin d'augmenter la robustesse de ces coefficients pour le calcul des distances cepstrales, une pondération en sinus est appliquée sur les coefficients MFCC C_i.

$$\widehat{C}_i = \left(1 + \frac{L}{2}\sin\frac{i\pi}{L}\right) C_i \qquad 1 \leq i \leq 12 \qquad \text{I. 16}$$

Où $\widehat{C}_i$ est le i^{eme} coefficient mel-cepstral liftré et L est le coefficient du liftering (L = 22).

L'organigramme de la Figure I-8 représente les étapes de la paramétrisation du signal parole par coefficients MFCC :

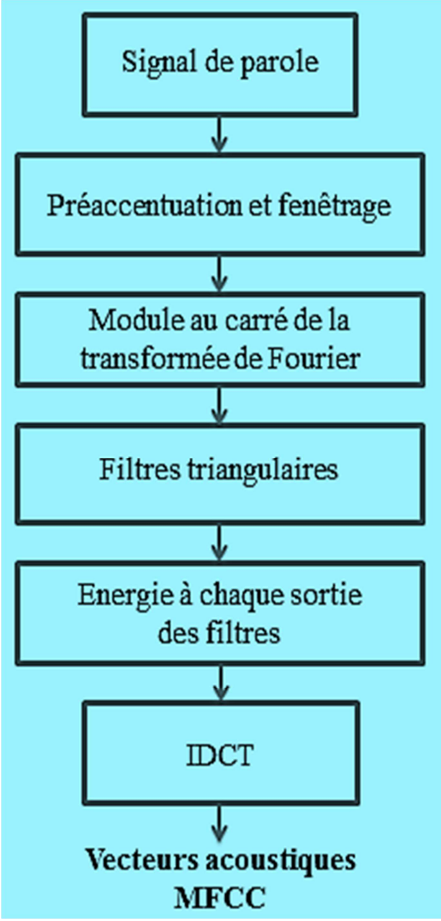

Figure I. 6: Extraction des vecteurs acoustiques MFCC.

I.7 **Conclusion :**

Au cours de ce chapitre, nous avons décrit l’anatomie du système phonatoire et le mécanisme responsable de la production et de la perception de la parole, ensuite nous avons cités les caractéristiques principales du signal de parole et les complexités liés à son traitement, et par la fin nous sommes passés au codage de la parole.

Les méthodes de codage traditionnelles (LPC, LPCC, MFCC et PLP) sont issues de connaissances sur la production et la perception humaine. Dans le cadre du modèle LPC, on cherche à modéliser le conduit vocal alors que le codage MFCC cherche à modéliser l’oreille. Le codage MFCC est la méthode de codage de référence. Il est utilisé dans un grand nombre d’applications : reconnaissance de la parole, du locuteur, de la langue, ...etc.

Le chapitre suivant présentera un état de l’art sur la reconnaissance automatique de la parole.

Chapitre II

ETAT DE L'ART SUR LA RECONNAISSANCE AUTOMATIQUE DE LA PAROLE

ETAT DE L'ART SUR LA RECONNAISSANCE AUTOMATIQUE DE LA PAROLE

La reconnaissance vocale a pour but de permettre à un utilisateur de s'adresser oralement à une machine pour des tâches diverses : transcription, commande, traduction, … ,l'utilisation de la reconnaissance de la parole possède donc un champ d'application très vaste qui nécessite pour la plupart encore des travaux de recherche. L'évolution des technologies a permis l'émergence de solutions entièrement logicielles et grands publics. Cependant, l'implantation d'un système automatique de la parole dans des conditions réelles pose de nombreux problèmes tels que le bruit, l'état du locuteur, la qualité d'enregistrement, la difficulté du vocabulaire, le mode d'élocution, l'intégration du système dans un équipement.

Dans ce chapitre, nous allons décrire et éclairer au mieux la composition d'un système de reconnaissance automatique de la parole, nous commençons par un historique, ensuite quelques définitions nécessaires à la compréhension de ce chapitre. Puis nous allons citer les diverses applications de la reconnaissance automatique de la parole. Et nous allons définir par la suite les techniques et les approches qui sont à la base de la plupart de ces systèmes. Nous terminerons ce chapitre par une vue générale sur les systèmes de reconnaissance de mots isolés et une conclusion.

Le chapitre suivant sera consacré à l'étude et réalisation de plusieurs SRAP par différentes méthodes pour quelques commandes isolées d'un robot mobile (Avance, Recule, Droite, Gauche, Stop). Ces système seront implémentés en se basant sur la paramétrisation acoustique par coefficients MFCC, alors la classification des vecteurs des coefficients sera réalisée par : l'alignement temporel dynamique DTW (Dynamic Time Warping), HMM (Hidden Markov Model) ,GMM (Gaussian Mixtures Model) et VQ (Vector Quantization).

II.1 Historique

La reconnaissance vocale est une discipline quasi contemporaine à l'informatique. Les premiers systèmes électroniques de reconnaissance vocale apparaissent vers 1950. Ils permettent la reconnaissance de chiffres ou de voyelles. Vers 1960, l'apparition des ordinateurs permet un essor de la recherche dans cette discipline. Cependant, les résultats restent modestes car la difficulté du problème a été sous-estimée. Vers 1970, la reconnaissance de la parole s'améliore grâce à l'intégration de contraintes linguistiques et vers la fin des années 70, l'apparition des chaînes de Markov cachées permet la

commercialisation des premiers systèmes de reconnaissance vocale. Avec les performances grandissantes de l'informatique et de l'électronique dans les années 80 et 90, les systèmes de reconnaissance de la parole deviennent de plus en plus fiables.

1952	reconnaissance de dix chiffres, pour un locuteur, par un dispositif électronique câblé.
1965	reconnaissance de phonème en parole continue.
1968	reconnaissance de mots isolés par des systèmes simulés sur gros ordinateurs. (500 mots)
1983	première mondiale de commande vocale à bord d'un avion de chasse en France.
1990	première véritable application de dialogue oral homme-machine.
1996	premières machines à dicter en parole continue commercialisées.

Tableau II 1 Quelques dates importantes dans l'histoire de reconnaissance de la voix

II.2 Reconnaissance automatique de la parole

La reconnaissance automatique de la parole est une technique informatique qui permet d'analyser la voix humaine captée au moyen d'un microphone pour la transcrire sous la forme d'un texte exploitable par une machine.

Le principe général d'un système de RAP peut être décrit par la figure II.1.

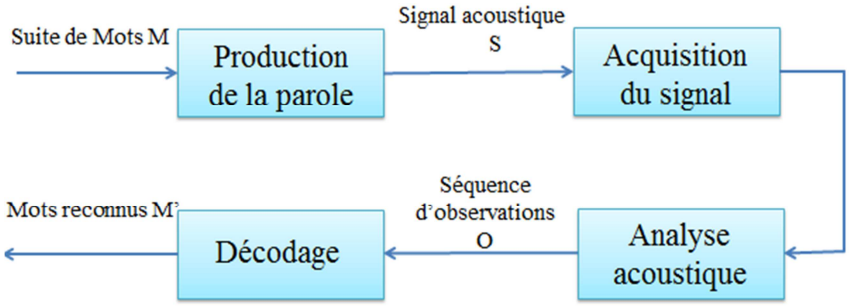

Figure II. 1: Principe de fonctionnement d'un SRAP.

La suite de mots prononcés M est convertie en un signal acoustique S par l'appareil phonatoire. Ensuite le signal acoustique est transformé en une séquence de vecteurs acoustiques ou d'observation O. Finalement, le module de décodage consiste à associer à la séquence d'observations O, une séquence de mots reconnus M', en adaptant une stratégie de comparaison bien définie.

II.2.1. Composition d'un SRAP

Un programme typique de reconnaissance automatique de la parole est composé des étapes suivantes :

- Prétraitement du signal qui inclut le pré-filtrage, le découpage, la quantification, le fenêtrage, la détection de point final et ainsi de suite.
- Extraction des paramètres caractéristiques (MFCC et LPC-CC).
- Choix d'une méthode de classification et de reconnaissance (DTW, VQ, HMM ,GMM).
- Evaluation de la reconnaissance.

La composition de base d'un SRAP est représentée dans la figure II.2.

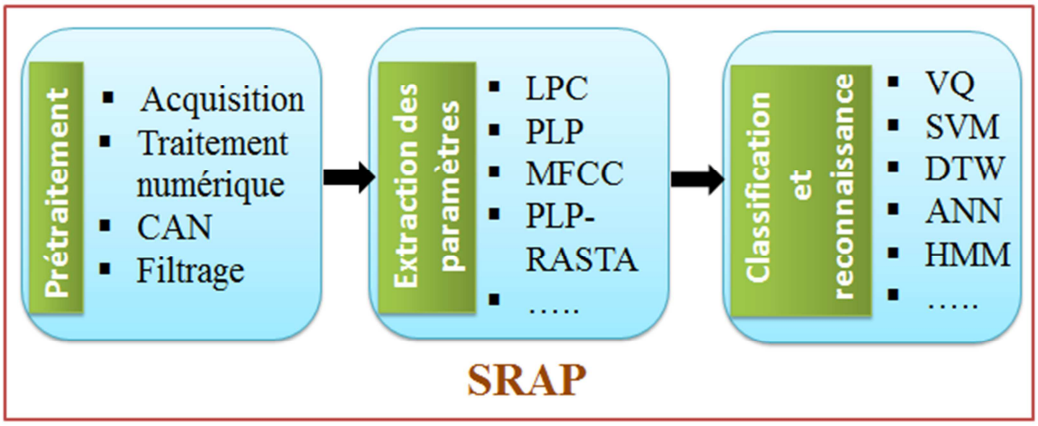

Figure II. 2: Composition de base des SRAP.

II.2.2. Mesures de performances d'un SRAP :

L'évaluation de la qualité des systèmes de reconnaissance de la parole est une tâche de compréhension. Le taux de reconnaissance permet la mesure d'efficacité d'un SRAP, ce dernier varie fortement selon le moyen d'acquisition utilisé, la taille du vocabulaire et le mode d'élocution.

Il existe plusieurs valeurs mesurant les performances d'un système de reconnaissance automatique de la parole :

- **Taux de reconnaissance :** le nombre ou le pourcentage de mots parfaitement reconnus.
- **Taux de substitution :** le nombre ou le pourcentage de mots pour lesquels le système fait erreur de reconnaissance.
- **Taux de rejet :** le nombre ou le pourcentage de mots que le système n'a pas compris.
- **Taux d'omission :** le pourcentage de mots non détectés.
- **Taux d'insertion :** le nombre ou le pourcentage de réponses inopinées.

II.2.3. Applications de la reconnaissance automatique de la parole :

Le choix d'une application doit faire l'objet d'une étude attentive, fondée sur un ensemble de critères objectifs. En particulier, il est important d'examiner si la voix apporte véritablement un accroissement des performances ou un meilleur confort d'utilisation. Par ailleurs, il ne faut pas trop attendre de la commande vocale mais la considérer comme un moyen complémentaire parmi d'autres moyens d'interaction Homme-Machine. Généralement pour des applications en reconnaissance de la parole, on jugera la qualité d'une application sur les critères suivants :

- Le début du flux de parole correctement reconnu. Si le locuteur prononce les mots séparément avec de petites pauses (environ 200 ms) entre chaque mot, on parlera de reconnaissance par mots isolés IWR, sinon ce sera de la reconnaissance de parole continue.
- La taille du vocabulaire correctement reconnu. Ce vocabulaire variera de quelques mots (la cabine téléphonique à entrée vocale) à plusieurs milliers de mots (la machine à écrire à entrées vocales).
- Les contraintes imposées par les systèmes sur l'environnement de fonctionnement : acceptation de bruit de fond et parasites divers. Des critères de qualité positifs dans certaines applications peuvent être négatifs dans d'autres : l'indifférence au locuteur est recherchée pour une cabine téléphonique à numération vocale alors qu'au contraire c'est la capacité de discrimination entre locuteurs qui déterminera la qualité d'une serrure à commande vocale.

Le tableau II.2 présente quelques exemples d'applications démontrant l'intérêt de la reconnaissance automatique de la parole :

Domaine	***Applications***
Automobile	Contrôle mains libres des équipements, les systèmes télématiques, etc.
Médical	Aide aux personnes handicapées.
Industriel	Contrôle vocal des machines, applications pour la gestion de stocks, etc.
Téléphonie	Automatisation de transactions téléphoniques, service téléphonique pour l'accès à des services d'informations, etc.
Multimédia	Logiciels de dictée vocale, interaction vocale dans les logiciels pédagogiques et ludiques, etc.

Tableau II 2: Quelques applications de la reconnaissance automatique de la parole.

II.2.4. Classification des systèmes reconnaissance automatique de la parole

Le domaine de la reconnaissance automatique de la parole peut être divisé en quatre sous domaines : la reconnaissance de mots isolés, la reconnaissance de mots enchainés, la reconnaissance et la compréhension de la parole continue avec un vocabulaire et une syntaxe limités et enfin, la reconnaissance et la compréhension du langage naturel.
Le domaine de la communication parlée homme-machine est illustré dans la Figure II.3.

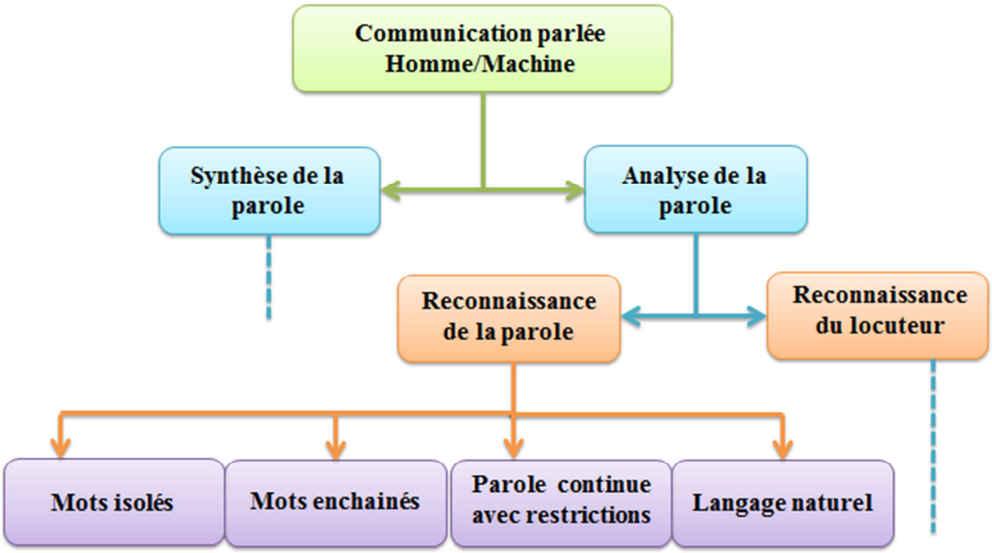

Figure II. 3: Classification des SRAP.

Il existe d'autres classifications, comme par exemple cette classification des systèmes de reconnaissance de la parole en trois types :

- Mots isolés, mots enchainés, reconnaissance de la parole continue, reconnaissance et compréhension du langage naturel.
- Grand et petit vocabulaire.
- Système de reconnaissance spécifique et non spécifique.

II.2.5. Approches de la reconnaissance automatique de la parole

Il existe trois approches permettant d'aborder la reconnaissance de la parole : l'approche globale, l'approche analytique et l'approche statistique [Bar96]. Dans l'approche globale, l'unité de base est le mot : le mot est considéré comme une entité indivisible. Une petite phrase, de très courte durée, peut être considérée comme un mot. Dans l'approche analytique, on tente de détecter et d'identifier les composantes élémentaires de la parole qui sont les phonèmes. L'approche statistique se fonde sur une formalisation statistique simple issue de la théorie de l'information [Cer06].

II.2.5.1. Approche analytique :

L'approche analytique cherche à résoudre le problème de la parole en isolant des unités acoustiques courtes en procédant à une segmentation en entités élémentaires de base étiquetées ou identifiées, comme les phonèmes, les syllabes ,…etc.[Cal93]. Les systèmes analytiques ont l'avantage de ne pas avoir lieu à se préoccuper de la taille du vocabulaire et de s'adapter facilement à tout nouveau locuteur, par contre, leur mise au point est plus délicate.

Il est nécessaire d'intégrer dans le même système un ensemble de niveau supérieur de reconnaissance : niveau lexical, syntaxique, sémantique et pragmatique.

Les principaux obstacles de la reconnaissance analytique résultent de la reconnaissance très limitée que l'on a de diffèrent niveaux notamment pour les niveaux phonétiques et sémantiques, mais aussi de les intégrer dans un même système. Les méthodes globales, développées pour la reconnaissance de mots isolés, ne font pas d'hypothèses sur la structure phonétique des mots, ce qui évite une erreur pénalisante au début du traitement [Hat91].

II.2.5.2. Approche globale :

L'approche globale s'applique aux systèmes pour lesquels l'unité de décision est le mot. On peut modéliser globalement une chaîne de reconnaissance des formes par le schéma suivant :

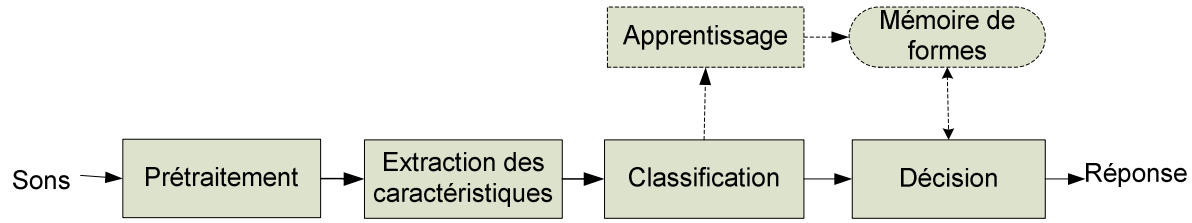

Figure II. 4 : Schéma de principe d'un système de reconnaissance des formes

Le prétraitement consiste d'abord à améliorer le signal d'entrée, c'est à dire à diminuer le bruit et ensuite, à extraire un vecteur caractéristique pour la classification.

Il faut d'abord délimiter le début et la fin de la parole afin d'améliorer la reconnaissance. Le signal acoustique variant au cours du temps, il est nécessaire d'effectuer le prétraitement sur une trame de courte durée (10 à 30 ms). On peut considérer ainsi le signal comme quasi stationnaire. On peut utiliser comme fenêtre pour la récupération des trames : les fenêtres de Hamming, Hanning, Kaiser ou de Blackman. L'analyse spectrale considère le signal comme périodique. Ces fenêtres permettent donc de diminuer l'influence des échantillons situés au

début et à la fin des trames. Ainsi, Ces fenêtres possèdent des propriétés qui permettent de minimiser l'introduction d'artefact dans le spectre.

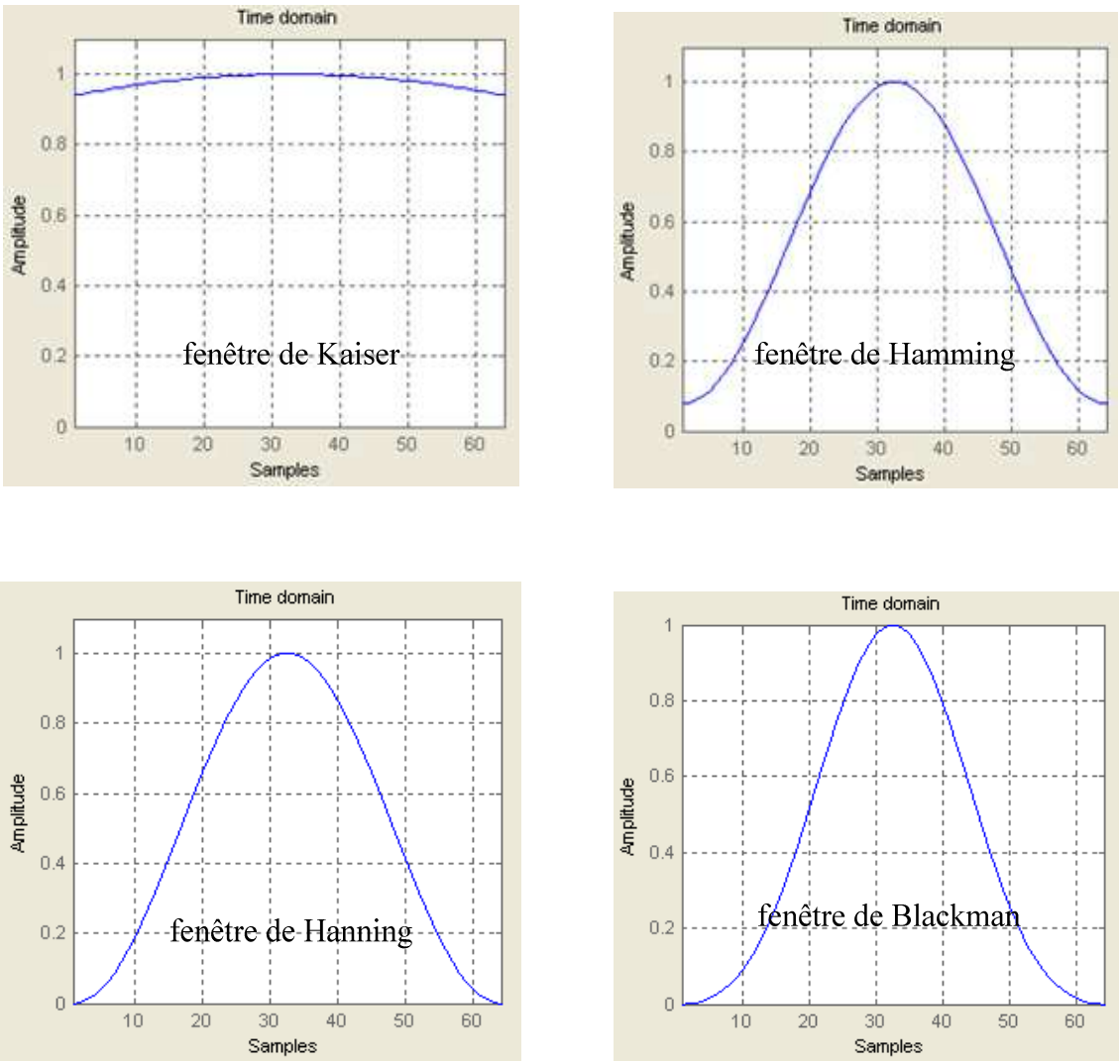

Figure II. 5: les fenêtres de Hamming, Hanning, Kaiser ou de Blackman .

Le vecteur de caractéristique est ensuite obtenu par une méthode d'analyse fréquentielle ou de modélisation explicite du signal. Une fois les paramètres obtenus par le traitement du signal, il est parfois nécessaire de réaliser une sélection des caractéristiques. En effet, si le vecteur obtenu possède une dimension trop grande, il est souvent intéressant de ne pas conserver toutes les caractéristiques, informations redondantes et non significatives, exemple : Paramètres liés à la reconnaissance du locuteur).

Lorsque les formes à reconnaître ont été acquises et paramétrées de façon satisfaisante, l'étape essentielle de la reconnaissance est la classification.

Pour être complet, un système de reconnaissance automatique de la parole peut posséder des analyseurs lexicaux, syntaxiques et sémantiques. Ces étapes sont rajoutées notamment pour des systèmes analysant la parole continue. Il améliore leur taux de reconnaissance en s'appuyant sur l'orthographe, la grammaire et le contexte de la phrase. En appliquant des probabilités basées sur la linguistique, il diminue le vocabulaire possible améliorant ainsi la reconnaissance du prochain mot prononcé. Cette partie liée au langage n'entre pas dans le cadre de ce projet.

II.2.5.3. Approche statistique :

Cette approche est fondue sur le principe de fonctionnement des méthodes globales mais avec l'exploitation des niveaux linguistiques. Ainsi une analyse acoustique est nécessaire pour convertir tout signal vocal en une suite de vecteurs acoustiques. Ces derniers sont considérés comme des observations dans la phase d'apprentissage des modèles statistiques et dans la phase de reconnaissance qui effectue une classification de chaque observation.

On considère une suite d'observations acoustiques O, résultant d'une analyse acoustique d'un signal de parole représentant une séquence de mots prononcés M. L'approche statistique consiste à chercher la séquence M' la plus probable parmi toutes les séquences de mots possibles E(M) sachant les observations O. Donc la séquence optimale est celle qui maximise la probabilité a posteriori P (M|O).

II.3 Méthodes de classification et de reconnaissance vocale

Apres avoir extraire le vecteur des paramètres du signal parole par l'une des techniques LPC, LPCC, PLP, **MFCC**, PLP-RASTA ou par ondelettes [Cal93], On passe à la deuxième étape qui est la classification ou la reconnaissance. La classification dans un SRAP regroupe les deux tâches d'apprentissage et de décision. L'objectif de cette phase est de chercher dans la base de données la classe d'appartenance la plus probable pour chaque mot prononcé.

Le principe de : la reconnaissance des mots isolés IWR est représenté sur la Figure II.6 :

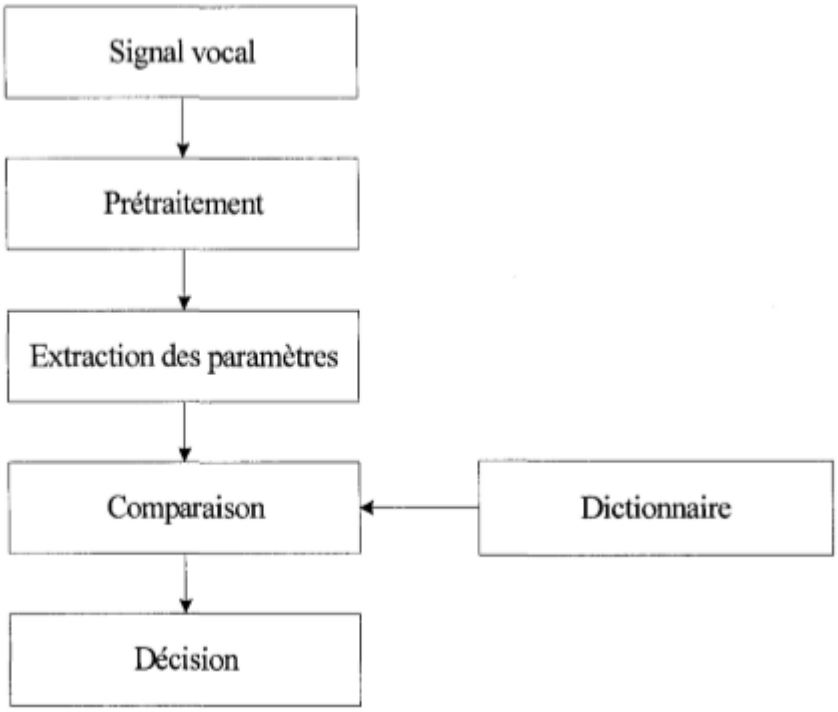

Figure II. 6: Reconnaissance des mots isolés IWR

- **L'apprentissage**

L'apprentissage a pour but la création d'un dictionnaire de référence représenté par les groupes de mots à reconnaître.

- **La décision**

On cherche dans cette phase les modèles acoustiques les plus proches, et cela en un temps aussi court que possible.

La décision peut conduire à un succès si la réponse est unique ou à une confusion dans le cas de réponses multiples. Sinon à un rejet de la forme si aucun des modèles ne correspond à sa description.

Il existe plusieurs méthodes de classification regroupées dans les catégories suivantes :

- Classification automatique.
- Classification statistique.
- Classification stochastique.
- Classification neuronale.

II.3.1 **Classification automatique**

La classification d'un ensemble de vecteurs consiste à les regrouper en classes. Bien entendu, une classe est un ensemble de vecteurs ayant des caractéristiques semblables. La

classification doit donc vérifier la compacité(les points représentant une classe sont plus proche entre eux que des points de toutes les autres classes) et la séparabilité (les classes sont bornées et il n y'a pas de recouvrement entre elles).

Ces deux propriétés sont rarement respectées en pratique, soit à cause du bruit, soit à cause de distorsion des signaux. La classification est fondue dans ce cas sur le principe de proximité [Boi87]. Parmi plusieurs méthodes de classification automatique on distingue la programmation dynamique [Hat91].

II.3.2 **Classification statistique**

Les méthodes de classification statistiques consistent à faire correspondre des vecteurs de caractéristiques de longueur fixe à un espace partitionné. Le principe dans ces méthodes est de comparer les caractéristiques de la forme à reconnaître avec la valeur moyenne des caractéristiques de chaque classe. Ensuite attribuer la forme à la classe ayant les valeurs les plus proches, par de nombreuses méthodes on peut citer à titre d'exemple :

- ✓ La décision bayésienne.
- ✓ La méthode des K plus proches voisins.

Il est à noter que ces méthodes sont basées sur la formalisation proposée par [Jel76].

II.3.3 **Classification stochastique**

L'approche stochastique utilise un modèle pour la comparaison, prenant en compte une plus grande variabilité de la forme. Cette variabilité est considérée comme un signal continu observable dans le temps à différents endroits constituant des états observables.

Le modèle stochastique décrit ces états à l'aide de probabilités de transitions d'état à état et de probabilités d'observation par état. La comparaison consiste à chercher dans ce graphe le chemin le plus probable correspondant à une suite d'éléments observés dans la chaîne d'entrée.

Ces méthodes sont robustes et fiables (du fait de l'existence de bon algorithme d'apprentissage) avec un calcul relativement faible. Parmi ces méthodes on peut citer les HMM (Hidden Markov Model).

II.3.4 **Classification neuronale**

L'utilisation d'un réseau de neurones en classification dépend du codage des sorties. Dans la méthode généralement employée, chaque unité de la couche de sortie représente une classe possible pour les formes en entrée. L'introduction d'une forme inconnue en couche d'entrée du réseau neuronal et après propagation des résultats du calcul élémentaire de chaque neurone

vers la couche de sortie, l'élément ayant la plus grande valeur permet de choisir à quelle classe affecter la forme en entrée [Wan93]. Le problème de classification peut être résolu en trouvant une fonction de transfert qui associe un ensemble de formes de départ à un ensemble de classes d'arrivées [Mor95].

Ces méthodes présentent les avantages suivants :

- Puissance d'approximation.
- Robustesse pour les tâches difficiles.
- Le traitement des données se fait d'une manière parallèle.

Cependant, les inconvénients de ces méthodes sont liés aux réseaux de neurones eux-mêmes, comme par exemple :

- Temps d'apprentissage très lent.
- Le choix de la topologie du réseau de neurones à utiliser (choix aléatoire ou par l'utilisation des heuristiques).
- Combien faut-il mettre de couches de neurones, et combien de neurones faut-il mettre dans chaque couche.

II.4 Les méthodes de reconnaissance de mots isolés implémentées.

Quatre Méthodes ont été implémentées pour la comparaison des vecteurs observés, aux différentes références stockées dans un dictionnaire.

Dans cette catégorie on trouve principalement :

- La méthode de la quantification vectorielle VQ (Vector Quantization en anglais).
- La méthode d'alignement temporel dynamique DTW (Dynamic Time Warping).
- La méthode des modèles de Markov cachées (HMM : Hiden Markov Model).
- La méthode des modèles de Mélange de gaussiennes GMM

II.4.1 La quantification vectorielle

La quantification vectorielle (VQ) [Rab83] est une technique de compression de données, qui consiste à coder de manière efficace des échantillons représentés par plusieurs valeurs (vecteurs). Ce codage se fait de la manière suivante :

On divise l'espace en classes adaptées à l'ensemble des échantillons et on calcule un représentant pour chaque classe. Ce représentant appelé centroïde ou noyau, représente la distance minimale intra-classes. L'ensemble des noyaux est appelé dictionnaire ou code-book. Imaginons qu'on a un ensemble d'échantillons, chacun représenté par un couple de valeur

(x1, x2), pour quantifier un échantillon X, on lui attribue les valeurs du représentant le plus proche (Figure II.7).

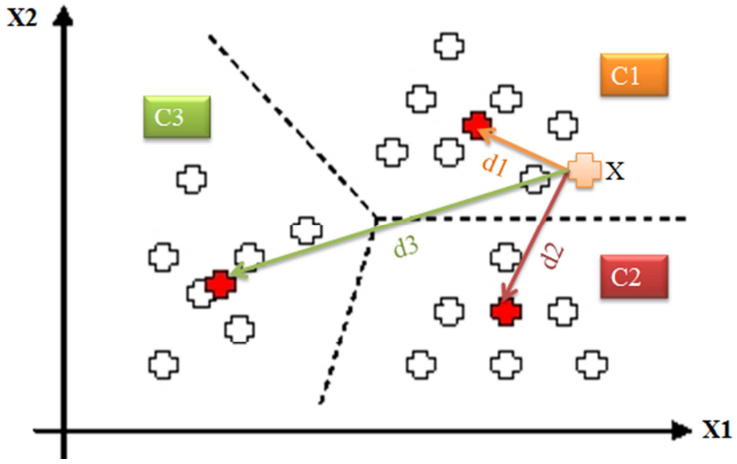

Figure II. 7: Quantification vectorielle d'un échantillon de dimension 2.

La Figure II.7 montre que le vecteur X appartient à la classe C1, car il est plus proche du noyau de cette classe (d1 < d2 < d3) distance euclidienne.

Malgré la simplicité du codage par quantification vectorielle, la conception d'un dictionnaire est plus compliquée et a donné lieu à de nombreux algorithmes, l'un des plus utilisés est l'algorithme K- means.

- **L'algorithme K-means**

L'algorithme K-means consiste à définir d'une manière itérative M classes à partir de L-vecteurs de paramètres qui constitue l'ensemble d'apprentissage [Fer83]. Chaque classe est concentrée autour d'un noyau, le dictionnaire des références est constitué de l'ensemble des noyaux des diverses classes, l'algorithme est décrit comme suite :

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

$$x_k \in C_i, SSi\ d(x_k, n_i) \leq d(x_k, n_j), avec\ j \neq i\ et\ 1 \leq j \leq M \qquad \text{II. 1}$$

d : la mesure de distorsion (il s'agit de la distance euclidienne dans la plupart des cas).

3. **Mise à jour :** calcul des nouveaux noyaux des classes, afin de minimiser la destorsion au sein de chaque classe ; n_i est defini par :

$$n_i = \frac{1}{N}\sum_{n=1}^{N} x_n \qquad \text{II. 2}$$

Ou N est le nombre d'elements de la classe C_i.

4. **Test d'arrêt :** si les contenus des classes restent stables et inchangés entre deux itérations consécutives alors fin de l'algorithme, sinon retour à l'étape 2.

Il est à noter que cet algorithme converge vers un optimum local, qui dépond des valeurs initiales des noyaux des classes [Eas92].

II.4.2 L'alignement temporel dynamique DTW

Pour pouvoir mesurer la distance de l'échantillon à identifier vis-à-vis des échantillons de références, il est à noter qu'afin d'éviter des calculs inutiles, les coefficients de références sont calculés une seule fois et sauvegardés dans un fichier.

Cependant, si l'on veut évaluer la distance entre deux vecteurs, il est nécessaire d'avoir des vecteurs de dimensions identiques. C'est pourquoi, avant de calculer la distance, on établit une correspondance linéaire des axes du temps. Cette solution consiste simplement à associer plusieurs vecteurs de référence à un seul vecteur d'entrée (ou inversement selon la séquence la plus longue).

Apres cette étape, on peut calculer la distance de l'échantillon à tester par rapport à l'ensemble des références. On utilise dans ce cas la distance euclidienne. D'autres distances plus performantes, tenant compte de la variance des coefficients par exemple comme la distance de « Mahalanobis », aurait pu être envisagée mais nous nous limitons à cette distance qui est la plus simple à calculer et qui permet d'obtenir des résultats satisfaisants.

Soient A et B deux images acoustiques (spectres) de longueur I et J respectivement.

La « distance » entre l'événement i ϵ [1, I] de A, et l'événement j ϵ [1, J] de B, se calcul avec une simple distance euclidienne :

$$d(i,j) = \sqrt{\sum_{k=1}^{n}(x_k - y_k)^2} \qquad \text{II. 3}$$

Avec : i =($x_1 \dots \dots . . x_n)^t$ et j =($y_1 \dots \dots . . y_n)^t$.

Une fois les distances obtenues, il suffit de rechercher la plus petite parmi celles-ci et de donner à l'utilisateur le résultat correspondant. Cette technique est la plus simple à mettre en œuvre, mais elle ne tient pas compte des variations temporelles de la voix. En effet, lorsqu'un locuteur même surentrainé, répète plusieurs fois la même séquence, il ne peut éviter les variations du rythme ou de vitesse d'élocution. Ces variations entraînent des transformations non linéaires dans le temps du signal acoustique, ce qui fait qu'on ne pourra comparer

directement point à point (calcul de la distance euclidienne) deux formes acoustiques sans correction temporelle au préalable. Pour établir une meilleure correspondance des axes temporels entre les deux mots, en même temps avec leurs comparaisons, on utilise une technique appelée technique d'alignement temporel dynamique ou DTW [Eas92]. C'est une technique basée sur la programmation dynamique qui consiste à essayer de trouver le chemin optimal à parcourir parmi l'ensemble des distances entre les vecteurs.

Cela suppose bien sûr que l'on considère la même plage de fréquence pour les deux signaux (entre 0 et N). On crée donc un chemin{C(K) = (n(k), m(k)), k ϵ [1, K]}. Il est nécessaire que les fonctions n(k) et m(k) soient croissantes et doivent correspondre à certaines contraintes :

- ✓ Les seuls chemins valides arrivants au point (i, j) sont ceux provenant des points de cordonnées (i -1, j), (i, j -1) et (i -1, j -1) (Figure II.8).
- ✓ De plus, on prend K tel que C(K) = (I, J). On pose C(1) = (1,1).

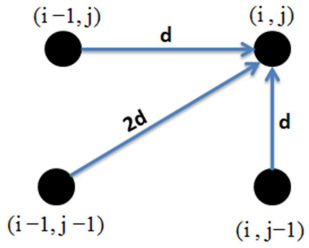

Figure II. 8: Exemple de contraintes de parcours pour la technique DTW.

La méthode consiste à choisir le chemin qui passe par les distances d (i, j) les plus petites, de sorte que la distance cumulée le long de ce chemin soit la plus petite possible.
On définit G (i, j) la distance cumulée au point (i, j) comme :

$$G(i,j) = \min \begin{cases} G(i-1,j) + d(i,j) \\ G(i-1,j-1) + 2d(i,j) \\ G(i-1,j-1) + d(i,j) \end{cases} \qquad \text{II. 4}$$

On remplit ensuite une matrice de taille I x J avec en $i^{éme}$ et $j^{éme}$ colonnes le résultat de G (i, j). Enfin on définit la distance normalisée entre deux prononciations du mot :

$$G_N = \frac{G(I,J)}{I+J} \qquad \text{II. 5}$$

On obtient une distance entre deux spectres. On effectue ce travail entre le mot à reconnaître et tous les mots du dictionnaire. On prend ensuite le mot du dictionnaire qui a la plus petite distance spectrale avec le mot à reconnaître.

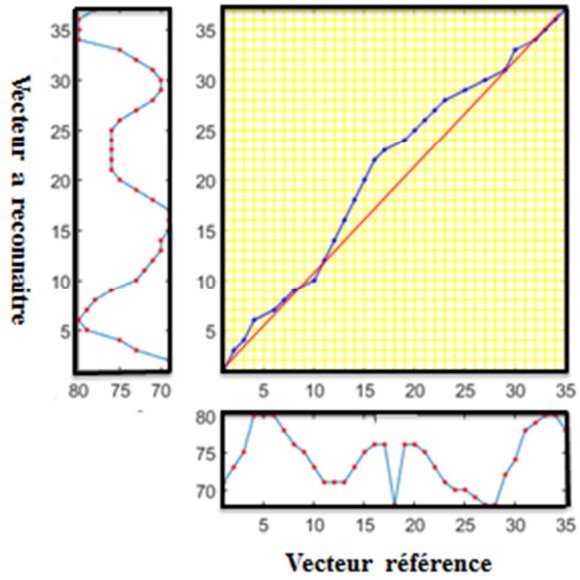

Figure II. 9: Chemin minimal entre deux vecteurs acoustiques.

La Figure II.9 illustre la notion de chemin entre deux spectres. Les différences entre les deux spectres « tordent » le chemin idéal (la diagonale).

II.4.3 Les modèles de Markov cachés

II.4.3.1 Définition des modèles de Markov cachés

Les modèles de Markov cachés (HMM) sont des approches stochastiques qui utilisent la probabilité à la place de la distance où le signal de la parole est représenté par une séquence d'états d'observations. Le principe de reconnaissance d'un mot avec HMM consiste à trouver un modèle qui reconstitue le mot avec une grande probabilité [Rab83].

Un modèle de Markov λ (*A*, *B*, π) est un automate probabiliste d'états finis constitué de N états.

Un processus aléatoire se déplace d'état en état à chaque instant, et on note q_t le numéro de l'état atteint par le processus à l'instant t. L'état réel q_t du processus n'est pas directement observable (caché), mais le processus émet après chaque changement d'état un symbole discret o_t qui appartient à un alphabet fini de n_v symboles $V = \{v_M\}, 1 \leq M \leq n_v$. Dans le cas d'un processus markovien du premier ordre, la probabilité de passer de l'état i à l'état j à l'instant t et d'émettre le symbole v_k ne dépend ni du temps, ni des états aux instants précédents. Un modèle de Markov caché ou HMM est alors défini par [Bel92]:

- Un ensemble $S = \{S_1, S_2, \dots , S_N\}$ de N états où un état est défini à l'instant t par : $q_t \in S$.
- Un ensemble $V = \{v_1, v_2, \dots , v_M\}$ qui contient M symboles d'observations. L'observation d'un symbole à l'instant t est notée $o_t \in V$.

- La matrice de probabilités $A = \{a_{ij}\}$ où a_{ij} est la probabilité de passage de l'état i vers l'état j. On a:

$$a_{ij} = P(q_{t+1} = S_j \mid q_t = S_i), \text{avec } 1 \leq i \text{ et } j \leq N \qquad \text{II. 6}$$

- La matrice de probabilités $B = \{b_j\}$ où b_j est la probabilité d'observation d'un symbole v_k , sachant qu'on est à l'état j. On a:

$$b_j(k) = P(o_t = v_k \mid q_t = S_j), \text{avec } 1 \leq j \leq N \text{ et } 1 \leq j \leq M \qquad \text{II. 7}$$

- Les probabilités initiales $\pi = \{\pi_1, \pi_2, \dots, \pi_N\}$ où π_i est la probabilité que le modèle commence par 1' état i. On a :

$$\pi_i = P(q_i = S_i), \text{avec } 1 \leq i \leq N \qquad \text{II. 8}$$

Un exemple d'un modèle de Markov caché à 5 états est représenté par la Figure II.10 [Rab93]:

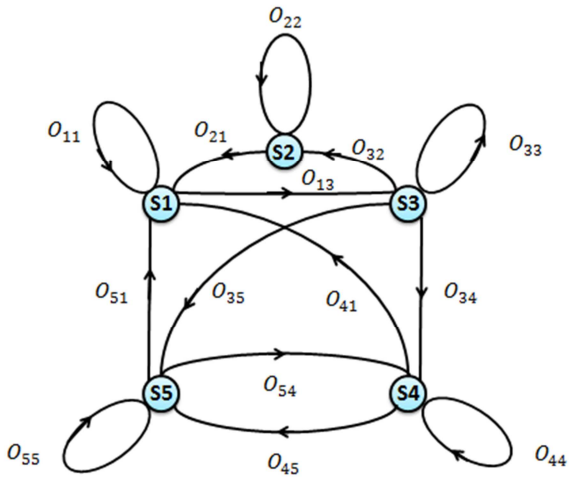

Figure II. 10: Modèle de Markov caché à 5 états.

Pour utiliser la méthode des HMMs, il faut résoudre les trois problèmes fondamentaux suivants :

- **Evaluation de la vraisemblance**

L'évaluation de la probabilité que la suite des observations ait été émise par un modèle. Lorsque plusieurs modèles existent, cette évaluation permet le choix du modèle le plus probable.

- **Le décodage**

La recherche de la séquence d'états d'un modèle ayant produit les observations. La séquence cachée de plus forte probabilité est déterminée par l'algorithme de Viterbi.

- **L’apprentissage**

L'apprentissage des paramètres d'un modèle. À partir d'un modèle donné a priori et d'observations supposées émises par ce modèle, on cherche les probabilités de transition et d'émission maximisant la vraisemblance des observations on utilise l’algorithme Baum Welch.

La solution du problème de l’évaluation de la vraisemblance donne un moyen de mesurer l’adéquation d’une séquence d’observation à un modèle. Ainsi on peut décider du meilleur modèle selon la règle de Bayes. Résoudre le problème du décodage permettra de segmenter les séquences par la recherche de la séquence d’états de vraisemblance maximale. Enfin, l’apprentissage doit permettre d’adapter automatiquement un HMM à un ensemble particulier de données.

II.5 **Conclusion**

Le signal acoustique de la parole présente une grande variabilité qui complique la tâche des SRAPs. Cette complexité provient de la combinaison de plusieurs facteurs, comme la redondance du signal acoustique, la grande variabilité intra et interlocuteurs, les effets de la coarticulation en parole continue, ainsi que les conditions d’enregistrement. Pour surmonter ces problèmes, différentes approches sont envisagées pour la reconnaissance de la parole telles que les méthodes analytiques, globales et les méthodes statistiques. Ainsi, dans ce chapitre, nous avons décrit brièvement un état de l’art sur les systèmes RAP, leur principe de fonctionnement, leur composition de base et leur classification. Après nous avons cité les diverses approches de RAP. Ensuite nous avons décrit les méthodes de classification en les classifiant dans des catégories telles que la classification automatique, statistique, stochastique et neuronale.

Par la fin, nous avons terminé ce chapitre par l’explication à titre d’exemples de quelques méthodes de reconnaissance de mots isolés, notamment la quantification vectorielle, l’alignement dynamique temporel et les modèles de Markov cachés.

Chapitre III

CONCEPTION DES SRAPs

CONCEPTION DES SRAPs

Le but de ce chapitre est de concevoir et de mettre en œuvre différents systèmes de reconnaissance automatique de la parole, afin de les utiliser pour le contrôle d'une simulation de plateforme mobile. Il s'agit de la partie logicielle du travail. Cette partie consiste à éliminer le bruit du signal original et ne traiter que la partie qui contient le signal parole en utilisant l'algorithme VAD , à extraire les paramètres MFCC pertinents du signal et à développer les différents algorithmes de reconnaissance de la parole (DTW, HMM, GMM,VQ).

III.1 Méthode de reconnaissance par DTW

III.1.1 Description du travail réalisé :

Dans ce travail, la reconnaissance vocale a été mise au point en utilisant l'algorithme de l'algorithme DTW et l'algorithme KNN l'algorithme VAD est appliqué au signal d'entrée issu du microphone, afin d'éliminer au maximum des moments de silence. Ensuite, on a procédé à l'extraction des paramètres pertinents du signal. Notre choix s'est porté sur la représentation par coefficients MFCC.

Le premier fichier de référence a été créé pour des signaux vocaux préenregistrés de différentes personnes. On compare les coefficients MFCC calculé du signal parole mis en test, aux coefficients MFCC préenregistrés dans la base de données en appliquant l'algorithme DTW.

Les scores de sortie de l'algorithme DTW ont été appliqués à l'algorithme KNN pour calculer le son commun le plus proche des cinq signaux vocaux préenregistrés. Finalement le résultat de la reconnaissance sera affiché sur l'écran de sortie du logiciel MATLAB.

La reconnaissance d'une commande, permet d'activer une action convenable de la plateforme mobile pendant une certaine période en attendant le prochain ordre, les cinq commandes possibles sont : Stop, Forward, Left, Right et Backward.

L'organigramme de la Figure III.1, représente la méthodologie suivie pour la réalisation ;

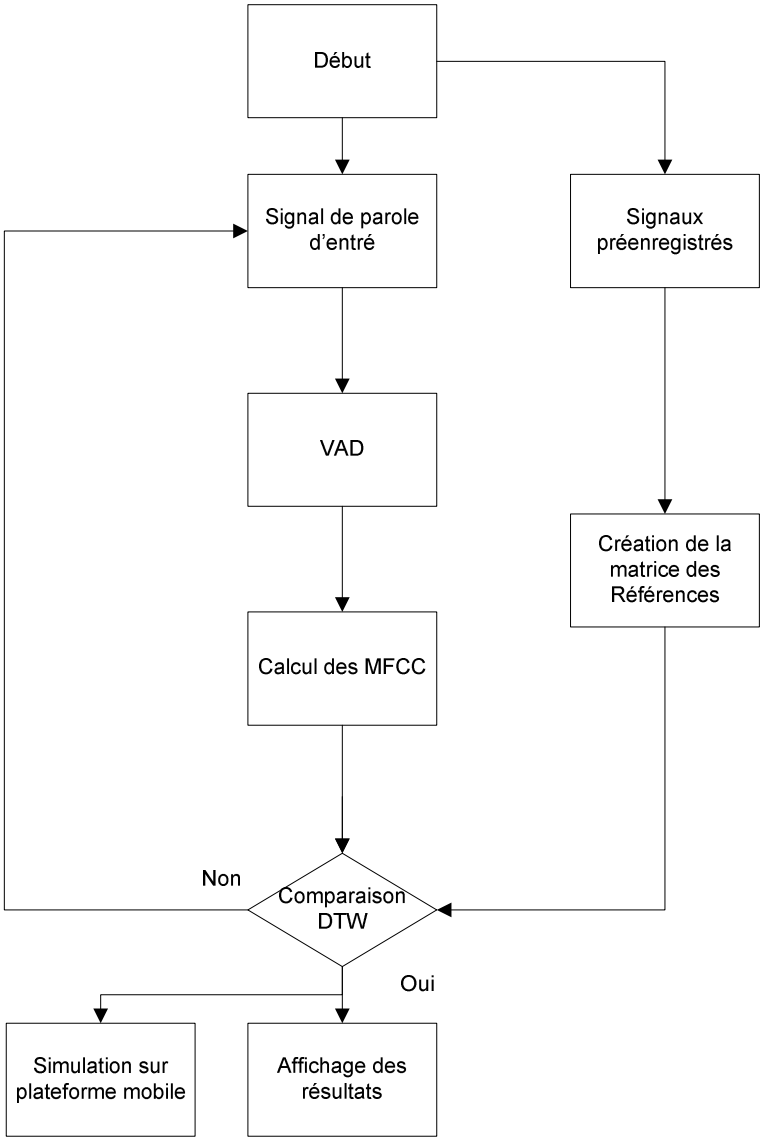

Figure III. 1 : Schéma de la reconnaissance MFCC + DTW.

III.1.2 Extraction des coefficients MFCC

La parole étant un signal constitué d’une infinité d’informations, il faut en extraire les informations les plus importantes. La représentation du signal parole dans le domaine temporel est illustrée dans la Figure III-2 (exemple des deux mots Stop et Backward).

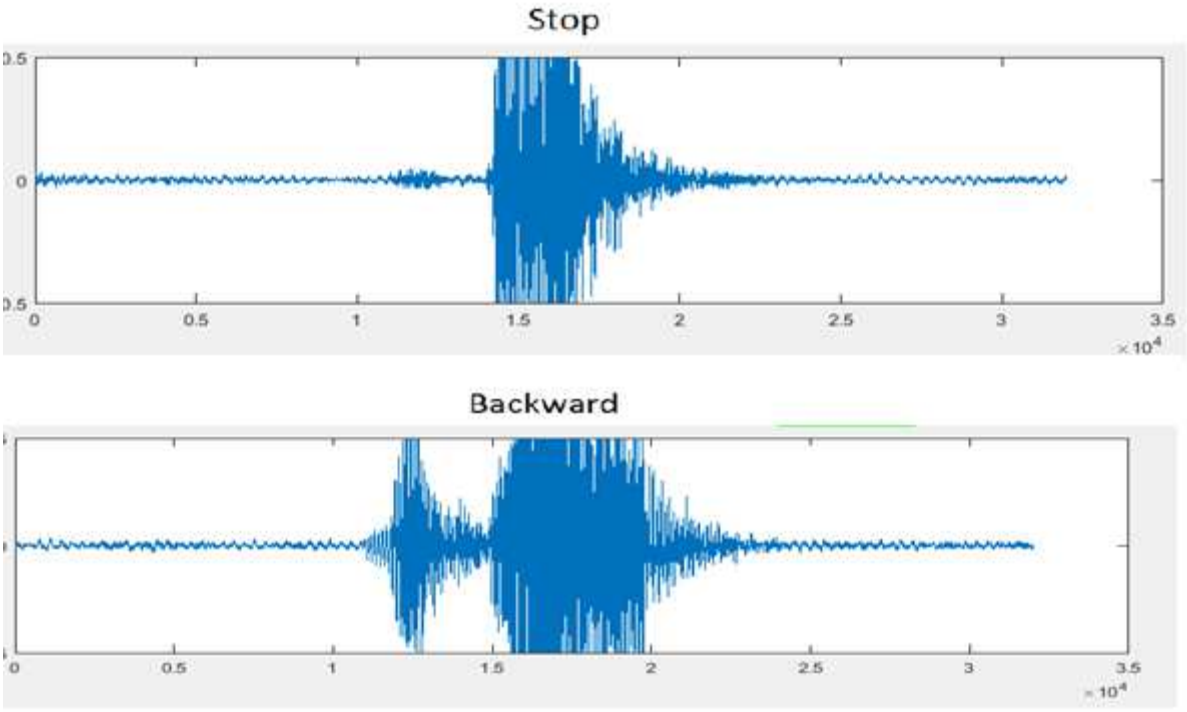

Figure III. 2: Représentation temporelle d’un signal de parole (durée de deux secondes).

Un traitement direct de comparaison sur ce genre de signal est impossible car il y a trop d’informations et surtout ces informations ne sont pas exploitables. À partir des spectrogrammes, on pourrait se contenter d’un simple calcul de distance spectrale entre le spectre du mot à reconnaître et celui du mot présent dans le dictionnaire. Mais le fait qu’il y ait des variations inévitables dans la prononciation nécessite l’utilisation d’un algorithme de comparaison dynamique. La comparaison dynamique directe de deux spectres donne des résultats peu convaincants et le temps de calcul est assez élevé (dû au fait qu’il y a encore beaucoup de données à traiter), c’est pour cela que on a eu recourt à des techniques d’extraction des caractéristiques pertinentes.

Plusieurs algorithmes d'extraction de caractéristiques peuvent être utilisées pour effectuer cette tâche, à titre d’exemples on peut citer les méthodes suivantes : Linear Predictive Coefficients (LPC), Linear Predictive Cepstral Coefficients (LPCC), Mel Frequency Cepstral Coefficients (MFCC) , Perceptual Linear Predictive (PLP) , PLP- Relative Spectra (PLP- RASTA) et human Factor Cepstral Coefficient (HFCC) .

La paramétrisation MFCC est la technique la plus fiable et la plus utilisée pour extraire les caractéristiques du signal parole. Cette technique est basée sur deux idées clés [Ver99] [Pic93] [Dav80] :

a. La première consiste à exploiter les propriétés du système auditif humain par la transformation de l'échelle linéaire des fréquences en échelle de Mel.
b. La deuxième consiste à effectuer une transformation cepstrale qui permet la décorrélation des composantes spectrales du signal de parole.

On résume le Procédé d'extraction comme suit :

- **Phase 1 :** Découper le signal en plusieurs fenêtres qui se recoupent entre elles. Par exemple si nous découpons un signal en X fenêtres de 256, avec un recoupement de 100, alors, la première fenêtre sera 0-255, la seconde 155-411,…etc. Nous appliquerons la MFCC à chaque fenêtre.
- **Phase 2 :** Afin de diminuer la distorsion spectrale créée par le recoupement nous multiplions le signal à transformer par une fenêtre de Hamming.
- **Phase 3 :** Appliquer ensuite la FFT à la fenêtre pour en ressortir la magnitude, on obtient donc le spectre.
- **Phase 4 :** On passe à l'échelle de Mel. En effet, Pour simuler l'oreille humaine, il faut passer par un Banc de Filtres, un filtre pour chaque fréquence qu’on cherche. Ces filtres ont une réponse de bande passante triangulaire. Pour connaitre l'intervalle entre chaque filtre, on utilise une constante: Mel-Frequency interval. Nous utilisons 24 filtres.
- **Phase 5 :** Pour finir, nous travaillons avec le Cepstre, nous convertissons le spectre logarithmique de Mel en temps au moyen de la DCT (Discret Cosinus Transform).Ainsi, nous réduisons le nombre de données caractérisant le signal et nous en ressortons 13 MFCCs.

L’étape d’extraction de paramètres est précédée d’un filtrage de préaccentuation du signal, le prétraitement et la représentation du signal sont illustrés dans la Figure III.3 :

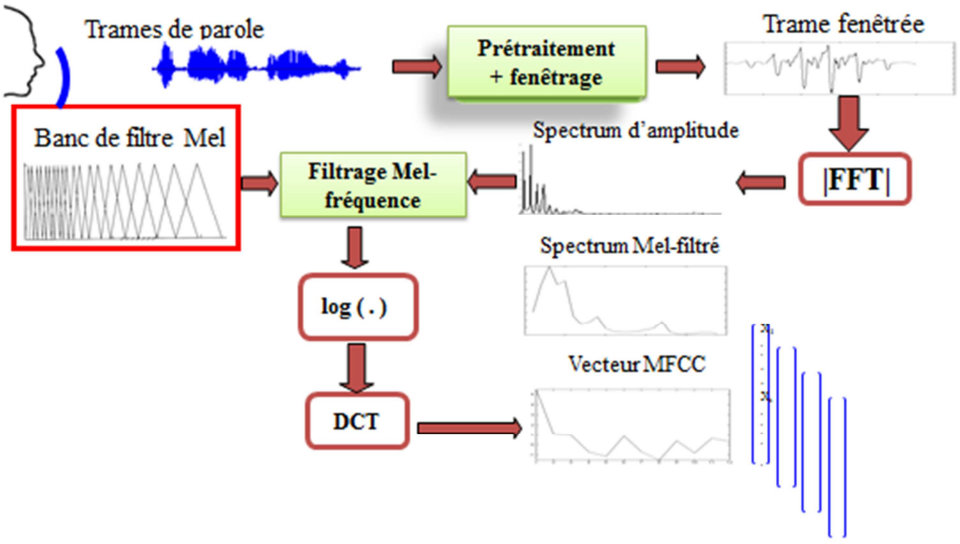

Figure III. 3: Extraction des caractéristiques du signal par la méthode MFCC.

III.1.3 Filtrage de préaccentuation :

Le signal vocal x (n) est envoyé à un filtre passe-haut de la forme:

$$y(n) = x(n) - a * x(n-1) \qquad \text{III. 1}$$

Où : y (n) est le signal de sortie et la valeur de a est habituellement entre 0,9 et 1,0.

La transformée en Z de cette équation est donnée par :

$$H(z) = 1 - a(z^{-1}) \qquad \text{III. 2}$$

L'objectif de la préaccentuation est de compenser la partie des hautes fréquences qui a été supprimée au cours de la production de la parole. Par ailleurs, il peut également amplifier l'importance des composantes à hautes fréquences [Rab78].

III.1.4 Segmentation et fenêtrage

Une fois la formule de préaccentuation du signal effectuée, nous découpons le signal en fenêtres. Le signal est découpé en tranches de 2n échantillons appelées trames ou encore fenêtres qui ont la particularité de se recouvrir de moitié dans l'objectif d'avoir un meilleur traitement par FFT (Fast Fourier Transform).

Le signal de la parole est de nature non stationnaire. Il est donc nécessaire, avant d'extraire les paramètres de la reconnaissance, de le subdiviser en segments. Cette étape permet d'obtenir pour chaque segment de parole un signal quasiment stationnaire [Rab78].

On utilise typiquement une fenêtre de N = 256 échantillons ou un nombre qui est une puissance de 2. Cela vient du fait que l'algorithme FFT que nous utilisons est bien plus rapide pour ces nombres. Dans notre programme principal nous utilisons des fenêtres de 256 échantillons.

III.1.5 Application d'une fenêtre de pondération

Une fenêtre de pondération est appliquée à chaque trame, ceci dans l'objectif d'harmoniser les échantillons pour permettre un meilleur traitement pour l'algorithme FFT. En effet, la FFT ne donne pas de bons résultats quand une pente trop importante est détectée dans une partie du signal. La fenêtre de pondération a pour objectif de minimiser les erreurs produites par FFT. Le concept ici est de minimiser la distorsion spectrale en utilisant la fenêtre en se rétrécissant le signal à zéro au début et à la fin de chaque trame.

Le résultat de fenêtrage est le signal [Var08] :

$$y_i(n) = x_i(n) * w(n), \quad 0 \leq n \leq N-1 \qquad \text{III. 3}$$

Où N est le nombre d'échantillons dans chaque trame.

Typiquement, la fenêtre de Hamming utilisée est de la forme :

$$w(n) = 0.54 - 0.46 * \cos\left(\frac{2\pi n}{N-1}\right) \qquad \text{III. 4}$$

La fenêtre de Hamming est représentée sur la Figure III.4 :

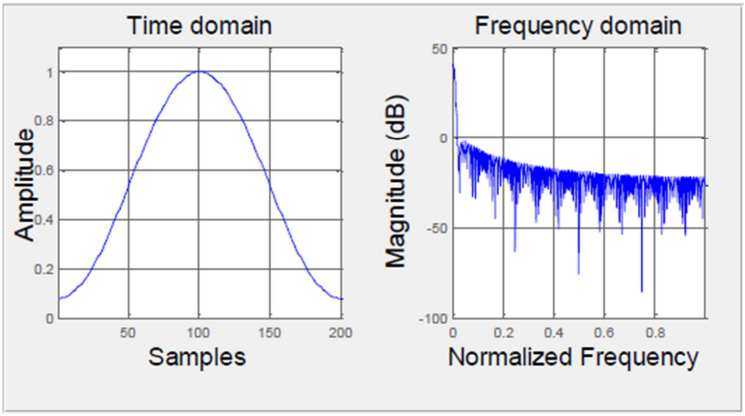

Figure III. 4: Représentation temporelle et fréquentielle d'une fenêtre de Hamming.

III.1.6 Transformée de Fourier Rapide

L'étape suivante est de calculer la transformée de Fourier rapide pour chaque trame acoustique, il s'agit d'une conversion de chaque trame de N échantillons du domaine temporel au domaine fréquentiel.

L'équation qui permet de calculer la TFR est donnée ci-dessous :

$$X_n = \sum_{k=0}^{N-1} x_k \, e^{\frac{-2\pi jkn}{N}}, \quad n = 0, \dots, N-1 \qquad \text{III. 5}$$

III.1.7 Banc de filtres à l'échelle des Mels

a. **L'échelle des Mels**

L'échelle des Mels est une échelle biologique. C'est une modélisation de l'oreille humaine. A noter que le cerveau effectue en quelque sorte une reconnaissance vocale complexe avec filtrage des sons. Prenons l'exemple suivant où une personne est à table en compagnie de nombreuses personnes, l'ensemble de ces personnes parlent en même temps et la personne discute avec son voisin. Malgré le bruit, la personne arrive à discerner clairement ce que lui dit son voisin, elle ignore de façon naturelle le bruit de fond et amplifie le son qui lui paraît le plus important. Le cerveau ne se contente non pas seulement de filtrer les sons et de les amplifier mais aussi de les prédire. Prenons l'exemple suivant où une personne discute avec une autre avec un volume sonore très bas, la deuxième personne n'ayant pas entendue une certaine partie de la phrase mais elle arrive à la reconstituer et à la comprendre.

On considère que l'oreille humaine perçoit linéairement le son jusqu'à 1000 Hz, mais après, elle perçoit moins d'une octave par doublement de fréquence. L'échelle de Mels modélise assez fidèlement la perception de l'oreille : linéairement jusqu'à 1000 Hz, puis logarithmiquement au-dessus. La formule donnant la fréquence en Mels **m** à partir de celle en Hz **f** est :

$$Mel(f) = 2595 log_{10}\left(1 + \frac{f}{700}\right) \qquad \text{III. 6}$$

La courbe de conversion est représentée sur la Figure III.5 :

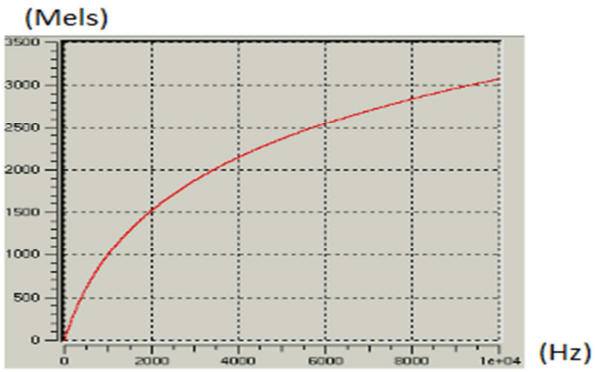

Figure III. 5: Représentation de l'échelle des Mels.

b. **L'analyse par banc de filtre**

Une méthode simple et peu couteuse pour obtenir une estimation de l'enveloppe spectrale du signal vocal est l'analyse par banc de filtres. Le principe revient à découper la bande passante utile en sous bandes contiguës dans lesquelles l'intensité du signal est évaluée à l'aide d'un filtre passe bande. On obtient ainsi une approximation du spectre de Fourier. Le spectrogramme obtenu par transformation de Fourier à court terme peut être alors vu selon deux perspectives [Cer06]:

- Soit comme une séquence de spectres fréquentiels correspondant aux segments temporels successifs de signal.
- Soit comme un ensemble de signaux temporels contenant une partie de l'information sur le signal vocal dans chacune des bandes de fréquences d'un banc de filtres.

Parce que l'étendue des fréquences présentes dans le spectre est encore très large, donc beaucoup de données à traiter, on a recours au banc de filtres dans l'échelle des Mels. On relie ainsi le système de reconnaissance vocale au fonctionnement de l'oreille humaine. Il s'agit de filtres passes bandes (fonction fenêtre de Hamming) centrés linéairement dans le domaine fréquentiel des Mels et de largeur telle qu'ils divisent l'espace des fréquences de manière égale dans le domaine des Mels et qu'ils se recouvrent chacun par moitié. Les bandes de fréquences des filtres sont espacées logarithmiquement selon l'échelle perceptive de Mel. Plus la fréquence centrale du filtre est basse, plus la bande passante du filtre est étroite. L'augmentation de la résolution pour les basses fréquences permet d'extraire plus d'informations dans ces zones où elle est plus dense. On a ainsi beaucoup de filtres pour les basses fréquences alors que les hautes fréquences sont disposées plus largement.

Chaque filtre va donner un coefficient cepstrale :

$$S_{i,k} = \sum_{n=0}^{\frac{N}{2}} Y_{i,n} M_{n,k} \ , \quad k = 0,1, \dots, K \qquad \text{III. 7}$$

Avec :

$Y_{i,n}$: Le $n^{ème}$ coefficient de la transformée de la $i^{ème}$ trame acoustique.

$M_{n,k}$: Le $n^{ème}$ coefficient du $k^{ème}$ filtre.

On a donc $S_{i,k}$, la matrice de sortie du $k^{ème}$ filtre pour la $i^{ème}$ trame acoustique. On a, à cette étape, ce qu'on appelle un Spectre Mel (Mel Spectrum).

Le nombre de filtres utilisés pour extraire les coefficients dans ce travail est 24 filtres, représentés sur la Figure III.6 :

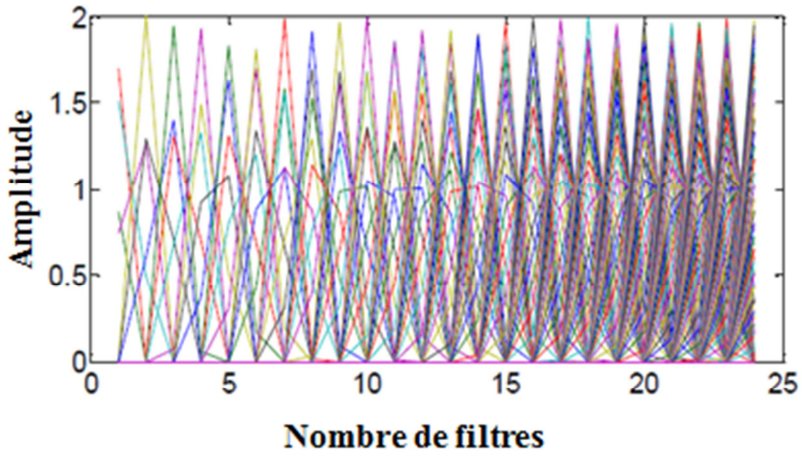

Figure III. 6: Représentation du banc de filtres à l'échelle des Mels.

III.1.8 Coefficients cepstraux

Il est possible d'utiliser directement les coefficients obtenus à la sortie des filtres pour la reconnaissance de la parole. Cependant, d'autres coefficients plus discriminants, plus robustes au bruit ambiant et surtout décorrelés entre eux sont préférés. Il s'agit d'un ensemble de coefficients cepstraux, généralement entre 10 et 15, calculés en effectuant un liftrage (filtrage dans le domaine cepstral) du spectre en puissance du signal. Dans cette étape finale, on transforme les données dans l'échelle des Mels (fréquentielle donc) vers l'échelle des temps. Le résultat de cette étape sera les MFCC proprement dit. Il suffit d'effectuer l'inverse de la transformée de Fourier. Dans la pratique, on effectue une transformée en Cosinus Discrète inverse (IDCT), ce qui revient au même puisque la transformée en Cosinus inverse donne la partie réelle de la transformée de Fourier ; Or ici on n'a que des réels. Il faut noter que la transformée en sinus donnera la partie imaginaire de la transformée de Fourier.

Les coefficients MFCC donnent une bonne représentation des propriétés spectrales locales du signal. Dans notre projet, nous avons utilisé 13 coefficients MFCC, donc un total de 39 traits acoustiques :

- ✓ 13 coefficients absolus : (Energie Absolue (1) et MFCC (12)).
- ✓13 coefficients Delta : (dérivée première des 13 coefficients absolus).
- ✓13 coefficients Delta Delta : (dérivée deuxième du 13 coefficients absolus).

Nous pouvons donc calculer les MFCC à partir de la formule suivante :

$$C_n = \sum_{k=1}^{K} (logS_k) \left[n \left(k - \frac{1}{2} \right) \frac{\pi}{K} \right] \quad , n = k = 1,2, \dots, K \qquad \text{III. 8}$$

Ou S_k sont les sorties de l'étape précédente et C_n sont les coefficients cepstraux.

Les coefficients MFCC sont représentés sur la Figure III.7 :

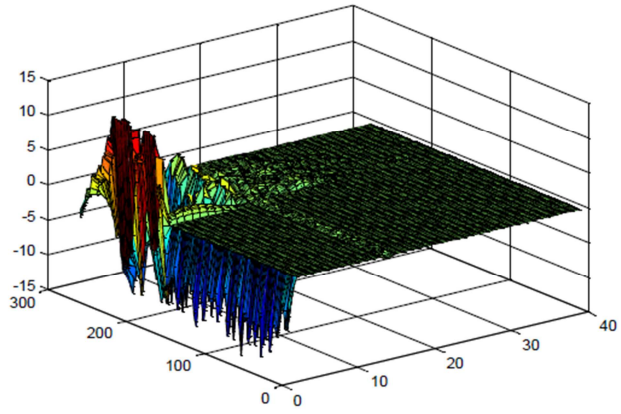

Figure III. 7: Représentation 3D des coefficients MFCC.

III.1.9 Création de la base de données

Le dictionnaire de références regroupe tous les modèles de mots du vocabulaire utilisés pour la reconnaissance et il est créé par apprentissage. Il est important d'avoir des modèles qui représentent bien les mots du vocabulaire pour obtenir une bonne performance de la reconnaissance.

Cette étape consiste à générer des modèles pour l'ensemble de commandes à exécuter. Cet ensemble comprend 4 exemplaires (quatre locuteurs différents) pour chacun des cinq commandes : " Left ", " Right", "Stop", " Forward " et " Backward ". Après avoir effectué l'extraction de caractéristiques du mot à tester, il est mis en correspondance avec l'ensemble des 20 commandes enregistrées dans la base de données afin de calculer la distance qui sépare les MFCCs du signal parole de test avec chacune d'elles.

III.1.10 Alignement Temporel Dynamique DTW

Dans ce type de technique de reconnaissance vocale, les données de test sont converties en modèles. Le processus de reconnaissance consiste alors de faire correspondre le mot entrant avec les modèles stockés. Le modèle, avec la mesure de distance la plus faible par rapport au mot d'entrée, est le mot reconnu. Le calcul de la meilleure ressemblance (la distance la plus petite) est basé sur la programmation dynamique.

Pour mieux comprendre la DTW, deux concepts doivent être pris en compte :

- **Caractéristiques :** les informations contenues dans chaque signal doivent être représentées d'une certaine manière.
- **Distances :** une certaine métrique doit être employée afin d'obtenir le chemin entre les deux vecteurs de caractéristiques.

Il existe deux types de distances :

- **Locale :** c'est la distance entre une caractéristique d'un signal et une caractéristique d'un autre signal .
- **Globale :** c'est la distance globale entre un signal entier et un autre signal de longueur éventuellement différente.

La mise en œuvre de l'algorithme DTW (détaillé dans le chapitre II), permet la comparaison dynamique d'un cepstre MFCC avec les cepstres préenregistrés dans la base de données. Un exemple de comparaison entre le cepstre du mot « Avance » et les cepstres des cinq commandes (Avance, Recule, Gauche, Droite et Stop) est donné sur la Figure III.8 :

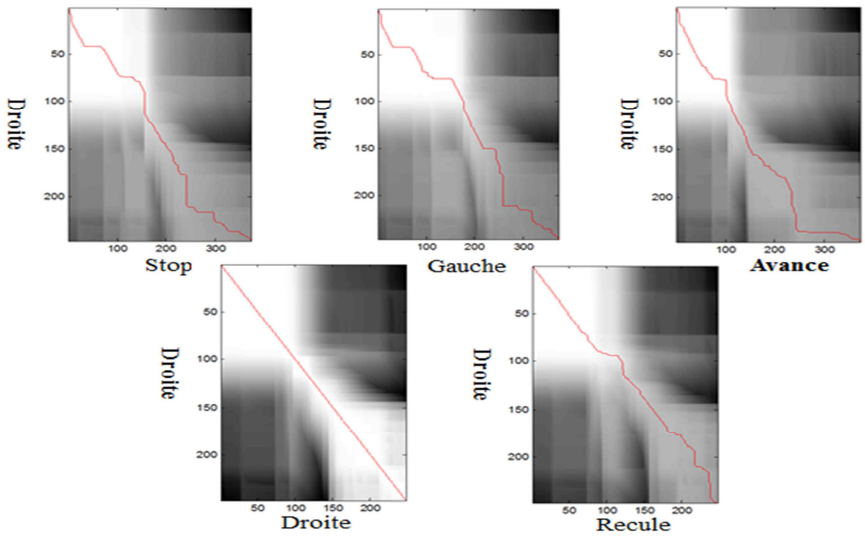

Figure III. 8: Chemin minimal entre les cepstres de deux mots.

Le résultat de la comparaison est une matrice nommée SCORES, elle contient les 20 valeurs de comparaison entre la commande à tester et les 20 commandes préenregistrées.

On constate donc, que la DTW fournie les mesures de similarité nécessaires à la classification. Et pour terminer le processus de reconnaissance de la parole on fait recours à l'algorithme d'apprentissage k plus proche voisin (k-ppv) ou KNN (K-NEAREST NEIGHBOR), qui donne le résultat final de la reconnaissance.

III.1.11 Résumé :

Les étapes de réalisation de notre système de commande vocale sont :

- Création d'une base de données composée des cepstres MFCC des cinq commandes (Stop, Forward, Right, Left et Backward).
- Mesure de similarité entre le cepstre du mot prononcé et les cepstres de la base de données (Rôle de la DTW).
- Attribution du cepstre à la classe la plus proche (Rôle de l'algorithme KNN).
- Interprétation des commandes vocales par des actions de la plateforme mobile.

Les résultats de simulation sont montrés sur la figure III.9 :

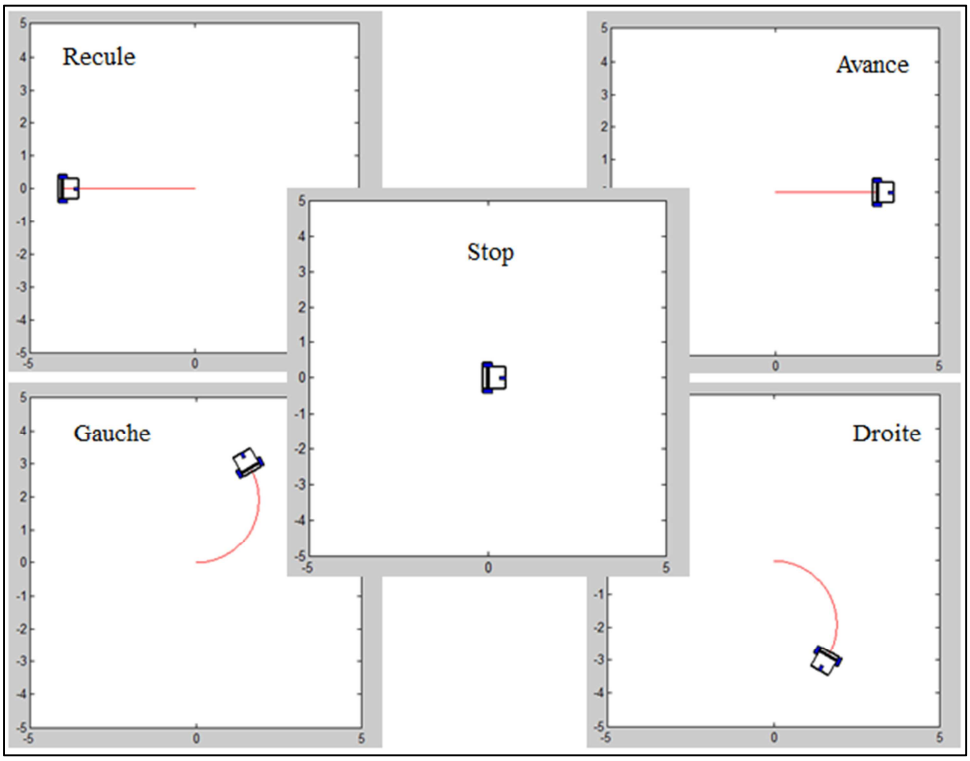

Figure III. 9: Simulation de la commande vocale d'une plateforme mobile.

L'algorithme DTW est un très bon outil capable de comparer deux spectres audio ayant des durées différentes, des débits ou des intensités différentes, et cela de façon optimale en recherchant le meilleur chemin pour passer d'un spectre à l'autre. Néanmoins d'autres méthodes existent, bien plus puissante que l'algorithme DTW mais bien plus complexe et gourmand en terme du temps de calcul. La même chose peut être dite sur la paramétrisation, car il existe des implantations très poussées des MFCC, comme par exemple le système Sphinx.

III.2 HMM Modèle de Markov caché

III.2.1 Introduction

Un modèle de Markov caché (HMM) ou plus correctement (mais non employé) automate de Markov à états cachés, est un modèle statistique dans lequel le système modélisé est supposé être un processus Markovien de paramètres inconnus. Dans un modèle de Markov caché, les états d'une exécution sont inconnus de l'utilisateur.

Les modèles de Markov cachés sont massivement utilisés notamment en reconnaissance de formes, en intelligence artificielle ou encore en traitement automatique du langage naturel.

On formalise un automate HMM comme étant un quadruplet des ensembles$\{S, \pi, A, B\}$ décrits comme suit:

- Si l'état i .
- π_i la probabilité que S_i soit l'état initial.
- a_{ij} probabilité de la transition.
- $b_i(k)$ la probabilité d'émettre le symbole k étant dans l'état S_i .

sous contraintes :

- $\sum_i \pi_i = 1$ la somme des probabilités des états initiaux est égale à 1 ;
- $\forall i, \sum_i a_{ij}$=1 la somme des probabilités des transitions partant d'un état est égale à 1 ;
- $\forall i, \sum_k b_i(k) = 1$ la somme des probabilités des émissions partant d'un état est égale à 1.

L'exemple suivant fournit plus d'explication pour un modèle de Markov à temps discret Imaginons un jeu simple, avec des sacs en papier (opaques) contenant des jetons numérotés. À chaque tour du jeu nous tirons un jeton d'un sac et, en fonction du jeton, passons à un autre sac. Après chaque tour, le jeton est remis dans le sac, nous notons enfin la séquence des numéros tirés.

Exemple 1: Nous disposons de deux sacs, appelés A et B, ainsi que d'un ensemble de jetons numérotés a et b.

Dans chaque sac nous plaçons un certain nombre de jetons a et un certain nombre de jetons b : dans cet exemple, nous plaçons dans le sac A 19 jetons b et un seul jeton a. Dans le sac B nous plaçons 4 jetons a et un seul jeton b.

- Nous commençons par piocher un jeton au hasard dans le sac A. Si l'on pioche un jeton a, on reste sur ce sac, si l'on pioche un jeton b, on passe au sac B. On note également quel jeton a été tiré et on le remet dans le sac.
- On recommence cette étape avec le sac en cours, jusqu'à ce que le jeu s'arrête (au bon vouloir du joueur).Nous avons les probabilités de passer à une station suivante :

	Tirage suivant en *a*	**Tirage suivant en *b***
Station courante en *A*	0,05	0,95
Station courante en *B*	0,8	0,2

En jouant plusieurs parties, nous sommes susceptibles d'obtenir les séquences suivantes :

- a b a b a b a a b a
- a b b a b a b a b a
- a b b a b b a b a b
- …

Ce jeu peut-être modélisé par une chaîne de Markov: chaque sac représente un **état**, la valeur du jeton donne la **transition**, la proportion de jeton d'une valeur est **la probabilité de la transition**.

Notre exemple du jeu du sac en papier est équivalent à l'automate de Markov suivant :

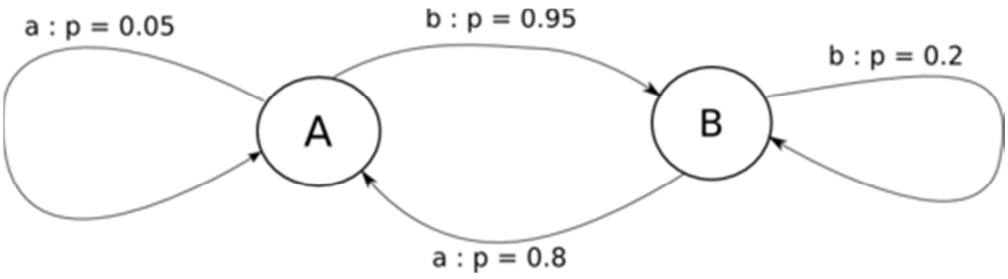

Figure III. 10: Modèle du jeu de sac en HMM

Nous reprenons en partie le modèle précédent mais introduisons de nouveaux types de sacs :
Des sacs pour savoir dans quel sac effectuer le prochain tirage ;
Des sacs de sortie pour générer la séquence.
À partir de la séquence générée, il sera généralement impossible de déterminer quels tirages ont conduit à quelle séquence, la séquence de tirage dans les sacs donnant les transitions est inconnue, c'est pourquoi on parle de sacs en papier cachés.

Exemple 2: Repartons de l'exemple précédent. Nous conservons les sacs A et B, qui donnent les transitions, et ajoutons deux sacs A' et B' (contenant des jetons j et k), situés juste à côté :

- A' contient quatre jetons j et un jeton k ;
- B' contient un jeton j et quatre jetons k.

Le jeu est le suivant :

- On part du groupe de sacs (A et A') , on tire un jeton dans le sac A', on consigne sa valeur (j ou k) et on le replace dans le sac ;
- On tire un jeton dans le sac A pour savoir dans quel groupe de sacs se feront les prochains tirages, on le replace ;
- Si le jeton sortant du sac A est un 'a' alors les prochains tirages se feront dans le groupe de sac (A et A'), si c'est un 'b', il se fera dans le groupe de sac (B et B') ;
- On recommence ces opérations autant de fois que le joueur le souhaite.

À chaque étape, on tire donc un jeton dans chaque sac d'un même groupe, à savoir A et A' ou B et B', ce qui permet d'avoir une valeur (j ou k) qui n'indique pas directement la transition.
Le jeu génère deux séquences :

- La séquence de sortie, connue, le résultat du jeu (ce sont les valeurs j ou k contenues dans les sacs A' et B') ;
- La séquence des transitions, inconnue (ce sont les valeurs a et b contenues dans les sacs A et B).

Pour cet exemple, nous avons pu générer les séquences suivantes :

Séquence de transition	A B A B	A B B A	A A B A	A B A B	A B A B
Séquences de sortie	j j k k	j k k j	k j j j	j k k j	k k j k

On observe que des séquences de transitions identiques peuvent donner des sorties différentes, et vice-versa.

Ce jeu peut être modélisé par un *Automate de Markov à états cachés* : les *groupes de sacs* sont les états, les tirages donnant le groupe de tirages suivant sont les **transitions** (avec la probabilité associée en fonction de la proportion des jetons dans les sacs *A* ou *B*), les sacs de sortie donnent les **valeurs de sortie** de l'automate (avec la probabilité associée en fonction de la proportion des jetons dans les sacs A' ou B').

Le jeu précédent correspond donc à l'**automate de Markov à états cachés** suivant :

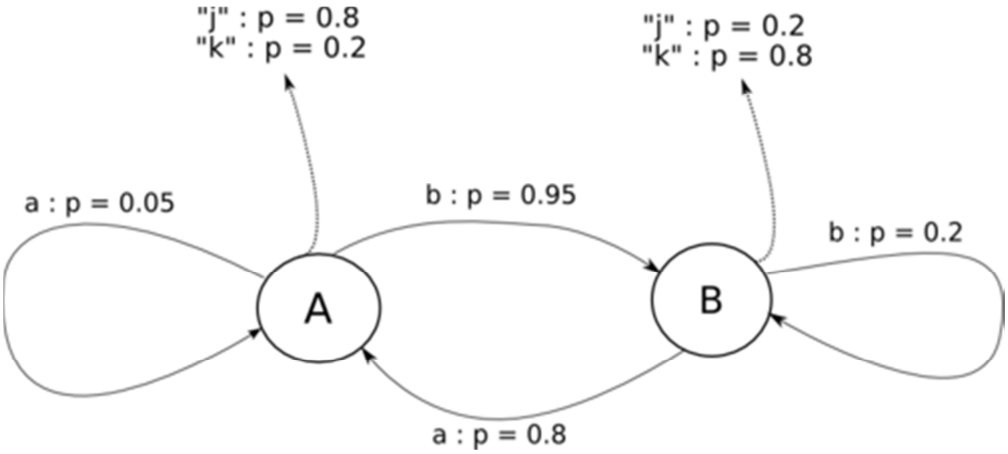

Figure III. 11: Modèle du jeu de sac en HMM

III.2.2 Utilisation des HMMs

Il y a trois exemples typiques de problèmes qu’on peu résoudre avec un HMM :

a. Connaissant l'automate, calculer la probabilité d'une séquence particulière (se résout à l'aide de l'algorithme Forward **(en)**) ;
b. Connaissant l'automate, trouver la séquence la plus probable d'état (caché) ayant conduit à la génération d'une séquence de sortie donnée (se résout avec l'algorithme de Viterbi) ;
c. Étant donné une séquence de sortie, retrouver l'ensemble d'états le plus probable et les probabilités des sorties sur chaque état. Se résout avec l'algorithme de Baum-Welch, appelé aussi algorithme forward-backward.

III.2.3 Définition du problème de reconnaissance de mots isolés IWR :

Dans un système d'IWR, le but est de préserver l'information temporelle et spectrale requise pour déterminer l'identité phonétique des unités de la parole et pour ignorer des facteurs

comme la déformation due au bruit de fond. Dans la pratique, un modèle est construit pour chaque mot du vocabulaire (Stop, Backward, Right, Left, Forward). Ainsi, chaque mot a un modèle probabiliste formé d'une séquence de symbole observable. La tâche du système IWR est de trouver le meilleur mot assorti parmi un vocabulaire dont le modèle est le plus susceptible de produire de la séquence donnée d'observation des vecteurs de caractéristique extraits à partir d'un son articulé .Le schéma de la figure III.12 montre le mécanisme simplifié d'un classificateur discret HMM.

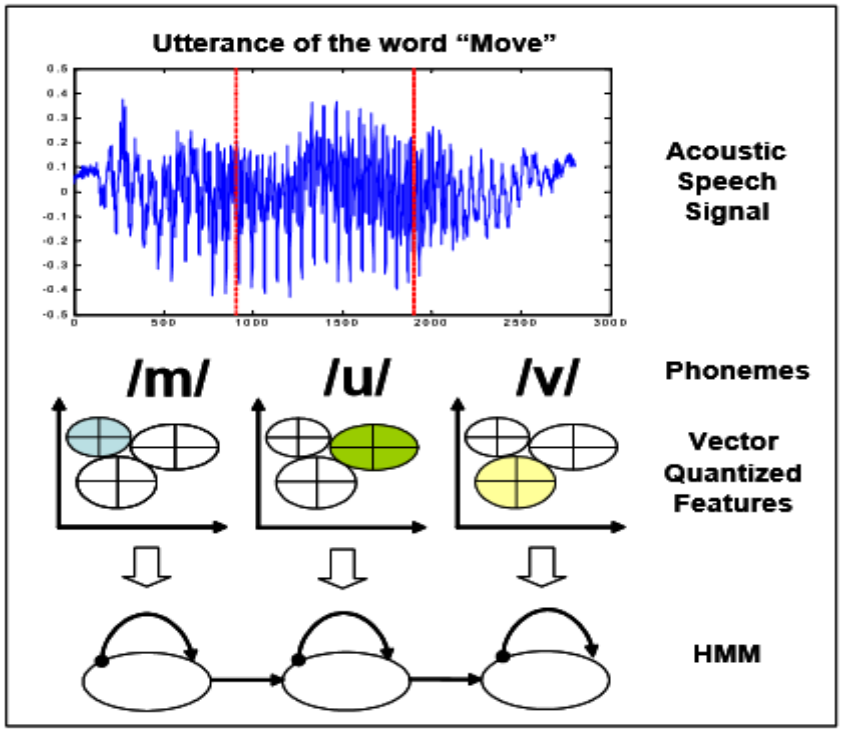

Figure III. 12: Une illustration d'un classificateur discret d'observation HMM pour IWR.

Supposons qu'une séquence d'observation (ou un ensemble de symboles) correspond à un ensemble de vecteurs de caractéristique extraits à partir d'un signal acoustique. Cette séquence d'observation est représentée par : $O = \{o_1, o_2, \ldots, o_M\}$. De plus, supposons que les modèles des mots à reconnaître correspondent à $W = \{w_1, w_2, \ldots, w_N\}$, l'équation suivante estime le mot le plus susceptible du vocabulaire de L-Mot, ayant la séquence d'observation O :

$$\hat{w} = argmax_{W \in L}[P(W|O)] \qquad \text{III. 9}$$

En appliquant la règle de Bayes

$$P(x|y) = \frac{P(y|x)P(x)}{P(y)} \qquad \text{III. 10}$$

à l'équation précédente on obtient :

$$\widehat{w} = argmax_{W \in L}[P(W|O)] = argmax_{W \in L}\left[\frac{P(O|W)P(W)}{P(O)}\right] \qquad \text{III. 11}$$

Dans cette équation $P(W)$ est la probabilité a priori d'avoir un modèle particulier W, qui produit le mot à reconnaître. Heureusement, $P(O)$, la probabilité de la séquence d'observation, peut être ignorée puisque cette probabilité est constante, en essayant de maximiser les probabilités de tous les modèles possibles de mot [Jel97]. Par conséquent, l'équation qui en resulte devient :

$$\widehat{w} = argmax_{W \in L}[P(O|W)P(W)] \qquad \text{III. 12}$$

$P(O|W)$, la probabilité d'avoir une séquence d'observation O ayant le model W, peut être obtenu par le Model du Markov Caché (HMM)

Notations :

Les différents ensembles de notations pour la modélisation cachée de Markov existent dans la littérature. Le tableau III.1 énumère la notation préliminaire qui représente HMM les concepts de base. Des concepts plus avancés suivent sur ces notations :

Notation	Définition
$Q = \{q_1, q_2, .. q_t \ldots., w_s\}$	Q l'ensemble des états possible q_t au temps t
$O = \{o_1, o_2, .. o_t \ldots., o_k\}$	O est l'ensemble des symboles observé, k le nombre des symboles dans le codebook et o_t est le symbole observé au temps t
$A = \begin{pmatrix} a(1\|1) & \cdots & a(1\|S) \\ \vdots & \ddots & \vdots \\ a(S\|1) & \cdots & a(S\|S) \end{pmatrix}$	A c'est la matrice de transition $a(i\|j) = P(q_t = i\|q_{t-1} = j)$ est la probabilité d'être à l'état i au temps t sachant qu'il était à l'état j au temps t-1 , la somme de chaque colonne est 1
$B = \begin{pmatrix} b(1\|1) & \cdots & b(1\|S) \\ \vdots & \ddots & \vdots \\ b(K\|1) & \cdots & b(K\|S) \end{pmatrix}$	B est la matrice des probabilités des observations où $b(k\|i) = P(o_t = k\|q_t = i)$

$\pi = \begin{pmatrix} P(q_t = 1) = \pi_{q_1} \\ \vdots \\ P(q_t = i) = \pi_{q_i} \\ \vdots \\ P(q_t = S) = \pi_{q_s} \end{pmatrix}$	π est le vecteur de probabilité de l'état 'initial Qui montre la probabilité d'avoir un état quelconque au temps t=1, la somme des éléments égale à 1
$\lambda = (A, B, \pi)$	λ est la notation d'un HMM
$\alpha_t(i) = P(o_1, o_2, .. o_t, q_t = i \mid \lambda)$	Le Variable Forward
$\beta_t(i) = P(o_{t+1}, o_{t+2}, .. \mid q_t = i, \lambda)$	Le Variable Backward

Tableau III. 1: les notations de base

III.2.4 Processus de Markov à temps discret

La chaîne de Markov est le procédé stochastique sous-jacent des modèles cachés de Markov. Une chaîne de Markov à états finis est un processus de Markov stochastique de temps discret dont la production est un ensemble S d'états distincts où chaque état correspond à une observation. [Rab89]. Les transitions d'état (la transition de nouveau à l'état original est également possible) dans un processus de Markov de temps discret peuvent se produire basées seulement sur deux paramètres :

- La probabilité de transition d'état :

$$a(i|j) = P(q_t = i | q_{t-1} = j), 1 \leq i, j \leq S \ , \sum_{i=1}^{S} a(i|j) = 1 \ \forall j \qquad \text{III. 13}$$

- La probabilité d'état initial :

$$P(q_1 = 1)\ 1 \leq i \leq S \quad \sum_{i=1}^{S} P(q_1 = i) = 1 \qquad \text{III. 14}$$

Il y a deux caractéristiques distinctes d'un processus de Markov. D'abord, il y a de la correspondance d'one-to one entre la séquence d'observation et la séquence d'état de chaîne de Markov. En d'autres termes, les observations sont déterministes. En second lieu, la séquence d'état de chaîne de Markov est également observable.

III.2.5 Les Trois problèmes fondamentaux du HMM :

Pour une exécution pratique de HMM, trois problèmes de base doivent être discutés dont les solutions efficaces mèneront à la l'entrainement du modèle, qui est l'estimation des paramètres du modèle, $\lambda = (A, B, \pi)$, et finalement le classificateur qui en résulte.

Problème 1 : Ayant une séquence d'observations et un HMM :

$O = \{o_1, o_2, .. o_t \ldots .., o_k\}$, $\lambda = (A, B, \pi)$

Quelle est la probabilité $P(O|\lambda)$, la probabilité que la séquence d'observation est produite par le modèle donné, calculée ? Le problème 1 est connu comme évaluation ou problème de marquage « scoring problem ».

Problème 2 : Ayant une séquence d'observation et un HMM $O = \{o_1, o_2, .. o_t \ldots., o_T\}$, $\lambda = (A, B, \pi)$ comment sera la séquence d'état $Q^* = \{q_1, q_2, .. q_t \ldots., q_T\}$ qu'explique les observations de façon optimale. Une solution qui satisferait $Q^* = argmax_{all\ Q}[P(Q, O|\lambda)]$ ce problème est connue comme le problème de décodage parce que le but est de découvrir la partie cachée du modèle, la séquence optimale d'état.

Problème 3: Quel est le paramètre du model $\lambda = (A, B, \pi)$ qui maximise $P(O|\lambda)$, la probabilité de la séquence d'observation connaissant le model. Ce problème représente le problème d'entrainement

III.2.5.1 Solution du problème de marquage « Scoring Problem » :

Le calcul de la probabilité de l'occurrence d'une séquence particulière d'observation connaissant le modèle peut également être considéré comme un problème de marquage dans lequel la tentative est de déterminer l'aptitude d'un modèle donné pour produire une séquence particulière d'observation. Ce point de vue devient utile en visant de choisir le meilleur modèle pour un mot indiqué parmi beaucoup de modèles de concurrence où chaque modèle représente un mot d'un petit vocabulaire [Rab89].

Fondamentalement, il y a deux options disponibles pour évaluer $P(O|\lambda)$; l'évaluation directe de force brutale et la procédure forward-backward. L'ancienne approche, appelée l'évaluation directe, est impraticable pour beaucoup d'applications du monde réel. La dernière méthode, appelée la procédure forward-backward, est intensivement employée pour aborder le problème d'évaluation dans HMM.

Forward-Backward procedure :

On définit la variable Forward :

$$\alpha_t(i) = P(o_1, o_2, .. o_t, q_t = i|\lambda) \qquad \text{III. 15}$$

la variable Forward $\alpha_t(i)$ est décrit comme probabilité conjointe de la séquence partielle d'observation jusqu'à instantané t, $o_1, o_2, .. o_t)$ et l'état i à l'instant t, connaissant le modèle. La récursion en avant peut être employée pour résoudre $\alpha_t(i)$ inductivement comme suit :

Initialisation : pour $1 \leq i \leq S$

$$\alpha_1(i) = P(o_1, q_1 = i|\lambda) = \pi_i b(o_1|q_1) \qquad \text{III. 16}$$

Récursion : pour $t = 1,2, \dots, T-1 \; et \; 1 \leq i \leq S$

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{S} \alpha_t(i) a(j|i)\right] b(o_{t+1}|j) \qquad \text{III. 17}$$

Arrêt : pour $1 \leq i \leq S$

$$P(O|\lambda) = \sum_{i=1}^{S} \alpha_t(i) \qquad \text{III. 18}$$

Dans la première étape, les probabilités Forward sont initialisées comme probabilité conjointe d'avoir l'observation O_t à l'état initial i pour tous les états possibles. Dans la deuxième étape, $\alpha_{t+1}(j)$, la probabilité conjointe de la séquence partielle d'observation jusqu'à l'instant t+1 et l'état j à l'instant t+1 est calculé des probabilités de $\alpha_t(i)$ des observations partielles précédentes, récursivement. La structure de treillis représentée sur le schéma de la figure suivante illustre comment l'état j au temps t+1 peut être atteint indépendamment de l'un des états de S au temps T. La troisième étape de la récursion Forward prouve que le calcul désiré de $P(O|\lambda)$ est obtenu à partir de l'addition des variables Forward finales indépendantes de S , $\alpha_t(i)'s$ qui réalisent la séquence d'observation [Rab93].

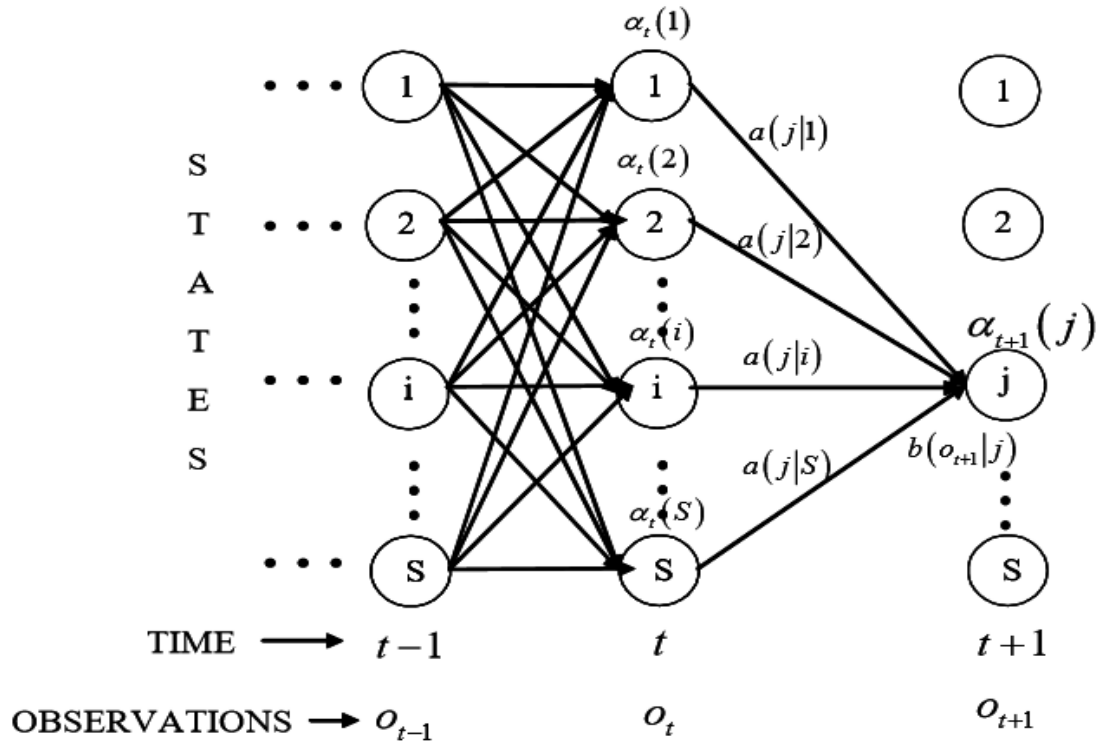

Figure III. 13: La structure de treillis employée pour dériver la récursion Forward daprès [Del00].

Après, la variable Backward $\beta_t(i)$ est défini d'une manière semblable comme a été fait avec la variable Forward ,comme suit :

$$\beta_t(i) = P(o_{t+1}, o_{t+2}, .. o_T | q_t = i, \lambda) \qquad \text{III. 19}$$

Initialisation : pour $1 \leq i \leq S$

$\beta_t(i) = 1$

Récursion : pour $t = T-1, T-2, \ldots, 1$ *et* $1 \leq j \leq S$

$$\beta_t(i) = \left[\sum_{j=1}^{S} a(j|i)\, b(o_{t+1}|j)\beta_{t+1}(j)\right].$$

Arrêt : pour $1 \leq i \leq S$

$$P(O|\lambda) = \sum_{i=1}^{S} \pi_i b(o_1|i)\beta_1(i)$$

III.2.5.2 Solution du problème de décodage :

La solution au problème 1 calcule $P(O|\lambda)$ en sommant tous les séquences d'état possibles par une structure de treillis dans le HMM, par conséquent il n'a pas comme conséquence la séquence optimale d'état qui explique la séquence d'observation. Dans le problème 2, l'unique séquence d'état (chemin) la plus vraisemblable $Q^* = \{q_1, q_2, .. q_t \ldots., q_T\}$ est cherchée pour une observation et model donnés.. En d'autres termes, le but est de trouver le chemin optimal qui satisfait $Q^* = argmax_{all\ Q}[P(Q, O|\lambda)]$, considérant cela qui maximise la probabilité $P(Q|O, \lambda)$ est équivalent au maximum $P(Q, O|\lambda)$. L'algorithme de Viterbi peut être employé pour déterminer une telle séquence optimale d'état comme une méthode de programmation dynamique appliquée à HMM [Rab93].

Algorithme de Viterbi :

L'algorithme de Viterbi choisit et maintient la séquence d'état la plus susceptible pour chaque état possible de S à chaque instant t récursivement à la différence de la Forward récursion qui additionne tous les transitions d'états entrants possibles atteignant le même état de destination. Une fois que l'algorithme de Viterbi atteint l'étape finale à la fin de la séquence d'observation, il emploie le retour-trace pour trouver le chemin le plus probable.

La reformulation suivante du problème de décodage fournit une analyse dans l'algorithme de Viterbi pendant qu'elle est appliquée à l'évaluation de la séquence d'état optimale dans HMM [Dug96]. Finalement, le but est de maximiser la probabilité $P(Q,O|\lambda)$.

Le principe de la probabilité conditionnelle est employé d'abord pour obtenir :

$$P(Q,O|\lambda) = P(Q|O,\lambda)P(Q|\lambda) \quad \text{III. 20}$$

$$P(Q,O|\lambda) = \pi_i b(o_1|q_1)a(q_2|q_1)b(o_2|q_2)\dots a(q_T|q_{T-1})b(o_T|q_T) \quad \text{III. 21}$$

Donc la fonction distance, associée avec toute séquence possible à travers le trajet allant du t=1 jusqu'au t=T, est calculée en prenant le négatif logarithme de $P(Q,O|\lambda)$

$$D_T = D(q_1,q_2,..q_t \dots .,q_T) = -\left[\log(\pi_{q_1} b(o_1|q_1)) + \sum_{t=2}^{T} \log(a(q_T|q_{T-1})b(o_T|q_T))\right] \quad \text{III. 22}$$

L'algorithme de Viterbi se présente comme suit :

1. **Initialization** : $for\ 1 \le i \le S$

$$\delta_1(\mathrm{i}) = -\log(\pi_\mathrm{i}) - \log\big(\mathrm{b}(\mathrm{o}_1|\mathrm{i})\big) \quad \text{III. 23}$$

$$\Psi_1(\mathrm{i}) = 0, \quad \text{III. 24}$$

2. **Recursion** : $for\ 1 \le t \le T$ and $1 \le j \le N$

$$\delta_\mathrm{t}(\mathrm{j}) = \min_{1\le \mathrm{i} \le \mathrm{S}}\left[\delta_{\mathrm{t}-1}(\mathrm{i}) - \log\big(\mathrm{a}(\mathrm{j}|\mathrm{i})\big)\right] - \log\big(\mathrm{b}(\mathrm{o}_\mathrm{t}|\mathrm{j})\big) \quad \text{III. 25}$$

$$\Psi_\mathrm{t}(\mathrm{i}) = \mathrm{argmin}_{1\le \mathrm{i} \le \mathrm{S}}\left[\delta_{\mathrm{t}-1}(\mathrm{i}) - \log\big(\mathrm{a}(\mathrm{j}|\mathrm{i})\big)\right] \quad \text{III. 26}$$

3. **Termination**:

$$\mathrm{P}^* = \min_{1\le \mathrm{i} \le \mathrm{S}}[\delta_\mathrm{T}(\mathrm{i})] \quad \text{III. 27}$$

$$\mathrm{q}_\mathrm{t}^* = \mathrm{argmin}_{1\le \mathrm{i} \le \mathrm{S}}[\delta_\mathrm{T}(\mathrm{i})] \quad \text{III. 28}$$

4. **Backtracking** : $for\ t = T-1, T-2, \dots, 2, 1$

$$\mathrm{q}_\mathrm{t}^* = \Psi_{\mathrm{t}+1}(\mathrm{q}_{\mathrm{t}+1}^*) \quad \text{III. 29}$$

L'argument P^*, le négatif du log de la probabilité du chemin avec le poids minimum, s'appelle le log de la vraisemblance. Bien que l'expression $\exp(-P^*)$ égale strictement à la probabilité désirée, l'argument du log de la vraisemblance est de préférence employé au lieu de la probabilité régulière pour la commodité dans les calculs sans n'importe quelle conversion. Notez que les petites mesures du log de la probabilité correspondent à de très grandes probabilités. En outre, la prise du logarithme négatif est un moyen pour éviter les problèmes de précision et de courant de fond qui se posent quand de petites valeurs de probabilité sont multipliées itérativement.

Le schéma de la figure III.14 montre que l'algorithme de Viterbi fonctionne dans une structure de treillis d'une manière semblable en tant que cela suivi dans la récursion Forward. Les flèches en gras représentent le choix de l'état le plus susceptible à chaque fois que l'étape et la meilleure séquence d'état en résultant :

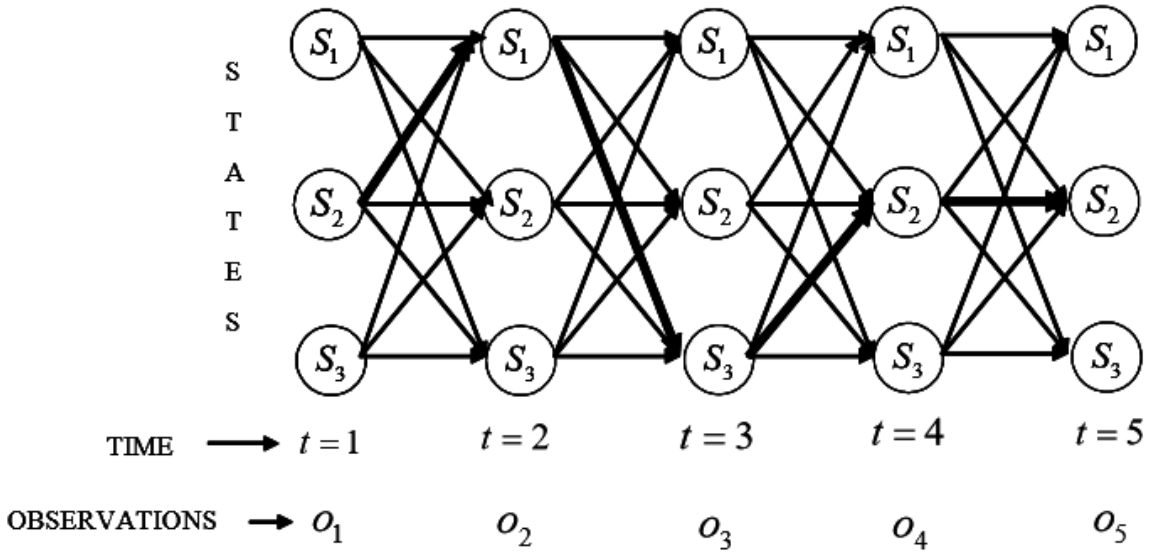

Figure III. 14: Le schéma structure de treillis illustrant l'algorithme de Viterbi pour un HMM à Trois-état

III.2.5.3 Solution du problème d'entrainement :

Le problème d'entrainement fait appelle à la ré-estimation des paramètres modèles $\lambda = (A, B, \pi)$ dans un effort d'apprendre et coder les caractéristiques de la séquence l'observation dans le modèle pour que ce modèle puisse identifier une séquence semblable à l'avenir [Dug96].

Malheureusement, l'optimisation des paramètres d'apprentissage de l'HMM est difficile à réaliser car il n'y a aucune méthode analytique connue qui maximise la probabilité conjointe des données d'apprentissage dans une forme close [Rab93]. Deux méthodes pour traiter ce problème seront discutées :

D'abord, il est possible d'optimiser les paramètres $\lambda = (A, B, \pi)$ pour que la vraisemblance, de l'occurrence pour les données de l'apprentissage $(O|\lambda)$, soit localement maximisée à l'aide de l'algorithme itératif de Baum-Welch, également connu sous le nom d'algorithme de Forward-Backward.

En second lieu, le problème peut être résolu en appliquant d'abord l'algorithme de Viterbi, et en maximisant après la probabilité $P(O|\lambda)$ à travers la plus probable séquence d'état.

a. L'algorithme de Baum Welch :

Cette méthode commence par un premier λ modèle arbitraire et cherche par la suite un nouveau modèle $\hat{\lambda}$ qui améliore la probabilité que la séquence d'observation donnée est produit par le nouveau modèle, à chaque itération jusqu'à ce qu'un maximum soit atteint, tel que :

$$P(O|\hat{\lambda}) \geq P(O|\lambda) \qquad \text{III. 30}$$

Ce critère d'optimisation est connu comme le critère de maximum de vraisemblance. Quelques nouvelles notations de [Rab93 ; Dug96] sont d'abord présentés pour expliquer l'algorithme de ré-estimation de Baum Welch . Le terme $\gamma_t(i)$ dénote la probabilité conditionnelle d'être dans l'état i au temps t, ayant donné toute la séquence d'observation le plein ordre d'observation $O = \{o_1, o_2, .. o_t \ldots., o_T\}$ et le modèle $\lambda = (A, B, \pi)$;

$\gamma_t(i) = P(\mathrm{q}_t = i|O, \lambda)$;

En utilisant la définition du variable Forward $\alpha_t(\mathrm{i})$, variable Backward $\beta_t(i)$, la règle de Bayes la relation suivante est obtenu :

$$\gamma_t(i) = \frac{P(\mathrm{q}_t=i|O,\lambda)}{P(O|\lambda)} = \frac{\alpha_t(i)\beta_t(i)}{P(O|\lambda)} \qquad \text{III. 31}$$

On définit La probabilité conditionnelle pour être à l'état i à l'instant t et faisant une transition vers l'état j à l'instant t+1, $\xi_t(i,j)$

$$\xi_t(\mathrm{i},\mathrm{j}) = \mathrm{P}(\mathrm{q}_t = \mathrm{i}, \mathrm{q}_{t+1} = \mathrm{j}|\mathrm{O}, \lambda) \qquad \text{III. 32}$$

En utilisant les variables Backward et Forward et la règle de Bayes on obtient

$$\xi_t(\mathrm{i},\mathrm{j}) = \frac{\alpha_t(\mathrm{i})\mathrm{a}(\mathrm{j}|\mathrm{i})\mathrm{b}(\mathrm{o}_{t+1})\beta_{t+1}(\mathrm{j})}{\sum_{\mathrm{i}=1}^{S}\sum_{\mathrm{j}=1}^{S}\alpha_t(\mathrm{i})\mathrm{a}(\mathrm{j}|\mathrm{i})\mathrm{b}(\mathrm{o}_{t+1})\beta_{t+1}(\mathrm{j})} \qquad \text{III. 33}$$

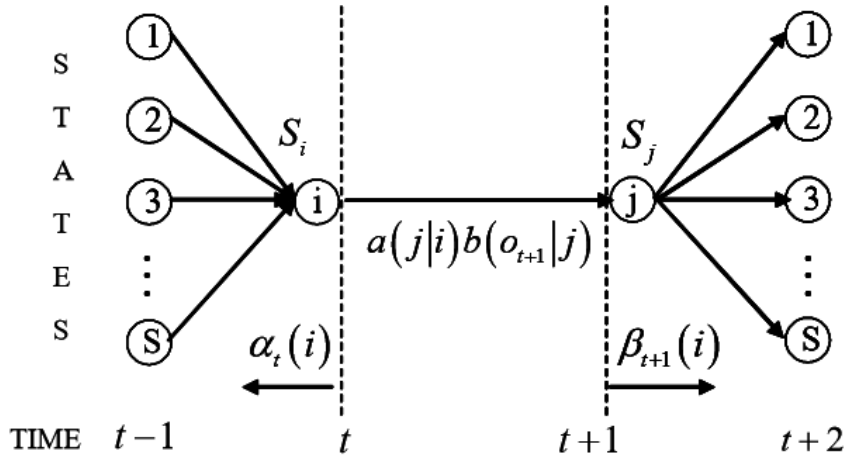

Figure III. 15: Illustration du calcul de $\xi_t(i,j)$

Il est à noter que $\gamma_t(i)$ est lié au $\xi_t(i,j)$ par une sommation à travers j

$$\gamma_t(i) = \sum_{j=1}^{S} \xi_t(i,j) \qquad \text{III. 34}$$

La sommation de $\gamma_t(i)$ est très utile, ça donne le nombre prévu de transitions de d'état i

$$\sum_{t=1}^{T-1} \gamma_t(i) = \text{le nombre prévu de transitions de d'état i} \qquad \text{III. 35}$$

Maintenant que tous les concepts sous-jacents ont été discutés et leurs notations respectives ont été données, les formules ré-estimation Baum-Welch sont omme suit [Rab93] :

$$\hat{\pi} = \text{le nombre prévu de transitions de d'état i à l'instant t} = 1 \qquad \text{III. 36}$$

$$\hat{\pi} = \gamma_1(i), 1 \le i \le S. \qquad \text{III. 37}$$

$$\hat{a}(j|i) = \frac{\text{nombre prévu de transitions de d'état i à l'état j}}{\text{nombre prévu de transitions de d'état j}} \qquad \text{III. 38}$$

$$\hat{a}(j|i) = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t} \qquad \text{III. 39}$$

$$\hat{b}(k|j) = \frac{\text{nombre prévu de fois à l'état j en observant le symbole } o_t=k}{\text{nombre prévu de fois à l'état j}} \qquad \text{III. 40}$$

$$\hat{b}(k|j) = \frac{\sum_{t=1, o_t=k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)} \qquad \text{III. 41}$$

$$P(O|\hat{\lambda}) - P(O|\lambda) \geq \varepsilon \qquad \text{III. 42}$$

ε est la tolérance

Cependant, l'algorithme de ré-estimation de Baum-Welch s'améliore vers les maximum les plus proches à proximité des paramètres modèles initiaux dû à la nature complexe de la fonction de probabilité de l'apprentissage $P(O|\lambda)$ qui a beaucoup de maximum locaux.

Une manière d'éviter ce problème est de répéter l'algorithme de ré-estimation de Baum-Welch plusieurs fois avec différents paramètres modèles initiaux aléatoires et de sélectionner le meilleur modèle ré-estimé qui renvoie le maximum $P(O|\lambda)$

b. L'algorithme de ré-estimation de Viterbi :

L'algorithme de décodage de Viterbi a été appliqué pour établir la séquence optimale d'état cachée. Cet algorithme peut également être appliqué au probleme de l'apprentissage pour la ré-estimation des paramètres modèles au-dessus de la séquence d'état la plus susceptible, à la différence de l'algorithme de ré-estimation de Baum-Welch qui fonctionne pour toutes séquences d'état possibles.

Dans un HMM installé de gauche à droite, l'état initial peut être indiqué comme état un pour se sauver de la ré-estimation les probabilités d'état initial. D'abord, la séquence d'état optimale Q^* est estimée puis la probabilité $P(O|Q^*, \lambda)$ est évaluée. Les arguments suivants sont également dépistés le long du chemin le plus probable [Del00] :

$n(j|i) =$ nombre de transitions de d'état i à l'état j le long du chemain optimal Q^*

$n(.|i) =$ nombre de transitions de d'état i le long du chemain optimal Q^*

$n(j|.) =$ nombre de transitions à l'état j le long du chemain optimal Q^*

$n(o_t = k, q_t = j) = nombre\ de\ fois\ d'observationsde\ k\ et\ l'état\ j$ le long du chemain optimal optimal Q^*

les formules de ré-estimation de Viterbi sont comme suit :

$$\hat{a}(j|i) = \frac{n(j|i)}{n(.|i)} \qquad \text{III. 43}$$

$$\hat{b}(k|j) = \frac{n(o_t=k, q_t=j)}{n(j|.)} \qquad \text{III. 44}$$

Notez que bien que l'algorithme de ré-estimation de Viterbi semble plus simple et en terme de calcul plus efficace que la ré-estimation de Baum Welch, il n'incorpore pas tous les séquences d'état possibles à la mise à jour du modèle.

III.2.6 Implémentation de l'IWR utilisant HMM :

La modélisation du HMM a été discutée, aussi bien que les paramètres des modèles, leurs structures conjointes et les questions de base d'implémentation. Également était présentés les deux fameux algorithmes d'entrainement pour estimer les paramètres optimaux du modèle. Du point de vue de la modélisation acoustique du signal, HMM peut être considérée comme une combinaison de deux procédés stochastiques. C'est une chaîne de Markov cachée, qui représente la variabilité temporelle d'un signal acoustique dans la séquence caché d'état, et un processus observable qui explique les propriétés spectrales d'un signal acoustique. Basé sur ces connaissances de base, un système d'IWR utilisant la classification discrète de symbole HMM, est maintenant présenté, afin d'identifier cinq ensembles distincts de signaux acoustiques rassemblés du canal d'oreille externe.

Le symbole discret HMM fonctionne avec un ensemble fini de symboles de nombre entier issu d'un codebook. Les K-means clustring algorithme permet de produire d'un tel codebook des vecteurs de caractéristique d'un ensemble de données d'entrainement. Les séquences discrètes d'observation sont dérivées des index du codebook à l'aide des vecteurs de caractéristique d'un mot indiqué du vocabulaire. Le but est d'identifier les mots inconnus indiqués que ces séquences d'observation représentent, à l'aide de la modélisation cachée de Markov. HMM une structure de gauche à droite essaye de modeler le modèle séquentiel vraisemblablement pour être présente dans les symboles d'un ensemble donné d'observation. Ainsi, il devrait préciser qu'un modèle de gauche à droite est plus approprié pour l'IWR [Rab93].

Par conséquent, la structure modèle choisie pour la reconnaissance est une HMM « gauche à droite » avec des symboles discrets d'observation.

Une des unités linguistiques de base utilisées généralement dans la modélisation cachée de Markov est « le phonème ». En fait, un phonème n'est rien davantage qu'une unité abstraite qui pourrait avoir de diverses formes acoustiques d'un contexte à l'autre ou parfois d'un locuteur à un autre, dû aux variations naturelles de prononciation.

Des chercheurs proposent que le nombre d'états soit au moins égal au nombre de phonèmes contenus dans le mot indiqué. D'autres rapportent qu'il vaut mieux assigner deux ou parfois trois états par phonème prenant en considération le phonème lui-même et les transitions venant à ou partant de lui [Den03 ; Bec99]. Dans la plupart des applications réelles des HMM, les résultats expérimentaux suggèrent la taille modèle appropriée [Del00]. Sélectionnant trop petit un certain nombre d'états auraient comme conséquence la classification pauvre due à la

capacité manquante dans le modèle. En outre, il convient de noter que les grands modèles aussi ont un coût accru en terme de calcul et de mémoire. Pendant la phase d’entrainement, des tailles modèles de cinq à huit états ont été expérimentées pour déterminer la taille modèle appropriée. Les résultats expérimentaux de cette étude ont prouvé que huit états sont suffisants pour modeler les unités linguistiques des phonèmes actuels dans le vocabulaire.

L'hypothèse est que le nombre d'états devrait correspondre rudement au nombre moyen des sons distincts (des phonèmes ou des syllabes) dans chaque mot. Ainsi, un huit-état HMM a été sélectionné.

Il y a principalement deux approches pour mettre en application un système d’IWR utilisant HMMs. La première technique modèle chaque mot dans un petit vocabulaire avec un HMM distinct. La deuxième est plus appropriée pour de plus grands vocabulaires dont le but est de modèler les unités de base de sous-mot telles que des phonèmes, des diphones ou des triphones. Dans cette méthode, tous ces modèles devraient être combinés pour expliquer le mot entier. L'ancienne technique a été choisie en modelant chacun des sept mots dans le vocabulaire par un HMM distinct suivant les indications du schéma de la figure III.16.

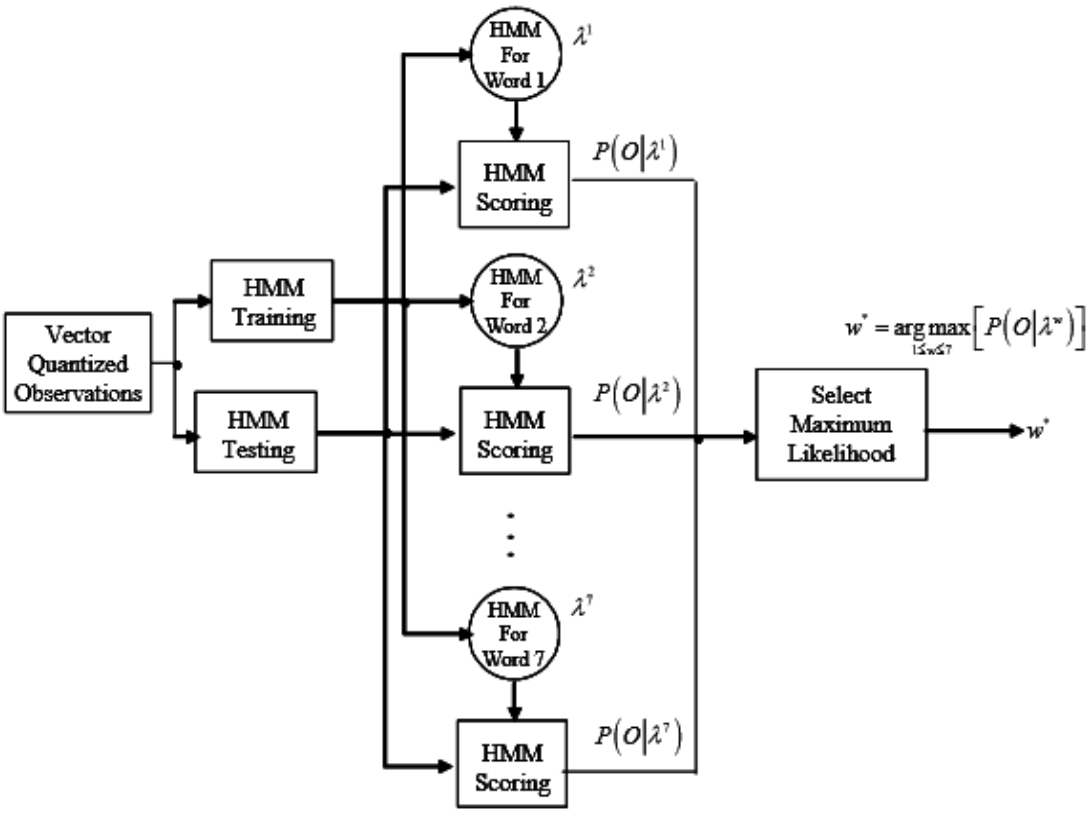

Figure III. 16: Diagramme en Bloc du système IWR

La procédure suivie dans l'IWR peut être divisée en deux étapes :

D'abord, un HMM distinct est construit pour chaque mot dans le vocabulaire, où chaque modèle de mot a le même nombre d'états. Cette étape implique la phase d'entrainement pour estimer les paramètres du modèle $\lambda^w = (A, B, \pi)$ en utilisant l'algorithme de réévaluation de Viterbi .

En second lieu, le HMMs entrainés sont employés pour identifier chaque mot inconnu dans l'ensemble d'essais. La phase de reconnaissance implique le calcul du vraisemblance $P(O|\ \lambda^w)$ pour tous les modèles possibles connaissant la séquence d'observation appartenant au mot inconnu indiqué. Spécifiquement, la récursion Forward-Backward est employée pour calculer $P(O|\ \lambda^w)$ pour les cinq modèles entrainés et l'index, du modèle avec le maximum de vraisemblance ,est identifiée comme mot indiqué tel que :

$$w^* = \text{argmax}_{1\leq w\leq 7}[P(O|\ \lambda^w)\ ] \qquad \text{III. 45}$$

III.2.7 Aperçu de système :

L'objectif principal de l'étude était d'identifier un ensemble de cinq mots parlés, dans lequel les signaux acoustiques ont été rassemblés utilisant un système d'IWR. Le petit vocabulaire se compose de cinq mots : {Stop, Right, Left, Backward, Forward} et la base de données de la parole a été produite utilisant 4 sujets adultes qui ont poussé chaque mot 10 fois. Un système de reconnaissance de HMM de symbole discret a été choisi parce que le vocabulaire à l'étude est petit et se compose des mots courts. Le système d'IWR suppose que le son articulé est une réalisation d'un certain message composé d'unités de base de sous-word, par exemple, les phonèmes, qui peuvent être considérés comme séquence des symboles d'un codebook unique. Le schéma III.17 représente le schéma fonctionnel global du système d'IWR HMM utilisé dans l'étude, le Toolbox Matlab utilisé c'est le HMM Toolbox

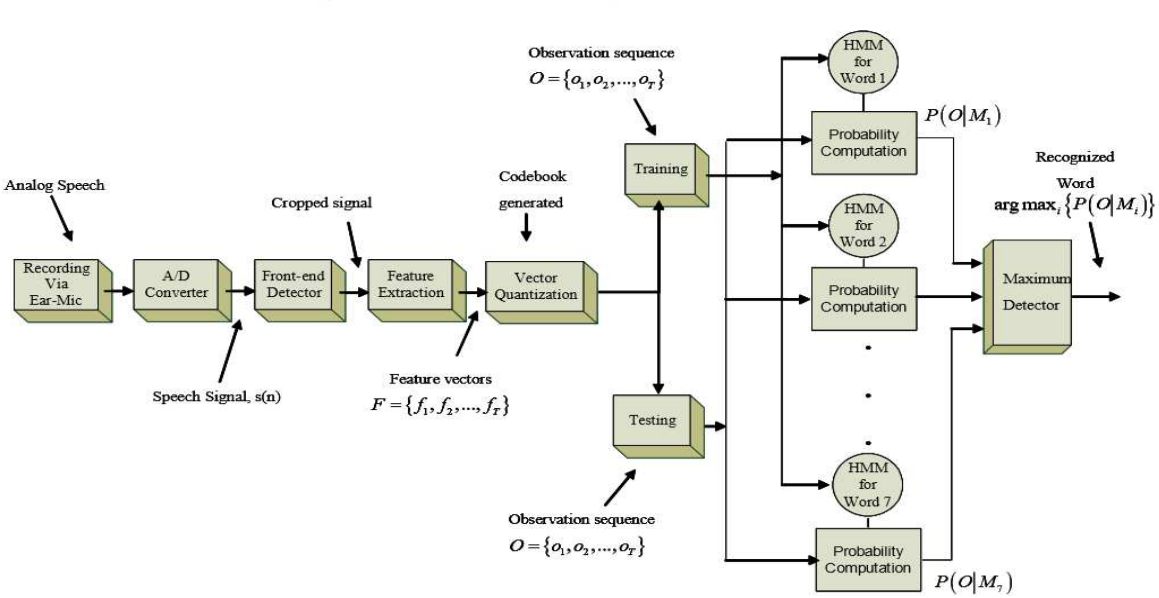

Figure III. 17: Schéma fonctionnel global du système d'IWR HMM

III.3 Le modèle GMM :

III.3.1 Introduction :

La modélisation par mélange de gaussiennes (GMM - Gaussian Mixture Model) est une méthode statistique qui a été utilisée dans des domaines variés que celui de la voix [Gis99; Ros95], la compression d'images [Aiy01], la classification des sons respiratoires en vue de détecter automatiquement des sibilants ou les crises d'asthme [Pell06] ou celui des finances et de l'économie pour la prévision de la bourse et du taux de change [Lis04]. Récemment les GMM ont été utilisés dans la reconnaissance automatique des émotions à partir de la parole dans [Fuj04; Hung, Qué04; Nei06].

L'utilisation des GMM dans le domaine du traitement du signal de la parole en général et celui de la Reconnaissance de mots isolés IWR en particulier est motivée par la notion intuitive que chaque densité de composante d'un mélange de gaussiennes permet de modéliser une ou un certain nombre de classes acoustiques telles les voyelles ou les fricatives par exemple. Ces classes acoustiques reflètent un aspect général de la configuration du système de la production de la parole (poumons, conduit vocal et cordes vocales) sous l'effet de l'état émotionnel éprouvé. Etant donné que les données d'apprentissage et de test ne sont pas « phonétiquement » annotées, les classes acoustiques sont considérées comme cachées dans le sens où la classe des données observées est inconnue. Par conséquent, la densité des vecteurs de traits générés de ces classes acoustiques cachées prête bien à un mélange de gaussiennes.

III.3.2 Propriétés et définition :

Les données dont la fonction de densité de probabilité est unimodale et symétrique peuvent être convenablement modélisées par une seule courbe gaussienne. Cependant, dans plusieurs cas de problèmes réels, les données ne peuvent être modélisées adéquatement par un seul paramètre de variance et de moyenne d'où l'intérêt de l'utilisation d'un modèle à mélange de gaussiennes. Les GMM permettent de réaliser une approximation d'une fonction de densité de probabilité, de complexité quelconque, en choisissant un nombre suffisant de composantes gaussiennes avec un choix éclairé des valeurs de ses paramètres.

La Figure III.18(b) montre un exemple de fonction de densité de probabilité de mélange de gaussiennes obtenue par la combinaison de trois gaussiennes pondérées par w1, w2 et w3 de la FigureIII.18(a).

Les GMM peuvent être considérés comme une approche hybride entre les modèles de densités paramétriques et non paramétriques. À l’instar des modèles paramétriques, les GMM possèdent une structure et des paramètres qui contrôlent le comportement de la densité d'une manière connue sauf qu'ils sont libres de toute contrainte relative à une distribution spécifique des données. De la même manière que pour les modèles non paramétriques, les GMM possèdent plusieurs degrés de liberté, ce qui offre la possibilité de modéliser des données ayant une densité quelconque sans demander une capacité exorbitante de calcul ou de stockage [Rey00].

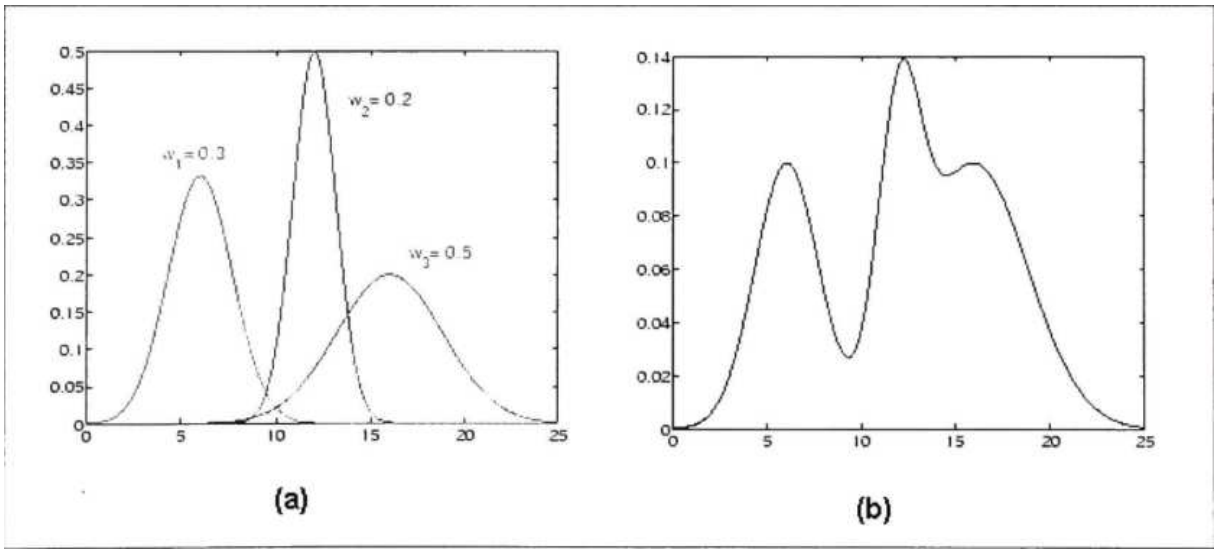

Figure III. 18: Exemple de mélange de trois gaussiennes (b), obtenue par la combinaison de trois gaussiennes pondérées par W1, W2 et W3 (a) tiré de [Res08].

Un GMM peut être également vu comme étant un HMM, à un seul état, ayant un mélange de gaussiennes comme densité d'observation. Une densité de probabilité d'un modèle de mélange de gaussiennes est une somme pondérée de M composantes de densités et s'écrit sous la forme mathématique suivante :

$$P(x|\lambda) = \sum_{i=1}^{M} w_i \, b_i(x) \qquad \text{III. 46}$$

Où x est un vecteur de données de dimension d, λ est le modèle GMM, les w_i représentent les pondérations des mélanges de gaussiennes avec les contraintes $\sum_{i=1}^{M} w_i = 1$et $w_i > 0$ pour $i = 1, \ldots . M$, et $b_i\ (x)$ sont les densités normales multidimensionnelles données par :

$$b_i(x) = \frac{1}{2\pi^{d|2} |\sum_j|^{1|2}} e^{-1|2(x-\mu_i)^T \sum_j{}^{-1}(x-\mu_i)} \qquad \text{III. 47}$$

μ_i et $\sum_j$ représentent respectivement le vecteur de la moyenne et la matrice de covariance de la $i^{ième}$ gaussienne, et l'exposant $[.]^T$ désigne la transposée du vecteur ou de la matrice.

Le modèle GMM λ, est défini par :

$$\lambda = \{w_m, \mu_m, \sum_m\} \qquad \text{III. 48}$$

Où $w_m, \mu_m, \sum_m$ représentent respectivement la pondération, le vecteur de la moyenne et la matrice de covariance de chacune des M composantes gaussiennes constituant le mélange de gaussiennes λ.

Les matrices pleines et les matrices diagonales sont les deux formes de matrice de covariance les plus largement utilisées dans la modélisation avec les GMM. Le modèle GMM avec une matrice de covariance pleine est le modèle le plus puissant, car il permet de mieux ajuster les données. L'inconvénient de ce type de matrice de covariance est qu'il nécessite un grand volume de données pour l'estimation des paramètres et dépend du schéma de régularisation pour obtenir des estimations précises. Le nombre de paramètres à estimer lors de la phase d'apprentissage est égal à $\frac{M}{2}(d^2 + 3d + 3)$ où d représente la dimension du vecteur de traits caractéristiques. D'autre part, la matrice de covariance diagonale est largement utilisée et permet d'obtenir des performances semblables aux matrices de covariance pleines en utilisant un nombre plus élevé de mélanges de gaussiennes [Rey95]. Le nombre de paramètres à estimer pour le cas d'un modèle avec une matrice de covariance diagonale est égal $M(2d + 1)$.

III.3.3 Estimation des paramètres du GMM

Avec les GMM, l'objectif de la phase d'apprentissage est d'estimer les paramètres λ. Pour l'ensemble des données d'entraînement, c.-à-d. trouver les valeurs des paramètres qui modélisent le mieux la distribution des données d'apprentissage. Il existe plusieurs techniques pour la l'estimation des paramètres d'un GMM et la méthode la plus populaire et bien établie est la méthode de l'estimation du maximum de vraisemblance (ML, Maximum Likelihood estimation).

Le but de la méthode ML est de trouver les paramètres du modèle qui maximisent la vraisemblance du GMM étant donné les données d'apprentissage [Mcl88;Rey95]. En supposant l'indépendance des vecteurs d'entraînement

$X = \{x_1, \dots, x_n, \dots, x_N\}$, la vraisemblance du modèle λ, s'écrit comme :

$$P(X|\lambda) = \prod_{n=1}^{N} P(x_n|\ \lambda) \qquad \text{III. 49}$$

Malheureusement, il n'existe pas de méthode analytique connue pour résoudre le problème de maximisation de cette fonction non linéaire du paramètre λ. Cependant, nous pouvons choisir $\lambda = \{w_m, \mu_m, \sum_m\}$ telle que la vraisemblance P(X|λ) est un maximum local en utilisant une méthode itérative telle que la méthode Estimation-Maximisation connue aussi sous le nom Baum-Welch pour les HMM ou en utilisant les techniques du gradient.

III.3.4 L'algorithme Estimation-Maximisation (EM) :

Introduit initialement par Baum [Bau72; Bau66; Dem77], l'algorithme EM est la méthode la plus utilisée pour l'apprentissage statistique faisant intervenir des variables manquantes (missing variables). Il permet de déterminer, suivant un processus itératif, les paramètres du modèle λ en maximisant dans l'espace des paramètres λ la fonction de la vraisemblance $P(X|\lambda)$ de l'ensemble des observations X conditionné sur l'ensemble des paramètres λ.

Pour des raisons analytiques, il est plus facile de travailler avec le logarithme de la vraisemblance qu'avec la vraisemblance elle-même. Étant donné que le logarithme est croissant et monotone, la valeur de λ qui maximise le logarithme de la vraisemblance maximise également la vraisemblance.

$$\log P(X|\lambda) = \sum_{n=1}^{N} \log P(x_n|\lambda) = \sum_{n=1}^{N} \log \sum_{i=1}^{M} w_i\, b_i\,(x) \qquad \text{III. 50}$$

Souvent la valeur moyenne du logarithme de la vraisemblance, obtenu par la division du $\log P(X|\lambda)$ par N est utilisée. Ceci a pour effet de normaliser le logarithme de la vraisemblance par rapport à la durée.

Soit

$X = \{x_1, \dots, x_n, \dots, x_N\}$, l'ensemble des observations,

$Y = \{y_1, \dots, y_n, \dots, y_N\}$, les variables manquantes supposées connues afin de simplifier le problème. Les valeurs y_n peuvent représenter la composante gaussienne qui se réalise pour la donnée observable x_n dans le cas des GMM, ou la séquence d'états cachés associée aux observations x_n dans le cas des modèles de Markov cachés;

$Q(\lambda, \hat{\lambda})$ est une fonction auxiliaire incluant les paramètres $\lambda = \{w_m, \mu_m, \sum_m\}$ du modèle courant et leurs valeurs estimées $\hat{\lambda} = \{\hat{w}_m, \hat{\mu}_m, \hat{\sum}_m\}$ à l'itération t . Elle est définie comme étant l'espérance mathématique du logarithme de la vraisemblance jointe des variables observées et des variables cachées :

$$Q(\lambda, \hat{\lambda}) = \sum_Y P(Y|X, \hat{\lambda}) \log P(X, Y|\lambda) \qquad \text{III. 51}$$

Maximiser la fonction $Q(\lambda, \hat{\lambda})$ est équivalent à maximiser (le logarithme de) la vraisemblance des données observées, étant donné que :

$$Q(\lambda, \hat{\lambda}) \geq Q(\lambda, \lambda) \Rightarrow \log P(X|\hat{\lambda}) \geq \log P(X|\lambda) \qquad \text{III. 52}$$

C'est-à-dire que nous avons trouvé un nouveau modèle $\hat{\lambda}$, plus probable que λ, à partir duquel, il est plus probable que la séquence d'observation soit générée.

En se basant sur cette procédure, si nous procédons au remplacement de λ par $\hat{\lambda}$ d'une manière itérative, et que nous répétions le calcul de ré-estimation, nous pouvons alors améliorer la probabilité que X soit observée à partir du modèle, et ce, jusqu'à ce qu'un point limite soit atteint.

Algorithme EM

1. **Initialisation** : Choisir une estimation initiale λ.
2. **Estimation**: Calculer la fonction auxiliaire $Q(\lambda, \hat{\lambda})$ qui représente une estimation $\log P(X|\lambda)$, en se basant sur les données observables.
3. **Maximisation**: Calculer $\hat{\lambda} = \text{argmax}_{\hat{\lambda}}\, Q(\lambda, \hat{\lambda})$ afin de maximiser la fonction auxiliaire Q sur λ.

4. **Itération** : Mettre $\lambda = \hat{\lambda}$, répéter étape 2 et 3 jusqu'à ce qu'il y ait convergence.

Dans le cas d'un mélange de gaussiennes, l'algorithme EM réalise un apprentissage non supervisé des paramètres de la densité du GMM, c'est-à-dire les moyennes, les matrices de covariances et les coefficients de pondération, à travers les vecteurs de données

$X = \{x_1, \dots, x_n, \dots, x_N\}$. Aucune donnée x_n n'est associée exclusivement à une gaussienne unique, mais plutôt sera considérée comme étant générée par chacune des gaussiennes avec une certaine vraisemblance. Les valeurs des paramètres λ sont données par les formules suivantes :

La probabilité a posteriori :

$$\widehat{P}(j|x_n, \hat{\lambda}) = \frac{P(x_n|j,\hat{\lambda})P(j,\hat{\lambda})}{P(x_n|\hat{\lambda})} \qquad \text{III. 53}$$

La pondération d'une gaussienne :

$$\bar{p}_j = \frac{1}{N}\sum_{n=1}^{N} \widehat{P}(j|x_n, \hat{\lambda}) \qquad \text{III. 54}$$

La moyenne :

$$\hat{\mu}_j = \frac{\sum_{n=1}^{N} x_n P(j|x_n,\hat{\lambda})}{\sum_{n=1}^{N} P(j|x_n,\hat{\lambda})} \qquad \text{III. 55}$$

La covariance :

$$\widehat{\sum}_j = \frac{\sum_{n=1}^{N} x_n P\left(\frac{j}{x_n},\hat{\lambda}\right)(x_n-\mu_j)(x_n-\mu_j)^T}{\sum_{n=1}^{N} P\left(\frac{j}{x_n},\hat{\lambda}\right)} \qquad \text{III. 56}$$

Avant l'utilisation de l'algorithme EM, il est nécessaire de déterminer deux facteurs importants pour l'apprentissage des modèles GMM :

- L'ordre M des mélanges de gaussiennes
- L'initialisation des paramètres du modèle.

Il n'existe pas de bons moyens théoriques qui peuvent guider la sélection de ces deux paramètres. Le meilleur choix demeure donc la solution empirique. Pour l'initialisation des paramètres du modèle de départ, diverses méthodes sont possibles. Indépendamment de la méthode utilisée, EM garantit de trouver le maximum local de la vraisemblance du modèle. Cependant, l'équation de la vraisemblance du GMM possède plusieurs maximums locaux et différents modèles de départ mènent vers différents maximums locaux [McL88].

Parmi les méthodes d'initialisation, nous retrouvons la méthode LBG qui réalise des regroupements d'un ordre égal à une puissance de deux, ou encore la méthode k-moyennes (k-means) qui permet d'obtenir des regroupements d'un ordre quelconque. Dans les travaux de Reynolds [Rey95] sur l'identification du locuteur en utilisant des modèles à mélange de gaussiennes, trois méthodes différentes pour l'initialisation du modèle ont été expérimentées.

III.4 Quantification vectorielle :

III.4.1. Introduction :

La Quantification Vectorielle (VQ) qui est l'une des nombreuses techniques utilisées en reconnaissance automatique de la parole, a été introduite pour créer des références statistiquement plus représentatives et en même temps économiques au stockage. L'idée essentielle de cette technique résulte du fait que dans l'espace de représentation de la parole, les vecteurs n'occupent que des sous-espaces sous forme de nuages. Ces derniers peuvent être représentés par leurs représentants (prototypes) sans trop de perte d'information.
La quantification vectorielle consiste à extraire un « dictionnaire » de vecteurs représentatifs (ensembles des centroïdes) d'un ensemble de vecteurs caractéristiques. Le dictionnaire doit respecter le mieux possible leur répartition dans l'espace. Une telle représentation permet d'exploiter la corrélation existante entre les composantes d'un vecteur et ainsi, de diminuer sa dimension.

III.4.2. L'algorithme Des K-Moyennes

L'algorithme des k-moyennes ou k-means (encore appelée méthode des centres mobiles) [Mac67] est une méthode de classification non-supervisée.
C'est un algorithme classique de quantification vectorielle permettant d'identifier les clusters d'individus similaires en se basant sur une mesure de similarité pour grouper les données. Un cluster dans l'algorithme des k-means est un sous-ensemble de l'espace des données identifié par son centre de gravité.

Son principe est le suivant : on dispose de points de l'espace des observations que l'on souhaite rassembler en classes, sans que l'on dispose de connaissance a priori de propriétés particulières sur ces classes (on ne connaît pas les classes a priori : elles sont à découvrir automatiquement), seul leur nombre k est fixé a priori.
Dans le cadre du clustering, on cherche généralement à partitionner un espace en classes concentrées et isolées les unes des autres. Dans cette optique, elle construit k partitions

et les corrige jusqu'à l'obtention d'une similarité satisfaisante. L'algorithme des k-moyennes vise à maximiser la similarité intra-classes et minimiser la similarité inter-classes.

III.4.2.1 Présentation De L'algorithme Des K-Moyennes :

1. Choix d'une métrique pour le calcul des distances (euclidienne, hamming…).
2. Définition d'un nombre k de classes sur un ensemble des éléments.
3. Initialisation aléatoire des $\mu_1, \mu_2, , , , , \mu_k$ le centre de gravité (centroïde) de chacune des k classes.
4. Affectation de chaque élément à la classe (cluster) le plus proche : dont le centre lui est le plus proche suivant la métrique choisie. (en utilisant par exemple une distance euclidienne)

$$C(x_i) = min_g d(x_i \mu_g).$$

5. Recalcule le centre μ_i de chaque cluster

$$\mu_g = \frac{1}{N} \sum_{i \in C_g} x_i$$

6. Répétition des étapes 4 et 5 jusqu'à convergence.

III.4.2.2 Choix Des Centres Initiaux

L'algorithme des k-moyennes est influencé par ses conditions initiales, il existe plusieurs méthodes d'initialisation, nous pouvons citer :

- L'initialisation aléatoire : Le dictionnaire le plus simple est celui qui contient les L premiers vecteurs de la suite d'apprentissage, où ces *L* vecteurs sont extraits aléatoirement de cette suite. Ces vecteurs peuvent bien sûr ne pas être du tout des représentants de la suite d'apprentissage, et dans ce cas, on aboutit à des résultats très médiocres.
- L'algorithme à seuil : Au lieu de prendre L vecteurs aléatoirement, on fixe une distance minimale entre les éléments du dictionnaire initial. Cette méthode permet d'obtenir une meilleure représentativité que dans le cas précédent.

III.4.3. L'algorithme de LINDE-BUZO-GRAY

L'algorithme de k-means présente un problème de choix d'initialisation, Linde-Buzo-Gray propose l'algorithme de « split » qui permet de résoudre le problème de choix de nombre de vecteurs-références et de l'initialisation de vecteur.
Cet algorithme de quantification vectorielle de type LBG est une variante des k-moyennes, son idée principale est de construire un Codebook pour toutes les périodes. Premièrement, le Codebook ne contient qu'un noyau. Pendant chaque itération, chaque noyau sera séparé en deux (séparation binaire) pour augmenter la taille du Codebook

III.4.3.1 Presentation de d'algorithme de Linde-Buzo-Gray :

1. **Initialisation** : choisir le centroïde de l'ensemble d'apprentissage, noté Y_1^0:

$$Y_0 = \{Y_1^0\} \quad et \quad n = 0$$

2. **Split ou éclatement** : on double la taille du codebook par éclatement de chaque centroïdes en suivant la règle :

$$\boldsymbol{y_0^+ = y_n + \varepsilon}$$

$$\boldsymbol{y_0^- = y_n - \varepsilon}$$

où $\boldsymbol{\varepsilon}$ est un vecteur de norme faible, n varie de 1 à la taille du codebook. On passe donc de 2^N élément à 2^{N-1}.

3. **Convergence** : pour avoir le meilleur ensemble de centroïdes pour le nouveau codebook, on applique l'algorithme de *k*-means.
4. **Arrêt** : on incrémente n, et on répète à l'étape 2 et 3 jusqu'à ce que le codebook de taille *M* fixée à l'avance soit calculé.

Chapitre VI

TEST DE PERFORMANCES DES DIFFERENTES APPLICATIONS

TEST DE PERFORMANCES DES DIFFERENTES APPLICATIONS

Les systèmes de reconnaissance réalisés ont été testés, afin d'avoir une idée sur leur taux de reconnaissance. Le test est limité à 50 échantillons, composés d'énonciations aléatoires des cinq commandes (10 répétitions pour chacune. Les résultats obtenus sont indiqués dans des tableaux et cela pour les test en deux langues, l'anglais {Backward, Forward, Stop, Left, Right} et le français {Recule, Avance, Arrête, Gauche, Droite}, les tests ont été accomplis pour les quatre méthodes par deux types de microphone ; le microphone du PC (enregistrements hautement bruités) et microphone d'un Smart Phone à travers une connexion Bluetooth en utilisant l'application WO MIC installée dans un pc (comme server) et le téléphone (hôte) ,ce dernier a fourni une qualité très nette des enregistrements ce qui a amélioré certainement le taux de reconnaissance et apporter plus de mobilité pour les locuteurs.

Avant de présenter les tableaux des résultats nous présentons les différentes interfaces avec descriptions des différentes figures et boutons.

Il est à noter que des vidéos ont été enregistrées pour les tests des différentes méthodes et sont jointes à ce document.

VI.1 Interfaces graphiques :

VI.1.1. Page principale

Dans ce projet il a été mis un seul modèle d'interface graphique pour les quatre méthodes traitées comportant quatre boutons dans la page principale:

- Un bouton pour l'apprentissage , nommé « system training » en cliquant au-dessus une fenêtre s'ouvre pour vous demander le fichier de l'ensemble des audio d'enregistrements que vous vouliez que le système s'entraine sur eux.
- Un bouton pour la création d'un nouveau locuteur, cela lors de l'exécution, doit être accompagné par le remplissage du nom du nouveau locuteur sur une zone de texte située juste au-dessus de ce bouton, en faisant ça l'application crée toute l'arborescence des fichiers nécessaires et laissées vide pour d'éventuels nouveaux enregistrements.
- Un bouton pour les nouveaux enregistrements, nommé « new recording » en cliquant au-dessus la page de la figure VI.2 s'ouvre.

- Un bouton pour donner le feu vert pour parler et donner l'ordre parmi une liste ,tout de dépend de la langue choisie (anglais ou français) et cela en regardant le fichier avec lequel le système a été entrainé, par exemple pour « ….\Data_wo_fr », il s'agit du français et la technique d'enregistrement est faite à travers le microphone de l'application WO du smartphone, en cliquant au-dessus de ce bouton une zone se colore en rouge en alertant que les deux secondes d'enregistrement viennent de commencer et quand la couleur change en vert c'est la fin d'enregistrement, une fois ratée ou mal faite il est possible de cliquer de nouveau et de recommencer de nouveau, le signal de parole s'affiche graphiquement et se joue en audio pour donner assurer le locuteur sur la qualité de l'enregistrement qu'il vient de l'achever.
- Un bouton pour l'exécution de la reconnaissance et le résultat s'affiche sur une zone texte nommée « l'ordre » suivit par une simulation du mouvement du robot mobile comme le montre la figure La figure VI.2.

  La figure VI.1 montre la manière dont est conçue la première page.

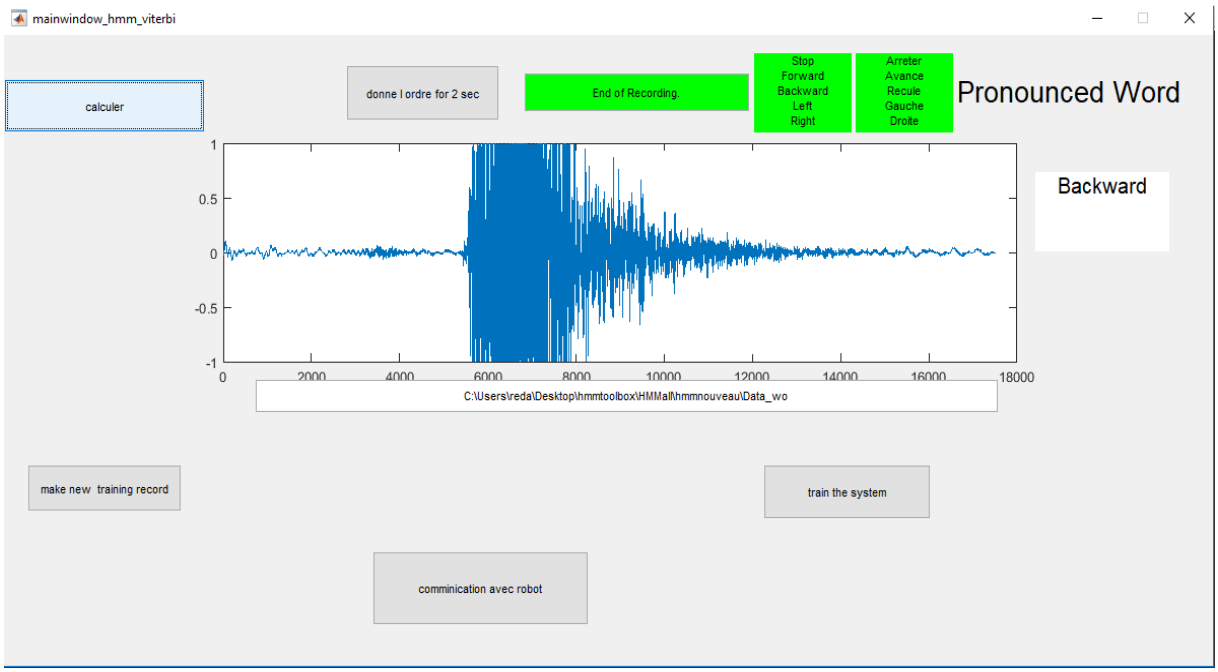

Figure VI. 1 Interface graphique de la page principale du SRAP

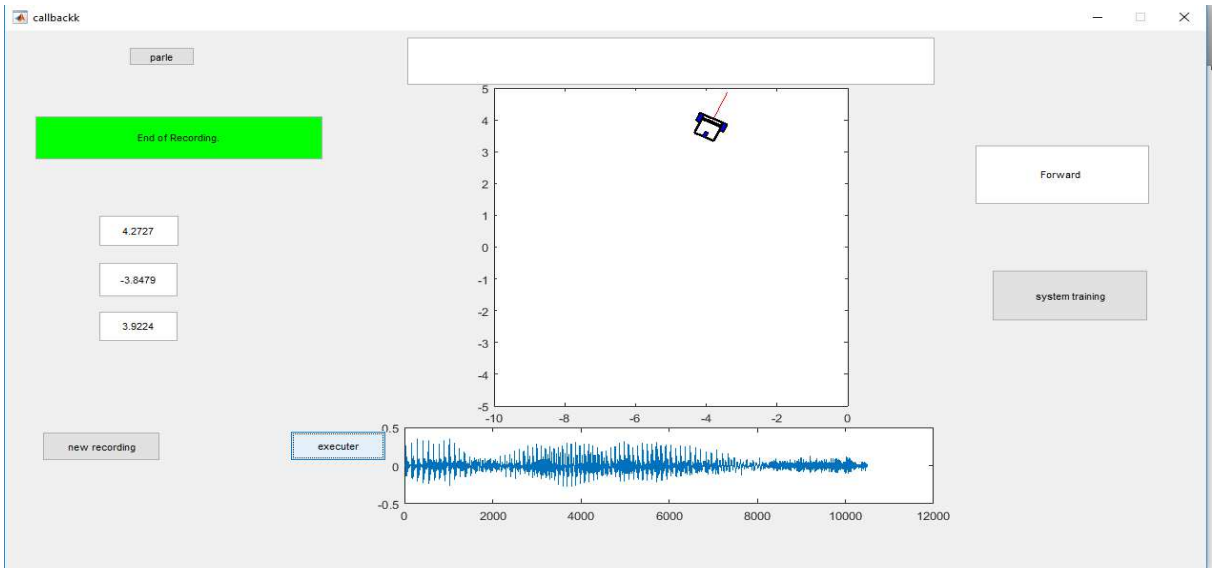

Figure VI. 2 La simulation de la plateforme mobile entrain de répondre l’ordre du locuteur

VI.1.2. **Page d’un nouvel enregistrement** :

Comme le montre la figure VI.2, le bouton « parle » vous invite à commencer de de parler et une zone de texte se colore en rouge pour une durée de deux secondes ensuite ça revient à sa couleur originale en vert indiquant la fin de l’enregistrement, trois boutons pour trois possibilités chacun ;

- Le bouton « play » permet de rejouer le fichier audio.
- Le bouton « delete » pour le supprimer.
- Le bouton « save » pour le sauvegarder.

L’enregistrement doit être accompagné par le choix du mot à enregistrer parmi une liste déroulante et le choix du locuteur dont l’enregistrement fait partie.

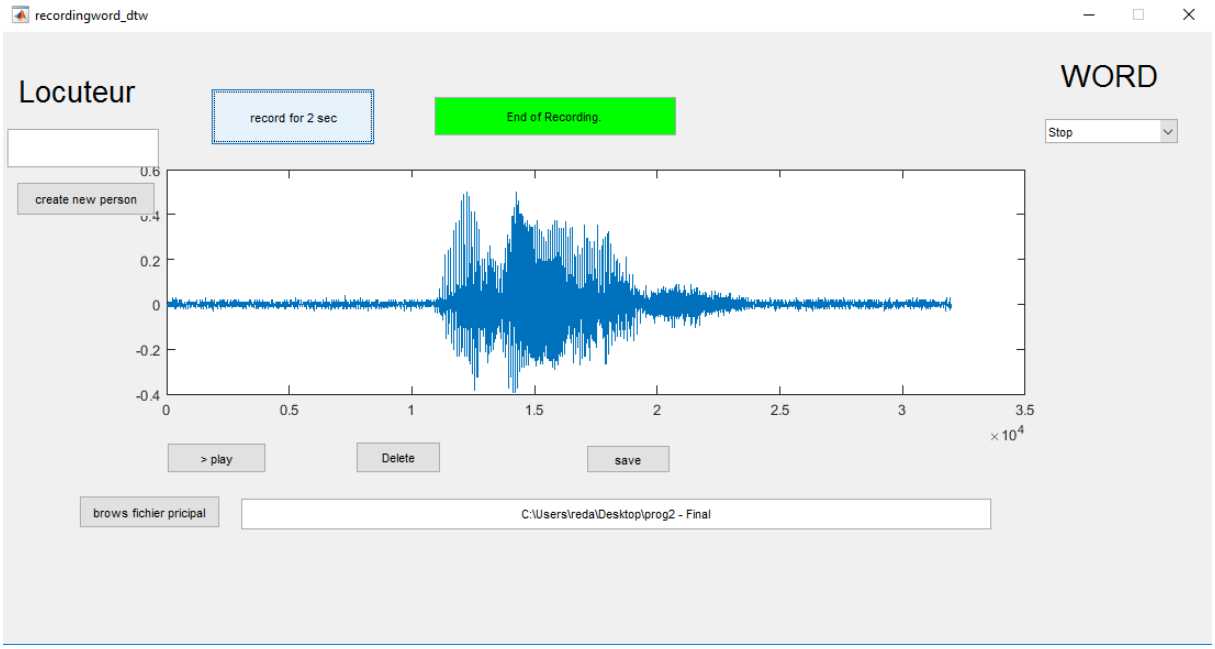

Figure VI. 3: Interface graphique d’un nouvel enregistrement

VI.2 Présentation des résultats :

VI.2.1 Exemple d’enregistrements d’un locuteur :

Les figures VI.3 et VI.4 montrent les signaux des cinq commandes dictées en français respectivement en anglais, après passage par le VAD , pour éliminer les moments de silence là où il n’y a pas d’activité audio . Un premier constat c’est les ressemblances entre les signaux des mots « Left et Right », « Forward et Backward », « Arrêter et Recule » et « Droite et Avance ». La chose qui peut être néfaste sur la qualité de reconnaissance de ces mots surtout dans un milieu bruité et d’un locuteur à un autre.

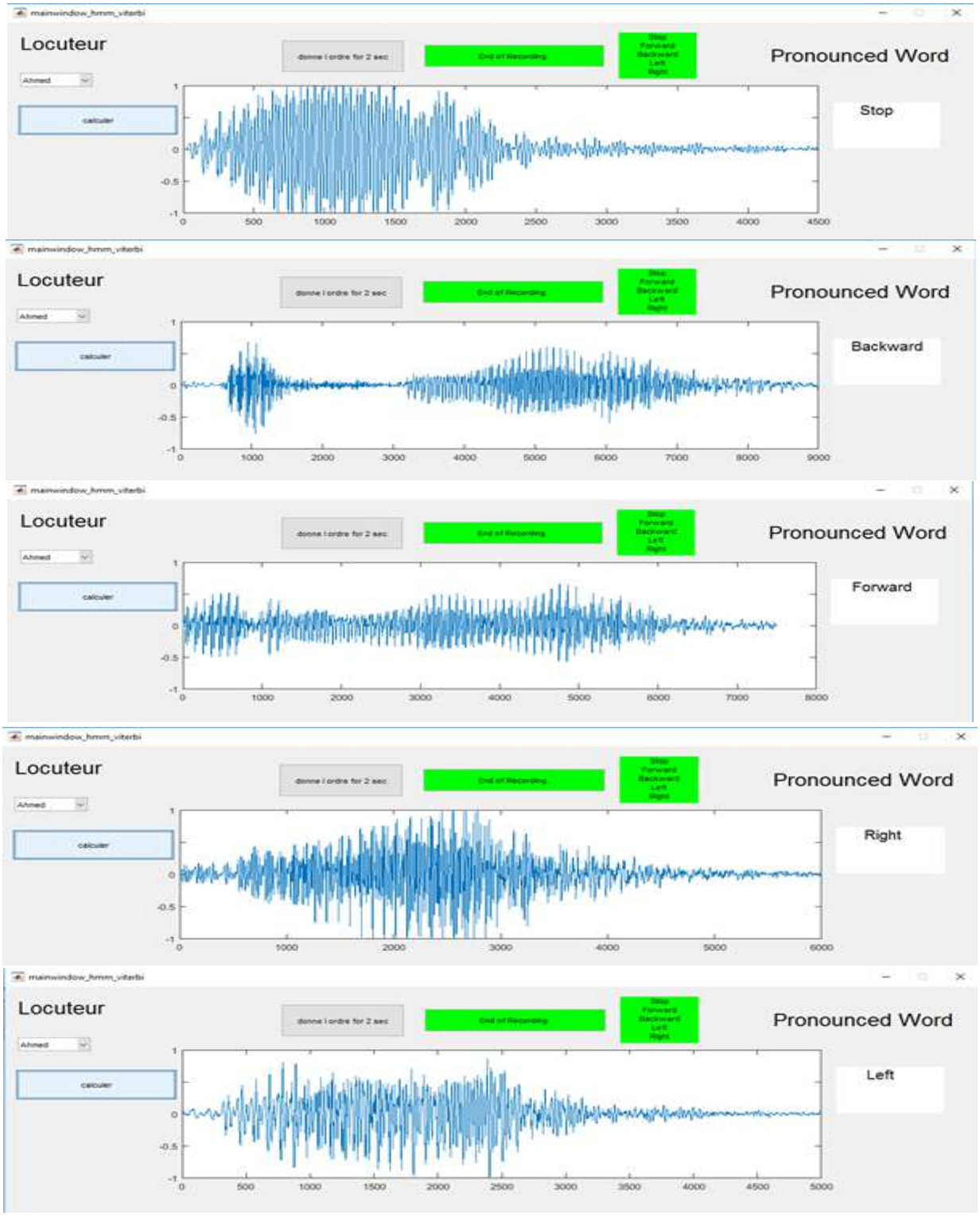

Figure VI. 4 Signaux des cinq commandes prononcés en Anglais

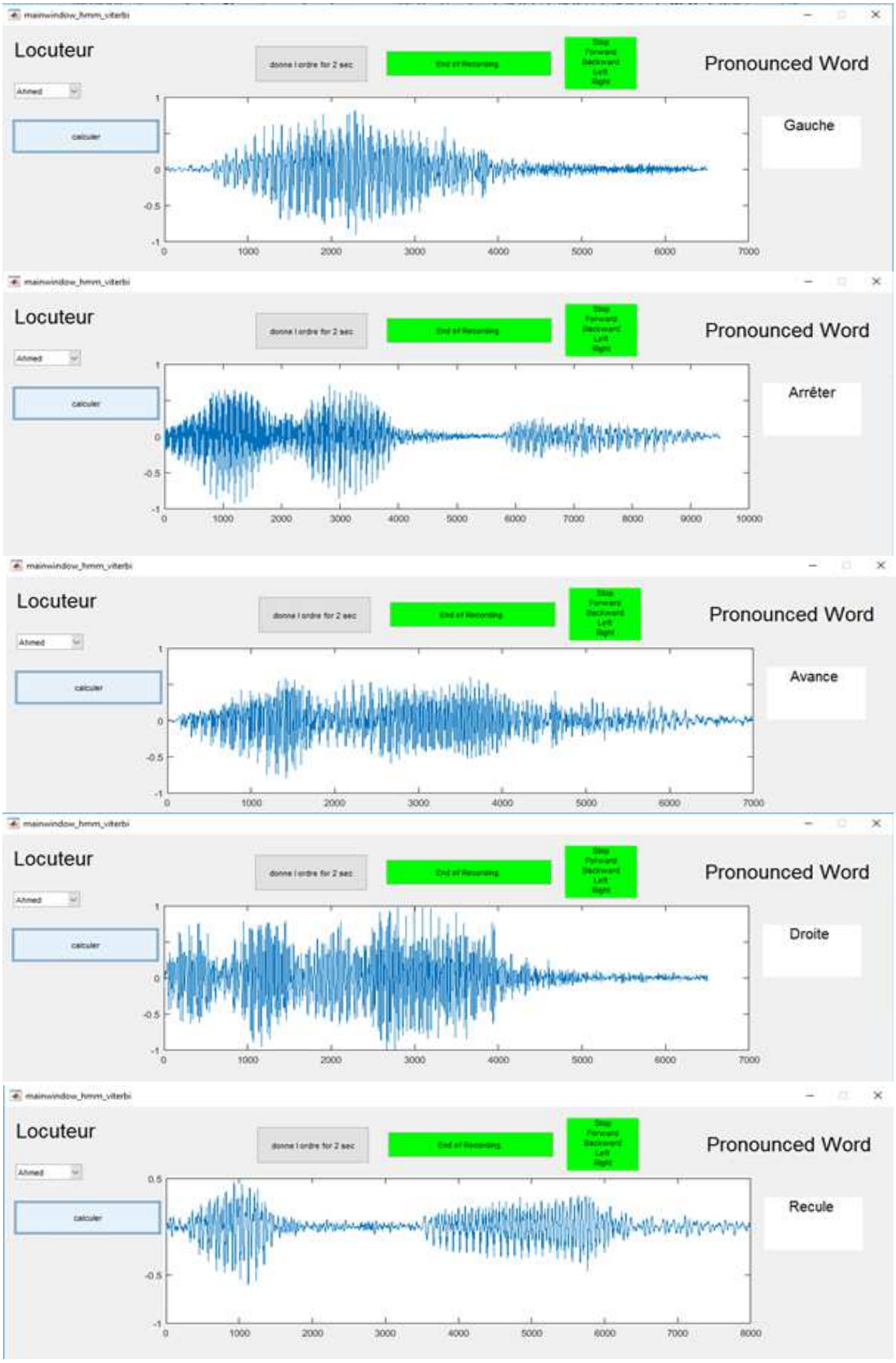

Figure VI. 5 Signaux des cinq commandes prononcés en Français

VI.2.2 Résultats des performances enregistrés avec microphone du pc (avec bruit) :

Commande	*Taux de reconnaissance*
Forward	10|10
Backward	10|10
Stop	2|10
Right	10|10
Left	10|10

Tableau VI. 1 Performance DTW (anglais).

Commande	*Taux de reconnaissance*
avance	10|10
recule	9|10
arrêter	10|10
droite	10|10
gauche	10|10

Tableau VI. 2 Performances DTW (Français)

Commande	*Taux de reconnaissance*
Forward	40|10
Backward	40|10
Stop	90|10
Right	50|10
Left	40|10

Tableau VI. 3 Performance VQ (anglais).

Commande	*Taux de reconnaissance*
avance	10|10
recule	10|10
arrêter	8|10
droite	5|10
gauche	4|10

Tableau VI. 4 Performances VQ (Français)

Commande	*Taux de reconnaissance*
Forward	10|10
Backward	09|10
Stop	10|10
Right	07|10
Left	10|10

Tableau VI. 5 Performance HMM (anglais).

Commande	*Taux de reconnaissance*
avance	07|10
recule	09|10
arrêter	10|10
droite	09|10
gauche	10|10

Tableau VI. 6 Performances HMM (Français)

Commande	*Taux de reconnaissance*
Forward	04|10
Backward	10|10
Stop	10|10
Right	10|10
Left	10|10

Tableau VI. 7 Performance HMM GMM (anglais).
M=1 Q=6

Commande	*Taux de reconnaissance*
avance	10|10
recule	10|10
arrêter	10|10
droite	09|10
gauche	10|10

Tableau VI. 8 Performances HMM GMM (français)
M=4 Q=6

* M : le nombre du Mélange des gaussiennes, Q : le nombre d'état des valeurs choisies conformément aux convergences des calculs.

VI.2.3 Résultats des performances enregistrés avec microphone WO MIC (qualité nette) :

Commande	*Taux de reconnaissance*
Forward	10|10
Backward	10|10
Stop	10|10
Right	10|10
Left	10|10

Tableau VI. 9 Performances DTW (anglais).

Commande	*Taux de reconnaissance*
avance	10|10
recule	10|10
arrêter	10|10
droite	10|10
gauche	10|10

Tableau VI. 10 Performances DTW (Français)

Commande	*Taux de reconnaissance*
Forward	08|10
Backward	09|10
Stop	10|10
Right	06|10
Left	09|10

Tableau VI. 11Performances VQ (anglais).

Commande	*Taux de reconnaissance*
avance	10|10
recule	10|10
arrêter	10|10
droite	10|10
gauche	08|10

Tableau VI. 12 Performances VQ (Français)

Commande	*Taux de reconnaissance*
Forward	10|10
Backward	10|10
Stop	10|10
Right	10|10
Left	10|10

Tableau VI. 13 Performances HMM (anglais).

Commande	*Taux de reconnaissance*
avance	10|10
recule	10|10
arrêter	10|10
droite	10|10
gauche	10|10

Tableau VI. 14 Performances HMM (Français)

Commande	*Taux de reconnaissance*
Forward	07|10
Backward	08|10
Stop	10|10
Right	10|10
Left	10|10

Tableau VI. 15 Performances HMM GMM (anglais).
M=1 Q=6

Commande	*Taux de reconnaissance*
avance	10|10
recule	10|10
arrêter	10|10
droite	10|10
gauche	10|10

Tableau VI. 16 Performances HMM GMM (Français)
M=8 Q=6

VI.3 Commentaires et comparaison des résultats :

La plupart du temps, l'inexactitude de la reconnaissance est due à des impulsions soudaines du bruit, ou à un changement soudain et radical dans le ton de la voix ou surtout à la ressemblance, citée auparavant dans la section précédente, par exemple « Forward et Backward ». Les remarques suivantes ont été enregistrées :

- Le choix de la langue est décisive, les résultats en français sont nettement mieux que ceux en anglais sauf si cela ne s’oppose pas à la confusion due à l’une des ressemblances citées préalablement, néanmoins cette confusion est moins rude en français qu’en anglais, cela s’explique par le fait que le locuteur algérien est francophone, la plus part prononce les commandes par la façon correcte qui ne laisse pas confusion, à l’inverse du l’anglais où on est faible dans la vraie prononciation donc , chacun sa façon de la prononcer pour les anglo-saxonnes les résultats seront pour l’anglais.
- La nature du microphone est un autre paramètre décisif, mieux est le microphone meilleurs sont les résultats, la qualité nette du microphone du smartphone s’est imposée clairement et pour toutes les méthodes jusqu’au point qu’elle a envahie la différence entre elles.
- Les deux méthodes HMM et GMM sont nettement mieux que le DTW et VQ en dernier emplacement, néanmoins elles sont gourmandes en terme de calcul, la chose qui n’apparait pas clairement dans ce projet vue que le vocabulaire est petit et concerne la reconnaissance de mots isolés IWR.
- Le vocabulaire est petit (5cinq commandes) donc n’est pas très suffisant de juger la différence entre les quatre méthodes, de telle façon que tous les résultats sont proches entre eux.
- Il a été constaté que plus base de donnés est riche mieux sont les résultats de la reconnaissance, pour cette raison et pour plus d’efficacité on suggère pour le locuteur de test de faire des enregistrements pour faire entrainer le système sur sa voix.

CONCLUSION GENERALE

CONCLUSION GENERALE

Nous avons testé quatre différentes méthodes de la reconnaissance automatique de la parole. Au vu des résultats des applications, on s'aperçoit que toutes les méthodes fonctionnent relativement très bien. Mais le problème majeur étant les erreurs de reconnaissance concernant les mots très proches ou mal prononcés, pour le français ce dernier problème ne se pose pas donc les résultats sont presque parfaits et s'approchent du 100|100. De plus le système réalisé ne permet pas de reconnaître des phrases mais plutôt des mots isolés et vue que le dictionnaire est limité à cinq mots les résultats était parfaits, sans oublier l'effet de la qualité des enregistrements qui a joué un rôle très remarquable dans les performances. Néanmoins ces méthode peut être parfaitement utilisées dans des appareils d'utilisation courante comme les téléphone portables (utilisation type appel à numérotation automatique), les consoles automobiles ou pourquoi pas dans le domaine de la domotique et même pour la commande vocale portant assistance aux handicapés. D'après notre programme la taille des dictionnaires est très faible, la reconnaissance est très rapide, inférieure à une seconde même.

Perspectives :

A la fin de ce modeste travail et comme perspectives :

- L'augmentation de la banque des ordres pour répondre à d'autres situations.
- Elargir la base des données pour un meilleur apprentissage.
- Pousser la reconnaissance à la personnalisation, de sorte à ce que le robot ne répond qu'aux ordres du maitre dont il a fait la reconnaissance ce qui rentre dans le cadre de la reconnaissance du locuteur.
- L'utilisation des modèles hybrides avec les modèles HMM tel que les réseaux de neurones ou les algorithmes génétiques.
- L'utilisation des Champs de Markov Aléatoire (CMA) comme outil probabiliste décrivant l'aspect discriminant pour une tâche de classification.
- L'augmentation du taux de reconnaissance par l'utilisation des algorithmes d'élimination du bruit plus puissants.
- L'utilisation d'informations supplémentaires comme des informations visuelles sur la géométrie des lèvres pour aider et corriger la reconnaissance.

- Réaliser les différentes applications sous format APK exécutable sur système d’exploitation Android.
- Ajouter des modules d’identification du locuteur.
- On propose de faire suivre ce travail par implémentation de ces applications sur cartes FPGA et ARDUINO comme le montre la figure VI.4

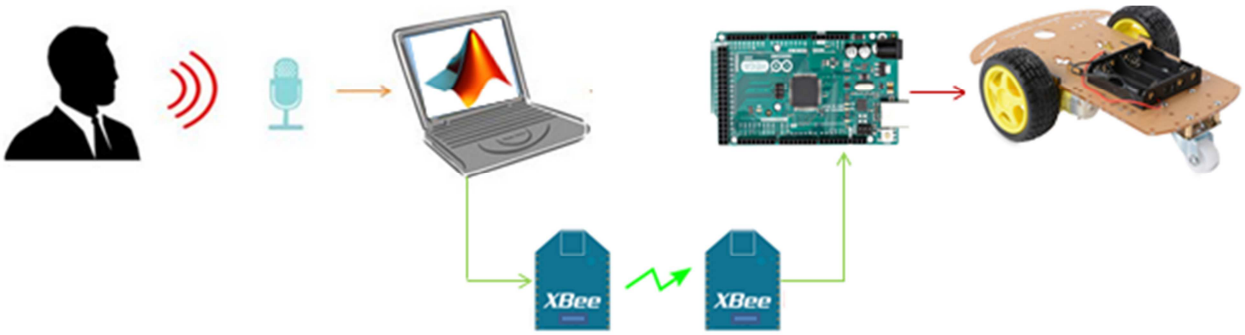

Figure 1 Commande de la plateforme à partir d’une station de base.

1. Principe de la décision bayesienne :

Le théorème de Bayes est un résultat de base en théorie des probabilités, issu des travaux de Thomas Bayes (1702-1761) et retrouvé ensuite par Laplace. Dans son unique article, Bayes cherchait à déterminer ce qu'on appellerait actuellement la distribution a posteriori de la probabilité P d'une loi binomiale.

La théorie de la décision bayesienne constitue une approche fondamentale de la reconnaissance de formes, leur but est d'introduire de l'information statistique dans un problème sous-contraint. Elle suppose que le problème puisse être entièrement spécifié en termes de probabilités et sous ces hypothèses, la décision bayesienne peut être considérée comme optimale.

2. Règle de bayes :

On considère un ensemble, de c classes, noté $\{w_1, w_2, \dots w_c\}$ et un échantillon représenté par un vecteur de caractéristiques x, il s'agit de déterminer la classe w_i qui maximise la probabilité a posteriori $P(w_i|x)$

Pour chaque classe w_i on suppose connaître :

- $P(w_i)$: la probabilité a priori de cette classe.
- $p(x|w_i)$: la densité de probabilité de x conditionnée par cette classe, aussi appelée vraisemblance de w_i par rapport à x.

La règle de Bayes permet de calculer la probabilité a posteriori de chaque classe, c'est-à-dire la probabilité conditionnée par l'observation de x, soit :

$$\boldsymbol{P(w_i|x) = \frac{p(x|w_i)P(w_i)}{\sum_i p(x|w_i)P(w_i)}} \qquad \text{V. 1}$$

3. Estimation au maximum de vraisemblance :

L'estimation au maximum de vraisemblance (ML pour Maximum Likelihood) est une méthode statistique pour déterminer un paramètre inconnu, en maximisant une probabilité.

Pour estimer les paramètres θ d'un modèle en fonction des données X que ce modèle est censé représenter, l'estimateur de maximum de vraisemblance est celui qui maximise la probabilité des données dans le modèle.

$$\boldsymbol{\theta_{ML} = argmax_\theta p(X|\theta)} \qquad \text{V. 2}$$

avec

$$\boldsymbol{p(X|\theta) = \prod_i p(x_i|\theta)} \qquad \text{V. 3}$$

4. Loi normale :

En probabilité, une variable aléatoire x suit une loi normale (ou loi gaussienne) $N(\mu, \sigma^2)$ d'espérance μ et d'écart type σ si sa densité de probabilité est :

$$\boldsymbol{p(x) = \frac{1}{\sigma\sqrt{2\pi}} exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right]} \qquad \text{V. 4}$$

Une telle variable aléatoire est dite variable gaussienne.

5. Loi normale multidimensionnelle :

On appelle loi normale multidimensionnelle ou loi multinormale une loi de probabilité qui est la généralisation multidimensionnelle de la loi normale.

Contrairement à la loi normale classique, paramétrée par un scalaire μ correspondant à sa moyenne et un second scalaire σ^2 correspondant à sa variance, elle est paramétrée par un vecteur μ de R^Dreprésentant son centre et une matrice Σ de R^DxR^D représentant sa matrice de variance-covariance.

Chaque élément de μ_i de μ représente l'espérance de la variable aléatoire x_i et chaque élément Σ_{ij} de Σ représente la covariance des variables aléatoires x_i, x_j et en particulier, chaque élément diagonal Σ_{ij} de Σ représente la variance σ_i^2 de la variable aléatoire x_i.

Comme toute matrice de variance-covariance, la matrice Σ est symétrique réelle, à valeurs propres positives ou nulles; lorsque la loi multinormale est non dégénérée (c.-à-d. qu'il n'existe aucune relation affine presque sûre entre les composantes du vecteur aléatoire), la matrice Σ est à valeurs propres strictement positives : elle est définie positive. Dans ce cas, la loi multinormale admet une densité sur R^D .

Un vecteur aléatoire X de R^D a une distribution normale multidimensionnelle de moyenne μ et de matrice de variances-covariances Σ si sa fonction de densité est définie de R^D dans R de la manière suivante :

$$p(x) = \frac{1}{(2\pi)^{D/2}\sqrt{|\Sigma|}} exp\left[-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right] \qquad \text{V. 5}$$

Qu'on note $N(\mu, \Sigma)$ par analogie avec la notation $N(\mu, \sigma^2)$ de la loi normale univariée.

6. Mélange de lois :

Depuis l'introduction des modèles de mélanges de gaussiennes GMM (Gaussian Mixture Model) par Douglas Reynolds en 1992 [Rey92], ils ont été largement utilisés dans le domaine de la reconnaissance des formes car ils correspondent à une situation où les données appartiennent à un ensemble de classes distinctes, avec une probabilité d'appartenance propre

à chaque classe. Le cas particulier considéré ici est celui où dans chaque classe les données suivent une loi gaussienne. Ce choix tient essentiellement du fait que la loi gaussienne appartienne à une famille de distributions dites exponentielles pour lesquelles le problème de l'identification des composantes du mélange se trouve simplifié.

BIBLIOGRAPHIE

[Abd06] S. H. ABBOU,(2006), "Une application de la transformée en ondelettes à la reconnaissance des commandes vocales en milieu bruité et sa mise en œuvre par processeur dédié au traitement du signal "

[Alt68] R. ALTER, (1968), "Utilization of Contextual Constraints in Automatic Speech Recognition", IEEE Transactions on Audio and Electroacoustics 16.1, p. 6–11.

[Ama13] F.AMAN, M. VACHER, S. ROSSATO et F. PORTET, (2013), "Speech Recognition of Aged Voices in the AAL Context : Detection of Distress Sentences", The 7th International Conference on Speech Technology and Human-Computer Dialogue (SpeD), p. 177–184.

[Azi12] L. Azib , (2012), "Application des Modèles de Markov Cachés et les Modèles de Mélanges de Gaussiennes pour la Classification Phonétique ".

[Bak75] J. K. BAKER, (1975), "The DRAGON System - An Overview", IEEE Transactions on Acoustics, Speech and Signal Processing 23.1, p. 24–29.

[Bar87] J.Barrett and T. Moir, (1987), "A Unified Approach to Multivariable, Discrete Time Filtering based on the Wiener Theory", Kyberbetika 23 pp 177-197.

[Bar96] C. Barras, (1996), " Reconnaissance de la parole continue : Adaptation au locuteur et controle temporel dans les modèles de Markov Cachés", these de Doctorat, Universite de Paris VI.

[Bau66] L. E. BAUM, and T. PETRIE, (1966), "Statistical inference for probabilistic functions of finite state Markov chains", Annals of Mathematical Statistics 37, p. 1554–1563.

[Bau67] L. E. Baum, and J. A. Eagon, (1967), "An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology", Bulletin of the American Mathematical Society 73, p. 360–363.

[Bel92] A. Belaid and Y. Belaid, (1992), "Reconnaissance des formes : méthodes et applications", Inter Edition.

[Ben 99] N. Ben Amara, (1999), " Utilisation des modèles de Markov cachés planaires en reconnaissance de l'écriture arabe imprimée", Thèse de doctorat ; Université de Tunis II.

[Ber12] K. Berbeche, (2012), "Modèles de Markov Cachés : Application à la reconnaissance automatique de la parole. "

[Boi87] R. Boite, M. Kunt, (1987), "Traitement de la parole", Presses Polytechnique Romandes, Lausanne.

[Boi00] R. Boite, H. Bourlard, (2000), "Traitement de la parole", Collection Electricité, Presses Polytechniques et Universitaires Romandes.

[Bro96] R. Brown and P. Hwang, (1996), "Introduction to Random Signals and Applied Kalman Filtering", Wiley, USA.

[Cal93] Caliope and J.P. Tibch, (1993), "La parole et son traitement automatique", Collection Technique et scientifique des télécommunications, Paris, Masson.

[Cer06] J.P. C. Ceresara, D. Fohr, Y. Laprie and K. Smaili, (2006), "Reconnaissance automatique de la parole : du signal à son interprétation ", Paris, Dunod.

[Cho87] Chow, Y., M. Dunham, O. Kimball, M. Krasner, G. Kubala, J. Makhoul, P. Price,

S. Roucos et R. Schwartz, (1987). "BYBLOS: The BBN continuous speech recognition system ", Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). T. 12, p. 89–92.

[Dav80] S. B. Davis and P. Mermelstein, (1980), "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences", IEEE Transactions on Acoustics, Speech and Signal Processing, vol. 28, no 4, pp 357-366.

[Dav52] K. H. Davis, , R. Biddulph and S. Balashek, (1952), "Automatic Recognition of Spoken Digits", The Journal of the Acoustical Society of America 24.6, p. 637–642.

[Den59] P. Denes, (1959), "The Design and Operation of the Mechanical Speech Recognizer", Journal of the British Institution of Radio Engineers 19.4, p. 219–234.

[Den60] P. Denes and M. V. Mathews, (1960), "Spoken digit recognition using time-frequency pattern matching", The Journal of the Acoustical Society of America 32.11, p. 1450–1455.

[Del00] John R. Deller, John G. Proakis, John H. L. Hansen, (2000), Discrete-Time Processing of speech Signals.

[Don02] G.Donald, B.Firesmith and S. Henderson, (2002), "The OPEN Process Framework : An Introduction", Pearson Education.

[Dre50] J.Dreyfus-Graf, (1950), "Sonograph and Sound Mechanics", The Journal of the Acoustical Society of America 22.6, p 731–739.

[Dug96] Dugad, R. and Desai, U. (1996) A Tutorial on Hidden Markov Models. Signal Processing and Artificial Neural Networks Laboratory, Department of Electrical Engineering, Indian Institute of Technology, Bombay Powai, Mumbai, 400 076, India.

[Eas92] S. Easwaran and J. N. Gowdy, (1992), "An improved initialization algorithm for use with the K-means algorithm for code book generation", Southeastcon 92, pp 471-474, Birmingham.

[Ezz02] H. Ezzaidi, (2002), "Discrimination Parole/Musique et étude de nouveaux parametres et modeles pour un systeme d'identification du locuteur", PhD thesis, Universite du Quebec.

[Fat60] R. Fatehchand, (1960), "Machine recognition of spoken words", Advances in Computers1, p. 193–229.

[Fer83] M.Ferretti and F. Cinare, (1983), "Synthèse, Reconnaissance de la parole", editestes, Paris.

[For59] J.W. Forgie, (1959), "Results Obtained from a Vowel Recognition Computer Program", The Journal of the Acoustical Society of America 31, p. 1480–1489.

[Fur05] S. Furui, (2005), "50 years of progress in speech and speaker recognition research", ECTI Transactions on Computer and Information Technology 1.2, p. 64–74.

[Fur86] S. Furui, (1986), "Speaker-independent isolated word recognition using dynamic features of speech spectrum", IEEE Transactions on Acoustics, Speech and Signal Processing 34.1, p. 52–59.

[Gal12] G.Galatas, G. Potamianos and F. Makedon, (2012), "Audio-visual speech recognition incorporating facial depth information captured by the Kinect", Proceedings of the 20th European Signal Processing Conference (EUSIPCO), p. 2714–2717.

[Gis94] H. Gish, M. Schmidt, (1994) Text-Independent Speaker Identification, IEEE Signal Processing Magazine

[God09] A.P.Godse and A.O.Mulani, (2009),"Embedded Systems", Technical Publications.

[Hat91] J. P. Haton, J. M. Pierrel, G. Perennou, J. Caelen, and J.L. Gauvain, (1991), "Reconnaissance automatique de la parole", Afcet Dunod, Bordas, Paris.

[Her90] H. Hermansky, (1990), "Perceptual linear predictive (plp) analysis of speech", The Journal of the Acoustical Society of America, pages 1738–1752.

[Her94] H. Hermansky and N. Morgan, (1994), "Rasta processing of speech", *IEEE Trans. On ASSP,2(4):587–589.*

[Hin06] G. E. Hinton, and S. Osindero, (2006), "A fast learning algorithm for deep belief nets", Neural Computation 18.7, p. 1527–1554.

[Jan07] B. Jane and A.David, (2007), "A Brief Historical Perspective of the Wiener-Hopf Techniques", University of Manchester Springer, USA pp 351-356.

[Jel75] F.Jelinek, L. R. Bahl and R. L. Mercer, (1975), "Design of a linguistic statistical decoder for the recognition of continuous speech", IEEE Transactions on Information Theory 21.3, p. 250–256.

[Jel76] F. Jelinek, (1976), "Continuous speech recognition by statistical methods", proc. IEEE, Vol 64, no 4, pp 532-556.

[Jel90] F.Jelinek, B. Merialdo, S. Roukos and M. Straussi, (1990), "Self-organized language modeling for speech recognition", Readings in Speech Recognition. Morgan Kaufmann, p. 450–506.

[Jot14] S. Jothilakshmi, (2014), "Spoken keyword detection using autoassociative neural networks", International Journal of Speech Technology 17.1, p. 83–89.

[Kur06] R. S. Kurcan ,(2006), "Isolated word recognition from in-ear microphone data using Hidden Markov Models (hmm) "

[Lee90a] C.H. Lee, L. R. Rabiner, R. Pieraccini and J. G. Wilpon, (1990a), "Acoustic modeling for large vocabulary speech recognition", Computer Speech and Language 4.2, p. 127–165.

[Lee90b] K.F. Lee, , H.W. Hon and R. Reddy (1990b), "An overview of the SPHINX speech recognition system", IEEE Transactions on Acoustics, Speech and Signal Processing 38.1, p. 35–45.

[Les77] V. R. Lesser and L. D. Erman, (1977), "A Retrospective View of the HEARSAY-II Architecture", Proceedings of the Fifth International Joint Conference on Artificial Intelligence, p. 790–800.

[Lin65] N.Lindgren, (1965), "Machine Recognition of Human Language Part I. Automatic Speech Recognition", IEEE Spectrum 2.3, p. 114–136.

[Lip87] R. P. Lippmann, (1987), "An Introduction to Computing with Neural Nets", IEEE ASSP Magazine 4.2, p. 4–22.

[Low76] B. T. Lowerre, (1976), "The HARPY speech recognition system", Thèse en informatique, Carnegie Mellon University.

[Mac67] J. MacQueen, (1967), "Some methods for classification and analysis of multivariate observations. Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics, 281--297, University of California Press, Berkeley, Calif".

[Mak75] J.M. Makhoul, (1975), "Linear prediction: a tutorial review", Proc. IEEE, Vol. 63,No.4, pp.561-579

[Mcl88] G.J. McLachlan, D. Peel, and P. Prado , (1988) , "Clustering Via Normal Mixture Models",

[Med76] M. Medress, (1976), "Speech Understanding Systems: Report of a Steering Committee", SIGART Newsletter 62, p. 4–8.

[Mik11] T.Mikolov, S. Kombrink, L. Burget, J. Cernocký and S.Khudanpur,(2011), "Extensions of recurrent neural network language model" , Proc of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), p. 5528–5531.

[Mud10] L. Muda, M. Begam and I. Elamvazuthi, (2010), "Voice Recognition Algorithms using Mel Frequency Cepstral Coefficient (MFCC) and Dynamic Time Warping (DTW) Techniques", Journal of Computing, Volume 2, issue 3.

[Ngi11] J.Ngiam, , A. Khosla, M. Kim, J. Nam, H. Lee and A. Y. Ng (2011), "Multimodal deep learning", Proceedings of the 28th International Conference on Machine Learning (ICML), p. 689–696.

[Pat10] I. Patel et Y. S. Rao2 , (2010), "Speech recognition using hmm with MFCC- an analysis using frequency specral decomposion technique "

[Pau89] D. B. Paul, (1989), "The Lincoln robust continuous speech recognizer", Proc of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). T. 1, p. 449–452.

[Pdf01] traitement_parole_p.perrot_part2

[Pic93] J.W. Picone, (1993), "Signal modeling techniques in speech recognition", Proc. IEEE, Vol 81 no 9, pp 1215-1247.

[Pov11] D. Povey and al, (2011), "The Kaldi Speech Recognition Toolkit", IEEE Workshop on Automatic Speech Recognition and Understanding.

[Qi07] Z. Qi and T. Moir, (2007), " An Adaptive Wiener Filtre for an Automotive Application with Non-Stationary Noise", 2nd Interntional Conference on Sensing Technology, USA pp 300-305.

[Rab78] L.R. Rabiner and R.W. Schafer, (1978), "Digital processing of speech signals", Prentice-Hall.

[Rab79] L.Rabiner, S. Levinson, A. Rosenberg and J. G. Wilpon (1979), "Speaker independent recognition of isolated words using clustering techniques", Proc of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 4, p. 574–577.

[Rab83] L. R. Rabiner, S.E. Levinson, and M. M. Sondhi, (1983), "On the Application of Vector Quantization and Hidden Markov Models to Speaker-Independent, Isolated Word Recognition", The Bell System Technical Journal, vol. 62, pp. 1075-1105.

[Rab93] L. Rabiner and B. H. Juang, (1993), "Fundamentals of speech recognition", Prentice Hall, New Jersey.

[Rea12] J.Read, R. Dridan, S. Oepen et J. L. Solberg, (2012), "Sentence Boundary Detection: A Long Solved Problem? ", Proceedings of COLING, p. 985–994.

[Red66] D. R. Reddy, (1966), "Approach to computer speech recognition by direct analysis of the speech wave", The Journal of the Acoustical Society of America 40.5.

[Reg12] C.Regenbogen, D. A. Schneider, R. E. Gur, F. Schneider, U. Habel and T. Kellermann, (2012), "Multimodal human communication - Targeting facial expressions, speech content and prosody", NeuroImage 60.4, p. 2346–2356.

[Ros95] Richard C. Rose ,A. Reynolds, (1995), "Speaker Verification Using Adapted Gaussian Mixture".

[Roy90] G.Roy and B. Eng, (1990), "Low-rate analysisby- synthesis wideband speech coding", Department of Electrical Engineering McGill université de Montreal, Canada.

[Sch07] H. Schwenk, (2007), "Continuous space language models", Computer Speech & Language 21.3, p. 492–518.

[Seg16] M. Seghiri, (2016), "Implémentation d’une commande vocale à distance d’une plateforme mobile ".

[Som14] T. Somaia, K. Waffaa, T. Hesham and M.Eman , (2014), " The effect of using integrated signal processing hearing aids on the speech recognition abilities of hearing impaired Arabic- speaking children, " Egyptian Journal of Ear; Nose, Throat and Allied Sciences.

[Tab94] J. Taboada, S. Feijoo, R. Balsa, C.Hernandez, (1994), "Explicit estimation of speech boundaries", IEEE Proc. Sci. Meas. Technol., vol. 141, pp. 153-159.