REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Mohamed Khider – Biskra جامعـــــة محمد خيضر بسكـــــرة

Faculté des Sciences ET de la Technologie كليـــة العلوم و التكنولوجيـــــــــا

Département: Genie Electrique قســم: الهندسة الكهربائيـــــــــة

Filière: Electronique ف رع : إليكترونيـــــــك

Réf: …….. المرجع……

Mémoire

Présenté en vue de l'obtention du diplôme de magister en Electronique

Option : **Signaux et Communications.**

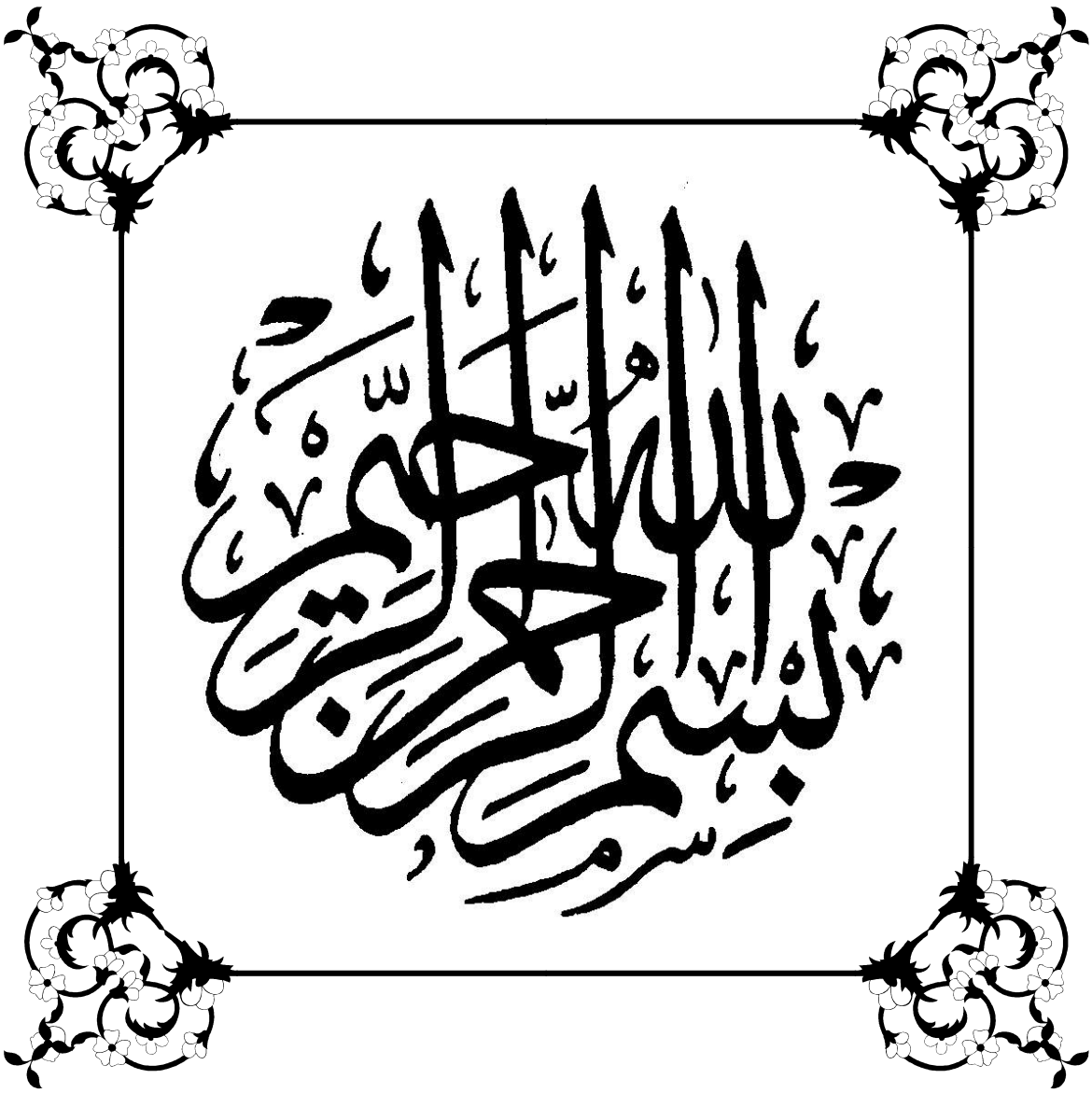
RECONNAISSANCE VOCALE BASÉE SUR LES SVM

Par: BENCHENIEF Abderezek

Soutenu le: 07/12/2011

Devant le jury :

DJEDI Nour Eddine	Professeur	Université de Biskra	Président
BEN OUDJIT Nabil	Professeur	Université de Batna	Examinateur
BAARIR Zine Eddine	Maître de conférences A	Université de Biskra	Examinateur
CHERIF Foudil	Maître de conférences A	Université de Biskra	Rapporteur


بسم الله الرحمن الرحيم

Dédicace

Tout d'abord, je veux rendre grâce à Dieu, le Clément et le Très Miséricordieux pour son amour éternel. C'est ainsi que je dédie ce mémoire à :

ma mère « ouannassa» pour sa tendresse et mon père pour sa patience et encouragement

mes très chers frères et mes chères soeurs pour leurs conseils,(adel , toufic, foudhil, ammar, kamel,abdehamid, messaoud,aissa, nadia, fatima, akila, rehaioua ,fatiha)

mes cousins et cousines,

tous ceux que j'aime,

tous mes amies.

Remerciements

A Dieu, le tout puissant, nous rendons grâce pour nous avoir donné santé, patience, volonté et surtout raison.

En premier lieu, je tiens à remercier mon encadreur Mr. Foudil Cherif M.C qui m'a aidé et conseillé durant ce travail.

Je remercie également tous les enseignants du département de l'électronique de l'université de BISKRA pour leur aide et encouragement.

Enfin, je remercie tous ceux qui m'ont soutenu, encouragé et donné l'envie de mener à terme ce travail.

Résumé

Dans ce travail, nous allons présenter un système de reconnaissance automatique de la parole (RAP) indépendant du locuteur basé sur une combinaison parallèle des classifieurs *Multi-Class Support Vector Machine* (SVM multiclasse). Ce système proposé utilise comme moteur de reconnaissance les deux Stratégie principales, un contre un, et un contre tous pour éviter des ambiguïtés et comme méthode de fusion l'approche par combinaison basée sur *l'intégrale floue de shoquet*.

Pour être combinés des classifieurs dans un système de reconnaissance automatique de la parole, ils doivent être différents. La diversité entre ces classifieurs est crée par changement des données d'apprentissage (*Entraînement discriminant*). Ce pondant, les techniques SVM exigent des vecteurs d'entrée de taille fixe. Pour lever cette difficulté, nous avons proposé un algorithme de normalisation des entrées basé sur les valeurs de la *kurtosis* des trames.

Nous cherchons à fiabiliser la reconnaissance en utilisant la complémentarité qui peut exister entre les classifieurs. Les expériences réalisées pour la reconnaissance des chiffres anglais, indiquent que l'utilisation de la combinaison de classifieurs augmente la performance du système de RAP en milieu réel, meilleur taux de reconnaissance obtenu par le système est de **99.72%.**

Mots clés: *reconnaissance automatique de la parole –la langue anglaise –MFCC –normalisation des entrées – combinaison parallèle de classifieurs –méthode de combinaison –Multi-Class Support Vector Machine (SVM multiclasse).*

Abstract

In this work, we present a system for automatic speech recognition (ASR) independent of the speaker based on a parallel combination of classifiers Multi-Class Support Vector Machine (SVM multiclass). The proposed system uses as recognition engine the two main strategies, one against one, and one against all to avoid ambiguity and as a method of fusion, the approach by combining based on Choquet Fuzzy Integral.

Classifiers to be combined in a system of automatic speech recognition, they must be different. The diversity of these classifiers is created by changing the training data (Discriminate training). The corresponding, SVM techniques require input vectors of fixed size. To overcome this difficulty, we proposed an algorithm for standardization of inputs based on the values of the kurtosis of the frames.

We seek to make reliable recognition using the complementarily that may exist between the classifiers. Experiments for the recognition of English digits indicate that the use of the combination of classifiers increases the performance of ASR system in a real environment; the better recognition rate obtained by the system is **99.72**%.

Keywords: *automatic speech recognition –English language –MFCC –standardization of inputs –parallel combination of classifiers –combination method –Multi-Class Support Vector Machine (SVM multiclass).*

ملخص

في هذا العمل، نقدم نظام التعرف الآلي على الكلام **(ASR: automatic speech recognition)** مستقل عن المتكلم، على أساس مزيج موازية للمصنفات **SVM** (مصنف المتجهات الداعمة) متعدد الطبقات. النظام المقترح يستخدم كمحرك الاعتراف الإستراتيجيتين الرئيسيتين، واحد ضد واحد، واحد ضد كل لتجنب الغموض وكوسيلة من وسائل الاندماج من خلال الجمع بين النهج القائم على التكامل الضبابي لـــ **Choquet**.

المصنفات لتكون مجتمعة في نظام التعرف الآلي على الكلام، يجب أن تكون مختلفة. يتم إنشاء تنوع هذه المصنفات عن طريق تغيير بيانات التدريب (تدريب متميز). في المقابل، تتطلب تقنيات SVM، ناقلات المدخلات من حجم ثابت. للتغلب على هذه الصعوبة، اقترحنا خوارزمية لتوحيد المدخلات التي تستند إلى قيم التفرطح من الإطارات.

نحن نسعى إلى جعل اعتراف موثوق به من خلال استغلال التكامل الذي قد يكون موجود بين المصنفات. تجارب للاعتراف الأرقام الإنجليزية، تشير إلى أن استخدام مزيج من المصنـــفات يزيد من أداء نظام ASR في بيئة حقيقية، وأفضل نـ سـ بة الاعتراف التي حصـــــل عليها هذا النـــظام هـــــو **99.72 %**.

كلمات مفتاحيه: التعرف الآلي على الكلام – اللغة الإنجليزية – MFCC – توحيد المدخلات – مزج موازي للمصنفات – طريقة المزج – SVM (مصنف المتجهات الداعمة) متعدد الطبقات.

Sommaire

Chapitre 5 : Expérience et Résultats

Annexe

Liste des figures

Liste des tableaux

Liste des Symboles et abréviations

T	*période*
A	*Amplitude*
Φ	*phase*
f	*fréquence fandamantale*
w	*pulsation*
f (t)	*la série de Fourier*
$\mathbf{f_0}$	*fréquence d'un signal périodique*
$\mathbf{f_n}$	*les harmoniques*
$\mathbf{C_0}$	*est la composante continue.*
TF	*la transformée de fourrier*
F (f)	*la transformée de fourrier de la fonction f (t)*
$\mathbf{\|f(t)\|^2}$	*la densité temporelle d'énergie*
$\mathbf{\|F(f)\|^2}$	*la densité spectrale d'énergie.*
*	*l'opérateur de la convolution*
τ	*Une variable*
$\mathbf{\theta_{xy}}$	*la fonction d'intercorrélation*
$\mathbf{\theta_{xx}}$	*la fonction d'autocorrélation*
$\mathbf{\theta_x(f)}$	*la densité spectrale d'énergie par l'autocorrection*
W	*l'énergie*
Fe	*fréquence d'échantillonnage*
***Conv* =**	*convolution*
$\mathbf{\delta^2}$	*la variante*
b (t)	*Le bruit blanc*
$\mathbf{B_0}$	*constante*
δ (τ)	*impulsion de Dirac*
$\mathbf{R_b}$	*La fonction d'autocorrélation est de la forme du bruit blanc*
x (n)	*signal vocal échantillonné*
z	*variable de la transformée en Z*
sgn	*la fonction sgn*
VAD	*Voice Activity Detector*
LPC	*Linear Predicting Coding*

u (n)	*signal d'excitation unitaire*
G:	*Un gain*
^:	*symbole de l'estimation*
LPCC	*Linear Prediction Cepstral Coefficients*
PLP	*Perceptually based Linear Prediction analysis*
RASTA	***R**el**A**tive **S**pec**T**r**A**l*
PCA (ACP)	*Analyse en composante principale*
LDA (ADL*)*	*Analyse discriminante linéaire*
DTW	*Comparaison dynamique (dynamic time warping)*
Hmm	*Modèle de Markov Caché*
Gmm	*Modèle de Mélange de lois Gaussiennes*
RN	*Réseau de neurones*
EMP	*Empirical Risk Minimisation*
SRM	*Structural Risk Minimisation*
VC	*la dimension Vapnik-Chervonenkis*
B	*biais*
<, >	*Produit scalaire*
C	*un paramètre du classifieur*
Φ(x)	*fonction noyau*
SVM	*pour Support Vector Machines*
RDF	*reconnaissance de formes*
RAP	*racornissement automatique de la parole*
SVM Multi classe	*problème de classification proposé, pour **K** classes différentes*
$\mathbf{S_1}$	*scores issus de classifieur SVM (stratégie Un contre UN)*
$\mathbf{S_2}$	*scores issus de classifieur SVM (stratégie UN contre Tous)*
$\mathbf{S_1^{tanh}}$	S_1 *normalisés par la méthode tangente hyperbolique*
$\mathbf{S_2^{tanh}}$	S_2 *normalisés par la méthode tangente hyperbolique*
$\mathbf{S^{fusion}}$	*Fusion de scores issus des stratégies SVM Multi classe*
MFCC	*Mel-Frequency Cepstral Coefficients*
μ	*la moyenne arithmétique*
σ	*l'écart type*
MAD	*l'écart absolu médian*
Lim	*la limite*
$\mathbf{g^i}$	*la fonction de densité floue*

$\mathbf{g_\lambda}$	*mesure floue*
$\mathbf{p_1}$ *et* $\mathbf{p_2}$	*taux de classifications*

Introduction

La reconnaissance automatique de la parole par les machines est depuis longtemps un thème de recherche qui fascine le public, mais qui demeure un défi pour les spécialistes. Pour le grand public, un archétype de la communication homme-machine reste probablement le dialogue avec l'ordinateur. Le dialogue en langage naturel avec une machine aussi intelligente que l'homme semble l'aboutissement normal des progrès technologiques. Dans le domaine de la recherche, il était possible d'imaginer à l'époque des progrès rapides des systèmes de reconnaissance automatique de la parole (RAP), en dépit de l'opinion sceptique de quelques spécialistes **[81].** Mais le projet de "compréhension de la parole " lancé en 1971 par le département de la défense américaine (ARPA Speech Understanding Project) **[82]** a contraint les chercheurs à tempérer leur optimisme: malgré des directions prometteuses, le sujet nécessite un effort à long terme.

Aujourd'hui, l'impact des systèmes de RAP est encore minime dans la vie courante, et la commande des ordinateurs ne s'effectue toujours pas par la voix, malgré les promesses de fabricants de logiciel ou de matériel informatique (Microsoft, Apple).L'annonce de la commercialisation du système de dictée vocale d'IBM pour les ordinateurs PC en 1994 a suscité de l'intérêt, mais aussi des réserves quant aux performances actuelles du système. Pourtant les progrès réalisés depuis 25 ans en RAP sont très importants, grâce à un grand nombre de recherches traitant du problème sous tous ses aspects. Les limitations de la capacité des systèmes de reconnaissance, imposées à l'origine par la complexité de la tâche, sont progressivement repoussées, et des systèmes efficaces pour des applications spécialisées sont maintenant disponibles et commercialisés.

Récemment, la combinaison de classifieurs a été proposée comme une voie de recherche permettant de fiabiliser la reconnaissance en utilisant la complémentarité qui peut exister entre les classifieurs. Sur ce point, la littérature abonde de travaux présentant des méthodes de combinaison qui se différencient aussi bien par le type d'informations apportées par chaque classifieur que par leurs capacités d'apprentissage et d'adaptation.

Notre étude s'intègre dans le cadre du développement d'un système de reconnaissance de chiffres parlés anglais, en mode indépendant du locuteur, basé sur une combinaison parallèle de classifieurs SVM multiclasse (Les deux approches classiques, one vs. one et one vs. all ont été mises en oeuvre pour éviter des ambiguïtés). Nous cherchons à fiabiliser la

reconnaissance en utilisant la complémentarité qui peut exister entre les classifieurs. Les méthodes de classification s'orientent vers la combinaison des classifieurs. Cette approche a montré son aptitude de concevoir des systèmes puissants et performants.

Alors que les premières expériences en combinaison parallèle de classifieurs ne datent que des années 80 **[83]**, cette technique est devenue une voie de plus en plus utilisée pour améliorer la qualité des systèmes de reconnaissance dans plusieurs applications: reconnaissance d'images médicales, reconnaissance de chiffres, de caractères et de mots manuscrits, de visages, vérification de signatures, reconnaissance de la parole, identification de formulaires. Ces systèmes diffèrent par le type de sorties des classifieurs combinés, par la nature des classifieurs utilisés ainsi que par les stratégies de combinaison choisies.

Pour être combinés dans un système de RAP, les classifieurs doivent être différents. Cette différence peut être crée en choisissant des classifieurs de divers types, ou par changement des données d'apprentissage dans le cas où les classifieurs sont de même type.

L'objectif de ce mémoire est de proposer un système de reconnaissance de chiffres parlés anglais. Ce système s'appuie sur une combinaison parallèle de classifieurs de même type SVM (Support Vector Machine) multiclasse, et utilise comme méthode de fusion l'approche par combinaison basée sur l'intégrale floue de shoquet pour fusionner les scores issus de chaque classifieur. Donc pour atteindre à ce but on a travaillé de la manière suivante:

Le premier chapitre contient une Généralité sur le Traitement du signal (Parole). On a rappelé quelques notions de bases, les outils importants d'analyse, d'interprétation des signaux, et les opérations qui peuvent effectuer sur un signal. Les processus aléatoires, et on a introduit un exemple de l'identification d'un système linéaire à l'aide d'un bruit blanc. Nous avons pu voir aussi au cours de ce chapitre, le phénomène de la production de la parole, les différentes sources permettant la génération des sons d'une langue donnée.

Au cours du deuxième chapitre on a présenté les méthodes classiques employées en reconnaissance de la parole. Les difficultés rencontrées pour la mise au point des systèmes de RAP proviennent de la variabilité du signal de parole et de la continuité du processus de production. Enfin, les différents types des systèmes de reconnaissance automatique de la parole, en discutant les avantages et inconvénients concernant chaque type.

Le troisième chapitre va mettre l'accent sur la méthode de classification choisie, par l'étude de la méthode de Machines à Vecteurs de Support (SVM). Le quatrième chapitre constitue notre contribution, il s'agit d'une conception convenable d'un système de reconnaissance de la parole basé sur la combinaison parallèle des classifieurs pour la classification des chiffres parlés anglais (*digit recognition*). Les différentes méthodes de normalisations et fusion, de scores ont été présentées à la fin de ce chapitre.

Les résultats obtenus par le système utilisé, décrites dans le cinquième chapitre. Finalement, nous terminons notre mémoire par une conclusion et les perspectives de notre projet.

CHAPITRE : 1

Généralités sur le Traitement du signal (parole)

1.1. Introduction

L'information portée par le signal parole peut être considéré de plusieurs façons.On distingue généralement plusieurs niveaux de description non exclusifs: *acoustique,phonétique* et *phonologique*, au niveau acoustique, on s'intéresse essentiellement au signal que l'on tentera de caractériser par son intensité, sa fréquence, son timbre et ses propriétés statistiques, au plan phonétique, on considère la génération des sons, les phonèmes qui composent un mot et les classes auxquels ils se rattachent, et enfin la phonologie s'attache à décrire le rythme, la prosodie, la mélodie d'une phrase.

Dans ce chapitre, nous allons parler sur le traitement du signal qui est la discipline qui développe et étudie les techniques de traitement (filtrage, amplification...), d'analyse et d'interprétation des signaux. Elle fait donc largement appel aux résultats de la théorie de l'information, des statistiques ainsi qu'à de nombreux autres domaines des mathématiques appliquées. Puis, Avant de vouloir reconnaître le signal de la parole, il est important de commencer par comprendre ce qu'est la parole, quel est son contenu spectral, quelles sont les parties qui la composent

1.2. Définitions de base

Un signal : est la représentation de l'information qu'il transporte de sa source à son destinataire .IL constitue la manifestation physique d'une grandeur mesurable (courant, tentions, force. Etc.)

La théorie du signal **:** a pour objectif fondamental la (description mathématique) des signaux. Cette représentation commode du signal permet de mettre en évidence ses principales caractéristiques (distribution fréquentielle, énergie, etc.) et d'analyse les modifications subies lors de la transmission ou l'exploitation des information véhiculées par ces signaux **[1]**.

1.3. Représentation des signaux

1.3.1. Les principaux types de signaux

Parmi les principaux signaux on peut citer :

1.3.1. 1. Les signaux déterministes

IL s'agit de signaux dont le modèle mathématique est connu. Leurs évolutions en fonction de temps peuvent donc être parfaitement prédite.

Les principaux signaux déterministes sont:

- périodiques

- sinusoïdaux
- non périodiques
- transitoires

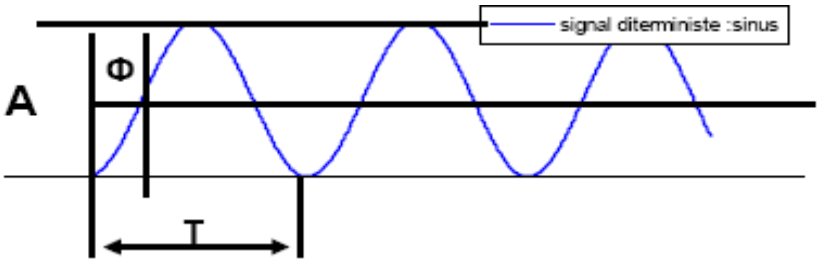


Figure 1.1 Exemple d'un signal déterministe

Y=A.sin (t +θ) T : période

A : Amplitude f=1/T : fréquence fandamantale

Φ : phase w=2. π f

1.3.1.2. Les signaux aléatoires

IL s'agit de signaux dont le modèle mathématique n'est pas connu. Leurs évolutions en fonction de temps sont imprévisibles. La description de ces signaux est sujette à des observations statistiques**.**

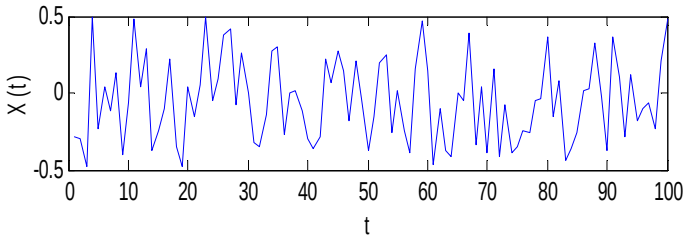


Figure 1.2 Exemple de signal aléatoire

1.3.2. Energie et puissance des signaux

On distingue deux grandes classes de signaux selon leurs natures énergétiques:

- énergie finie.
- les signaux à puissance moyenne fini non nulle qui ne sont pas physiquement réalisable**.**

Un signal x (t) à énergie finie non nulle est pour le quel l'équation suivante reste finie.

$$0 < \int_{-\infty}^{+\infty} \left|x(t)\right|^2 dt < \infty \qquad \textbf{(1.1)}$$

Un signal à puissance moyenne finie non nulle est pour le quel l'équation suivante reste finie.

$$0 \prec \lim_{T\to\infty} \frac{1}{T} . \int_{-T/2}^{T/2} |x(t)|^2 dt \prec \infty \qquad \textbf{(1.2)}$$

T : intervalle de temps.

1.4. Transformation de Fourier

Il y a deux domaines importants de description du signal salon la nature de la variable indépendant:

. Le domaine de description temporal de la forme x(t) dans lequel la variable indépendante est le temps t .dans ce domaine de représentation le x(t) peut être caractérisé par:

- La durée.
- Période fondamentale.
- Amplitude.

. Le domaine de description fréquentiel de la forme x(t) dans le quel la variable indépendant est la fréquence « F » dont la Dimension est l'inverse du temps. Dans ce domaine de représentation le X(f) peut être caractérisera:

- La bande passante.
- Fréquence fondamentale.
- La phase.

Ces deux domaines de description sont relies entre eux par la transformation de fourrier qui constitue une généralisation de la série de Fourier .Cette dualité entre les domaines de description temporel et fréquentiel est le fondement de la plu- part des méthodes du traitement du signal**.**

1.4.1. La série de Fourier

La série de Fourier est une méthode d'analyse de signaux périodique.

Un signal f (t) périodique s'il existe un certain intervalle de temps T tel que l'équation **(1.3)** est vérifiée:

$$f(t) = f(t+T) \quad \textbf{(1.3)}$$

Tout signal périodique borné et intégrable peut se représenter comme une série de fonctions sinusoïdales dont les fréquences sont des multiples de la fréquence fondamentale (série de Fourier).Soit P l'intervalle d'amplitude T (une période).

La décomposition d'un signal périodique en série de Fourier est donnée par l'équation suivante.

$$f(t) = \sum_{n=0,\infty} C_n \cdot \exp(j\pi n f_0 t) \quad \textbf{(1.4)}$$

f_0 **:** est la fréquence fondamentale

$f_n = \frac{n}{T} = nf_0 = \frac{nw}{2.\pi}$: sont appelés les harmoniques.

C_0 : est la composante continue.

Ce développement en série de Fourier permet d'obtenir une représentation fréquentielle discrète du signal ou chaque composante C_n est localisée à la fréquence fn=n/T voir les suivants.

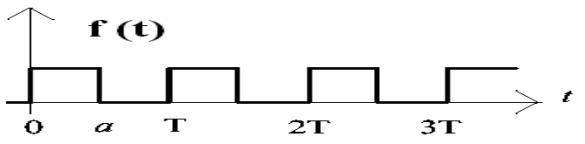


Figure 1.3 Train d'impulsion rectangulaire

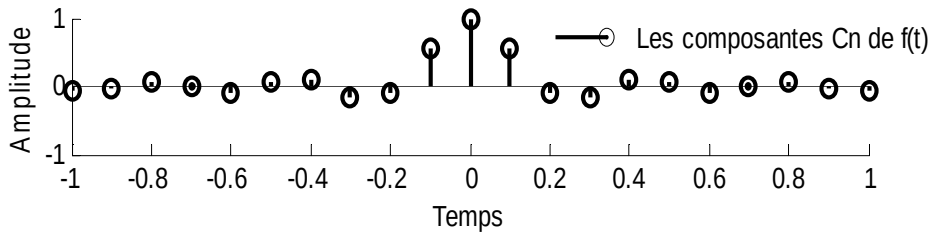


Figure 1.4 Représentation fréquentielle de f (t) « C_n »

1.4.2. La transformée de Fourier

La transformée de Fourier est une généralisation de la série de fourrier appliquée aux signaux non périodiques. Soit f(t) un signal non périodique défini est intégrable dans l'intervalle (-T/2 , +T/2) : **[2]**

la transformée de fourrier de la fonction f(t) dénotée TF est donné par l'équation suivante.

$$F(f) = \int_{-\infty}^{+\infty} f(t)\exp(-jwt)dt \qquad \textbf{(1.5)}$$

On remarque que F (f) est en fonction uniquement de la fréquence f. La relation **(1.6)** est la transformée de fourrier inverse.

La fonction F(f) analyse le signale f(t) en fournissant des informations sur sa distribution fréquentielle (énergie, amplitude, phase).

$$f(t) = \int_{-\infty}^{+\infty} F(f)\exp(jwt)df \qquad \textbf{(1.6)}$$

W =2. $\pi.f_0$: pulsation
f_0 : fréquence fondamentale

La transformée de fourrier F (f) est en générale une fonction complexe pouvant se mettre sous la forme suivante.

$$F(f) = \mathrm{Re}[F(f)] + j\,\mathrm{Im}[F(f))] \qquad \textbf{(1.7)}$$

Re : partie réelle
Im : partie imaginaire

Qui révélé une autre écriture de la transformée de Fourrier: $F(f) = |F(f)|\exp(j\theta(f))$

$$\begin{aligned} |F(f)| &= \sqrt{\mathrm{Re}\,F(f) + j\,\mathrm{Im}\,F(f)} \\ \vartheta(f) &= artg\,(\mathrm{Im}\,F(f) / \mathrm{Re}\,F(f)) \end{aligned} \qquad \textbf{(1.8)}$$

Les figures suivantes montrent la TF d'un signal :

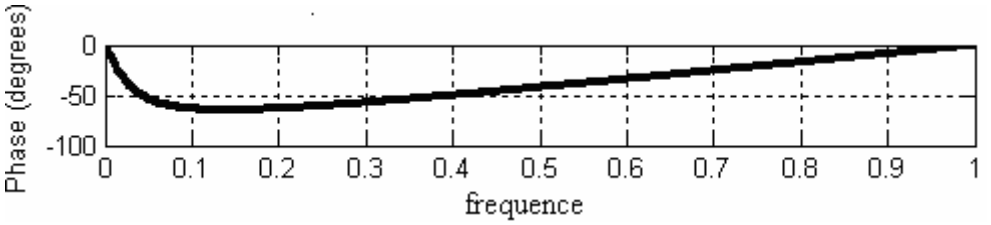
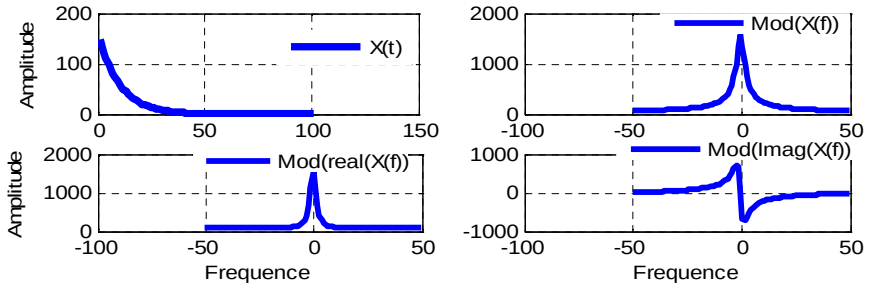


Figure : 1.5 Le spectre de la phase de **X (f)**



-A- **-B-**

Figure: 1.6 -A- La transformée de fourrier de x (t).-**B**-le module de X (f).

1.4.2.1. Théorème de Perceval

Énergie d'un signal = Énergie de sa Transformée. [3]

Voir la relation suivante :

$$\int_{-\infty}^{\infty} [f(t)]^2 dt = \int_{-\infty}^{\infty} [|F(f)|]^2 df \qquad \textbf{(1.9)}$$

Ce théorème important montre que l'énergie totale d'un signal f(t) peut être déterminée en considérant la puissance instantanée |f(t)|^2 ou l'énergie par unité de fréquence $|F(f)|^2$ .Les quantités $|f(t)|^2$ et $|F(f)|^2$ sont respectivement les la densité temporelle d'énergie et la densité spectrale d'énergie. Ce théorème montre donc que l'énergie du signal peut être répartie sur le spectre.

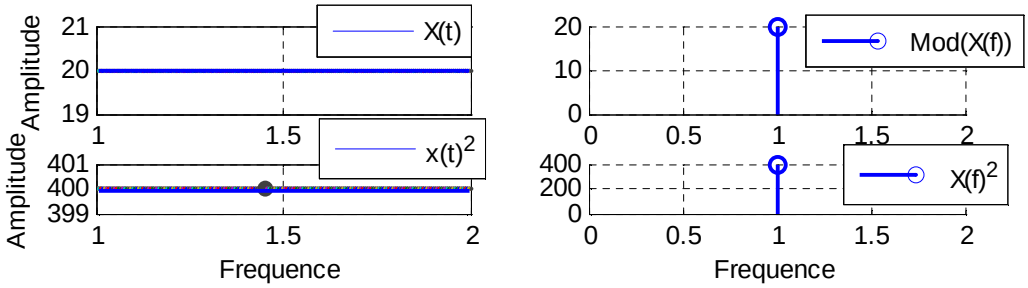


Figure 1.7 Théorème de Perceval: Les aires de $|f(t)|^2$ (gauche) et $|F(f)|^2$ sont égales (adroite)

1.5. Convolution

Une impulsion brève injectée à l'entrée d'un système linéaire, continu et stationnaire, donne en sortie un signal de durée finie. Cette réponse est appelée réponse impultionnelle du système et notée h (t).

Dans le cas général pour :

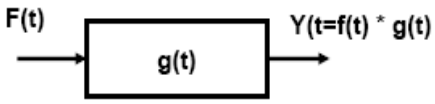


Figure 1.8 Système à la fonction Transfert .g (t).

$$[f * g](t) = \int_{-\infty}^{+\infty} f(\tau)\, g(t-\tau)\, d\tau \qquad \textbf{(1.10)}$$

* :l'opérateur de la convolution

τ **:** Une variable

Cette opération, est appelée «convolution» exprime la réponse du système à un signal quelconque à partir de celle à un signal type (réponse impultionnelle); la réponse dépend du système, caractérisé par h (t), et l'histoire du signal **[1].**

Remarque :

La transformée de fourrier du produit du Convolution est donnée par l'équation (1.11)

$$f(t)*g(t) \xrightarrow{TF} F(f)G(f) \quad \textbf{(1.11)}$$

TF : symbole de la transformée de fourrier

1.6. Corrélation

On appelle fonction d'inter corrélation de deux signaux réels **x** et **y** de carré sommable :

$$\theta_{xy}(\tau) = \int_{-\infty}^{\infty} x(t)\,y(t+\tau)\,dt \quad \textbf{(1.12)}$$

Et la fonction d'autocorrélation **[3] :**

$$\theta_{xx}(\tau) = \int_{-\infty}^{\infty} x(t)\,x(t+\tau)\,dt \quad \textbf{(1.13)}$$

Propriété fondamentale

On appelle densité spectrale d'énergie ou spectre d'énergie du signal réel x la transformée de Fourier de la fonction d'aucorrélation Equation 1.14 **:**

$$\theta_x(f) = \int_{-\infty}^{\infty} \theta_{xx}(t)\,e^{-j2\pi.f.t}\,dt \quad \textbf{(1.14)}$$

Pour tous signaux à énergie finie l'énergie totale est (Théorème de Perceval) équation suivante :

$$W = \int_{-\infty}^{+\infty} [x(t)]^2\,dt = \int_{-\infty}^{\infty} [X(f)]^2\,df = \int_{-\infty}^{\infty} \theta_{xx}(f)\,df \quad \textbf{(1.15)}$$

1.7. Echantillonnage et reconstitution du signal

1.7.1. Echantillonnage du signal

Echantillonner ou numériser un signal continu, revient à prendre des valeurs de ce signal (Échantillons) à des instants donnés (instants d'échantillonnage) et souvent de façon Régulière (période d'échantillonnage).

$$F_e >= 2F_M \quad \textbf{(1.16)}$$

Avec : F_M : la fréquence la plus haute contenue dans le Signal à numériser

F_e : la fréquence d'échantillonnage

Théorème de Shannon:

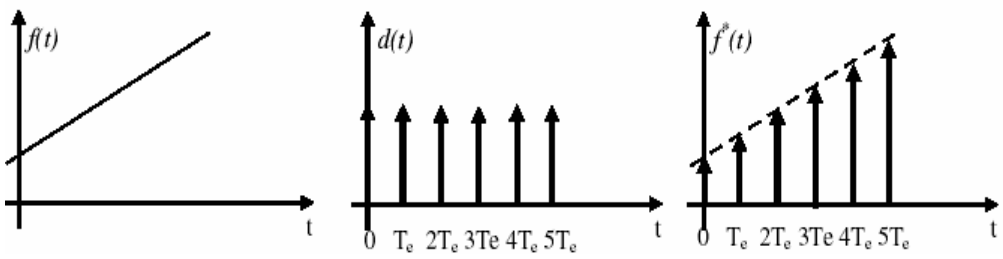
Pour pouvoir reconstituer un signal continu à partir d'un train d'échantillons de période Te, il faut que la fréquence d'échantillonnage Fe , soit au moins deux fois plus grande que la plus grande des fréquences contenues dans le signal continu que lui a donné naissance **[2].**

1.7.2. Echantillonneur Idéal

Un échantillonneur idéal prend, à intervalles réguliers, la valeur du signal de manière instantanée.

Par conséquent :

$$f^*(t) = f(t)\sum_{n=-\infty}^{\infty}\delta(t-nT_e) = \sum_{n=-\infty}^{\infty}f(nT_e)\delta(t-nT_e) \quad \textbf{(1.17)}$$



Signal x Train d'impulsions Dirac = signal échantillonné

Figure:1.9 Passage du signal continu f (t) Au signal discrétisé f^* (t)

Par la TF on a :

$$F_e(f) = Fe\sum_{-\infty}^{+\infty}F(f - nFe) \quad \textbf{(1.18)}$$

Fe : fréquence d'échantillonnage

Une complète reconstitution est donc possible en filtrant le signal échantillonné par un filtre passe bas idéal (filtre d'interpolation ou lissage) de fonction de transfert $H_r(f)$ Tel que :

$$H_r(f) = \text{Te.} \rightarrow pour../\ f\ / <= Fe \qquad \textbf{(1.19)}$$

Le filtrage du signal échantillonné donne le signal reconstitue :

$$fr\ (f) = Fe(f)\ H_r(f) \qquad \textbf{(1.20)}$$

Avec Fe >= $2F_M$

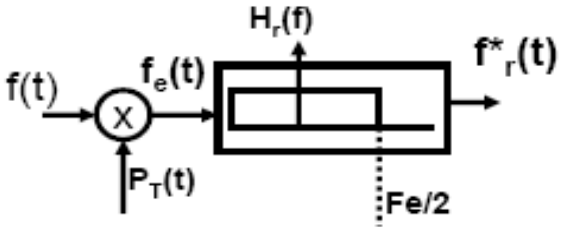


Figure 1.10 Echantillonnage et interpolation

1.8. Processus aléatoires et Bruit

De nombreux domaines utilisent des observations en fonction du temps (ou, plus exceptionnellement, d'une variable d'espace). Dans les cas les plus simples, ces observations se traduisent par une courbe bien définie. Malheureusement, des sciences de la Terre aux sciences humaines, les observations se présentent souvent de manière plus ou moins erratique. Il est donc tentant d'introduire des probabilités. Un processus aléatoire : généralise la notion de variable aléatoire utilisée en statistiques élémentaires. On le définit comme une famille de variables aléatoires $X(t)$ qui associe une telle variable à chaque valeur $t \in T$. L'ensemble des observations disponibles $x(t)$ constitue une réalisation du processus **[4].**

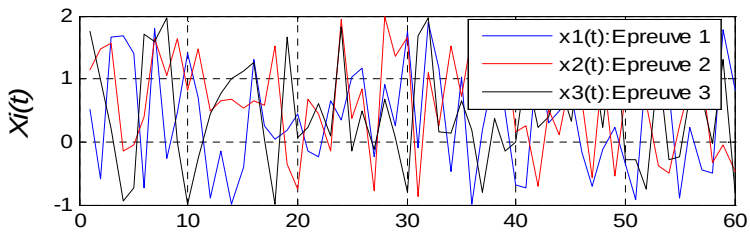


Figure 1.11 Réalisation d'un processus aléatoire provenant de plusieurs épreuves

1.8.1. Description d'un processus aléatoire

Les moments important d'un processus aléatoire x(t) sont :

- Sa moyenne (espérance mathématique) est :

$$\mu_x(t) = E[x(t)] = \int_{-\infty}^{+\infty} x.p(x,t).dx \qquad \textbf{(1.21)}$$

D'où $p(x,t)$: la densité de probabilité du x(t)

- Sa variance est :

$$\delta^2{}_x(t) = E[(x(t) - \mu_x(t))^2] = \int_{-\infty}^{+\infty} (x - \mu_x(t))^2.p(x,t).dx \qquad \textbf{(1.22)}$$

- Sa fonction d'autocorrélation est :

$$R_x = E[x(t), x(t+\tau)] = \int_{-\infty}^{+\infty} x(t).x(t+\tau)p(x,t).dx \qquad \textbf{(1.23)}$$

1.8.2. Caractéristiques statistiques

- Stationnarité : un processus aléatoire est dit stationnaire si les propriétés statistiques sont indépendants du temps (t) : $\mu_x = \mu_x(t)$, $\delta^2{}_x = \delta_x^2(t)$ et $R_x = R_x(\tau)$
- Ergodisme: un processus aléatoire est dit ergodique si les valeurs moyennes statistiques sont édentiques aux valeurs moyennes temporelles : $\mu_x(t) = \overline{x(t)}$, $\delta_x^2(t) = \overline{x^2(t)}$ et $R_x(\tau)=\varsigma_{xx}(\tau)$.

Moyenne temporelle.	Puissance temporelle.	Autocorrélation temporelle
$\overline{x(t)}$	$\overline{x^2(t)}$	$\overline{C_{xx}(\tau)}$

1.8.3. Bruit

1.8.3.1. Définition

Le bruit correspond à tout signal indésirable limitant l'intelligibilité d'un signal utile**.**

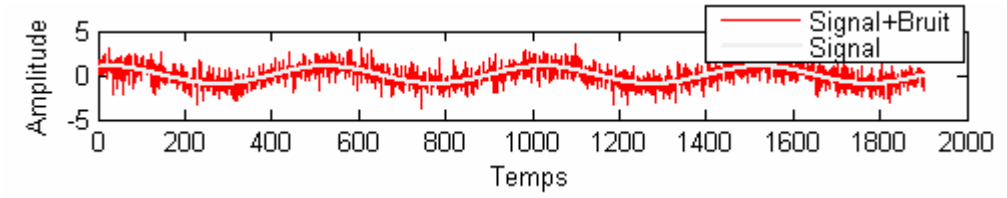


Figure.1.12 Représentation d'un Signal bruité

1.8.3.2. Classification des bruits

Les bruits peuvent être classés selon leur aspect : c'est à dire, leur répartition statistique, leur forme oscillatoire ou le son qui leur correspond **[5]** :

- le bruit blanc ;
- le bruit de scintillation, aussi nommé *bruit rose* ou *bruit en 1/f* ;
- le bruit en créneaux.

Les bruits peuvent être classés suivant leur origine physique :

- le bruit thermique, lié aux évènements de diffusion thermique ;
- le bruit grenaille, lié au déplacement des porteurs dans un champ électrique ;
- le bruit d'avalanche, causé par la génération en avalanche de porteurs ;
- le bruit de quantification (causé par la numérisation d'un signal).
- le bruit fantôme est un bruit qui n'est pas dû au milieu extérieur. Il s'agit en général d'un défaut du capteur ou dans l'électronique qui traite le signal.

1.8.3.3. Modèles de bruit

Le bruit dit gaussien dont la densité de probabilité à une répartition de type gaussien caractérisé par une valeur moyenne et un écart type.

Le bruit dit périodique formé d'une somme de signaux sinusoïdaux sans référence de phase.

1.8.3.4. Rapport signal sur bruit

Le rapport signal sur bruit est un indicateur pour mesurer la qualité de réception d'un signal. Ce nombre étant le rapport de deux puissances, est donc sans grandeur.

Soit P_x la puissance totale du signal utile et P_b la puissance de toutes les perturbations, le rapport (1.24) est le rapport signal sur bruit **[6]**.

$$RSB = 10.Log_{10}\frac{p_x}{P_b} \qquad \textbf{(1.24)}$$

1.8.3.5. Identification d'un système linéaire à l'aide d'un bruit blanc

Le bruit blanc b (t) est processus aléatoire stationnaire au sens large et ergodique qui possède une densité spectrale uniforme ou blanche **[2].**

$$G_b(f) = B_0 \rightarrow -\infty < f < +\infty \qquad \textbf{(1.25)}$$

B_0 : constante

La fonction d'autocorrélation est de la forme :

$$R_b = B_0.\delta(\tau) \qquad \textbf{(1.26)}$$

δ (τ) : impulsion de Dirac

La fonction d'autocorrélation du bruit blanc est différent de zéro que pour $\tau = 0$, ce qui signifie qu'une valeur de ce bruit n'est corrélé qu'avec elle-même : on dit que le bruit blanc n'est corrélé qu'avec lui-même.

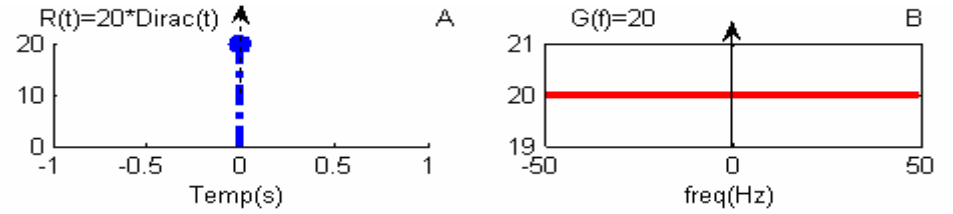


Figure 1.13 Caractéristique du bruit blanc. Bruit blanc (gauche), densité spectrale (adroite)

Soit un générateur de bruit blanc fournissant un signal e(t)=b(t), nous avons la propriété suivante :

$$R_{ee}(t) = e(t) \;\; conv \;\; e^*(-t) = R_{bb}(t) = B_0.\delta(t) \rightarrow [1]$$

Tel que e*: conjugué de e

conv=*

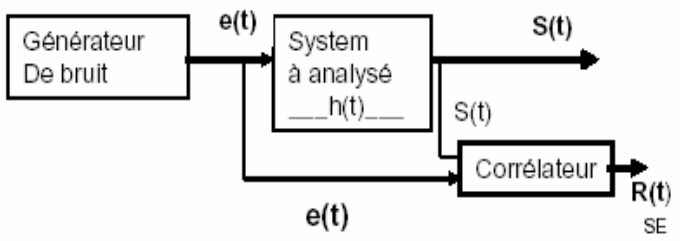


Figure 1.14 Principe de montage permettant de réaliser l'identification d'un système Linéaire à partir d'un bruit blanc

Le signal de sortie S(t) du système à analyser, système linéaire, continu et stationnaire à par définition la forme suivante :

$$S(t) = h(t) * e(t)$$

Le corrélateur réalise donc l'opération de corrélation entre e (t) et S (t), soit sous forme d'un produit de convolution :

$$R_{se}(t) = S(t) * e^{*}(-t) = h(t) * [e(t) * e^{*}(-t)]$$

$$= h(t) * [B_0 . \delta(t)] = B_0 . [h(t) * \delta(t)]$$

$$R_{se}(t) = B_0 . h(t)$$

Ainsi, à la sortie du corrélateur, nous obtenons la réponse impultionnelle à un constant pré.
La sortie du corrélateur donne donc directement la réponse impultionnelle h (t) du système à édentifier.

1.9. Le signal de parole

1.9.1. Q'est ce que c'est la parole ?

La parole est un moyen de communication très efficace et naturel de l'humain. La parole se distingue d'autres sons par ses caractéristiques acoustiques qui ont leur origine dans le mécanisme de production. La parole apparaît physiquement comme une variation de la pression de l'air causée et émise par le système articulatoire. Les sons de parole sont produits par les vibrations des cordes vocales ou par une turbulence crée par l'air dans une constriction ou un relâchement de l'occlusion du conduit vocal. L'unité de parole de plus petite taille est un phonème (voyelle ou consonne). Le nombre de phonèmes est toujours très limité, normalement inférieur à cinquante. Par exemple : la langue française comprend 36 phonème**s [7].**

1.9.2. Production de la parole

La parole est produite par le système articulatoire, présenté à la figure 1.15.

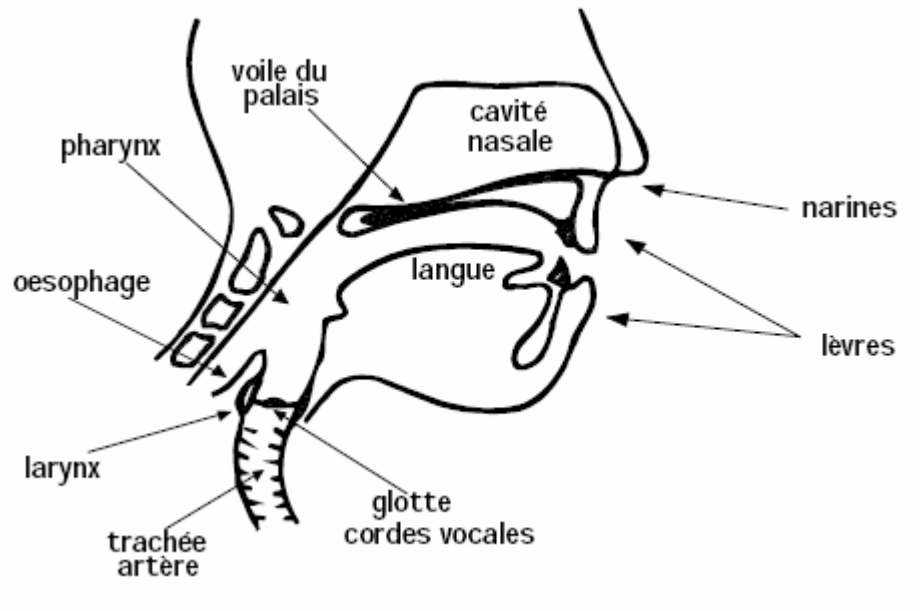
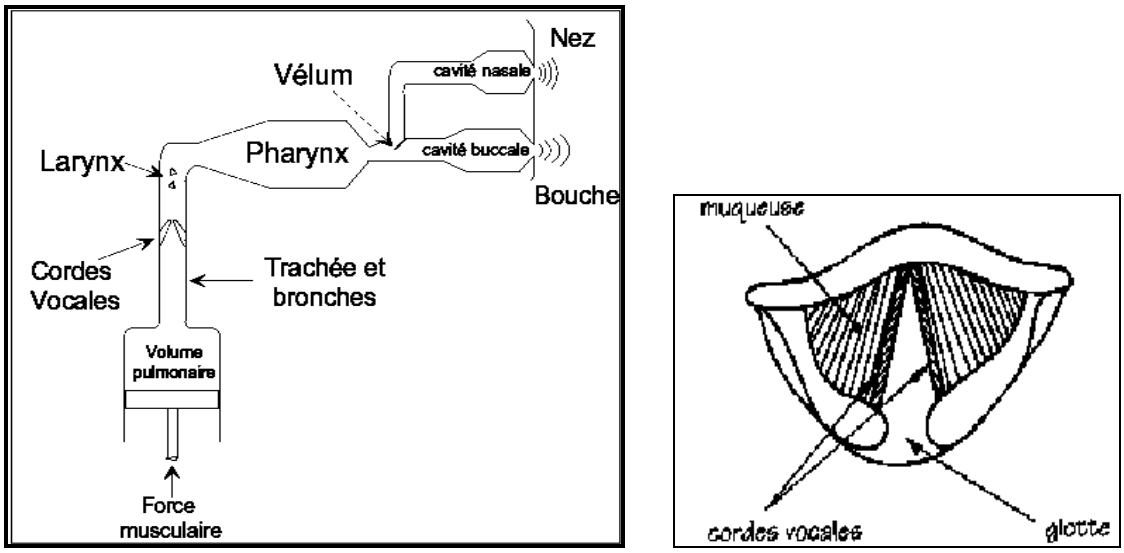


Figure 1.15 Appareil phonatoire **[8]**

La parole peut être décrite comme le résultat de l'action volontaire et coordonnée des appareils respiratoires et masticatoire. Cette action se déroule sous le contrôle du système nerveux central qui reçoit en permanence des informations par rétroaction auditive et par les sensations cénesthésiques.

L'appareil respiratoire fournit l'énergie nécessaire à la production de sons, en poussant de l'air à travers la trachée-artère. Le mouvement du flux d'air cause la vibration des cordes vocales. Cette vibration se propage à travers la cavité pharyngienne, la cavité buccale et la cavité nasale. Selon la position des articulateurs (mâchoire, langue, palais, lèvre, bouche), des sons différents sont produits.



- A - **- B**

Figure 1.16 : –**A-** : Modèle mécanique de production de la parole ; –**B-:** Cordes vocales **[9].**

L'intensité du son émis est liée à la pression de l'air en amont du larynx, sa hauteur est fixée par la fréquence de vibration des cordes vocales, appelée fréquence fondamentale (ou pitch). La fréquence fondamentale peut varier selon le genre (masculin ou féminin) et l'âge du locuteur. La fréquence du fondamental peut varier **[10]** :

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

Deux sons de même intensité et de même hauteur se distinguent par le timbre, qui est déterminé par les amplitudes relatives des harmoniques du fondamental.

Les **sons voisés** résultent d'une vibration périodique des cordes vocales et ce sont les signaux quasi périodiques. Par contre les **sons non voisés** ne présentent pas de structure périodique, ils sont considérés comme des bruits blancs filtrés par la transmittance de la partie du conduit vocal située entre la constriction et lèvres.

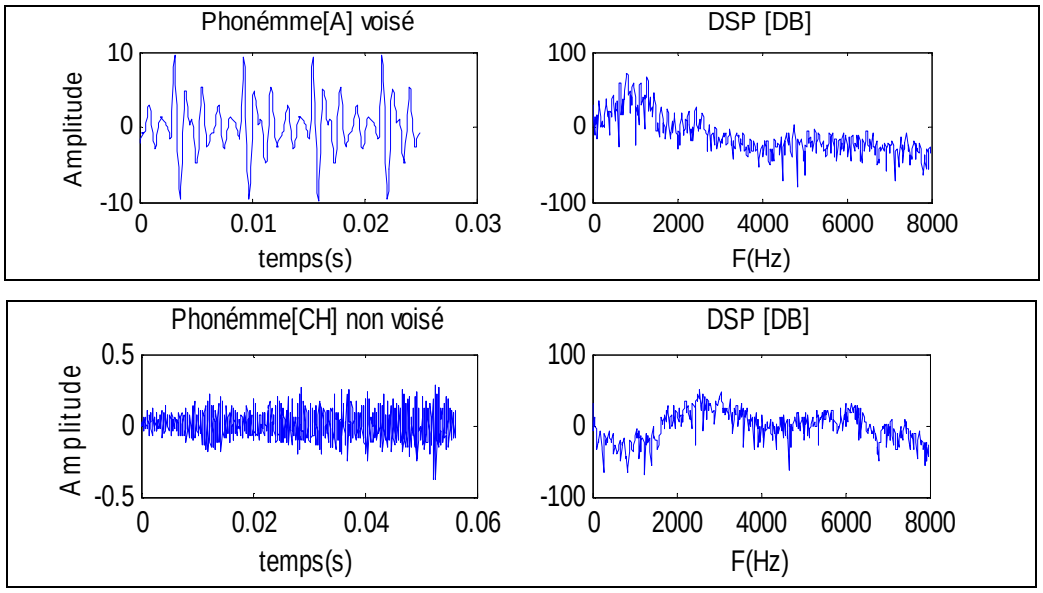


Figure 1.17 Exemple de son voisé (haut) et non – voisé (bas)

La figure 1.17 donne un exemple de son voisé et non voisé. On y constate le son voisé (en haut) représente des zones assez périodiques, appelées zones voisées, tandis que le son non voisé (en bas) représente des zones bruitées, appelées zones non voisées.

Un son voisé est défini par, sa fréquence fondamentale (=hauteur), son timbre (rapport entre fondamental et harmonique).Voir figure 1.18.

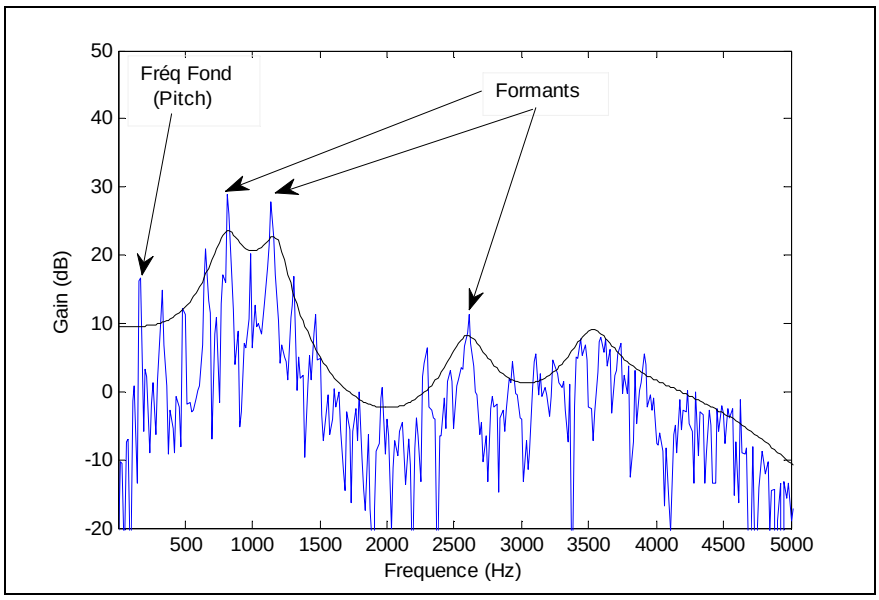


Figure 1.18 Un **son voisé** et ces caractéristiques (Pitch- timbre)

1.9.3. Caractéristiques phonétiques

Les caractéristiques phonétiques sont : les phonèmes, voyelles, consonnes **[7].**

1.9.3.1. Phonème

Un phonème est la plus petite unité présentée dans la parole .Le nombre de phonèmes est toujours très limité (normalement inférieur à cinquante) et ça dépend de chaque langue.

1.9.3.2. Voyelles

Les voyelles sont des sons voisés qui résultent de l'excitation du conduit vocal par des impulsions périodiques de pression liées aux oscillations des cordes vocales. Il y a deux types de voyelle : les ***voyelles orales*** (i, e, u, ...) qui sont émises sans intervention de la cavité nasale et les ***voyelles nasales*** (ã, e~, …) qui font intervenir la cavité nasale. Chaque voyelle se caractérise par les résonances du conduit vocal qu'on appelle "***les formants***". En général, les trois premiers formants sont suffisants pour caractériser toutes les voyelles.

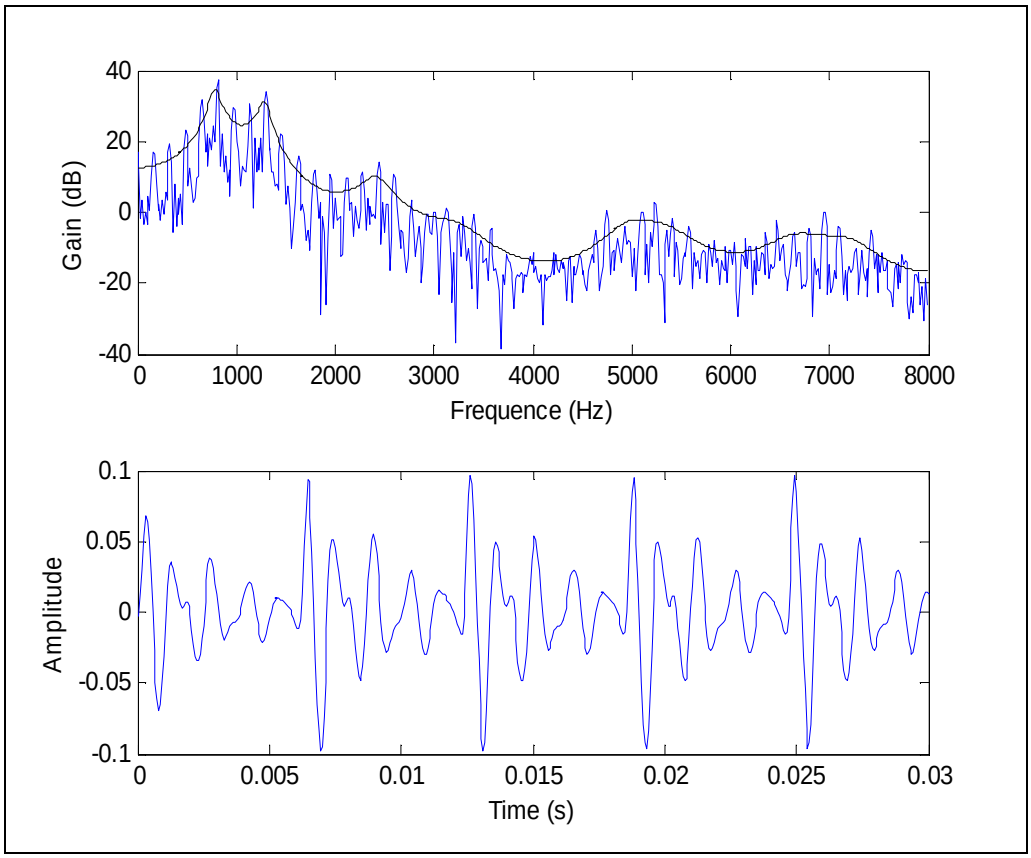
1.9.3.3. Consonnes

Les consonnes sont des sons qui sont produits par une turbulence créée par le passage de l'air dans une constriction du conduit où une source périodique liée à la vibration des cordes vocales s'ajoute à la source de bruit (les consonnes voisées) ([v] et [z]).

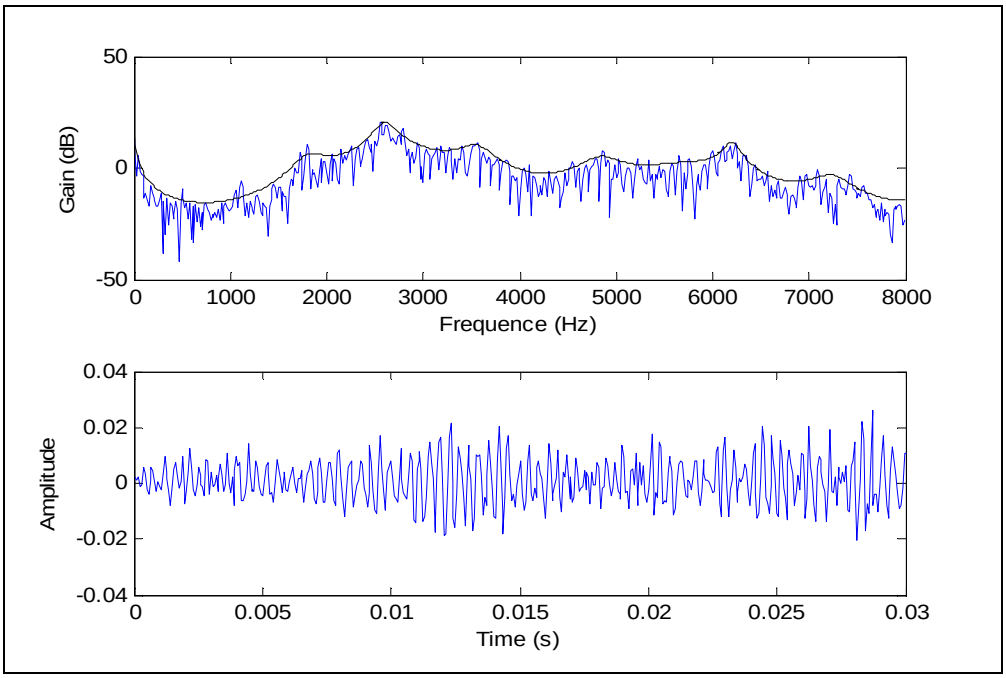
1.9.4. Spectre

Le spectre d'un signal est le résultat de la transformation mathématique (comme la Transformée de Fourier à court terme) de ce signal. À partir d'un spectre on peut savoir la fréquence et l'amplitude des composantes présentes dans le signal analysé. La forme générale des spectres, appelée enveloppe spectrale, présente elle-même des pics et des creux qui correspondent aux résonances et aux anti-résonances du conduit vocal et sont appelés formants et anti-formants.

L'évolution temporelle de leur fréquence centrale et de leur largeur de bande détermine le timbre du son. Il apparaît en pratique que l'enveloppe spectrale des sons voisés est de type passe bas, avec environ un formant par kHz de bande passante, et dont seuls les trois ou quatre premiers contribuent de façon importante au timbre. Par contre, les sons non voisés présentent souvent une accentuation vers les hautes fréquences **[7].**



----- [a]-----



----- [ch] ----

Figure1.19 Exemple du spectre du [a] (haut) et du [ch] de 'baluchon' (bas)

La figure 1.19 illustre le spectre du [a] et [ch] dans le mot 'baluchon'. On peut y voir les parties voisées du [a] (son voisé) apparaissant sous la forme de successions de pics spectraux marqués, dont les fréquences centrales sont multiples de la fréquence fondamentale. Par contre, le spectre du [ch] (son non voisé) ne présente aucune structure particulière.

1.9.5. Spectrogramme

Il est souvent intéressant de représenter l'évolution temporelle du spectre à court terme d'un signal, sous la forme d'un spectrogramme. L'amplitude du spectre y apparaît sous la forme de niveaux de gris dans un diagramme en deux axes : temps et fréquence. Ils mettent en évidence l'enveloppe spectrale du signal, et permettent par conséquent de visualiser l'évolution temporelle des formants. Les périodes voisées y apparaissent sous la forme de bandes verticales plus sombres**.**

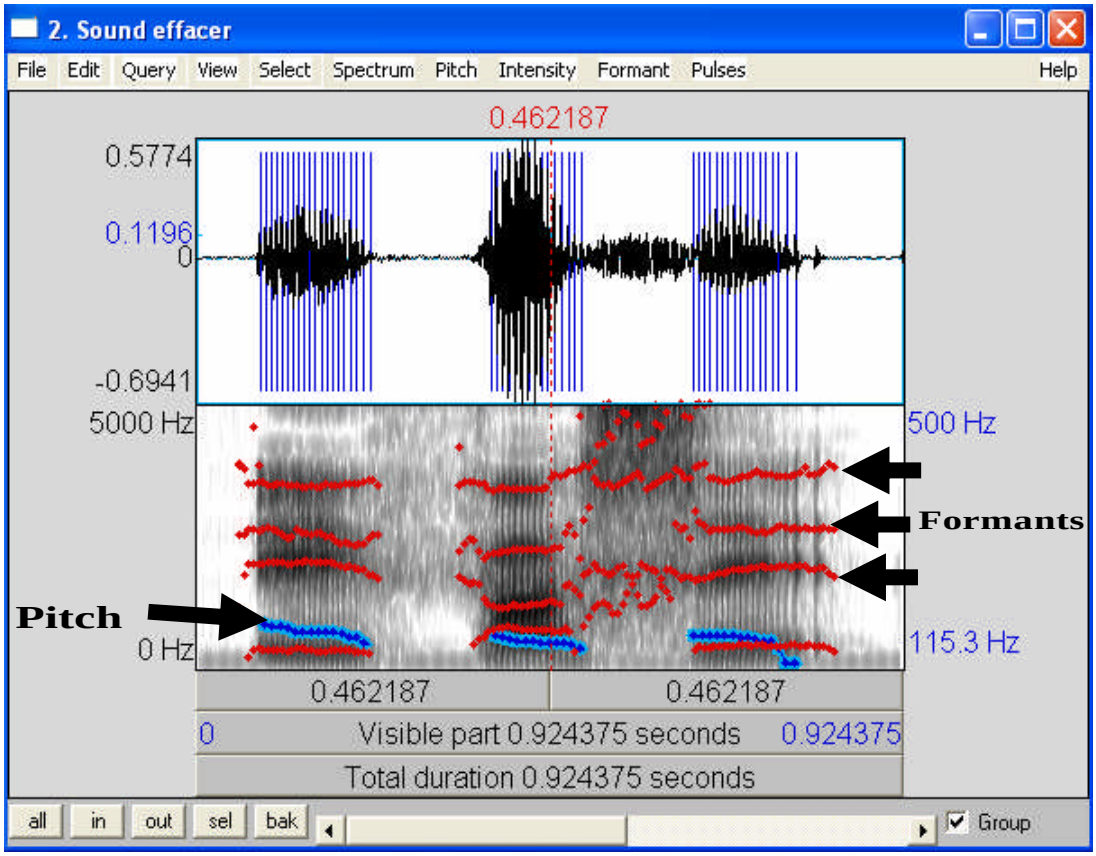


Figure 1.20 Spectrogramme du mot « Effacer » .Editer par le logiciel **[praat 5201]**

1.9.6. Formant

Le flux laryngé étant modulé par un résonateur pharyngo-buccal, le conduit modifie la distribution de l'énergie du spectre de la source vocale et présente plusieurs fréquences de résonance (et d'antirésonance). Il en résulte plusieurs zones de fréquences renforcées appelées formants. La représentation d'un segment de parole périodique dans le domaine fréquentiel (figure 1.21) forme des harmoniques dont l'amplitude est modulée par l'effet de filtrage du conduit vocal. Notons les trois pics d'amplitude dans le spectre à environ 800, 1100 et 2600 Hz, qui correspondent aux résonances du conduit.

Si la fréquence fondamentale de la vibration des cordes vocales est plus haute que la fréquence des premiers formants du système, alors le caractère du son donné par les fréquences des formants seront souvent perdus. Par exemple, pour des chanteurs d'opéra de soprano, qui chantent assez haut, il devient très difficile de distinguer leurs voyelles.

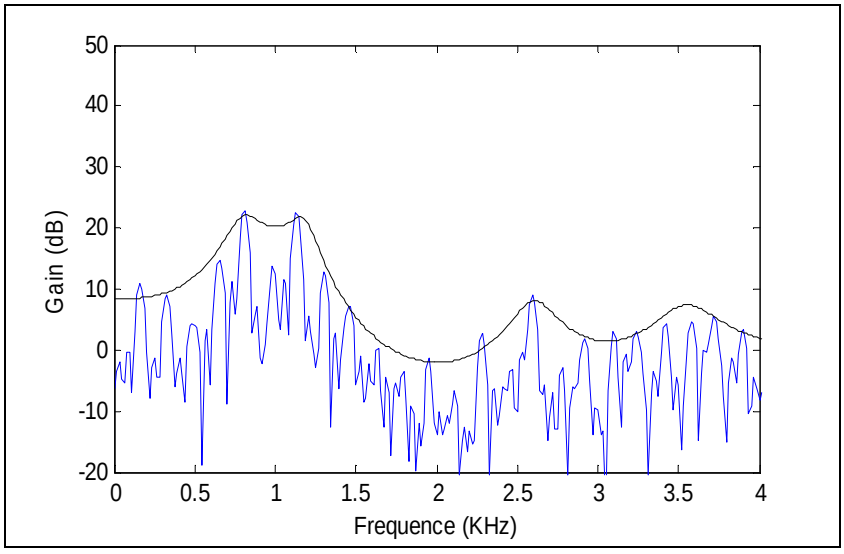


Figure 1.21 Spectre d'un signal voisé, présente quatre résonances formantiques.

Le nombre, les amplitudes et les fréquences des formants varient dans le temps suivant l'articulation des sons voisés. Les deux premiers formants suffisent généralement à la reconnaissance d'une voyelle **[11].**

1.10. Conclusion

Le traitement du signal est une discipline indispensable que pour tout électronicien doit connaître au moins dans ses grandes lignes. Parce que Son champ d'application se situe

dans tous les domaines concernés par la perception, la transmission ou l'exploitation des informations véhiculées par les signaux.

Nous avons pu voir au cours de ce chapitre, le phénomène de la production de la parole, les différentes sources permettant la génération des sons d'une langue donnée. Nous avons aussi remarqué que le signal vocal est très complexe, du fait de sa grande variabilité, ce qui rend toute tentative de le modéliser ou de reconnaître très délicate. Un signal de parole est une séquence de sons correspondant à une suite d'états de l'appareil phonatoire. Le signal de parole est un processus aléatoire non stationnaire à long terme.

CHAPITRE : 2

La reconnaissance de la parole

2.1 Introduction

Si l'homme a la faculté de comprendre un message vocal provenant d'un locuteur quelconque, dans des environnements souvent perturbés par le bruit, quelques soient son mode d'élocution, la syntaxe et le vocabulaire utilisés, la machine est-elle capable d'en faire autant ? Une solution peut-elle répondre en globalité à ces difficultés ? Le problème de la reconnaissance vocale est un sujet d'actualité et pour l'instant, seules les solutions partielles sont aptes à répondre aux différentes tâches que la machine doit effectuer.

La reconnaissance automatique de la parole (RAP) est une branche de la reconnaissance des formes. Grâce à cette technologie, on peut communiquer oralement avec la machine au lieu d'utiliser les gestes ou les commandes des automatismes, ce qui facilite considérablement l'interaction homme/ machine.

Le signal de la parole ne peut être exploité directement. En effet, le signal contient de nombreux autres éléments que le message linguistique : des informations liées au locuteur, aux conditions d'enregistrement, etc. Toutes ces informations ne sont pas nécessaires lors du décodage de la parole et rajoutent même du bruit De plus, la variabilité et la redondance du signal de la parole le rendent difficilement exploitable tel quel. Il est donc nécessaire d'en extraire uniquement les paramètres qui seront dépendants du message linguistique.

Dans un premier temps on expliquera ce que l'on entend par reconnaissance automatique de la parole afin de mieux appréhender le sujet. Puis, on montrera l'évolution de la discipline depuis ses débuts jusqu'à nos jours. Suivra ensuite un chapitre abordant les différentes méthodes utilisées pour reconnaître la parole. Enfin les différents types d'applications basées sur cette technologie.

Dans un deuxième temps on détaillera le prétraitement du signal qui permet de compenser les déformations dues à la transmission du signal de parole, tels que le micro ou le canal téléphonique. Puis, on décrira les techniques de paramétrisation les plus utilisées, du signal vocal, afin d'obtenir une projection du signal de la parole dans un espace de dimension plus restreint où des paramètres pertinents liés à la parole peuvent être facilement extrait. Finalement, les diverses méthodes de modélisation utilisées dans la conception des systèmes de reconnaissance automatique de la parole.

2.2. Reconnaissance de la parole

2.2.1. Introduction

Notre Dieu tout puissant, nous a délégué un organisme biologique très complexe et très développé, notre espèce humaine est l'unique à être privilégiée de « la pensé », le message représentant cette dernière, en générale, peut prendre trois aspects, l'aspect écrit, l'aspect signé et celui verbal, en prenant le dernier aspect, la forme la plus simple qui le concrétise est **la parole**. L'expression répandue 'ce ne sont que des paroles' banalise ce terme, cependant nous apercevons son véritable poids du point de vue phénomène à étudier, seulement chez les chercheurs de ce domaine, car la parole pour eux est un phénomène très complexe, non seulement en tenant compte de la difficulté du mécanisme interne qui la génère et celui qui la transmet, mais aussi de l'entrave de l'organisme qui la reconnaît.

Sans sa reconnaissance, la parole n'a aucun sens, car elle est produite pour être reconnue dans le but de transmettre une pensé précise afin de satisfaire un certain besoin.

Par l'esprit scientifique, de savoir, de développement et de création, les chercheurs optent pour dépasser la compréhension de la communication entre Hommes, en se dirigeant vers un nouvel horizon qui englobe la reconnaissance automatique de la parole.

La reconnaissance automatique de la parole est la manière évoluée pour établir un dialogue artificiel « Homme-Machine », dans le but d'adapter une machine à un vocabulaire limité, qui traduit un besoin issu d'un locuteur.

Le domaine de cette application peut atteindre plusieurs domaines tels que : le pilotage d'avion, la composition du numéro téléphonique du correspondant, faire acquérir des informations à un PC, différentes aides à des handicapés …etc.

Ce domaine fait l'objectif des chercheurs depuis de longues années, par conséquent un bon nombre de méthodes sont incorporées, telles que les méthodes globales, analytiques, probabilistes et encore les méthodes connexionnistes qui sont adoptées depuis les années quarante **[12].**

2.2.2. Définition

La reconnaissance automatique de la parole est l'un des deux domaines du traitement automatique de la parole, l'autre étant la synthèse vocale. La reconnaissance automatique de la parole permet à la machine de comprendre et de traiter des informations fournies oralement

par un utilisateur humain. Elle consiste à employer des techniques d'appariement afin de comparer une onde sonore à un ensemble d'échantillons, composés généralement de mots mais aussi, plus récemment, de phonèmes (unité sonore minimale : voir plus loin). En revanche, le système de synthèse de la parole permet de reproduire d'une manière sonore un texte qui lui est soumis, comme un humain le ferait**.**

Ces deux domaines et notamment la reconnaissance vocale, font appel aux connaissances de plusieurs sciences : l'anatomie (les fonctions de l'appareil phonatoire et de l'oreille), les signaux émis par la parole, la phonétique, le traitement du signal, la linguistique, l'informatique, l'intelligence artificielle et les statistiques.

Il faut bien distinguer ces deux mondes : un système de synthèse vocale peut très bien fonctionner sans qu'un module de reconnaissance n'y soit rattaché. Evidemment le contraire est également tout à fait possible. Par contre, dans certains domaines bien précis, l'un ne va pas sans l'autre. Il est bien entendu que l'étude se portant sur la reconnaissance automatique de la parole, l'autre aspect du traitement de la parole ne sera pas traité dans ce rapport.

Le traitement automatique de la parole ouvre des perspectives nouvelles, compte tenu de la différence considérable existant entre la commande manuelle et vocale. L'utilisation du langage naturel dans le dialogue personne/machine met la technologie à la portée de tous et entraîne sa vulgarisation, en réduisant les contraintes de l'usage des claviers, souris et codes de commandes à maîtriser. En simplifiant le protocole de dialogue personne/machine, le traitement automatique de la parole vise donc aussi un gain de productivité puisque c'est la machine qui s'adapte à l'homme pour communiquer, et non l'inverse. De plus, il rend possible l'utilisation simultanée des yeux ou des mains à une autre tâche. Il permet d'humaniser les systèmes informatiques de gestion de l'information, en axant leur conception sur les utilisateurs.

A la base, les logiciels de reconnaissance vocale servent surtout à entrer du texte en masse tout en se passant du clavier (qui offre un débit de 50 mots par minute contre plus de 150 pour la parole), le clavier reste cependant encore nécessaire aux corrections de texte et à l'utilisation de l'ordinateur **[13]**.

2.2.3. Historique

On va montrer l'évolution de la reconnaissance automatique de la parole depuis ses débuts jusqu'à les dernières années.

2.2.3.1. La naissance

Les premières tentatives de création d'une machine capable de comprendre le discours humain eurent lieu aux USA à la fin des années 40, au sein du Ministère de la Défense américain. Le but était de traduire et d'interpréter des messages russes interceptés.

Ces premières expériences s'appuyaient sur une approche descendante, c'est-à-dire fournissant une recherche mot à mot. Pendant ces premières années de vie de la reconnaissance vocale, il a fallu énormément de temps et de ressources informatiques pour enregistrer et emmagasiner la représentation de chaque mot dans chaque langue. Malgré tous les efforts fournis, les résultats sont médiocres et peu fiables, mais laissaient la porte ouverte à de nombreuses recherches.

2.2.3.2. L'avancée des années 70

Les années 70 sont une période charnière. D'abord, elle voit la première réalisation commerciale en reconnaissance vocale : «le Voice Command system» de *J.J.W. Glenn* et *M.H. Hitchcok,* appareil autonome qui reconnaît de manière fiable 24 mots isolés après cinq cycles d'apprentissage par le même locuteur. L'analyse du message est effectuée par un banc de seize filtres ; chaque mot est représenté par huit événements prélevés aux instants de plus grande variation interne du message. Cette normalisation temporelle, ainsi que les traitements d'apprentissage et de reconnaissance, sont confiés à un mini calculateur incorporé.

Aux Etats-Unis, l'importance des recherches sur la parole a beaucoup varié au cours des dernières années. A l'effort de recherche particulièrement intensif correspondant au projet SUR (Speech Understanding Research, Recherche sur la compréhension de la parole) de l'Arpa (Advanced Research Projects Agency ou Agence de projets de recherche avancés), succède maintenant un effort plus mesuré.

En ex-URSS, les recherches dans ce domaine ont commencé très tôt et restent à l'heure actuelle très actives. Mais à la différence des équipes américaines qui ont développé rapidement d'énormes systèmes de compréhension de la parole, les équipes soviétiques n'ont que très récemment abordé l'étude des niveaux syntaxique et sémantique ; elles sont à l'origine de l'utilisation de la technique de «programmation dynamique» dont l'emploi s'est maintenant partout généralisé.

En France, les recherches ont démarré vers 1970, et plusieurs laboratoires de recherches ont pu mettre au point différents systèmes de reconnaissance vocale avec plus ou

moins de succès, ces laboratoires mettant l'accent sur le support de reconnaissance : mots isolés, syllabes, grands vocabulaires...

2.2.3.3. La reconnaissance du langage

Dès lors, les recherches dans le domaine de la reconnaissance de la parole n'ont cessé de progresser dans le sens de la compréhension du langage parlé et des phrases structurées.

Aujourd'hui, le taux d'erreur ainsi que le temps d'apprentissage des systèmes de reconnaissance ne cesse de diminuer pour atteindre de nos jours des résultats proche de 95%. Ce taux est évidement variable selon la difficulté du langage. En effet la machine a parfois du mal à éviter certains pièges linguistiques **[14].**

Néanmoins nous verrons plus loin que de nos jours nous disposons d'une technologie très aboutie.

Dates	Grandes lignes de l'histoire de a reconnaissance vocale
1950	apparaît le premier système de reconnaissance de chiffres, appareil entièrement câblé et très imparfait.
1951	S.P. Smith présente un détecteur de phonèmes
1952	K.H. Davis, R Biddulph et S.Baleshek annoncent la première ; machine à aborder la reconnaissance de manière globale : les dix chiffres «zéro» à «nine» sont reconnus analogiquement avec un bon taux de réussite pour une seule voix.
1960	utilisation des méthodes numériques : P.B. Denes et M.V. Matthews, pour reconnaître les dix premiers chiffres, comparent globalement les représentations temps fréquence, numérisées et normalisées en durée totale
1965	reconnaissance de phonèmes en parole continue
1968	reconnaissance de mots isolés par des systèmes implantés sur gros ordinateurs (jusqu'à 500 mots)
1971	lancement du projet ARPA aux USA (15 millions de dollars) pour tester la faisabilité de la compréhension automatique de la parole continue avec des contraintes raisonnables
1972	premier appareil commercialisé de reconnaissance de mots
1978	commercialisation d'un système de reconnaissance à microprocesseurs sur une carte de circuits imprimés
1983	première mondiale de commande vocale à bord d? Un avion de chasse en France
1985	commercialisation des premiers systèmes de reconnaissance de plusieurs milliers de mots
1986	lancement du projet japonais ATR de téléphone avec traduction automatique en temps réel
1988	apparition des premières machines à dicter par mots isolés
1990	Dragon lance DragonDictate 30K, le premier système de reconnaissance vocale capable de piloter un PC en utilisant des commandes vocales
1991	Microsoft utilise la technologie de Dragon Systems avec son VoicePilot.
1993	Dragon Systems commercialise le premier logiciel de reconnaissance vocale compatible avec les cartes son standard sur marché (sous Windows).
1994	Lancement du premier système de reconnaissance vocale d'IBM sur PC.
1997	-en juillet, La société Dragon lance la première version de Dragon NaturallySpeaking, un programme capable de reconnaître un langage continu. -En août IBM propose également un système de dictée vocale continue : ViaVoice
1998	-La société Phillips commercialise un logiciel de reconnaissance vocale nommé Freespeech (pour les applications téléphoniques) - La même année, l'entreprise Belge Lernout & Hauspie lancera ses premiers produits nommés VoiceXpress, VoiceXpress + et VoiceXpress Pro
2000	l'entreprise Belge Lernout & Hauspie concrétise, Le géant absorbe Dragon Systems, La société belge devient l'acteur principal de la reconnaissance vocale à l'échelle planétaire. Elle commer- cialise le Dragon Naturally Speaking sous la marque L&H.
2006	Le DARPA prépare une super machine dédiée à la traduction. Cet organisme ambitionne de développer un logiciel capable de traduire de façon quasi simultanée l'arabe et le mandarin avec une précision accrue.

Tableau 2.1 Les Grandes lignes de l'histoire de a reconnaissance vocale

2.2.3.4. Grandes lignes de l'histoire de la reconnaissance vocale

La Reconnaissance Automatique de la Parole (RAP) est un domaine d'étude actif depuis le début des années 50. De nombreux progrès ont été réalisés ces dix dernières années dans ce domaine. Il existe d'ailleurs des logiciels vendus actuellement dans le commerce, capable d'effectuer une reconnaissance de la parole continue pour un vocabulaire important La reconnaissance vocale est une technologie particulièrement récente. On peut résumer en quelques dates les grandes étapes de la reconnaissance de la parole **[14]**. Voir le tableau 2.1 ci-dessus.

2.2.4. Principe de fonctionnement

2.2.4.1. Problématique

Pour bien appréhender le problème de la reconnaissance automatique de la parole, il est bon d'en comprendre les différents niveaux de complexité et les différents facteurs qui en font un problème difficile. Le système doit-il être optimisé pour un unique locuteur ou est-il destiné à devoir se confronter à plusieurs utilisateurs ?

On peut aisément comprendre que les systèmes dépendants d'un seul locuteur sont plus faciles à développer et sont caractérisés par de meilleurs taux de reconnaissance que les systèmes indépendants du locuteur étant donné que la variabilité du signal de parole est plus limitée.

Cette dépendance au locuteur est cependant acquise au prix d'un entraînement spécifique à chaque utilisateur. Ceci n'est néanmoins pas toujours possible. Par exemple, dans le cas d'applications téléphoniques, on comprend bien que les systèmes puissent être utilisés par n'importe qui et donc être indépendants du locuteur. Bien que la méthodologie de base reste la même, cette indépendance au locuteur est obtenue par l'acquisition de nombreux locuteurs (couvrant si possible les différents dialectes) qui sont utilisés simultanément pour l'entraînement de modèles susceptibles d'en extraire toutes les caractéristiques majeures. Une solution intermédiaire parfois utilisée consiste à développer des systèmes capables de s'adapter rapidement (de façon supervisée ou non) au nouveau locuteur. Le système est-il robuste ?

Autrement dit, le système est-il capable de fonctionner proprement dans des conditions difficiles? En effet, de nombreuses variables pouvant affecter significativement les performances des systèmes de reconnaissance ont été identifiées :

- Bruits d'environnement (dans une rue, un bistrot, etc.…).
- Déformation de la voix par l'environnement (réverbérations, échos, etc.…).

- Qualité du matériel utilisé (micro, carte son, etc.…).
- Bande passante fréquentielle limitée (fréquence limitée d'une ligne téléphonique).
- Elocution inhabituelle ou altérée (stress, émotions, fatigue, etc.…) **[12].**

2.2.4.2. Fonctionnement

Le problème de la reconnaissance automatique de la parole consiste à extraire l'information contenue dans un signal de parole, typiquement par échantillonnage du signal électrique obtenu à la sortie d'un microphone, afin qu'il puisse être comparé à des modèles sous forme numérique. Parmi plusieurs techniques de reconnaissance, il y en a deux qui sont majoritairement utilisées afin de parvenir à résoudre ce problème : la comparaison à des exemples et la comparaison d'unités de parole **[12].**

A. Reconnaissance par comparaison à des exemples

Les premiers succès en reconnaissance vocale ont été obtenus dans les années 70 à l'aide d'un paradigme de reconnaissance de mots. L'idée, très simple dans son principe, consiste à faire prononcer un ou plusieurs exemples de chacun des mots susceptibles d'être reconnus, et à les enregistrer sous forme de vecteurs acoustiques (représentation numérique du signal sonore).

Puisque cette suite de vecteurs acoustiques caractérise complètement l'évolution de l'enveloppe spectrale du signal enregistré, on peut dire qu'elle correspond à l'enregistrement d'un spectrogramme.

L'étape de reconnaissance proprement dite consiste alors à analyser le signal inconnu sous la forme d'une suite de vecteurs acoustiques similaires, et à comparer la suite inconnue à chacune des suites des exemples préalablement enregistrés. Le mot «reconnu» sera alors celui dont la suite de vecteurs acoustiques «spectrogramme» colle le mieux à celle du mot inconnu. Il s'agit en quelque sorte de voir dans quelle mesure les spectrogrammes se superposent,voir la figure 2.1 **[12].**

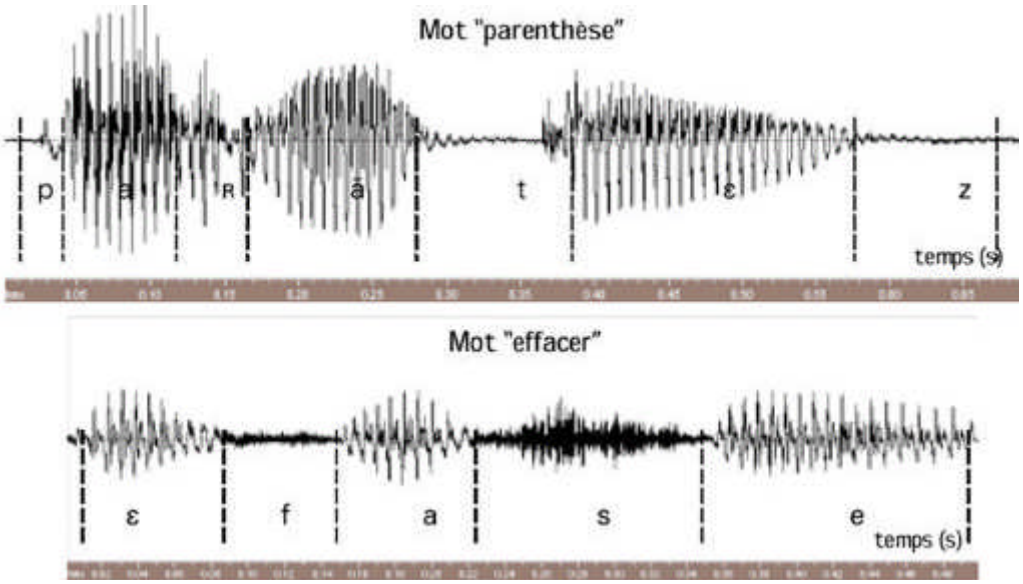


Figure 2.1 : Spectrogramme des mots « parenthèse » et « effacer »

L'étape de reconnaissance proprement dite consiste alors à analyser le signal inconnu sous la forme d'une suite de vecteurs acoustiques similaires, et à comparer la suite inconnue à chacune des suites des exemples préalablement enregistrés. Le mot «reconnu» sera alors celui dont la suite de vecteurs acoustiques (le «spectrogramme») colle le mieux à celle du mot inconnu. Il s'agit en quelque sorte de voir dans quelle mesure les spectrogrammes se superposent.

Ce principe de base n'est cependant pas implémentable directement : un même mot peut en effet être prononcé d'une infinité de façons différentes, en changeant le rythme de l'élocution. Il en résulte des spectrogrammes plus ou moins distordus dans le temps. La superposition du spectrogramme inconnu aux spectrogrammes de base doit dès lors se faire en acceptant une certaine «élasticité» sur les spectrogrammes candidats. Cette notion d'élasticité est formalisée mathématiquement par un algorithme nommé : l'algorithme DTW (Dynamic Time Warping ou déformation dynamique temporelle).

On comprend donc qu'une telle technique soit limitée par la taille du vocabulaire à reconnaître (une centaine de mots tout au plus) et qu'elle soit plus propice à la reconnaissance mono-locuteur (une reconnaissance multi-locuteur imposerait d'enregistrer, de stocker, et surtout d'utiliser pour la comparaison de nombreux exemples pour chaque mot)

Les résultats obtenus, dans le contexte mono-locuteur/petit vocabulaire, sont aujourd'hui excellents (proches de 100%) mais ne correspondent pas aux attentes actuelles en matière de reconnaissance vocale **[14]**.

B. Reconnaissance par modélisation d'unités de parole

La plupart des systèmes de reconnaissance de la parole sont de nos jours basés sur ce mode là. Dés que l'on cherche à concevoir un système réellement multi locuteur, à plus grand vocabulaire et s'adaptant facilement à une application, il devient nécessaire de mener la reconnaissance sur base d'unités de parole de plus petite taille, que l'on appelle phonèmes.

En effet, la parole est constituée d'une suite de sons élémentaires : «a», «é», «ss». Ils sont produits par la vibration des cordes vocales. Ces sons mis bout à bout composent des mots. On ne se contente plus alors d'exemples de ces unités, mais on cherche plutôt à en déduire un modèle (un modèle par unité), qui sera applicable pour n'importe quelle voix. Il apparaît ainsi dans de nombreuses publications que l'on peut décomposer la reconnaissance de la parole en quatre modules.

Un module d'acquisition et de modélisation du signal qui transforme le signal de parole en une séquence de vecteurs acoustiques. Pour être utilisable par un ordinateur, un signal doit tout d'abord être numérisé. Cette opération tend à transformer un phénomène temporel analogique, le signal sonore dans notre cas, en une suite d'éléments discrets, les échantillons. Ceux-ci sont obtenus avec une carte spécialisée courante de nos jours dans les ordinateurs depuis l'avènement du multimédia. La numérisation sonore repose sur deux paramètres : la quantification et la fréquence d'échantillonnage.

La quantification définit le nombre de bits sur lesquels on veut réaliser la numérisation. Elle permet de mesurer l'amplitude de l'onde sonore à chaque pas de l'échantillonnage. Le choix de la fréquence d'échantillonnage est aussi déterminant pour la définition de la bande passante représentée dans le signal numérisé.

Un module acoustique qui peut produire une ou plusieurs hypothèses phonétiques pour chaque segment de parole (par exemple de 10 ms, pour chaque vecteur acoustique), associées en général à une probabilité. Ce générateur d'hypothèses est généralement basé sur des modèles statistiques de phonèmes, qui sont entraînés sur une grande quantité de données de parole (par exemple, enregistrement de nombreuses phrases) contenant plusieurs fois les différentes unités de parole dans plusieurs contextes différents. Ces modèles statistiques sont

le plus souvent constitués de lois statistiques paramétriques dont on ajuste les paramètres pour «coller» au mieux aux données ou de réseaux de neurones artificiels.

Un module lexical dans le cadre de la reconnaissance de la parole continue, même si le système acoustique est basé sur des phonèmes, il faut obtenir, pour chaque entrée du dictionnaire phonétique, un modèle qui lui est propre. Un tel module lexical embarque en général des modèles des mots de la langue (les modèles de base étant de simples dictionnaires phonétiques , les plus complexes sont de véritables automates probabilistes, capables d'associer une probabilité à chaque prononciation possible d'un mot). A l'issue de ce module, il peut donc y avoir plusieurs hypothèses de mots qui ne pourront être départagées que par les contraintes syntaxiques.

Un module syntaxique qui interagit avec un système d'alignement temporel pour forcer la reconnaissance à intégrer des contraintes syntaxiques, voire sémantiques. Les connaissances syntaxiques sont généralement formalisées dans un modèle de la langue, qui associe une probabilité à toute suite de mots présents dans le lexique. Ainsi le système est capable de choisir entre plusieurs mots selon le contexte de la phrase ou du texte en cours, et de son modèle lexical. On peut ajouter à cela un module de filtrage pouvant corriger le signal après l'acquisition afin de retirer les distorsions ou les bruits provenant du matériel ou de l'environnement du locuteur. Ce module est aussi appelé «traitement du canal de transmission». Du fait de sa complexité et du peu d'amélioration qu'il apporte, ce module n'est pas toujours intégré aux systèmes. Cependant la recherche de meilleurs traitements du canal de transmission sera sûrement nécessaire à l'amélioration des systèmes de reconnaissance vocale **[14].**

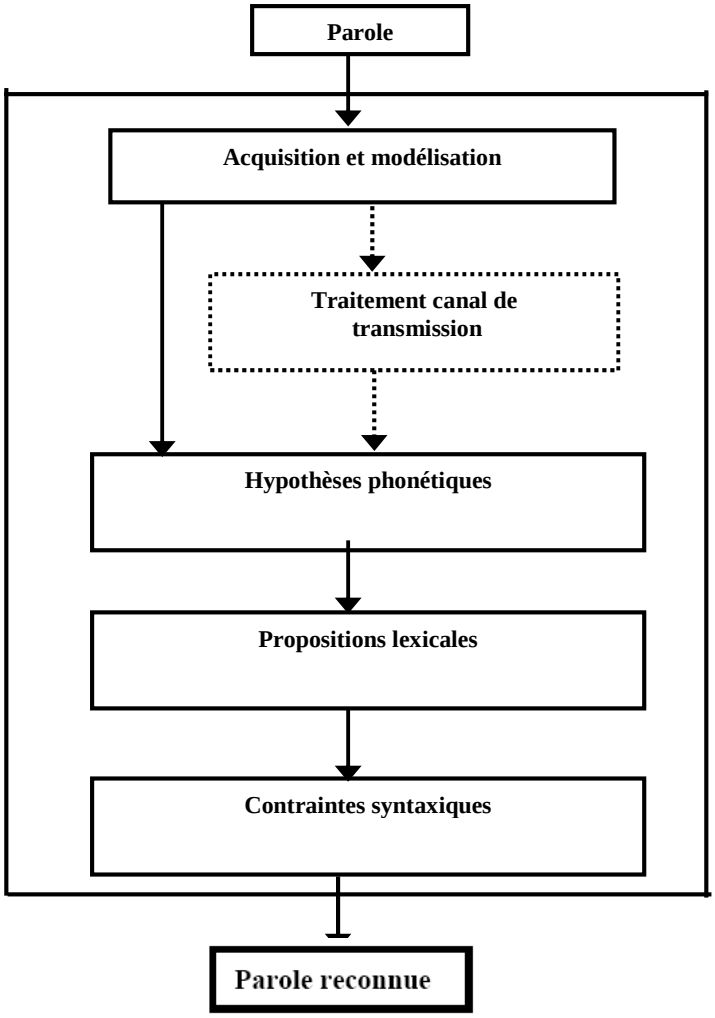


Figure 2.2: Les modules de la comparaison par unité de parole.

2.2.5. Reconnaissance de petits vocabulaires

Ça concerne la reconnaissance de mots isolés, multi locuteurs dans des conditions difficiles, par exemple : reconnaissance de chiffres à travers le réseau téléphonique **[12].**

2.2.6. Reconnaissance de petits vocabulaires de mots isolés

La reconnaissance de mots isolés, le plus souvent mono locuteur, pour des vocabulaires de quelques dizaines jusqu'à quelques centaines de mots est un problème assez bien résolu. Les premiers systèmes commerciaux de cette catégorie sont apparus il y a un peu plus de vingt ans **[13].**

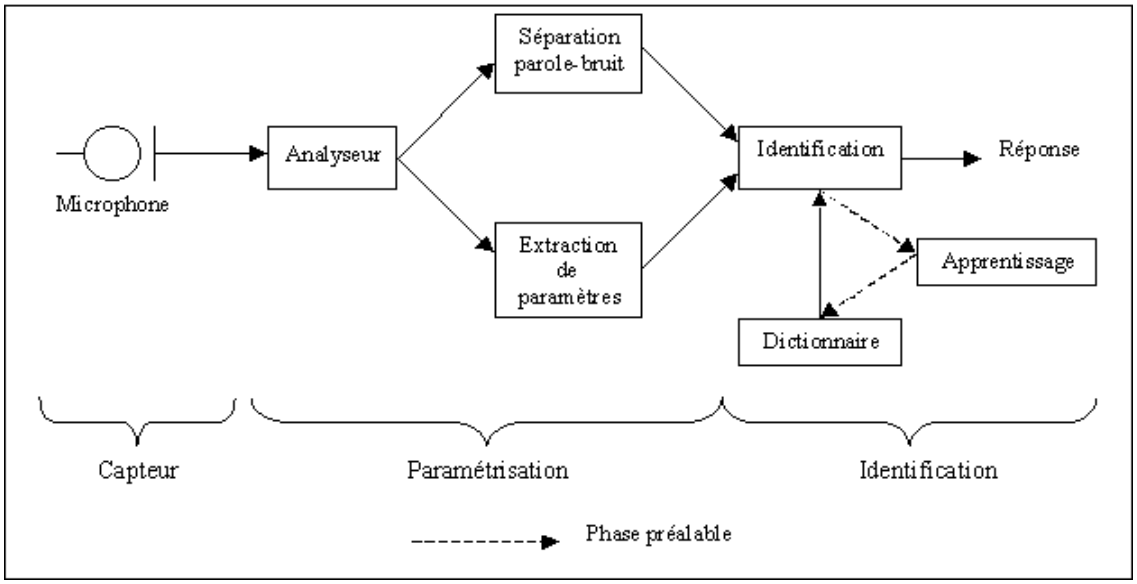


Figure 2.3 : Système de reconnaissance de mots isolés **[15].**

2.2.7 Reconnaissance de grands vocabulaires

Par exemple par IBM, Kurzeil et Dragon systèmes mono locuteur en particulier pour des tâches de dictée de textes dans des domaines d'application fixés .Des systèmes de ce type sont présentés, fondés sur une modélisation stochastique de la parole, méthode actuellement la plus performante. Dans cette catégorie apparaissent aussi des systèmes utilisant une reconnaissance phonétique des mots, c'est notamment le cas d'un produit de speech systems **[13].**

Microsoft, en passant par Apple et IBM, de nombreux industriels travaillent sur des projets de reconnaissance vocale, généralement en complément d'une activité de recherche sur la synthèse de la parole, le tout s'insérant dans des projets plus généraux d'interface Homme Machine.

Il faudra attendre encore plus longtemps avant que la machine remplace purement et simplement la secrétaire dactylo pour la saisie de textes sur ordinateur. Les systèmes de reconnaissance vocale actuels sont encore bien trop grossiers pour comprendre toutes les finesses qui peuvent se glisser dans la **syntaxe** et dans les intonations de la langue parlée en continu et non plus sous la forme de mots clés ou de petites phrases sommaires **[13].**

2.2.8 Reconnaissance de la parole continue

Tout d'abord, qu'est ce que la parole continue ? C'est un discours, des phrases où les mots s'enchaînent sans moyen de séparer, contrairement aux mots isolés. Le but de cette partie n'est pas de rentrer dans les détails de la programmation d'un logiciel de reconnaissance de la parole continue, cela serait trop long et fastidieux. On va donc présenter les " ficelles " de la reconnaissance de la parole continue de manière très générale.

Les objectifs de cette partie étant donc éclaircis, on peut entamer la réflexion autour de la reconnaissance de la parole continue. Pourquoi, après tout, s'évertuer à attribuer à une machine de telles capacités ? Est-ce par pure fantaisie que les auteurs de science-fiction inventent des dialogues entre un héros et sa machine ? Non, ceci relève d'un besoin qui pourrait se résumer à une chose : la recherche d'un confort et d'amélioration de l'interaction de l'homme avec la machine. Les avantages d'un tel progrès sont simples à imaginer.

Cette partie du dossier va donc s'attacher à comprendre les mécanismes mis en jeu dans la reconnaissance de la parole continue, et plus précisément, les stratégies à mettre en œuvre pour aboutir à un bon résultat. Nous avons pu voir dans la partie précédente, intitulée " reconnaissance de mots isolés ", les méthodes pour reconnaître un mot. Dans une phrase, les mots s'enchaînent sans aucun moyen apparent de les dissocier. C'est là qu'intervient la notion de stratégie. La problématique à résoudre est comment découper un signal afin de reconnaître les différents mots ou phonèmes qui le composent **[12].**

2.2.9. Quelques applications

De façon générale, 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, en tout état de cause, comme un moyen complémentaire parmi d'autres moyens d'interaction Homme-Machine plus traditionnels**.** Bien entendu, à chaque type d'application correspondent des critères de performance différents. Ainsi, pour des applications en reconnaissance de la parole, on jugera la qualité d'une application sur les quatre critères principaux suivants :

- Le débit 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, 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 entrée vocale).
- Les contraintes imposées par le système sur l'environnement de fonctionnement : acceptation de bruits 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 numérotation vocale alors qu'au contraire c'est la capacité de discrimination entre locuteurs qui déterminera la qualité d'une serrure à commande vocale.
- Les contraintes imposées par le système sur l'utilisateur : est-il unique ou multiple, doit-il s'astreindre à une phase d'apprentissage préalable **[15].**

2.2.9.1 Dictée vocale

L'orientation actuelle des logiciels tend de plus en plus à offrir un contrôle total de l'environnement permettant de se passer du clavier et de la souris pour utiliser l'ordinateur. Les nouveaux systèmes d'exploitations couplés aux logiciels à venir devraient enfin permettre d'offrir un ordinateur fonctionnant réellement «sans les mains» **[15].**

2.2.9.2 Contrôle de qualité, saisie des données

Dans de nombreux environnements de travail la possibilité de décharger le travailleur, grâce à une interface vocale, apporte un gain incontestable de liberté et de rapidité de mouvement. Pendant qu'il observe un processus complexe, il peut par exemple décrire des informations visuelles. Il a aussi la possibilité de commander à distance un automate évoluant en milieu hostile (apesanteur, sous-marin, industrie pétrolière) **[16].**

2.2.10. Conclusion

Au terme de ce bilan rapide sur la reconnaissance vocale, on a pu constater que ce domaine est particulièrement vaste et qu'il n'existe pas de produit miracle capable de répondre à toutes les applications. Le bruit, par exemple, non traité par ce document, reste un frein à la généralisation des systèmes de reconnaissance.

La reconnaissance vocale reste un compromis entre la taille du vocabulaire, ses possibilités multi locuteur, son encombrement physique, sa rapidité, temps d'apprentissage, et...

La puissance des outils de calcul actuels et les capacités d'intégration des systèmes ont provoqué un regain d'intérêt depuis ces dernières années chez les industriels. En effet, ces

derniers voient dans la reconnaissance vocale, « le plus commercial », permettant de faire la différence avec la concurrence.

2. 3. Prétraitement et extraction des paramètres acoustiques

De manière à atténuer les déformations du signal dues à l'environnement (p.ex. échos, bruits de fond) et à tous les éléments intermédiaires nécessaires à le capter (p.ex. micros), à le transmettre (p.ex. lignes téléphoniques) ou à l'enregistrer (p.ex. convertisseurs analogique/numérique, déformations dues aux têtes d'enregistrement magnétique), un certain nombre de stratégies, de méthodes et d'algorithmes sont déployés. Pour la plupart, ce sont ceux utilisés dans le domaine du traitement du signal, avec cependant quelques particularités dues au signal de parole lui-même, citons-en quelques-unes ici:

- La plus grande partie de l'énergie du signal de parole se trouve entre 0 et 4000 [Hertz]
- Le signal de parole est très redondant **[17].**

2. 3.1. Extraction des vecteurs acoustiques

Presque la plus part des informations qui peuvent être extraites d'un signal de paroles se trouvent dans la bande fréquentielle 200Hz-8KHz. Les étapes principales pour extraire les vecteurs acoustiques sont : *le prétraitement, le fenêtrage, l'extraction de paramètres.* La figure 2.4 regroupe ces étapes.

Figure 2.4 : Les étapes de prétraitement.

A partir d'un signal vocal échantillonné *x(n),* on peut trouver les vecteurs de paramètres f_x (n; m), dont *m=0,1,…, M-1* et *n=0,1,…, N-1*, i.e. M vecteurs de taille N. par la suite, les étapes précédentes seront décrites en détail.

2. 3.2. Le prétraitement

C'est la première étape du processus du calcul des vecteurs acoustiques. L'objectif du prétraitement est de modifier le signal de parole, *x*(*n*) , pour qu'il soit plus convenable à l'étapes de l'extraction de paramètres. Les opérations de prétraitement (élimination de bruit, préaccentuation et l'élimination de silence) peuvent être vues dans la figure 2.5.

Figure 2.5. Les étapes de prétraitement.

2. 3.2.1. La préaccentuation

Le spectre d'un signal de parole a une décroissance globale de l'énergie. Pour compenser cette décroissance, on effectue une préaccentuation en utilisant un filtre passe-haut. Le filtre le plus utilisé est le filtre à réponse impulsionnelle finie décrit ci-dessous:

$$H(z) = 1 - 0.95.z^{-1} \qquad \textbf{(2.1)}$$

La réponse de ce filtre peut être vue dans la figure 2.6. Le filtre dans le domaine temporel est **h(n)** = **{1, −0.95}** et le filtrage dans le domaine temporel donnera le nouveau signal $\mathbf{s_1(n)}$:

$$s_1(n) = \sum_{k=0}^{M-1} h(k)\,\hat{s}(n-k) \qquad \textbf{(2.2)}$$

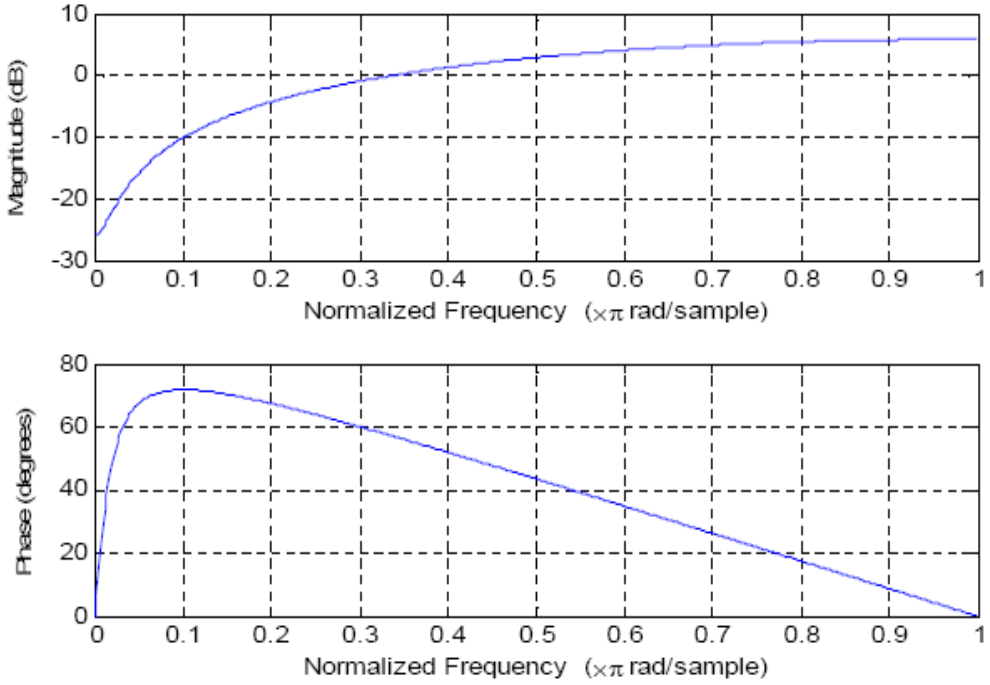


Figure 2.6 : Le filtre de la préaccentuation.

2. 3.2.2. L'élimination du silence

L'élimination des zones de silence qui existent dans un signal de parole est une tâche très importante. Cette tâche semble relativement triviale, mais elle présente quelques difficultés dans la pratique. Les mesures les plus utilisées pour trouver et éliminer le silence sont : l'énergie du signal, la puissance du signal et le rapport de passage par zéro. Pour un signal de parole **$s_1(n)$** ces mesures sont calculées comme suit :

$$E_{s_1}(m) = \sum_{n=m-L+1}^{m} s_1^2(n) \qquad \textbf{(2.3)}$$

$$P_{s_1}(m) = \frac{1}{L} E_{s_1}(m) \qquad \textbf{(2.4)}$$

$$Z_{s_1}(m) = \frac{1}{L} \sum_{n=m-L+1} \left|\mathrm{sgn}(s_1(n)) - \mathrm{sgn}(s_1(n-1))\right| \qquad \textbf{(2.5)}$$

$$\mathrm{sgn}(s_1(n)) = \begin{cases} +1, s_1(n) \geq 0 \\ -1, s_1(n) \prec 0 \end{cases} \qquad \textbf{(2.6)}$$

Il est à noter que l'index pour ces fonctions est *m* et pas *n*, car ces mesures ne sont pas calculées pour chaque échantillon. L'énergie s'accroît quand le signal **$s_1(n)$** contient de la parole et c'est le cas aussi pour la puissance. Le rapport de passage par zéro donne une mesure du nombre de fois où le signal **$s_1(n)$** change de signe. Ce rapport est en général plus grand dans les régions non voisées **[18].**

Ces mesures auront besoin des indicateurs pour prendre la décision du moment où la parole commence et le moment où elle se termine. Pour trouver ces indicateurs, on a besoin d'information concernant le bruit. Cela est faite en supposant que les 5 premières trames sont des bruits. Avec cette supposition la moyenne et la variance de la mesure W seront calculées, telle que W est définie comme suit :

$$W_{s_1}(m) = P_{s_1}(m).(1 - Z_{s_1}(m)).s_c \qquad \textbf{(2.7)}$$

A l'usage de cette fonction, la puissance et le rapport de passage par zéro sont pris en compte. $\mathbf{S_c}$ est un facteur utilisé pour annuler les petites valeurs. Dans une application typique $\mathbf{S_c}$ = **1000.** L'indicateur pour cette fonction peut être calculé comme suit :

$$t_W = \mu_{w+} \alpha\delta_w \quad \textbf{(2.8)}$$

μ_w et δ_w sont respectivement la moyenne et la variance de W_{s1} (m) calculée pour les cinq premiers trames. Le terme α est une constante qui dépend des caractéristiques du signal. Après quelques tests, l'approximation ci-dessous de α donnera de bons résultats pour l'élimination du silence avec plusieurs niveaux de bruit.

$$\alpha = 0.2.\delta w - 0.8 \quad \textbf{(2.9)}$$

La fonction d'élimination du silence, VAD (m) peut être définie comme suit :

$$\mathrm{VAD(m)} = \begin{cases} 1, \mathrm{W}_{s_1}(m) \geq t_w \\ 0, \mathrm{W}_{s_1}(m) \prec t_w \end{cases} \quad \textbf{(2.10)}$$

Avec la fonction VAD (n) le calcul de $x_1(n)$ est simplement $s_1(n)$ quand VAD(n) est à un. Après l'étape de prétraitement, le signal $x_1(n)$ est préparé pour l'étape suivante : Voir la figure 2.7 et la figure2.8.

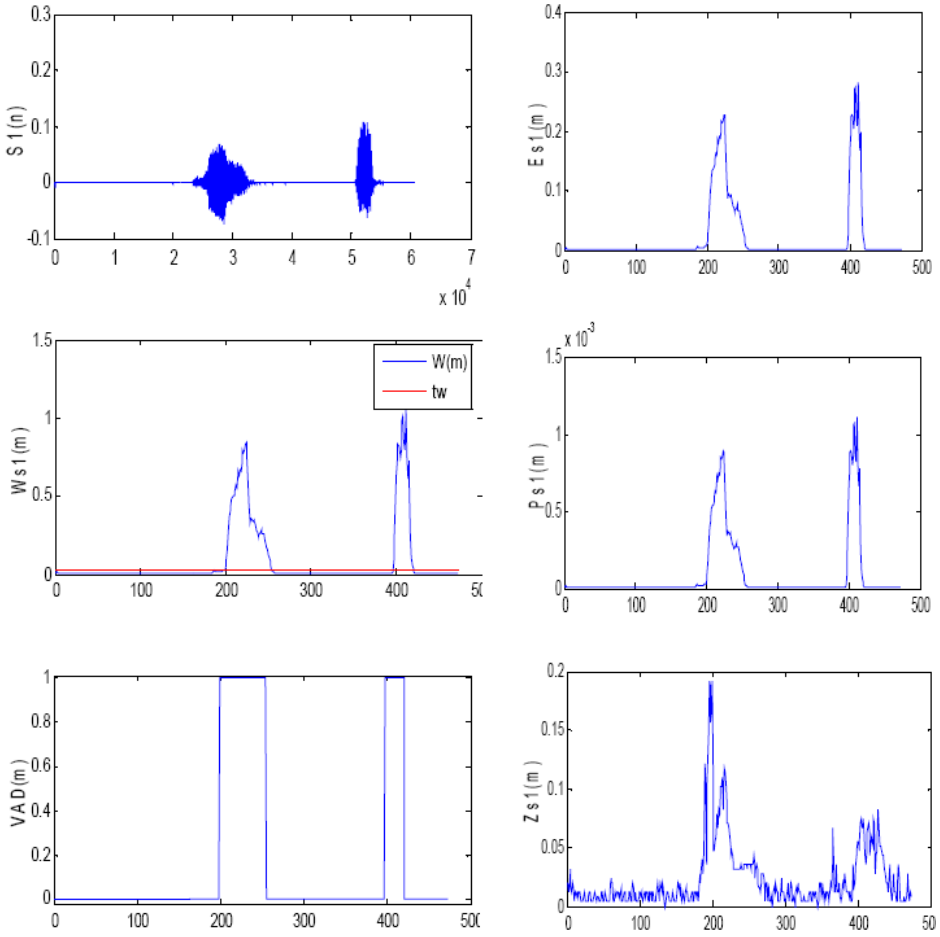


Figure 2.7 : Les différentes mesures utilisées pour éliminer le silence.

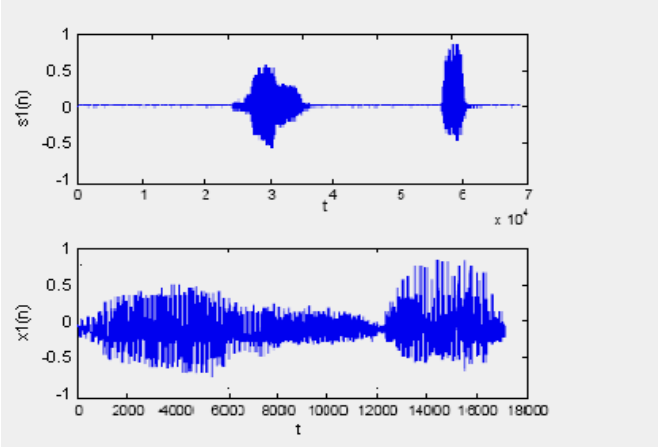


Figure 2.8 Du signal $s_1(n)$ avec silence au signal $x_1(n)$ sans silence en utilisant la fonction *VAD (n).*

2. 3.3. Le fenêtrage

L'étape suivante consiste à découper **$x_1(n)$** en trames et d'appliquer une fenêtre pour chacune d'elles. Voir la figure 2.9.

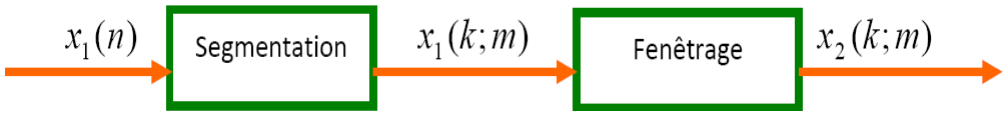


Figure 2.9: Les étapes du fenêtrage.

Chaque trame est de longueur de K échantillons, tel que les trames adjacentes sont séparées par P échantillons, voir la figure 2.10.

Ensuite, on applique une fenêtre à chaque trame pour réduire la discontinuité à la fin de chacune d'elles. La fenêtre la plus utilisée est la fenêtre de Hamming, elle est définie comme suit :

$$W(k) = 0.54 - 0.46\cos\left(\frac{2\pi k}{k-1}\right) \qquad \textbf{(2.11)}$$

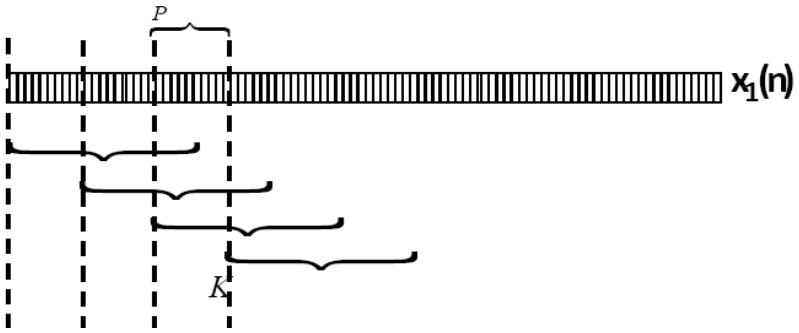


Figure 2.10 : Le découpage en trames.

2.3.4. Extraction de paramètres caractéristiques

Le signal de la parole ne peut être exploité directement. En effet, le signal contient de nombreux autres éléments que le message linguistique : des informations liées au locuteur, aux conditions d'enregistrement, etc. Toutes ces informations ne sont pas nécessaires lors du décodage de la parole et rajoutent même du bruit. De plus, la variabilité et la redondance du signal de la parole le rendent difficilement exploitable tel quel. Il est donc nécessaire d'en extraire uniquement les paramètres qui seront dépendants du message linguistique.

Généralement, ces paramètres sont estimés via des fenêtres glissantes sur le signal. Cette analyse par fenêtre permet d'estimer le signal sur une portion du signal jugée

stationnaire : généralement 10 à 30 ms en limitant les effets de bord et les discontinuités du signal via une fenêtre de Hamming **[19].**

La majorité des paramètres représentent le spectre fréquentiel et son évolution sur une fenêtre de taille donnée. Les techniques de paramétrisation les plus utilisées sont : PLP (Perceptual Linear Prediction : domaine spectral) **[20],** LPCC (Linear Prediction Cepstral Coefficients : domaine temporel**) [21],** MFCC (Mel Frequency Cepstral Coefficients : domaine cepstral).

2. 3.4.1. Les paramètres calculés par le modèle de prédiction linéaire

C'est un modèle basé sur les corrélations entre les échantillons successifs du signal vocal. Cette méthode se base sur l'hypothèse que le canal buccal est constitué d'un tube cylindrique de section variable. La LPC (Linear Predicting Coding) pour un ordre p se définit de la manière suivante **[22]**:

$$\hat{s}(n) \approx a_1 s(n-1) + a_2 s(n-2) + \ldots + a_p s(n-p) = \sum_{i=1}^{p} a_i s(n-i) \qquad \textbf{(2.12)}$$

On considère que le signal de la parole à l'instant **n** peut être représenté par une combinaison linéaire des p échantillons précédents. Les a_i sont les coefficients de prédiction et sont supposés constants sur une fenêtre d'analyse. On introduit le terme d'excitation unitaire u(n) et un gain G.

$$s(n) = \sum_{i=1}^{p} a_i s(n-i) + Gu(n) \qquad \textbf{(2.13)}$$

On réalise la transformée en z de l'expression, on obtient :

$$S(z) = \sum_{i=1}^{p} a_i z^{-i} S(z) + GU(z) \qquad \textbf{(2.13)}$$

La fonction de transfert est donc la suivante :

$$H(z) = \frac{1}{G} \times \frac{S(z)}{U(z)} = \frac{1}{1 - \sum_{i=1}^{p} a_i \times z^{-i}} = \frac{1}{A(z)} \qquad \textbf{(2.14)}$$

On peut modéliser le système par la figure suivante et peut être rapproché au modèle acoustique linéaire de production de parole. La fonction u(n) est soit un train d'impulsions quasi périodiques pour les sons voisés (produit par les cordes vocales) ou une source de bruit aléatoire pour les sons non voisés.

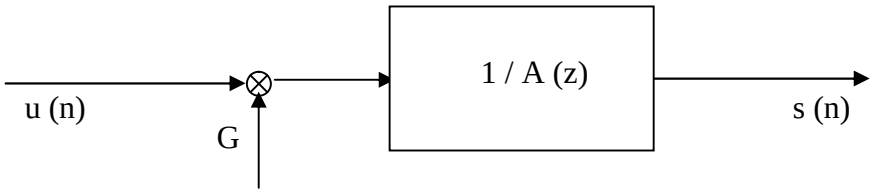


Figure 2.11 : Représentation du canal buccal.

Pour définir le signal s(n), on définit l'erreur de prédiction :

$$e(n) = s(n) - \hat{s}(n) = Gu(n) \qquad \textbf{(2.15)}$$

Pour déterminer les coefficients a_i, on utilise la méthode des moindres carrés sur une fenêtre de temps de longueur m :

$$E_m = \sum_m e^2(m) = \sum_m \left[s(m) - \sum_{i=1}^{p} a_i s(m-i) \right]^2 \qquad \textbf{(2.16)}$$

On cherche à minimiser E_m, deux méthodes peuvent être réalisées pour résoudre le système d'équation :

- la méthode de covariance
- la méthode d'autocorrélation

Une résolution rapide des modèles autorégressifs est donnée par l'algorithme de *Levinson* et *Schur* **[22].**

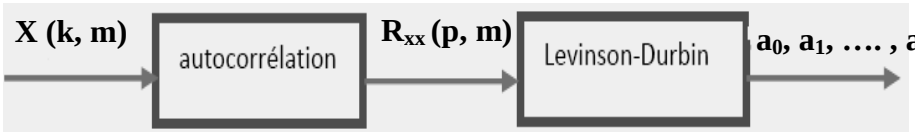


Figure 2.12 : Les étapes de la prédiction linéaire, X (k, m) représente les trames du signal de parole et R_{xx} (p, m), l'autocorrélation des fenetres

A. LPCC (Linear Prediction Cepstral Coefficients)

L'un des ensembles de paramètres les plus importants que nous pouvons déduire en profitant des coefficients a_i sont les coefficients cepstraux LPCC en utilisant la procédure récursive suivante **[17].**

$$
\begin{aligned}
c_0 &= r(0) \\
c_m &= \mathrm{a}_m + \sum_{k=1}^{m-1} (\frac{k}{m}) c_k \mathrm{a}_{m-k}, \quad 1 \le m \le p \\
c_m &= \sum_{k=1}^{m-1} (\frac{k}{m}) c_k \mathrm{a}_{m-k}, \quad m > p
\end{aligned}
\qquad \textbf{(2.17)}
$$

B. Les paramètres calculés par l'analyse Mel cepstral

Au lieu d'utiliser la prédiction linéaire, une autre méthode très utilisée dans l'extraction des paramètres acoustiques d'un locuteur, à savoir, l'analyse mel-*cepstral*. Cette méthode se compose de deux parties : le calcul cepstral et une méthode nommée échelle Mel **[18]**.

B.1. Banc de filtres

Les techniques de filtrages sont des transformations qui permettent d'obtenir une nouvelle représentation du signal dans un nouvel espace dans lequel le traitement est réalisé. Cela revient à multiplier le signal d'origine dans ce nouvel espace par une fonction de transfert. La transformation inverse permet d'observer le résultat de l'opération. Les filtres permettent de sélectionner des fréquences particulières.

Un banc de filtres est un ensemble de filtres conçus pour partitionner le spectre d'un signal en bandes de fréquences, appelées « bandes critiques ». Ces bandes se chevauchent et dont les fréquences centrales ont la plus forte amplitude. Chaque bande critique correspond à l'écartement en fréquence nécessaire pour que deux harmoniques soient discriminées. Cette

analyse se base sur le système de perception humain. Leur étagement en fréquence imite la répartition et la forme des filtres de la cochlée.

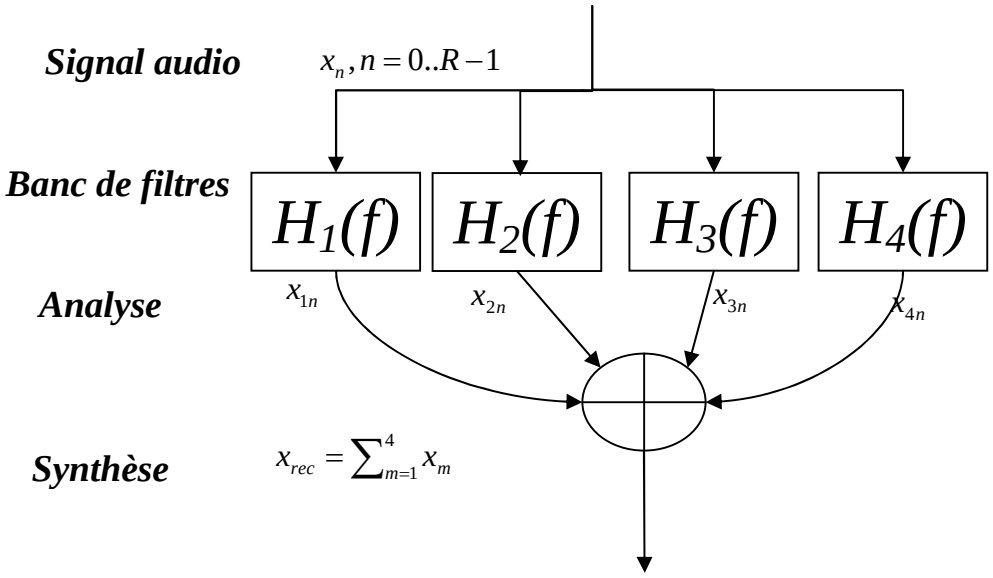


Figure2.13. Schéma d'un banc de filtre

La répartition des fréquences des filtres est différente selon les échelles choisies soit linéaire ou logarithmiques. Il existe plusieurs implémentations de bancs de filtres comme **l'échelle MEL** ou l'échelle Bark **[22].**

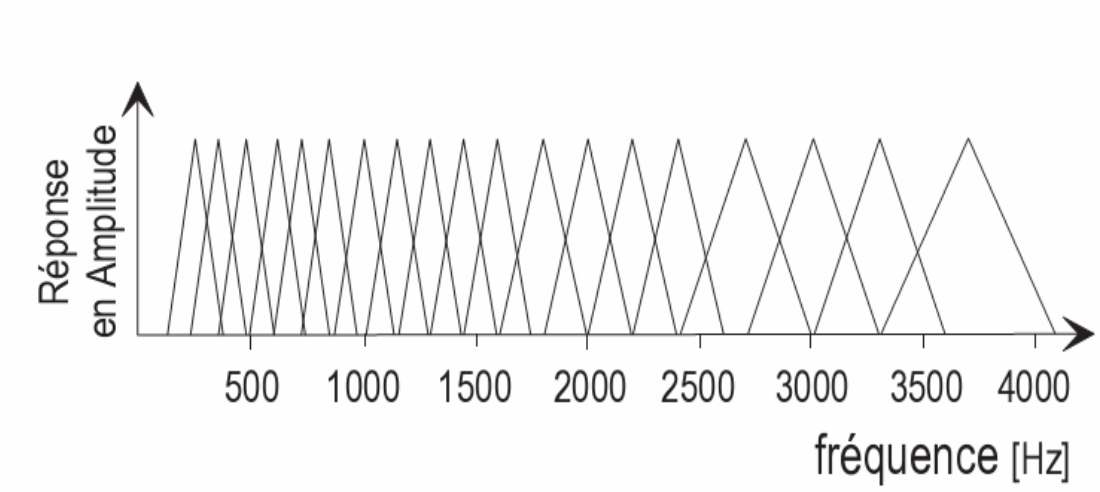


Figure2.14 : Implémentation de bancs de filtres selon l'échelle MEL avec 19 canaux répartis entre 0 et 4000 Hz

B.2. Analyse Cepstrale

Les coefficients en sortie des bancs de filtres ou les coefficients $\mathbf{a_i}$ (*LPC*) peuvent être utilisés pour mesurer des différences entre deux trames. Ils présentent cependant des inconvénients comme de dépendre de l'énergie du signal ou de l'excitation. La transformation cepstrale permet d'obtenir une information normalisée.

L'analyse cepstrale est une méthode basée sur le modèle de production de la parole. Le signal de la parole peut être représenté par la convolution de la source (cordes vocales) et du filtre (canal buccal) dans le domaine temporel **[22]** comme suit :

$$s(t) = e(t) \otimes h(t) \qquad \textbf{(2.18)}$$

On passe dans le domaine fréquentielle pour obtenir l'enveloppe spectrale qui permet de faire apparaître les différences de fréquences. La convolution devient donc une multiplication.

$$S(f) = E(f) \,.\, H(f) \qquad \textbf{(2.19)}$$

On souhaite séparer la source du filtre pour récupérer l'enveloppe spectrale du signal. Pour cela, on utilise la fonction log :

$$\log (| S(f) |) = \log (| E(f) |) + \log (| H(f) |) \qquad \textbf{(2.20)}$$

On applique ensuite la transformée inverse pour obtenir les coefficients temporels appelés coefficients cepstraux. Les premiers coefficients donnent les paramètres de l'enveloppe spectrale, les coefficients les plus élevés fournissent les variations de l'excitation. Si l'enveloppe spectrale est obtenue à partir d'une analyse en banc de filtres sur une échelle *MEL*, les coefficients sont appelés *MFCC* (*Mel Frequency Cepstrum Coefficients*). Autrement si l'analyse du signal est obtenue par *LPC*, les coefficients sont dénommés *LPCC* (*Linear Predicting Coding Cepstrum*). De plus, les coefficients cepstraux $\mathbf{c_m}$ *LPCC* peuvent être obtenus à partir des coefficients $\mathbf{a_p}$ de la *LPC*, comme on a vu dans la section précédente.

2. 3.4.3. Analyse Perceptive PLP (Perceptually based Linear Prediction analysis)

La prédiction linéaire perceptuelle exploite les connaissances du système auditif humain pour paramétrer la parole en introduisant des mécanismes psycho acoustiques de l'oreille humaine.

L'analyse perceptive repose sur l'analyse par prédiction linéaire (*LPC*). Dans cette analyse, les coefficients sont calculés suite à la résolution d'équations obtenus par la transformée de Fourier inverse du module au carré de la transformée de Fourier du signal.

L'analyse perceptive introduit au niveau fréquentiel des bandes critiques. Cette intégration se fait avec un banc de 17 filtres dont les fréquences centrales sont espacées linéairement selon l'échelle *Bark*. Cette intégration se justifie par le fait que l'oreille se comporte comme un banc de filtres.

En sortie, on effectue une préaccentuation à l'aide d'un filtre du premier ordre pour rendre compte des courbes d'isotonie. Ces courbes indiquent les intensités (en dB) nécessaires pour obtenir une même sensation de volume sonore à différentes fréquences : Par exemple, pour reproduire le volume perçu d'un son de 1000 Hz à 40 dB, il faut ± 70 dB à 40 Hz et ± 50 dB à 10 kHz. Ensuite, on applique une phase de compression d'intensité en sonie par l'utilisation de la fonction racine cubique pour simuler la loi de perception humaine en puissance sonore **[22].**

Enfin, on réalise la transformée de Fourier inverse et on calcule les coefficients *PLP* de la même manière que les coefficients *LPC*.

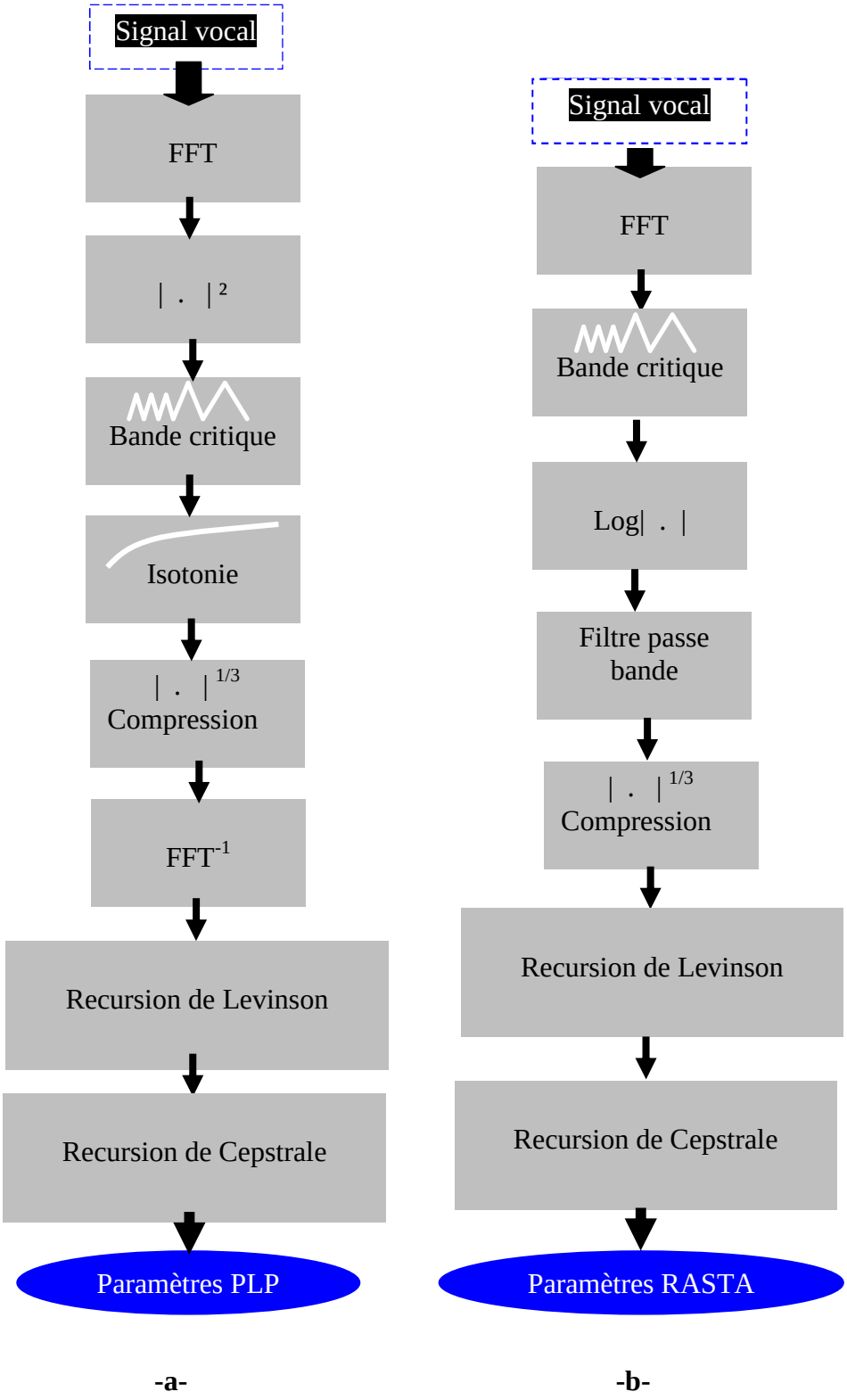
Les expériences ont montré que les coefficients *PLP* permettent d'annuler les différences entre locuteurs et possèdent une meilleure robustesse au bruit **[23].**

2. 3.4.4. **Analyse RASTA (RelAtive SpecTrAl)**

L'analyse RASTA a pour but de supprimer les variations temporelles trop lentes ou trop rapides correspondantes au bruit. Elle se base sur le fait que la perception humaine réagit aux valeurs relatives plus qu'aux valeurs absolues.

L'analyse RASTA repose sur l'analyse PLP. En effet, après avoir effectuée la transformée de Fourier discrète à court terme, on calcule le spectre d'amplitude en bandes critiques. On applique le logarithme pour récupérer l'enveloppe spectrale du signal comme pour une analyse cepstrale. On effectue ensuite un filtre passe bande qui a pour conséquence de supprimer les composantes constantes ou lentes du signal. On réalise après une compression de l'amplitude par l'application d'une racine cubique. Enfin, on calcule les coefficients selon la méthode *LPC* classique.

L'analyse RASTA-PLP a pour but d'améliorer la robustesse du système de reconnaissance en milieu bruité. Elle est basée sur l'analyse PLP **[24].**



Algorithme 2.1 Les étapes à suivre pour l'analyse. **-a-** de PLP. **-b-** de RASTA

2.3.5. Analyse de données et sélection de caractéristiques

Les vecteurs caractéristiques issus de l'analyse fréquentielle disposent souvent d'un nombre de dimensions très importantes. Il est primordial de diminuer leur nombre de dimensions pour réduire le temps d'apprentissage des algorithmes de classification. C'est pour cela que l'on couple aux techniques d'extraction de caractéristiques des méthodes d'analyse de données et de quantification vectorielle.

2.3.5.1. Analyse en composante principale *(ACP)*

L'analyse en composante principale est une technique mathématique qui consiste à rechercher les directions de l'espace qui représentent le mieux les corrélations entre n variables aléatoires.

L'ACP permet donc de transformer un vecteur de n paramètres en un ensemble de n valeurs propres avec leur poids permettant de déterminer leur importance. On ne garde que les premières valeurs propres, celles qui expliquent le mieux l'information c'est à dire dont le vecteur propre maximise l'inertie du nuage projeté. Les **k** vecteurs propres choisis (**k<n**) définissent les axes principaux du sous espace dans lequel on projette les n variables du vecteur pour obtenir une représentation compactée appelée composantes principales. On diminue ainsi la dimension des vecteurs caractéristiques qui passe de **n** à **k** composantes tout en minimisant la perte d'information.

Cette technique a été utilisée en reconnaissance de la parole et a permis d'obtenir de bonne performance **[25].**

2.3.5.2. Analyse discriminante linéaire *(ADL)*

L'analyse discriminant linéaire est une méthode statistique qui permet de séparer les classes par minimisation de la distance intraclasses et par maximisation de la distance interclasses. Comme pour l'*ACP*, il est possible de réduire la dimensionnalité de l'espace en éliminant les valeurs propres les plus faibles.

L'espace acoustique est reparti en un ensemble de classe, chaque classe étant représentée par une matrice de covariance. On cherche à trouver les axes qui permettent au mieux d'éloigner les classes des unes des autres. Ces axes sont les vecteurs propres de la matrice de covariance interclasses. Ainsi un sous ensemble de ces axes correspondant aux directions des grandes variances (les grandes valeurs propres) est utilisé pour former les nouveaux vecteurs caractéristiques du signal.

Cette technique a montré de meilleures performances en reconnaissance de la parole que l'analyse en composantes principales **[26].**

2.3.6. Conclusion

L'extraction des paramètres acoustiques est une étape très importante dans les systèmes de reconnaissance automatique du locuteur. Son but essentiel est d'extraire les données pertinentes à l'étape de modélisation statistique, et minimise ainsi les données redondantes et le bruit qui se présentent dans les signaux vocaux. Plusieurs expériences ont montré que les paramètres *LPCC* et *MFCC* donnent de meilleures performances aux systèmes de reconnaissance de la parole. Les expériences ont montré que les coefficients *PLP* permettent d'annuler les différences entre locuteurs et possèdent une meilleure robustesse au bruit. L'analyse RASTA-PLP a pour but d'améliorer la robustesse du système de reconnaissance en milieu bruité. Elle est basée sur l'analyse PLP.

2.4. Modèle de reconnaissance de la parole

La reconnaissance des formes a pour but d'identifier et de classer des formes dont la structure varie. L'étape du choix de la méthode de classification est donc très importante dans la conception d'un système de reconnaissance de la parole.

2.4.1. Comparaison dynamique *(dynamic time warping : DTW)*

Cette méthode est la solution la plus simple à mettre en oeuvre en reconnaissance de la parole. Elle consiste à effectuer une comparaison dynamique entre un vecteur de référence et un vecteur de test. On cherche à apparier l'ensemble des points du premier vecteur avec le second. On associe un poids à chaque appariement possible. L'utilisation de la programmation dynamique permet de trouver l'appariement minimisant ce coût. Les formes références sont obtenues par extraction des caractéristiques. La forme test peut avoir subit des transformations (rotations, translation, changement de taille). On utilise pour cela une méthode de mesure de similarité. La distance finale entre une forme référence et une forme test est calculée comme la somme des distances partielles entre vecteurs de paramètres le long du chemin optimal. Des poids liés à la suppression et à l'insertion d'informations, sont imposés pour calculer le meilleur chemin.

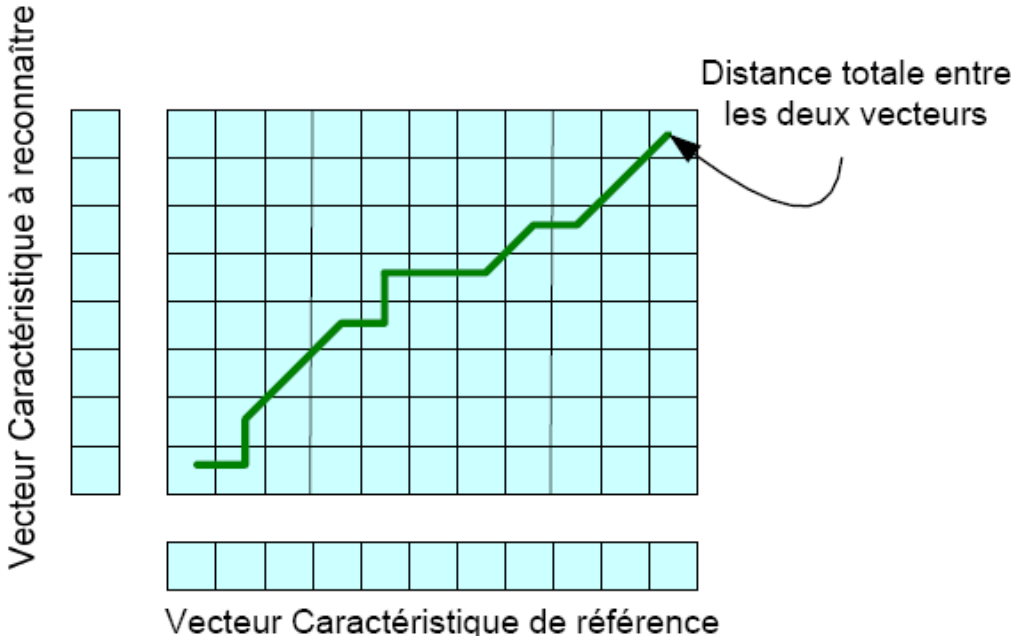


Figure 2.15 Comparaison élastique entre deux vecteurs caractéristiques

Cette méthode a l'avantage d'être simple et de nécessiter que peu de données d'apprentissage. Par contre, elle est limitée par la taille du vocabulaire à reconnaître et à la reconnaissance monolocuteur **[22].**

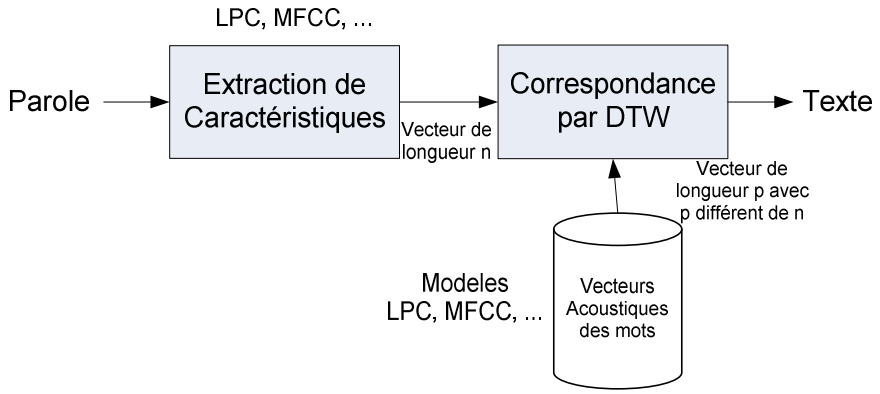
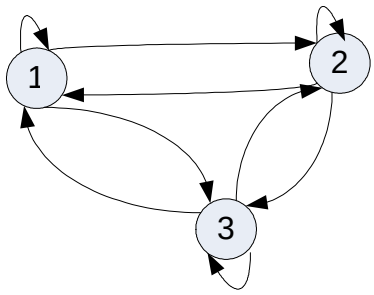


Figure 2.16 Schéma d'un système de reconnaissance basé sur la comparaison dynamique

2.4.2. Modèle de Markov Caché

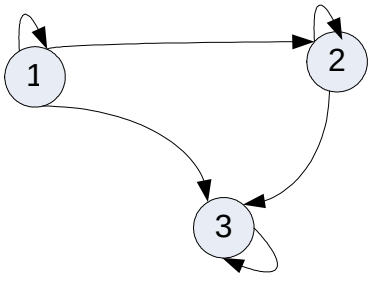
C'est l'une des méthodes les plus utilisée en reconnaissance de la parole. Ce modèle combine les avantages d'une machine à état et des distributions de probabilités. Un modèle de Markov Caché est constitué d'états dont le nombre est à priori inconnu, d'une matrice de

transitions A, d'une matrice d'émissions B et d'un vecteur d'initialisations π. La matrice de transitions A donne les probabilités de transitions entre état. La matrice d'émissions B indique les probabilités d'apparitions de chaque symbole dans chaque état. Le vecteur d'initialisations π indique les probabilités de l'état de départ.



$$A = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{6} & \frac{1}{2} & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{6} & \frac{1}{2} \end{bmatrix} \quad B = \begin{bmatrix} \frac{1}{4} & \frac{3}{4} \\ \frac{1}{6} & \frac{5}{6} \\ \frac{1}{3} & \frac{2}{3} \end{bmatrix} \quad \pi = \begin{bmatrix} \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \end{bmatrix}$$

Le plus souvent on utilise des HMM (*Hidden Markov Models*) dit gauche-droite, c'est à dire qu'il est impossible de revenir à un état précédent. Cette organisation permet de modéliser des contraintes temporelles. Chaque état représente un morceau de signal représenté dans l'espace des paramètres acoustiques par un certain nombre de vecteurs alignés temporellement. Le passage d'un état à un autre s'effectue en tenant compte d'une probabilité de transition d'un état à l'autre.



$$A = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 1 \end{bmatrix} \quad B = \begin{bmatrix} \frac{1}{4} & \frac{3}{4} \\ \frac{1}{6} & \frac{5}{6} \\ \frac{1}{3} & \frac{2}{3} \end{bmatrix} \quad \pi = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$

2.4.2.1. Les problèmes fondamentaux

Il existe trois problèmes fondamentaux liés aux chaînes de Markov Cachés.

Le premier est de retrouver la probabilité qu'une séquence d'observations a été générée par un modèle de Markov caché. Ce problème traite de la reconnaissance, on cherche à calculer la vraisemblance. C'est à dire qu'à partir d'un modèle représentant un mot, il est possible en fonction des caractéristiques en entrée représentées sous forme d'observations de

déterminer quel est le pourcentage de correspondance de la séquence avec le mot. Les algorithmes qui permettent de calculer la probabilité sont des algorithmes de programmation dynamique appelés **Forward et Backward.** A chaque instant et pour chaque état, on calcule la probabilité d'avoir émis un symbole et d'avoir passé une transition. La probabilité d'avoir générée la séquence est donc la somme des probabilités de chaque état.

Le second problème des modèles de Markov Caché est de retrouver la séquence d'états cachés la plus probable pour une séquence d'observation donnée. On utilise pour résoudre ce problème, l'algorithme de **Viterbi** qui est similaire à l'algorithme **Forward**. On ne prend en compte que les transitions entre état qui maximisent la probabilité.

Le dernier problème concerne l'apprentissage des modèles de Markov Cachés. L'apprentissage consiste à déterminer les matrices de transitions, d'émissions et le vecteur d'initialisations. La résolution de ce problème est donnée par l'algorithme de **Baum Welch** qui réestime les probabilités de transitions et d'émissions de l'avant vers l'arrière (**forward-backward**). L'algorithme se termine lorsque la variation du maximum de vraisemblance atteint un certain seuil **[22]**.

1.4.2.2. La reconnaissance de la parole

Pour résumer brièvement, on construit un modèle de Markov Cachés pour tous les mots. Puis, on réalise l'estimation des paramètres du modèle qui maximisent la vraisemblance des vecteurs d'apprentissages pour un mot m.

Lorsque l'on veut reconnaître un mot inconnu, on applique l'algorithme **forward-backward** sur tous les modèles. Le modèle qui maximise la vraisemblance correspond au mot reconnu.

Dans le cadre de la reconnaissance de la parole de mots isolés, les modèles de Markov cachés sont utilisés pour représenter le signal acoustique. Les états peuvent représentés grossièrement un son (phonème). Dans un autre ordre d'idée, les états peuvent représentés les différentes versions de prononciation d'un mot.

Les transitions représentent les différentes possibilités d'enchaîner les sons. Cette intégration de la dimension temporelle dans le modèle explique pourquoi les chaînes de Markov Cachées sont souvent utilisées dans les systèmes de reconnaissance de la parole.

Pour un vocabulaire relativement petit, contenant environ 100 mots, le modèle de Markov Caché pour représenter un mot, possède entre 5 et 10 états et environ 40 observations **[27].**

Pour de grands vocabulaires, l'association d'un modèle de Markov Caché à un mot est impossible car le nombre de modèles et le volume d'apprentissage seraient trop importants. C'est pourquoi la reconnaissance de grands vocabulaires est toujours effectuée à partir de modèles de Markov Cachés d'unité comme le phonème, les diphones ou les triphones. L'approche fréquemment utilisée consiste à prendre un modèle à 3 états par phonème, en faisant l'hypothèse que l'état du milieu modélise la partie stationnaire du phonème et les états extérieurs modélisent la coarticulation avec les phonèmes voisins. Le nombre de phonèmes choisis est généralement de l'ordre de 35 pour le français.

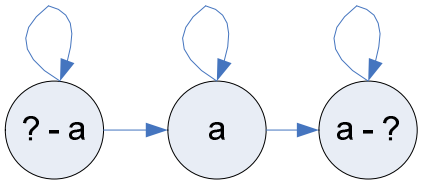


Figure 2.17 Représentation du phonème « a »

Un autre modèle souvent utilisé est le triphone ou l'allophone. Il tient compte des phonèmes suivant et précédent. Le nombre d'allophones dans la langue française est d'environ 7500 et son apprentissage nécessitent de grandes bases de données.

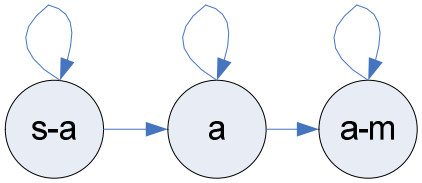


Figure 2.18 Représentation du triphone « s-a-m »

La représentation d'un mot est alors obtenue par la concaténation de plusieurs modèles de Markov Cachés. Ainsi, un long apprentissage est évité et il possible d'ajouter des nouveaux mots en considérant seulement la séquence de triphones.

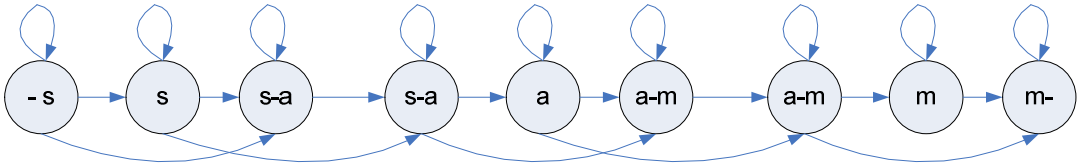


Figure 2.19 Représentation du mot « sam » par concaténation de phonèmes.

Le modèle de Markov caché est très utile pour la reconnaissance de la parole. Il est aussi utile pour les systèmes de reconnaissance automatique du locuteur dépendante de texte. Cependant, due à la complexité de son apprentissage, d'autres méthodes ont été utilisées et qui ont donné de meilleurs résultats **[18].**

2.4.3. Modèle de Mélange de lois Gaussiennes

Les modèles de mélange de Gaussiennes font partis des méthodes de classification paramétrique globale. C'est à dire que la loi de distribution est supposée connue et l'ordre des données n'est pas significatif. Cette méthode de classification est un cas spécifique des modèles de Markov Cachés, c'est le modèle à un état.

Elle repose sur le fait que l'on peut modéliser la distribution des données par une fonction de densité de probabilité en combinant plusieurs fonctions Gaussiennes.

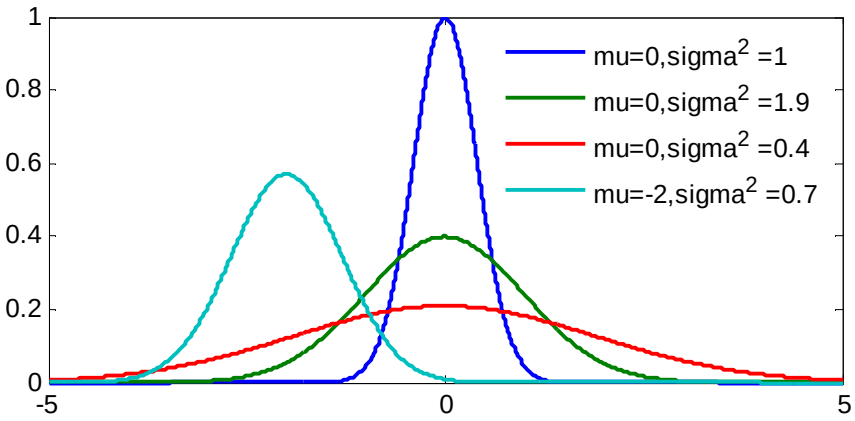


Figure 2.20 Exemple d'un mélange de gaussiennes monodimensionnelle

On considère que les données en entrées, les vecteurs de caractéristiques, sont supposées être des réalisations de variables aléatoires mutuellement indépendantes qui suivent la loi suivante :

$$f(x) = \sum_{i=1}^{M} \pi_i f_i(x) \qquad \textbf{(2.21)}$$

avec **M** : le nombre de composantes, $\boldsymbol{\pi_i}$**, i ∈ {1...M}** : les probabilités de chaque composante et $\mathbf{f_i(x)}$, **i ∈ {1...M}** : les densités de probabilités de chaque composante. Ainsi, on cherche à décomposer une fonction inconnue et à priori complexe f sur un ensemble de fonctions plus simples $\mathbf{f_i}$.

On peut également décrire cette équation comme un modèle dans lequel on suppose que les données sont réparties aléatoirement (et indépendamment les unes des autres) en **M** classes qui sont caractérisées par une distribution différente $\mathbf{f_i}$.

Chaque composante $\mathbf{f_i}$ correspond à des lois normales multidimensionnelles. $\boldsymbol{\mu_i}$ et $\boldsymbol{\Sigma_i}$, **i ∈ {1...M}**, sont les vecteurs moyens et les matrices de covariances de chaque composante. La fonction $\mathbf{f_i}$ de densité de probabilité gaussienne multidimensionnelle est donnée par :

$$f_i(x) = \frac{1}{(2\pi)^{\frac{p}{2}}\sqrt{\Sigma_i}} \exp\left[-\frac{1}{2}(x-\mu_i)^t \Sigma_i^{-1}(x-\mu_i)\right] \qquad \textbf{(2.22)}$$

Où **p** désigne la dimension des vecteurs.

L'estimation des paramètres du modèle se fait généralement au sens du maximum de vraisemblance. On utilise pour cela l'algorithme *EM* (*Expectation-Maximisation*) **[22]**.

2.4.3.1. La reconnaissance de la parole

L'utilisation d'une telle méthode en reconnaissance de la parole se justifie par le fait que l'on peut répartir les vecteurs de caractéristiques en plusieurs classes qui constituent le mélange (son voisé / non voisé, ou plus finement en fonction du phonème).

Elle est utilisée dans les modèles de Markov Cachés dans le cas d'observations continues. Le modèle de Mélange de Gaussiennes est utilisé pour représenter le vecteur caractéristique à l'intérieur d'un état. La probabilité d'émission d'une observation O au sein d'un état j devient donc :

$$b_j(O) = f(x) = \sum_{i=1}^{M} \pi_i f_i(x) \qquad \textbf{(2.23)}$$

Concrètement, on calcule le modèle de Mélange de Gaussiennes pour chaque coefficient des vecteurs caractéristiques associés à un état. Par exemple, le modèle de Markov Caché dans le cas d'observations continues en considérant qu'un seul coefficient peut être modélisé de cette manière :

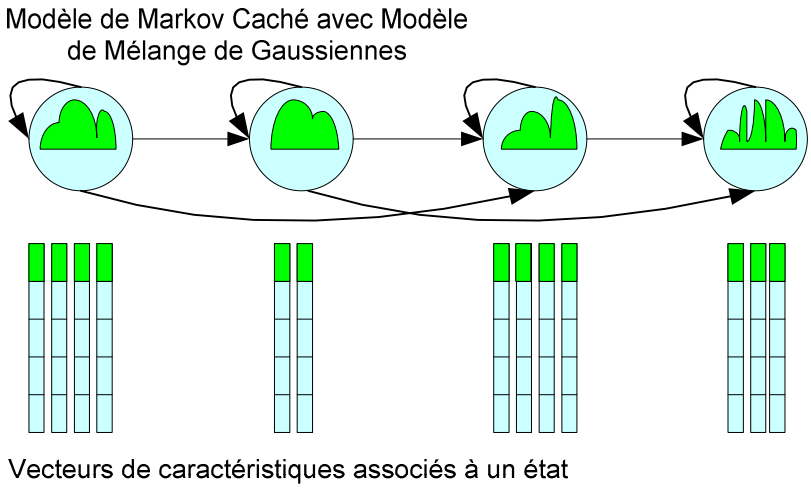


Figure 2.21 Modèle de Markov Caché en cas d'observations continues

2.4.4. Réseau de neurones

Un réseau de neurones est un ensemble d'éléments simples, appelés neurones formels et reliés les uns avec les autres, qui se transmettent l'information par l'intermédiaire de ces liens ou connexions. Chaque neurone réalise une somme pondérée des valeurs de ses entrées. Sa sortie est une modulation de cette somme.

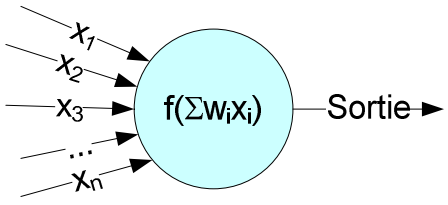


Figure 2.21 Principe du neurone artificiel

Comme nous pouvons le voir sur la figure 2.21, pour n entrées, la sortie est donnée par la formule générale suivante :

$$S_i = f_{act}\left(\sum_{i=1}^{n} w_i x_i \right) \qquad \textbf{(2.24)}$$

La fonction f_{act} est appelée fonction d'activation, elle génère en général des valeurs entre [0, 1]. Une fonction d'activation très courante est la fonction sigmoïde :

$$f_{act}(E) = \frac{1}{1 + e^{-E}},\ 0 \leq f_{act}(E) \leq 1 \qquad \textbf{(2.25)}$$

2.4.4.1. Réseau de neurones multicouche

Un réseau de neurones classiquement utilisé est le perceptron. Il appartient à la catégorie des classifieurs à apprentissage supervisé. Ils sont définis à partir d'une répartition des neurones en couche. Les sorties des neurones de la couche i forment les entrées de la couche i+1.

Généralement les neurones d'entrées du perceptron correspondent aux coefficients du vecteur caractéristique et les neurones de sorties donnent la classe d'appartenance. La figure suivante montre un perceptron à trois couches dont une couche cachée, appliqué à un problème de reconnaissance globale de mots :

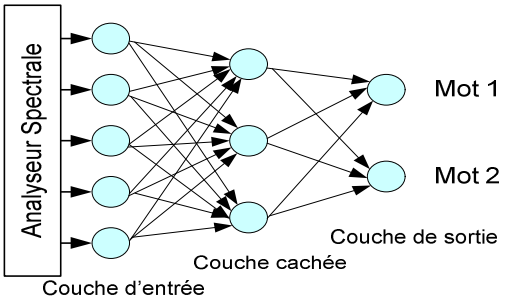


Figure 2.22 Structure d'un perceptron à trois couches

Ce type de classification est souvent employé pour la fusion d'informations issues de plusieurs sous systèmes.

2.4.4.2. Réseau Multicouche à retard

La parole est un processus dynamique qui évolue dans le temps. Cependant, les réseaux de neurones de base n'intègrent pas la dimension temporelle des données.

Les réseaux multicouches à retard sont des perceptrons où l'on a introduit des retards temporels fixes sur les entrées. A chaque instant, chaque neurone traite une coordonnée du vecteur qui se présente sur la couche d'entrée ainsi qu'une coordonnée des n vecteurs précédents. L'apprentissage des poids des différentes connexions est effectué à l'aide d'un algorithme dérivé de l'algorithme de rétro propagation du gradient d'erreur.

2.4.4.3. Réseau Récurrent

Les réseaux de neurones récurrents sont caractérisés par le fait que les entrées d'un neurone peuvent être aussi bien des entrées que des sorties d'autres neurones.

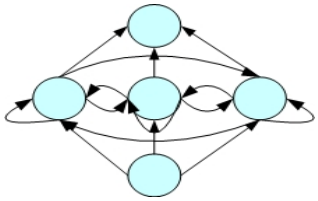


Figure 2.23 Réseaux récurrents à trois couches

Les réseaux de neurones récurrents sont connus pour avoir produit un système considéré comme l'un des meilleurs systèmes de reconnaissances de la parole **[28]**. **Robinson** améliora son système jusqu'en 1994, ainsi il reporte une précision de reconnaissance des phonèmes de 80%. Son système sans optimisation possède un taux de reconnaissance de 70,4%. Le système utilise un réseau de neurone récurrent à délai. La spécificité de ces systèmes est de posséder des connexions entre les neurones avec des délais différents. Ainsi, les futures trames sont prises en compte avant qu'une décision ne soit faite. De plus, les précédentes prédictions sont replacées en entrée du réseau pour ajouter une information de contexte. Le réseau de neurones combiné avec plusieurs métriques pour l'évaluation des prédictions, permet de créer un classifieur efficace pour la gestion de la parole en données segmentées. Il a été étendu pour incorporer les connaissances du langage et permet de rivaliser avec les systèmes commerciaux.

En 1998, **Chen** et **Jamieson** ont repris une approche identique à celle de **Robinson** mais en introduisant un nouveau critère qui maximise la précision de la classification des

trames. Le taux de reconnaissance des phonèmes observés est de 79,9%, c'est à dire un peu moins bien que le précédent mais possède un taux de classification des trames de 74,2% ce qui représente l'un des meilleurs taux de reconnaissance à l'heure actuelle.

2.4.4.4. La reconnaissance de la parole

Les réseaux de neurones disposent de propriétés qui sont intéressantes du point de vue de la reconnaissance de la parole.

Ils peuvent apprendre des fonctions fortement non linéaires qui permettent de reconnaître une classe de forme et, simultanément, de rejeter les autres classes. Ils constituent des approximateurs de fonctions très puissants.

De plus, ils ne nécessitent aucune donnée statistique à l'initialisation contrairement aux modèles de Markov Cachés.

Enfin, ils présentent des structures parallèles régulières facilement implantables sous forme logicielle et matérielle **[22]**.

2.4.5. Machines à vecteurs de support (SVM)

Les SVM sont des classifieurs basés sur la théorie de l'apprentissage statistique supervisé, c'est-à-dire que pour la classification il faut fournir, préalablement, un ensemble d'entraînement à l'SVM. Le principe des SVM est de séparer les données par un hyperplan. Une description plus détaillée est donnée dans le **3eme** chapitre.

Les SVM présentent des résultats meilleurs par comparaison avec d'autres méthodes comme le « maximum likelyhood » ou les réseaux neuronaux **[29].** Dans le contexte de classification d'image hyperspectrale, des études ont montré l'efficacité des SVM sans nécessiter une réduction préalable de l'ensemble de caractéristiques **[29].**

Les SVM ont été appliqués avec succès sur plusieurs problèmes pratiques : l'imagerie biomédicale, la compression d'image ou la reconnaissance d'objets 3D. L'intérêt des SVM par rapport aux autres approches est dû, principalement, à quatre raisons **[30]** :

- Méthode non linéaire.
- Le problème d'apprentissage se réduit à la résolution d'un problème d'optimisation
- quadratique convexe (question très traitée dans la littérature).
- Robustesse à la dimension des données utilisation d'un ensemble d'entraînement de petite taille

2.4.6. Comparaison : modèles utilisés en reconnaissance automatique de la parole

Chaque modèle de reconnaissance de la parole pressente des avantages et inconvénients. Le tableau suivant résume les points forts et les limites rencontrées, pendant l'utilisation de tel modèle.

Modèles utilisés en RAP	Avantages	Inconvénients
DTW (dynamic time waping)	- L'algorithme de DTW est rapide, bien adapté à la parole parce que capable de tenir compte des variations temporelles du signal. Il ne nécessite pas beaucoup de données pour fonctionner correctement	- La DTW est très sensible à la segmentation du signal. En effet, si le point de départ du calcul dynamique n'est pas bon, l'algorithme peut rapidement diverger du chemin optimum.
HMM (Hidden Markov Model)	-Cette technologie offre des algorithmes performants pour l'apprentissage et la reconnaissance, grâce auxquels les HMMs se sont avérés les mieux adaptés aux problèmes de la reconnaissance de la parole.	-les modèles de Markov Cachés, présentent certaines limites et difficultés, qui résident essentiellement dans le choix d'un bon modèle initial pour l'apprentissage, qui est généralement aléatoire, ce qui conduit souvent à un optimum local.
GMM (Gaussians Mixture Model)	-L'utilisation d'un mélange de plusieurs densités gaussiennes multidimensionnelles a permet de donner une très bonne représentation des vecteurs acoustiques -L'utilisation du modèle GMM permet d'estimer fidèlement des densités de probabilités aléatoires telles que celle des vecteurs acoustiques. -Le temps d'apprentissage et relativement petit par rapport à d'autre modèle tel que le modèle HMM.	- Bien qu'ils soient capables de capturer les informations à plus long terme d'un locuteur, ils ne contiennent pas d'aspects dynamiques. Pour une bonne modélisation (i.e. beaucoup de Gaussiennes) nécessitent beaucoup de données.
ANN (*Réseau de neurones*)	- Les réseaux de neurones se révèlent utiles pour la classification de formes statiques	- Les réseaux de neurones ne peuvent modéliser une évolution temporelle à long terme. Ils sont donc mal adaptés au traitement de signaux séquentiels tel que la parole
SVM *(Support Vector Machines)*	- Le grand avantage, par rapport aux autres techniques, est la capacité à généraliser la classification. - Cette méthode est adaptée aux applications présentant une grande variation intra-classes. - Les SVM ont un comportement plus performant pour un ensemble d'entraînement très réduit	- L'inconvénient des SVM est le choix empirique de la fonction noyau adaptée au problème - Un deuxième inconvénient est le temps de calcul qui croît de façon cubique en fonction du nombre de données à traiter

Tableau 2.2 Comparaison : modèles utilisés en RAP

2.4.7. Conclusion

Dans un système de reconnaissance automatique de la parole, les paramètres acoustiques sont utilisés pour estimer un modèle, qui peut être statistique ou basé sur le calcul d'une distance euclidienne.

Nous avons présenté les différents types des systèmes de reconnaissance automatique de la parole. Identifier ce qui est dit un locuteur est une tâche très importante. En exploitant les avantages des systèmes RAP, nous pouvons concevoir des systèmes de reconnaissance de la parole très performants. Dans le chapitre suivant, nous étudierons les Machines à Vecteurs Support, ainsi que la problématique de classification lorsque nous avons plusieurs classes. Finalement, la méthode pour sélectionner les paramètres du SVM est présentée.

CHAPITRE : 3

Les Support Vector Machines (SVM)

3.1. Introduction

Parmi les méthodes à noyaux, inspirées de la théorie statistique de l'apprentissage de Vladimir Vapnik, les SVM constituent la forme la plus connue. SVM est une méthode de classification binaire par apprentissage supervisé, elle fut introduite par Vapnik en 1995. Cette méthode est donc une alternative récente pour la classification. Cette méthode repose sur l'existence d'un classificateur linéaire dans un espace approprié. Puisque c'est un problème de classification à deux classes, cette méthode fait appel à un jeu de données d'apprentissage pour apprendre les paramètres du modèle. Elle est basée sur l'utilisation de fonctions dites noyau (kernel) qui permettent une séparation optimale des données. Dans la présentation des principes de fonctionnements, nous schématiserons les données par des « points » dans un plan.

3.2. Machine à vecteur Support et Kernel Machines

Les *SVM* sont des classifieurs binaires et statistique. C'est à dire qu'ils permettent de créer une surface de décision entre deux classes définies dans un même espace. Pour cela, ils construisent une frontière de décision par projection des caractéristiques provenant d'un espace d'origine dans un espace de caractéristiques de dimension supérieure (voir infini) dans le but de rendre les classes linéairement séparables.

L'hyperplan choisi est celui qui maximise la marge de séparabilité entre les deux ensembles de données. La sélection de l'hyperplan dans un espace de caractéristiques nécessite d'évaluer un produit scalaire dans cet espace. Ce qui peut être très coûteux en temps et en complexité si l'espace est de très grande dimension. Heureusement, ce calcul n'est pas obligatoire grâce à une opération mathématique appelé kernel. Le kernel calcule le produit scalaire de deux points dans l'espace de dimension supérieur sans avoir à les projeter.

De plus, cette technique n'est pas appliquée seulement au *SVM*, mais aux groupes des machines à apprentissage appelés kernel machines. Toutes utilisent des kernels pour appliquer les données d'un espace d'entrée en un espace de caractéristiques de dimension supérieur. Ils sont capables de créer une séparation non linéaire d'un espace d'entrée. Il existe de nombreuses kernel machines, les plus connues *Relevant Vector Machines, Bayes Point Machine, Gaussian processes, Least Squares SVM,* méthodes de *Boosting* et *Kernel Fisher Discriminant* **[22].**

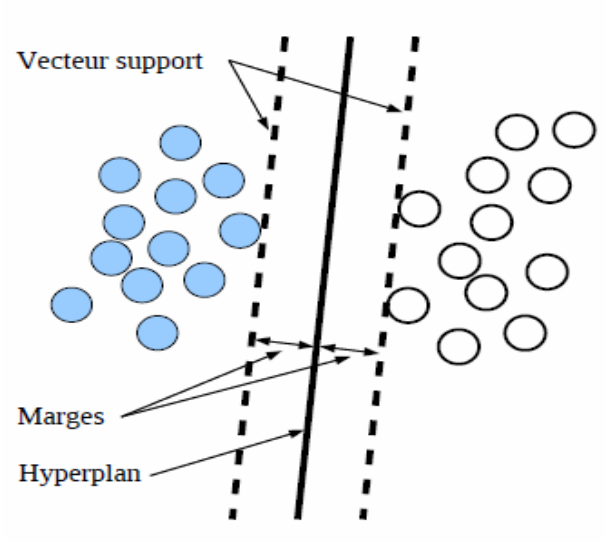


Figure 3.1 Séparation linéaire dans un espace à deux dimensions.

3.2. Apprentissage statistique et SVM

Les *SVM* sont des classifieurs qui réalisent un apprentissage statistique. Comme il est dit plus haut, les *SVM* sont des méthodes de résolutions de problèmes de classification binaires. Les performances de ces classifieurs sont calculées en fonction de l'erreur de classification. Si l'échantillon est correctement classé, son erreur vaut 0 sinon elle vaut 1.

Pour l'apprentissage, on cherche donc à minimiser empiriquement l'erreur (*Empirical Risk Minimisation* : *EMP*), c'est à dire on minimise la somme des erreurs de classifications pour chaque échantillon. Cependant, un des problèmes avec cette méthode d'apprentissage est que si la complexité du classifieur est élevée, il y a un risque de sur apprentissage des données.

Pour en prendre en compte la complexité du système d'apprentissage, on minimise le risque structurel (*Structural Risk Minimisation* : *SRM*). Dans cette approche, on prend en compte la probabilité de classer correctement un échantillon : **P(x,y)**. Cependant, cette probabilité est la plupart du temps, inconnue. Dans ce cas, le risque structurel est exprimé par le risque empirique additionné à la dimension **Vapnik-Chervonenkis** (*VC*) du système d'apprentissage **[31].** En généralisation, la dimension, VC est une borne supérieure de l'erreur commise par le système. Elle représente la dimension, dans laquelle tous les échantillons sont correctement assignés. De plus, l'algorithme d'apprentissage *SMO* détermine les vecteurs supports de manière optimale.

3.3. SVM principe de fonctionnement général

3.3.1. Notions de base: Hyperplan, marge et support vecteur

Pour deux classes d'exemples donnés, le but de SVM est de trouver un classificateur qui va séparer les données et maximiser la distance entre ces deux classes. Avec SVM, ce classificateur est un classificateur linéaire appelé hyperplan.

Dans le schéma qui suit, on détermine un hyperplan qui sépare les deux ensembles de points **[32].**

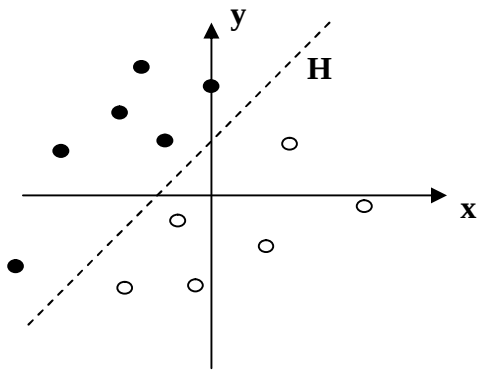


Figure 3.2 Exemple d'un hyperplan séparateur **[32].**

Les points les plus proches, qui seuls sont utilisés pour la détermination de l'hyperplan, sont appelés vecteurs de support.

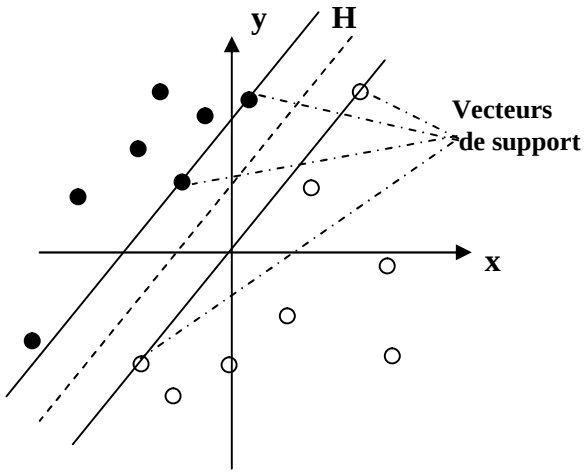


Figure 3.3 Exemple de vecteurs de support **[33]**.

Il est évident qu'il existe une multitude d'hyperplan valide mais la propriété remarquable des SVM est que cet hyperplan doit être optimal. Nous allons donc en plus chercher parmi les hyperplans valides, celui qui passe « au milieu » des points des deux classes d'exemples. Intuitivement, cela revient à chercher l'hyperplan le « plus sûr ». En effet, supposons qu'un exemple n'ait pas été décrit parfaitement, une petite variation ne modifiera pas sa classification si sa distance à l'hyperplan est grande. Formellement, cela revient à chercher un hyperplan dont la distance minimale aux exemples d'apprentissage est maximale **[34]**.

On appelle cette distance « marge » entre l'hyperplan et les exemples. L'hyperplan séparateur optimal est celui qui maximise la marge. Comme on cherche à maximiser cette marge, on parlera de séparateurs à vaste marge.

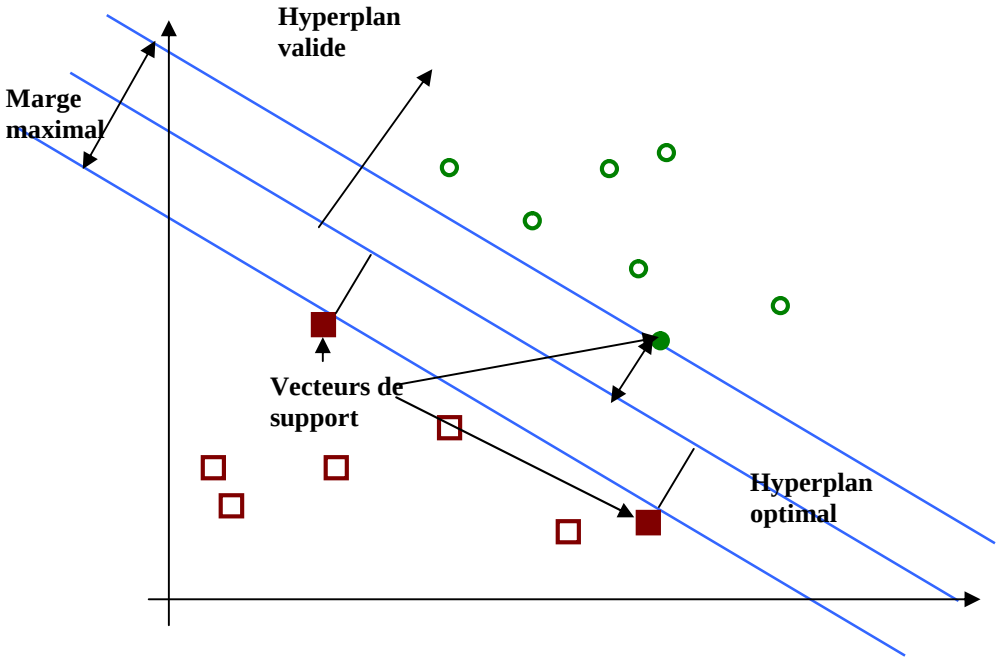


Figure 3.4 Exemple de marge maximal (hyperplan valide) **[34]**.

3.3.2. Pourquoi maximiser la marge ?

Intuitivement, le fait d'avoir une marge plus large procure plus de sécurité lorsque l'on classe un nouvel exemple. De plus, si l'on trouve le classificateur qui se comporte le mieux vis-à-vis des données d'apprentissage, il est clair qu'il sera aussi celui qui permettra au mieux de classer les nouveaux exemples. Dans le schéma qui suit, la partie droite nous montre

qu'avec un hyperplan optimal, un nouvel exemple reste bien classé alors qu'il tombe dans la marge. On constate sur la partie gauche qu'avec une plus petite marge, l'exemple se voit mal classé **[32].**

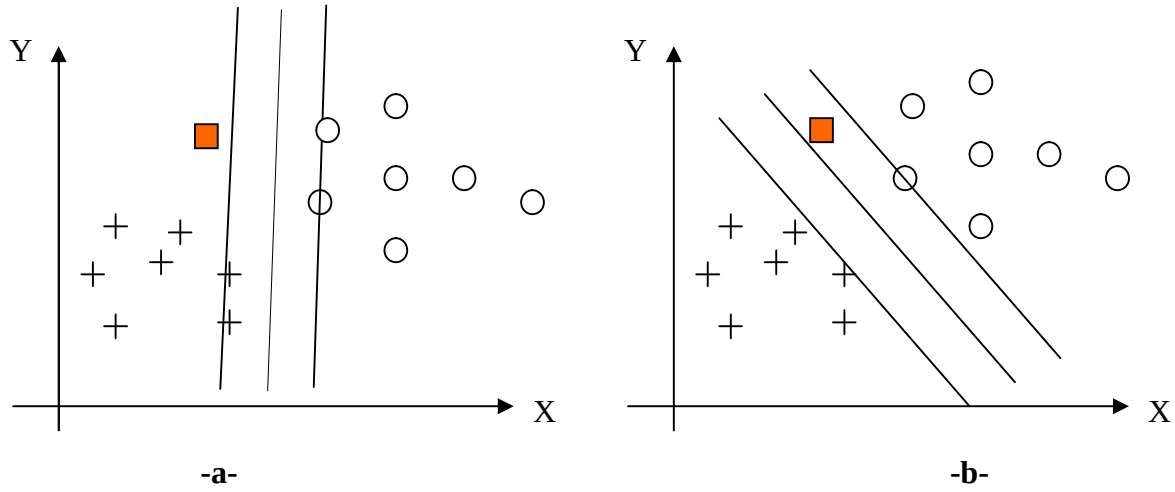


Figure 3.5 a) Hyperplan avec faible marge, **b)** Meilleur hyperplan séparateur **[32].**

En général, la classification d'un nouvel exemple inconnu est donnée par sa position par rapport à l'hyperplan optimal. Dans le schéma suivant, le nouvel élément sera classé dans la catégorie des « + ».

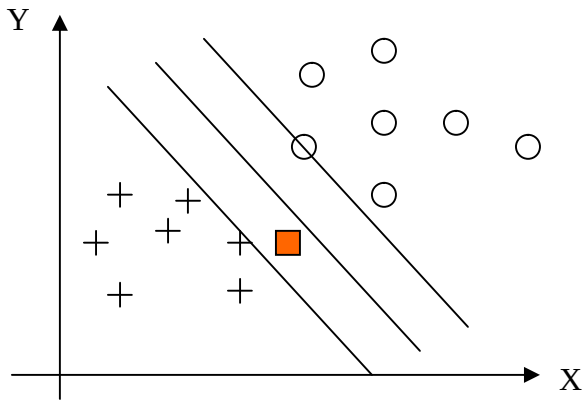


Figure 3.6 Exemple de classification d'un nouvel élément **[32].**

3.3.3. Linéarité et non-linéarité

Parmi les modèles des SVM, on constate les cas linéairement séparable et les cas non linéairement séparable. Les premiers sont les plus simples de SVM car ils permettent de

trouver facilement le classificateur linéaire. Dans la plupart des problèmes réels il n'y a pas de séparation linéaire possible entre les données, le classificateur de marge maximale ne peut pas être utilisé car il fonctionne seulement si les classes de données d'apprentissage sont linéairement séparables **[32].**

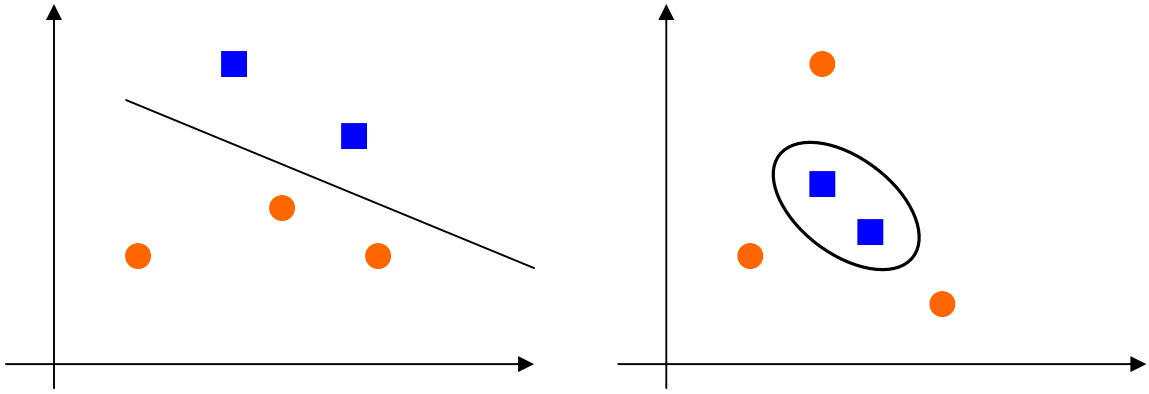


Figure 3.7 **a)** Cas linéairement séparable,**b)** Cas non linéairement séparable **[32].**

3.3.4. Cas non linéaire

Pour surmonter les inconvénients des cas non linéairement séparable, l'idée des SVM est de changer l'espace des données. La transformation non linéaire des données peut permettre une séparation linéaire des exemples dans un nouvel espace. On va donc avoir un changement de dimension. Cette nouvelle dimension est appelé « espace de re-description ».En effet, intuitivement, plus la dimension de l'espace de re-description est grande, plus la probabilité de pouvoir trouver un hyperplan séparateur entre les exemples est élevée. Ceci est illustré par le schéma suivant

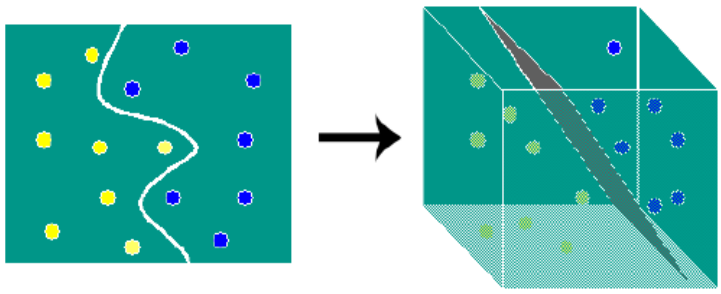


Figure 3.8 Exemple de changement de l'espace de données **[32].**

On a donc une transformation d'un problème de séparation non linéaire dans l'espace de représentation en un problème de séparation linéaire dans un espace de re-description de

plus grande dimension. Cette transformation non linéaire est réalisée via une fonction noyau. En pratique, quelques familles de fonctions noyau paramétrables sont connues et il revient à l'utilisateur de SVM d'effectuer des tests pour déterminer celle qui convient le mieux pour son application. On peut citer les exemples de noyaux suivants : polynomiale, gaussien, sigmoïde et laplacien **[32].**

3.4. Fondements mathématiques

Nous allons détailler dans les paragraphes ci-dessous les principes mathématiques sur lesquels repose SVM.

3.4.1. Transformation des entrées

Le principe des SVM consiste à projeter les données de l'espace d'entrée X (appartenant à deux classes différentes) non-linéairement séparables dans un espace Ψ de plus grande dimension appelé espace de caractéristiques de façon à ce que les données deviennent linéairement séparables. Dans cet espace, on construit, un hyperplan optimal séparant les classes tel que :

- Les vecteurs appartenant aux différentes classes se trouvent de différents côtés de l'hyperplan.
- La plus petite distance entre les vecteurs et l'hyperplan (la marge) soit maximale **[35].**

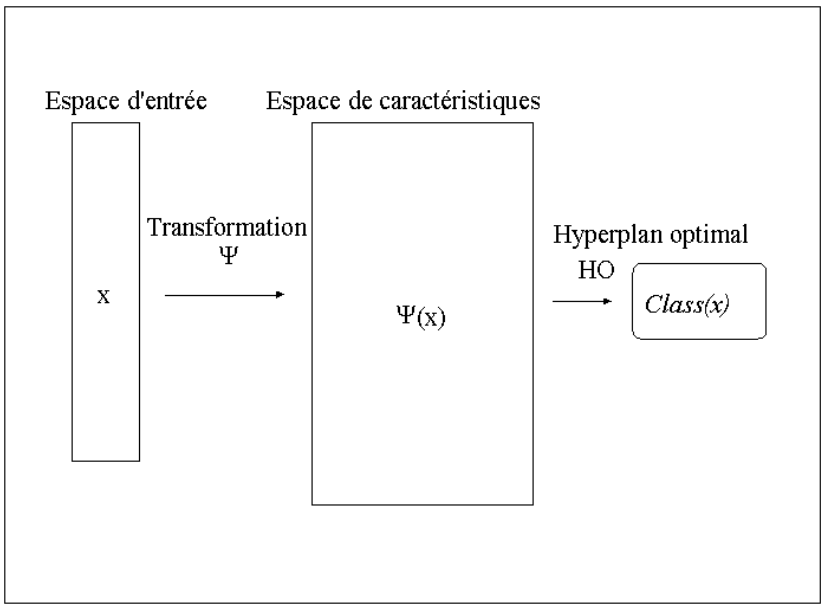


Figure 3.9 Principe des techniques SVM **[35].**

La figure 3.9 représente le principe de la technique SVM.

Dès lors, il s'agit de choisir l'hyperplan optimal qui classifie correctement les données (Lorsque c'est possible) et qui se trouve le plus loin possible de tous les points à classer.

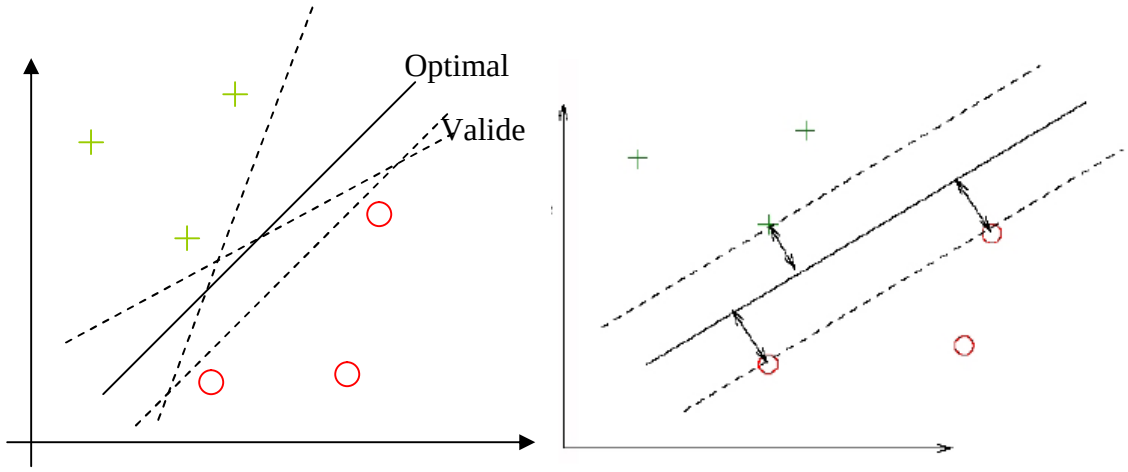


Figure 3.10 Exemple de recherche d'un hyperplan optimal **[36]**.

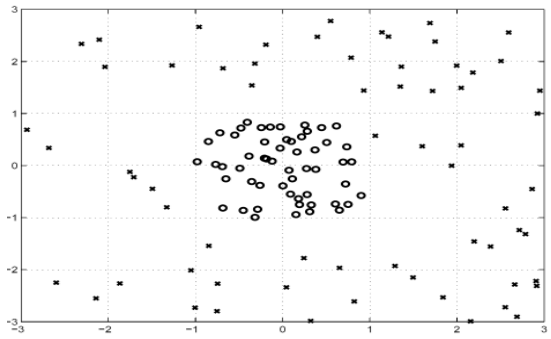
Mais l'hyperplan séparateur choisi devra avoir une marge maximale.

Exemple :

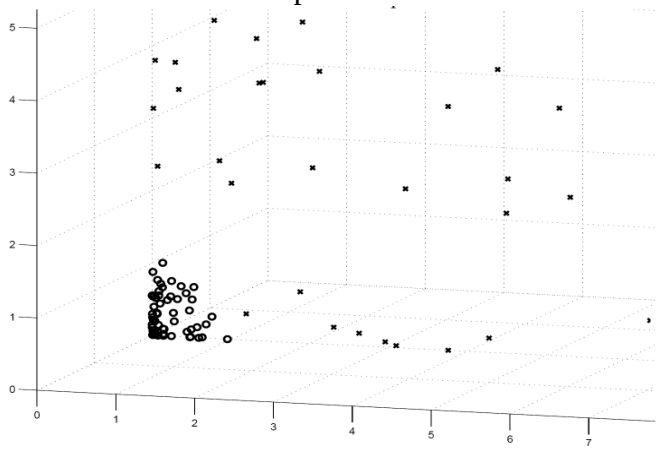
Pour avoir une idée plus claire sur les SVM, voici un exemple inspiré du travail de **[37]** qui met en pratique le principe des SVM. Dans cet exemple, les données non-linéairement séparables dans R^2 deviennent linéairement séparables dans R^3 grâce à la transformation Ψ définie par :

$$\Psi : R^2 \longrightarrow R^3$$
$$(x_1 , x_2) \longrightarrow (x_1^2 , 2^{0.5} x_1.x_2 , x_2^2)$$

La figure 3.10 représente une simulation de cette transformation que nous avons, réalisée avec Matlab. Les données de la figure 3.11.a ont été tirées aléatoirement dans R^2 et la figure 3.11.b représente l'image de ces données dans R^3 suivant la transformation Ψ **[35].**



a : exemples tirés aléatoirement dans R^2 appartenant à deux classes non-linéairement séparables



b : l'image des exemples de la figure 3.11 (a) dans R^3 en utilisant la transformation Ψ

Figure 3.11 Exemple montrant l'efficacité d'une transformation dans un espace de plus grande dimension pour faciliter le classement **[35].**

3.4.2. Le classifieur linéaire

Les SVM font partie de la famille d'algorithmes de classification supervisé : étant donné un ensemble d'échantillons d'entraînement, chacun appartenant à une classe, le SVM cherche la fonction qui assigne chaque échantillon à sa classe correspondante. L'objectif de la théorie d'apprentissage statistique est de chercher une certaine fonction qui classifie de façon satisfaisante les échantillons qui n'ont pas étés utilisés pour l'entraînement, c'est-à-dire, que l'erreur de classification soit minimal **[30]**.

Pour le cas de la classification linéaire, on considère un ensemble d'entraînement de N vecteurs dans un espace de caractéristiques de **d** dimensions $x_i \in \mathbb{R}^d$ (i=1,.., N) avec $y_i \in \{-1,1\}$ associé à chaque vecteur x_i. Si les deux classes sont linéairement séparables, on

peut trouver un hyperplan (une surface linéaire) définie par le vecteur $w \in \mathbb{R}^d$ (vecteur normal au hyperplan) et un biais **b** $\in \mathbb{R}$ capable de séparer les deux classes sans erreur figure 3.12.

On définit, une fonction de décision linéaire, *f*, associée à l'hyperplan :

$$f(x) = \langle w, x \rangle + b \qquad \textbf{(3.1)}$$

Pour décider à quelle classe appartient un échantillon, il suffit d'évaluer le signe de cette fonction sgn [*f(x)*].

L'objectif, donc, est d'estimer w et b pour que

$$y_i(\langle w, x_i \rangle) + b) \geq 1, \quad i = 1,..., N \qquad \textbf{(3.2)}$$

Plusieurs solutions sont en général possibles, un critère permettant de choisir la meilleure est la marge géométrique, qui est la distance entre deux classes et vaut 2 / || *w* || La grandeur de cette marge est proportionnelle à la capacité de généralisation des SVM. Le plus large est la marge, plus performant sera le classifieur : l'erreur de classification sera minimal.

Ainsi, il faut maximiser 2 / || *w* ||, ou ce qui est équivalent, minimiser || *w* || / 2. Cela nous amène au problème d'optimisation quadratique suivant :

$$\begin{aligned} &\text{minimiser} \quad \frac{\|w\|^2}{2} \\ &\text{sous la contrainte} \quad y_i(\langle w, x_i \rangle) + b) \geq 1, \quad i = 1,..., N \end{aligned} \qquad \textbf{(3.3)}$$

Cette optimisation peut se transformer, grâce aux multiplieurs de Lagrange, en cette autre formulation **[30]** :

$$\begin{aligned} &\text{minimiser} \quad \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle \\ &\text{sous la contrainte} \quad \sum_{i=1}^{N} \alpha_i y_i = 0 \quad \text{et } \alpha_i \geq 0, \quad \text{i=1,…,N} \end{aligned} \qquad \textbf{(3.4)}$$

La résolution de ce problème d'optimisation se fait par méthodes de programmation quadratique. Des détails sur ce sujet peuvent se trouver dans **[38]** et **[39].**

Finalement, la fonction de discrimination associée à l'hyperplan optimal devient une équation dépendante des multiplieurs de Lagrange et des échantillons d'entraînement :

$$f(x) = \sum_{i \in S} \alpha_i y_i \langle x_i, x \rangle + b \qquad \textbf{(3.5)}$$

Où *S* est le sous-ensemble d'échantillons d'entraînement correspondant aux multiplieurs de Lagrange α_i's non nuls.

Les contraintes supposent que les données sont linéairement séparables, mais pour des ,applications réelles, cela n'est pas toujours vrai. Pour traiter les données non-linéairement séparables, l'introduction d'une certaine marge d'erreur est nécessaire, ce qui change l'équation (3.2) :

$$\begin{aligned} & y_i(\langle w, x_i \rangle) + b) \geq 1 - \xi_i, \quad i = 1,..., N \\ & \xi_i \geq 0 \qquad i = 1,..., N \end{aligned} \qquad \textbf{(3.6)}$$

Où les ξ_i sont les variables introduites pour neutraliser l'effet de la non-séparabilité des données. Le nouveau problème d'optimisation change légèrement et dévient :

$$\text{minimiser} \qquad \left[\frac{\|w\|^2}{2} + C \sum_{i=1}^{N} \xi_i \right] \qquad \textbf{(3.7)}$$

sous la contrainte $\quad y_i(\langle w, x_i \rangle) + b) \geq 1 - \xi_i, \quad \xi_i \geq 0$

Où la constante *C* contrôle le taux de pénalité et devient un paramètre du classifieur. De nouveau, ce problème d'optimisation se résout par programmation quadratique après la transformation par méthodes de Lagrange **[30]**.

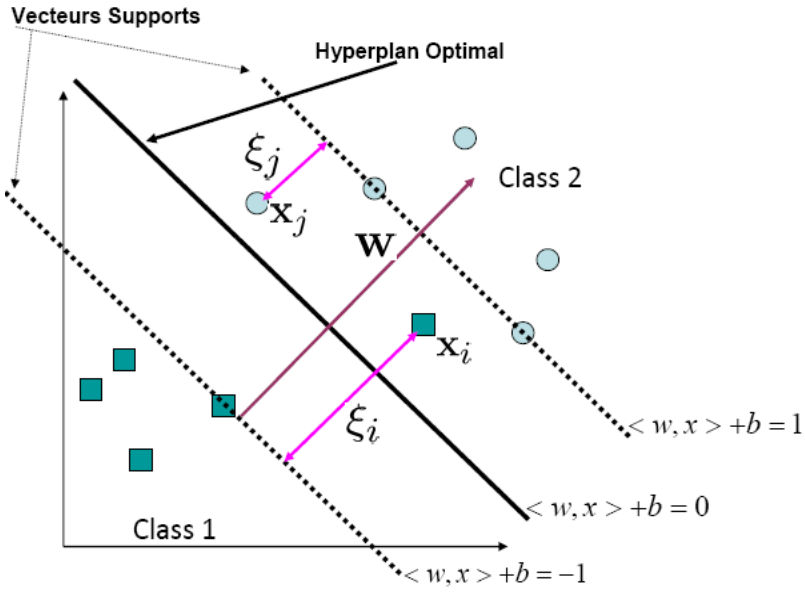


Figure 3.12 Hyperplans séparateurs dans le cas de données linéairement non séparables

3.4.3. Le classifieur non-linéaire

Pour améliorer les performances de classifications présentées ci-dessus il faut faire appel aux SVM non-linéaires. La base des SVM non-linéaires est l'utilisation des fonctions noyaux afin de projeter les données d'entrée sur un plus grand espace des caractéristiques pour pouvoir séparer les nouvelles variables par des hyperplans. Le choix du noyau ce fait à priori et a de l'influence sur les résultats de classification **[30]**.

On peut penser la transformation comme une fonction $\Psi(x)$, non-linéaire qui projette les données sur un espace $\mathbb{R}^{d'}$ (avec $d' > d$). Alors, la formulation faite pour l'espace à dimension d, doit se faire pour l'espace à dimension d', c'est-à-dire, que le produit scalaire $<x_i, x_j>$ se transforme en $\langle \Psi(x_i), \Psi(x_j) \rangle$. A ce point-là, le problème principal consiste en le calcul de la fonction $\Psi(x)$, qui peut supposer un coût calculatoire très grand et qui parfois est incalculable **[30]**

L'utilisation des fonctions noyau résout ce problème, car elles s'expriment selon :

$$K(x_i, x_j) = \langle \Psi(x_i), \Psi(x_j) \rangle \quad \textbf{(3.8)}$$

Ainsi, le problème est simplifié, car on évite le calcul du produit scalaire dans l'espace transformé $\langle \Psi(x_i), \Psi(x_j) \rangle$. Le problème d'optimisation s'exprime

$$\text{minimiser} \quad \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$

$$\text{sous la contrainte} \quad \sum_{i=1}^{N} \alpha_i y_i = 0 \text{ et } 0 \leq \alpha_i \leq C \qquad \textbf{(3.9)}$$

Le résultat final est la fonction *f(x)* discriminante exprimée en fonction des données originales dans l'espace de caractéristiques de dimension d

$$f(x) = \sum_{i \in S} \alpha_i y_i K(x_i, x) + b \qquad \textbf{(3.10)}$$

Pour plus de détails sur les fonctions noyau se trouvent dans **[40].**

Exemple

Prenons le cas trivial ou x = (x_1; x_2) dans $\mathbb{R}^2$ et $\Psi(x) = (x_1^2; 2^{½} x_1 x_2; x_2^2)$ est explicite. Dans ce cas, $\mathbb{R}^{d'} = \mathbb{R}^3$ est de dimension 3 et le produit scalaire s'écrit :

$$\begin{aligned} \langle \Psi(i), \Psi(y) \rangle &= x_1^2 y_1^2 + 2x_1 x_2 y_1 y_2 + x_2^2 y_2^2 \\ &= (x_1 y_1 + x_2 y_2)^2 \\ &= (x, y)^2 \\ &= K(x, y) \end{aligned}$$

Le calcul du produit scalaire dans $\mathbb{R}^{d'}$ ne nécessite pas l'évaluation explicite de Ψ .D'autre part, le plongement dans $\mathbb{R}^{d'} = \mathbb{R}^3$ peut rendre possible la séparation linéaire de certaines structures de données figure 3.13.

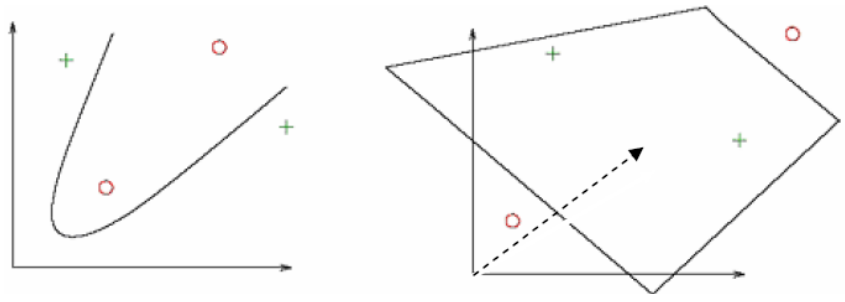


Figure 3.13 Rôle de l'espace intermédiaire dans la séparation des données.

3.4.3.1. Exemples de noyaux

- ***Linéaire*** $K(x, y) = \langle x, y \rangle$

- ***Polynomial*** $K(x, y) = (\langle x, y \rangle + q)^p$

où d est le degré du polynôme à déterminer par l'utilisateur.

- ***Le noyau RBF (Radial Basis Function)*** $K(x, y) = e^{-\|x-y\|^2/2\sigma^2}$

où σ est à déterminer et $\gamma = 1/2\sigma^2$.

Beaucoup d'articles sont consacres à la construction d'un noyau plus ou moins exotique et adapté à une problématique posée : reconnaissance de séquences, de caractères, l'analyse de textes... La grande flexibilité dans la définition des noyaux, permettant de définir une notion adaptée de similitude, confère beaucoup d'efficacité à cette approche à condition bien sur de construire et tester le bon noyau. D'ou apparaît encore l'importance de correctement évaluer des erreurs de prévision par exemple par validation croisée **[41]**.

3.4.4. Le classifieur multi-classe

Les SVM étant des classificateurs binaires, c'es-tà-dire, un problème à deux classes, la résolution d'un problème multiclasses est réalisée en le transformant en une combinaison de problèmes binaires **[45]**, ceci restant cependant un domaine de recherche très ouvert. Jusqu'à maintenant la meilleure méthode de construire SVM multiclasse n'est pas claire [**43**]. Schölkopf et al. [**42**] ont proposé un classificateur de type « un contre tous ». Clarkson et Moreno [**43**] ont proposé un classifieur de type «un contre un". Leurs structures sont présentées dans les figures 3.14 et 3.15.

Les deux types de classifieurs sont au fait de combinaisons des classifieurs binaires à base des sous-classifieurs SVM. Quand une entrée *x* vecteur de données pénètre le classifieur, un vecteur de K dimensions valeurs $f(x)^{(i)}, i = 1,..., K$ (une dimension pour chaque classe) est généré. Le classifieur classe ensuite *x* par le critère de classification suivant:

$x \in$ classe $\boldsymbol{i}$ si $f(x)^{(i)} = \max_{j=1,..,K} f(x)^{(j)}$. **(3.11)**

L'échantillon d'entrée *x*, est assigné à la classe qui a la plus grande valeur de la fonction de décision des *K* classifieurs antérieurs.

Où *K* est le nombre de classes, $f(x)^{(j)}$ est la sortie du j^iéme *SVM*

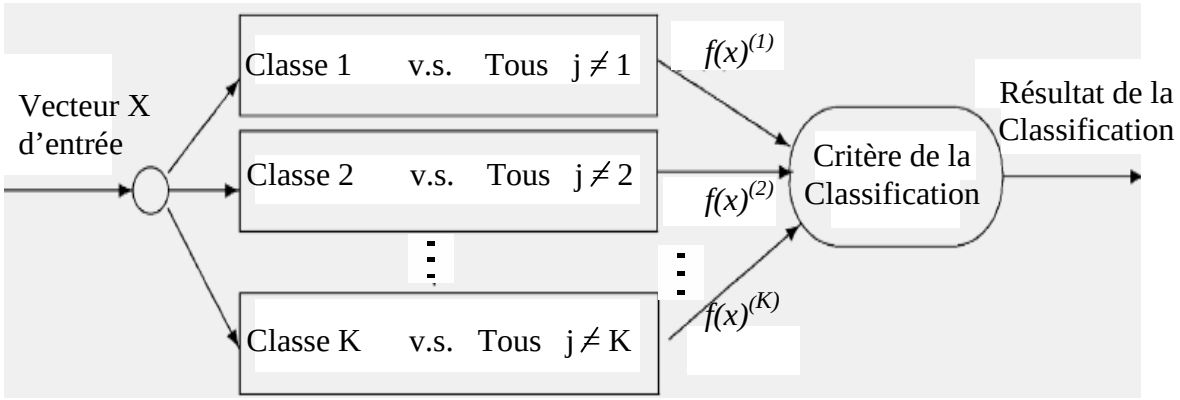


Figure 3.14 SVM Multi-classe : la stratégie "un contre le reste"

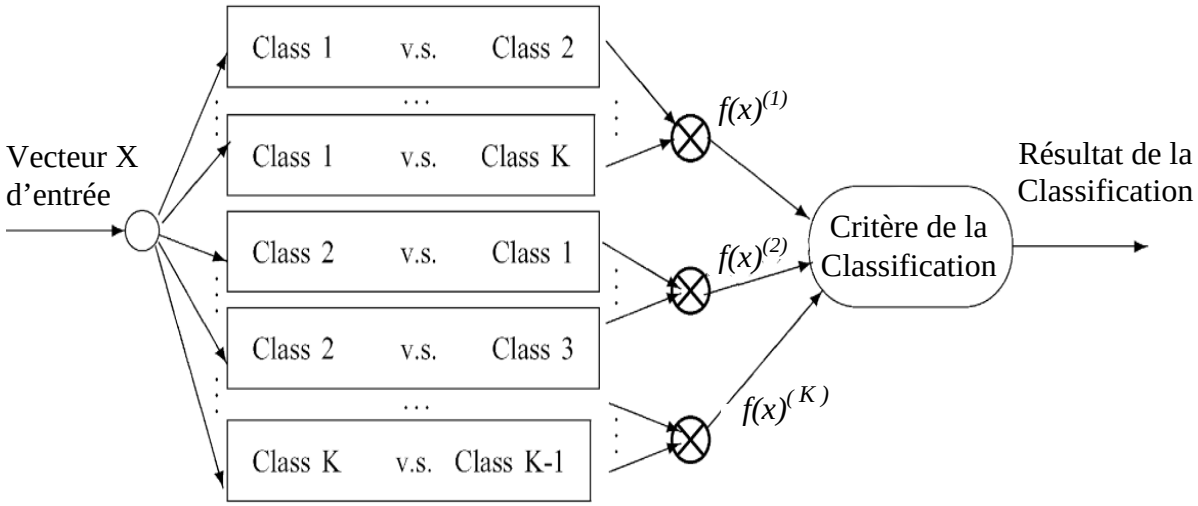


Figure 3.15 SVM Multi-classe : la stratégie "un contre un"

3.4.4.1. Comparaison entre techniques multi classes favorise

Une récente comparaison entre plusieurs techniques multi classes favorise l'approche *une contre toutes* vu sa simplicité et ses bonnes performances de classification. Concernant l'apprentissage, il est préférable d'avoir K *SVMs* que d'avoir K (K-1)/2 *SVMs* et le temps d'exécution des deux stratégies est similaire **[44].**

3.5. La reconnaissance de la parole

Les Kernel machines sont apparues comme des solutions viables et compétitives dans de nombreux domaines. Cependant, il existe de nombreux problèmes à leur utilisation pour la reconnaissance de la parole.

- Le premier problème qui est la classification multi classe, n'est pas spécifique au domaine de la reconnaissance de la parole. Cependant, les Kernel machines sont des classifieurs binaires. Il faut donc choisir une méthode pour étendre la classification binaire à une classification multi classes. De plus, il faut que la méthode soit capable de conserver la capacité de généralisation du classifieur binaire. Les méthodes de classification un contre tous et un contre un sont souvent utilisées. Il existe beaucoup de littérature qui traitent de ce problème, on peut donc citer **[45].**

- On se confronte également au problème d'estimation de probabilités à posteriori. La plupart des systèmes de reconnaissance de la parole combinent les informations acoustiques avec des informations statistiques du langage. Le classifieur doit donc retourner une probabilité conditionnelle pour chaque classe. Or le résultat en sortie d'un Kernel machine est la classe d'appartenance. Il faut donc choisir une méthode satisfaisante qui permet de calculer une probabilité à partir d'informations de distances fournies par le classifieur. Une possibilité de résolution de ce problème est de faire apprendre un modèle de mélange de Gaussiennes en sortie du classifieur *SVM*. Cette méthode est décrite dans **[46].**

- Le choix d'un kernel adapté. Une méthode pour choisir et paramétrer le kernel est nécessaire. Cependant, il n'existe pas de techniques pour trouver les meilleurs paramètres. En effet, ils dépendent fortement du problème étudié. Pour un problème de reconnaissance de chiffres, on pourra se référer à la documentation suivante : **[47].**

- Le travail dans un environnement temps réel. Les systèmes de la reconnaissance de la parole sont crées pour une communication temps réel. La reconnaissance de la parole doit être rapide **[48].**

- Le dernier problème est le problème des séquences de longueurs variables qui modélisent les variations de séquences des mots, l'accent, la vitesse d'élocution. Il existe plusieurs méthodes pour passer d'une séquence de longueur variable à un vecteur de dimension fixe. Beaucoup de méthodes ont le désavantage de supprimer de l'information

utile. Pour cela, il existe des méthodes à base de modèle génératif pour générer des scores, c'est à dire des vecteurs de taille fixe qui représentent des séquences de taille variables. Les scores sont considérés comme des vecteurs de caractéristiques et sont classifiées par une SVM multi classe **[49].** Un modèle génératif est un modèle qui s'appuie sur la règle de Bayes pour générer des probabilités. Les approches inspirées des modèles génératifs sont les chaînes de Markov Cachées et les Modèles de Mélange de lois Gaussiennes.

3.6. Les domaines d'applications

SVM est une méthode de classification qui montre de bonnes performances dans la résolution de problèmes variés. Cette méthode a montré son efficacité dans de nombreux domaines d'applications tels que le traitement d'image, la catégorisation de textes ou le diagnostiques médicales et ce même sur des ensembles de données de très grandes dimensions La réalisation d'un programme d'apprentissage par SVM se ramène à résoudre un problème d'optimisation impliquant un système de résolution dans un espace de dimension conséquente. L'utilisation de ces programmes revient surtout à sélectionner une bonne famille de fonctions noyau et à régler les paramètres de ces fonctions. Ces choix sont le plus souvent faits par une technique de validation croisée, dans laquelle on estime la performance du système en la mesurant sur des exemples n'ayant pas été utilisés en cours d'apprentissage. L'idée est de chercher les paramètres permettant d'obtenir la performance maximale. Si la mise en oeuvre d'un algorithme de SVM est en général peu coûteuse en temps, il faut cependant compter que la recherche des meilleurs paramètres peut requérir des phases de test assez longues **[32]**

3.7. Conclusion

Dans ce chapitre, nous avons donné une vision générale et une vision purement mathématiques des SVM. Cette méthode de classification est basée sur la recherche d'un hyperplan qui permet de séparer au mieux des ensembles de données. Nous avons exposé les cas linéairement séparable et les cas non linéairement séparables qui nécessitent l'utilisation de fonction noyau (kernel) pour changer d'espace. Cette méthode est applicable pour des tâches de classification à deux classes, mais il existe des extensions pour la classification multi classe.

Nous nous sommes ensuite intéressés aux différents domaines d'application. Il existe des extensions que nous n'avons pas présentées, parmi lesquelles l'utilisation des SVM pour

des tâches de régression, c'est-à-dire de prédiction d'une variable continue en fonction d'autres variables, comme c'est le cas par exemple dans la prédiction de consommation électrique en fonction de la période de l'année, de la température, etc. Le champ d'application des SVM est donc large et représente une méthode de classification intéressante.

CHAPITRE : 4

SVM multiclasse pour La reconnaissance de chiffres parlés anglais

4.1. Introduction

Dans le chapitre précédent nous avons présentés la méthode de classification binaire SVM, inspirées de la théorie statistique de l'apprentissage de Vladimir Vapnik introduite en 1995. Cependant, il existe de nombreux problèmes à leur utilisation pour la reconnaissance de la parole qui sont discutés d'une façon vaste antérieurement.

Dans ce chapitre, nous nous intéressons à la méthode d'apprentissage et de classification basée sur la théorie de l'apprentissage artificiel de Vapnik. Cette méthode : les Machines à Vecteurs de Support (SVMs pour Support Vector Machines) a été adaptée et appliquée au problème de la reconnaissance des chiffres parlés. L'avantage des SVMs est qu'un nombre restreint d'échantillon suffit à la détermination des vecteurs de support (SVs) permettant la discrimination entre les classes contrairement à l'estimation statistique.

4.2. SVM pour les Systèmes de reconnaissance de formes

L'objectif des systèmes de reconnaissance de formes est de classer la donnée d'entrée dans l'une des K classes. Tout système de reconnaissance des formes comporte toujours les trois parties suivantes :

- Un capteur permettant d'appréhender le phénomène physique considéré (dans notre cas un microphone),
- Un étage de paramétrisation des formes (par exemple un analyseur spectral),
- Un étage de décision chargé de classer une forme inconnue dans l'une des catégories possibles

Les SVM (Support Vector Machines) sont de nouvelles techniques d'apprentissage statistique supervisé : il faut fournir un ensemble de données d'entraînement pour construire le classifieur. Elles permettent d'aborder des problèmes très divers comme le classement, la régression, la fusion, etc... Depuis leur introduction dans le domaine de la Reconnaissance de Formes (RDF), plusieurs travaux ont pu montrer l'efficacité de ces techniques principalement en traitement d'image **[35].** L'idée essentielle des SVM consiste à projeter les données de l'espace d'entrée (appartenant à deux classes différentes) non-linéairement séparables dans un espace de plus grande dimension appelé espace de caractéristiques de façon à ce que les données deviennent linéairement séparables. Dans cet espace, la technique de construction de l'hyperplan optimal est utilisée pour calculer la fonction de classement séparant les deux classes.

Le système conventionnel de reconnaissance de formes basé sur SVM, peut se schématiser, comme indiqué dans la figure 4.1.

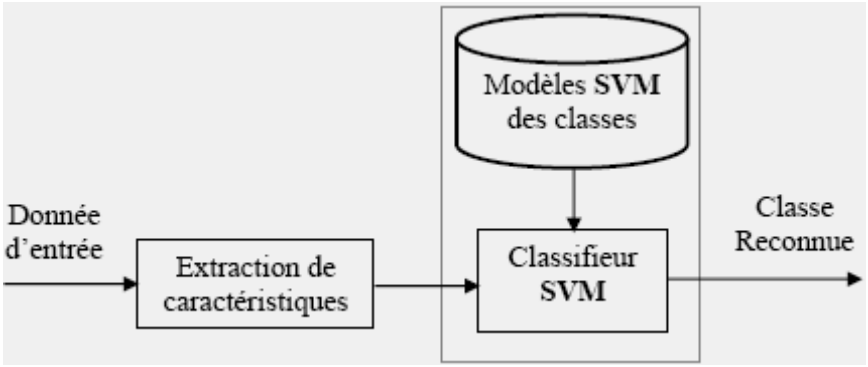


Figure 4.1 système conventionnel de reconnaissance de formes à base des SVM **[49].**

Après l'extraction des paramètres du signal vocal par la méthode MFCC, ces paramètres sont utilisés comme une donné d'entrée pour le composant de classification (SVM), qui va rechercher un hyperplan séparateur qui sépare les exemples dans la phase d'apprentissage et prend une décision de classification dans la phase d'identification.

Dans le module SVM, il y a deux phases : une pour l'apprentissage et l'autre pour la classification.

4.2. SVM multiclasse pour la reconnaissance automatique de chiffres parlés.

Le but de notre travail est la reconnaissance automatique de chiffres parlés (digit recognition en anglais), spécifiques à la langue anglaise, en mode indépendant du locuteur, en appliquant les séparateurs à vaste marge (SVM).

Pour atteindre cet objectif. Nous utiliserons les deux méthodes principales de la classification binaire multi classe (SVM Multi-classe) comme technique de reconnaissance de formes. Le système de reconnaissance résultant est basé sur, la fusion des scores, issus des deux stratégies de classification, un contre un, et un contre tous, afin d'augmenter la fiabilité du système résultant figure 4.3. Pour être combinés dans un système de reconnaissance automatique de la parole, les classifieurs doivent être différents. Cette différence peut être crée en choisissant des classifieurs de divers types, ou par changement des données d'apprentissage dans le cas où les classifieurs sont de même type comme notre cas.

Cependant, la combinaison de scores provenant de différents systèmes soit cohérente, les scores doivent d'abord être transformés dans un domaine commun : on parle alors de normalisation de score **[50].** Ils existent plusieurs méthodes de fusion de scores, qui seront présentées par la suite. Les chiffres anglo-saxons à reconnaître, avaient enregistrés par multi locuteurs et regroupaient dans une base de données disponible.

Si l'on veut appliquer les SVM au traitement de signal, on doit dépasser les limitations du modèle de base car le signal audio est dynamique et variable **[51].** On doit par exemple, être capable de représenter des séquences d'observation de taille variable par des vecteurs de taille fixe. Dans ce chapitre, on va essayer d'obtenir une solution à ce problème en insérant une méthode de normalisation des entrées (alignement temporelle). .

Le système de reconnaissance de chiffres parlés proposé peut être vus ou décomposé en quatre modules (composants):

- Acquisition,
- Paramétrisation (Segmentation et extraction des caractéristiques de voix),
- Apprentissage et classification,
- Décision (Calcul de résultat).

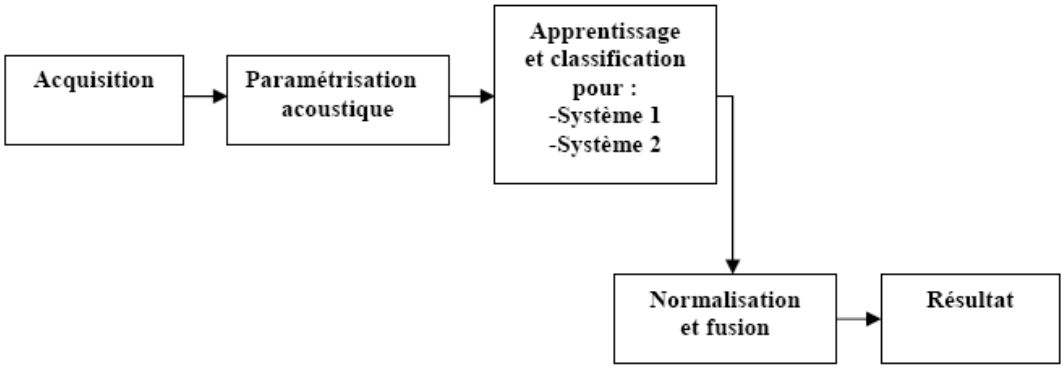


Figure 4.2 Les différents composants du système

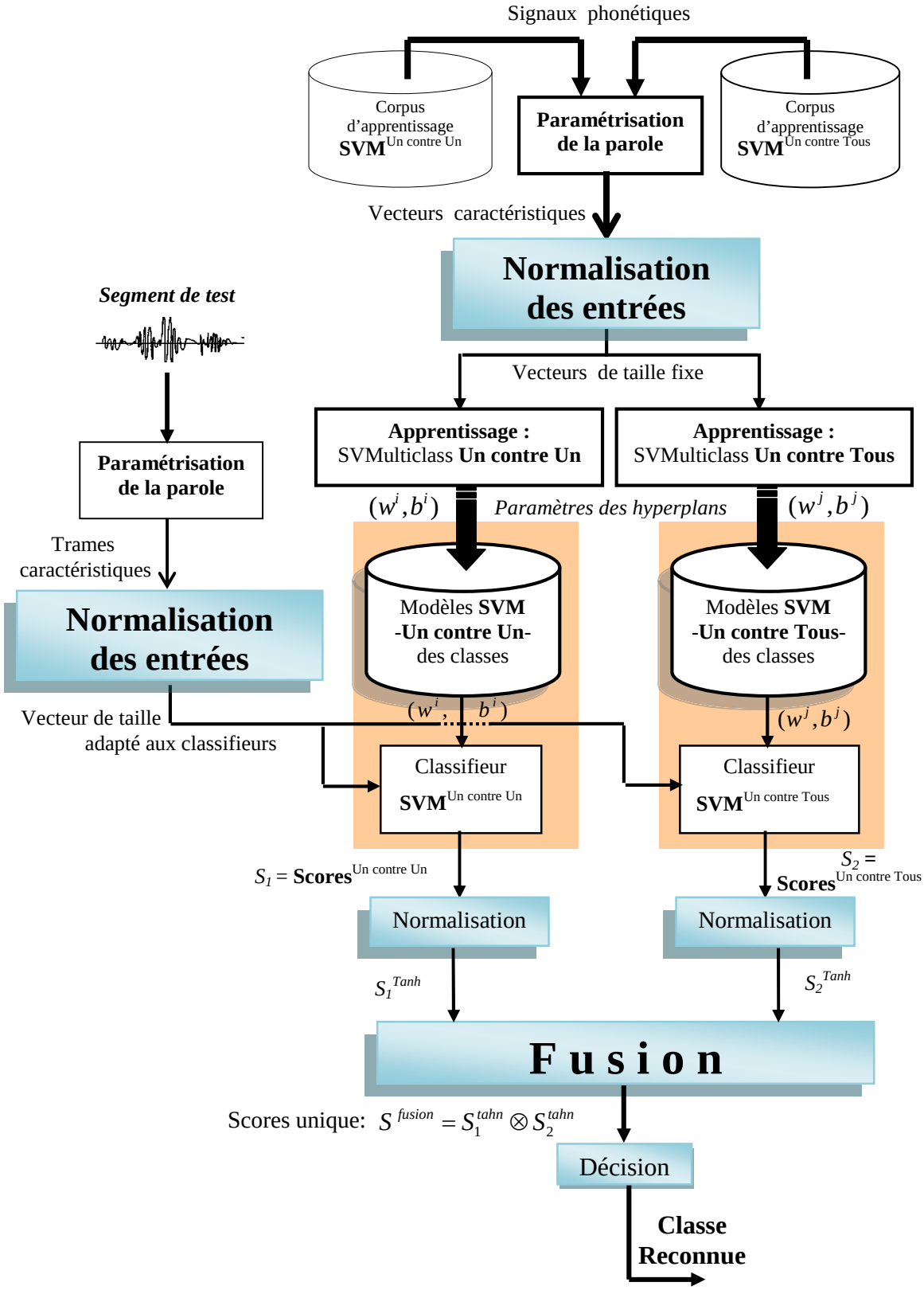


Figure 4.3 Architecture fonctionnelle du système

Dans les sections suivantes, nous allons consacrer à étudier, le système de reconnaissance de la parole proposé ; en détaillant ces différentes composantes.

4.3. Description des étapes

4.3.1. Acquisition

D'un sens, acquérir quelque chose c'est devenir propriétaire d'un bien. De ce sens l'acquisition du signal de parole (information) revient à l'appropriation des informations à un micro-ordinateur afin d'exécuter une tâche précise. L'acquisition est la première étape du processus de reconnaissance. Dans notre cas, les signaux sont disponibles sur une base de données (est un corpus anglais). Les enregistrements ont été effectués par multi-locuteur. La parole n'est pas bruitée. La fréquence d'échantillonnage choisie est de 8000 Hz, les échantillons ont été codés sur 16 bits par échantillon.

4.3.2. Paramétrisation

Durant cette phase, le signal vocal (segments phonémiques) est préaccentué pour lever les hautes fréquences qui sont moins énergétiques que les basses fréquences. Ce qui permet de compenser le niveau plus faible des sons. On utilise généralement un filtre passe-haut, dit de préaccentuation de 0.95. Puis une étape de décomposition parole/non-parole sera nécessaire afin de ne conserver que les zones de parole (correspondant aux locuteurs) (voir section 2. 3.2.).

Après la préaccentuation, le signal vocal qui est fortement non stationnaire est décomposé en une succession de tranches élémentaires supposées stationnaires. Ces tranches sont appelées fenêtres d'analyse ou trames. Typiquement, une analyse est appliquée toutes les 10 ms sur des fenêtres de 20 ms (par glissement et recouvrement des fenêtres d'analyse) pour générer un vecteur acoustique. Le découpage du signal en trames produit des discontinuités aux frontières des trames, qui se manifestent par des lobes secondaires dans le spectre. Pour composer ces effets de bord, nous appliquons une fenêtre de Hamming à chacune de ces tranches (Voir section 2. 3.3).

Après cette étape de prétraitement du signal d'entrée, une analyse acoustique est appliquée à chaque fenêtre à l'aide de la technique MFCC (mel-frequency cepstral coefficients) pour extraire les 14 premiers coefficients MFCCs pertinents du signal vocal. Ces coefficients sont calculés à partir des énergies extraites de 20 filtres triangulaires répartis

sur l'échelle de Mel (Voir section 2.3.4). Le choix d'utiliser cette technique est basé sur les points suivants :

1. Les coefficients MFCC (MEL Frequency Cepstral Coeficients) ont prouve leur efficacité en traitement automatique de la parole. Leur succès provient, entre autres, de l'utilisation de l'échelle MEL qui favorise les basses fréquences **[52].**

2. Ces coefficients MFCC ont été très utilisés en reconnaissance automatique de la parole du fait des bons résultats qu'ils ont permis d'obtenir. Nous voyons deux avantages à l'emploi de la méthode des MFCC. La première qualité est sa résistance reconnue au bruit. La deuxième qualité majeure de la méthode MFCC est sa plausibilité biologique puisqu'elle utilise une échelle psychoacoustique des fréquences similaire à celle de l'oreille interne **[53]** et **[54].**

3. Les paramétrisations basées sur une analyse en banc de filtres du signal (telles que MFCC) offre de bien meilleures performances aux systèmes de la reconnaissance automatique de la parole.

4.3.3. Normalisation des entrées

Le même message prononcé deux fois par un même locuteur dans des conditions identiques produit deux formes spectrales différentes. Cette variabilité est dite *intra-locuteur*. La qualité de la voix, le débit de parole, le degré d'articulation sont tous des facteurs à la base de variations acoustiques pour un signal donné. Ces variations entraînent des transformations non linéaires dans le temps du signal parole. La non linéarité vient du fait que les transformations affectent plus les parties stables du signal que les phases de transitions **[55].**

Donc, un locuteur ne prononce jamais deux fois le même mot de la même façon.

Chaque segment de parole contient un nombre variable de trames ce qui complique la gestion de la dynamique temporelle par les séparateurs à vaste marge (SVM). Pour lever cette difficulté, un alignement temporel est effectué après la phase d'analyse afin de garder une taille fixe pour le vecteur spectral, quelle que soit la longueur de son analyse. Pour cela une méthode spéciale est utilisée. Celle-ci consiste à calculer, le coefficient d'aplatissement supérieur à 3 (Kurtosis>3) pour les vecteurs caractéristiques pour chaque segment d'entrée, puis en gardant seulement, les (N) premières trames, qui ils possèdent des distributions type «leptokurtique». Le nombre N est un entier, étant constant à tous les énoncés d'entrées. à cette

façon, le nombre de paramètres présentés, à l'entrée de notre système est toujours fixe, quelle que soit la longueur du segment.

4.3.3.1. L'algorithme de l'alignement temporel dynamique proposé

Dans ce travail, une procédure particulière a été proposée et testée. Un système de la reconnaissance de la parole a été programmé, et utilise, un corpus de chiffres anglais parlés, afin d'enquêter ce nouvel algorithme. Cette étude s'est basée, sur le mode multi locuteurs en utilisant pour la classification les Machine à vecteur Support (Support Vector Machines en anglais) afin, de reconnaître les onze mots isolés. Avant de détailler notre algorithme, il faut définir quelques définitions probabilistes.

A. Kurtosis

En théorie des probabilités et en statistiques, le kurtosis (mot d'origine grecque), plus souvent traduit par coefficient d'aplatissement, ou coefficient d'aplatissement de Pearson, correspond à une mesure de l'aplatissement, ou a contrario de la pointicité, de la distribution d'une variable aléatoire réelle. C'est la seconde des caractéristiques de forme, avec le coefficient de dissymétrie. Elle mesure, hors effet de dispersion (donnée par l'écart-type), la disposition des masses de probabilité autour de leur centre, tel que donné par l'espérance mathématique, c'est-à-dire d'une certaine façon, leur regroupement proche ou loin du centre de probabilité **[56].**

Le moment centré d'ordre 4 permet de calculer le degré d'aplatissement d'une distribution à une variable. Afin d'obtenir un nombre sans dimension, on le divise par le carré de la variance. L'indicateur obtenu est appelé coefficient d'aplatissement de Pearson, ou kurtosis.

Pour résumer :

$$\text{Kurtosis} = \frac{m_4}{m_2^2} \qquad \textbf{(4.1)}$$

D'où $m_2 = \Sigma(X - \text{mean})^2/N$

$m_4 = \Sigma(X - \text{mean})^4/N$

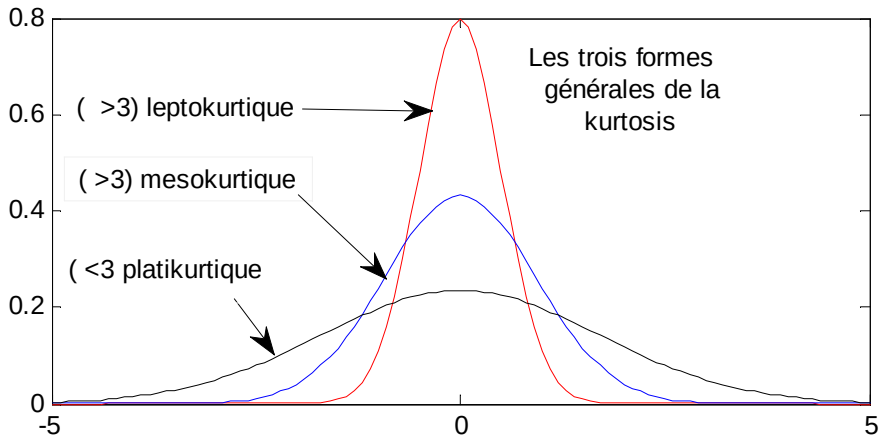
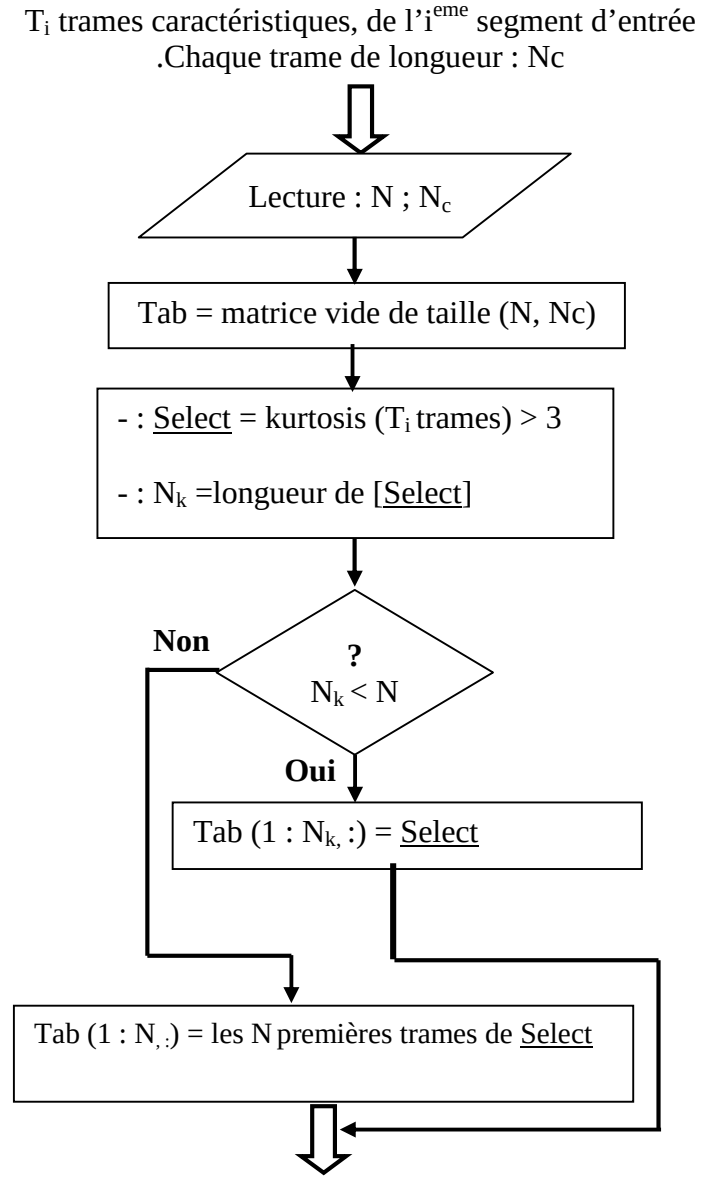


Figure 4.4 Les trois formes générales de la kurtosis

Une loi normale présente un coefficient de kurtosis égalant 3. Lorsque le coefficient est inférieur à 3, la distribution est aplatie par rapport à la distribution normale. La série est dite « platikurtique ». Lorsque le kurtosis est supérieur à 3, elle est dite «leptokurtique». Dans ce cas, la distribution est plus pointue que celle de la loi normale et les queues de distribution sont plus épaisses ce qui indique une plus forte probabilité des points extrêmes. Ainsi, plus la kurtosis est élevée, plus élevées sont les probabilités de subir des pertes ou des gains très importants **[57].** Voir la figure 4.4 ci-dessus.

B. Description de l'algorithme d'alignement

L'algorithme d'alignement temporel proposé est une procédure particulière, s'est basé sur les valeurs du coefficient d'aplatissement (kurtosis) des trames caractéristiques. Pour comprendre son comportement durant son exécution, il nous faut bien décrire quelques variables. N est un entier donné, représente le nombre des trames à extraire de chaque énoncé. N_k est le nombre des trames qui représentent un coefficient de kurtosis supérieur à 3. Nc est le nombre des coefficients mfcc. Voir l'algorithme ci-dessous.



Algorithme 4.1 Les étapes essentielles de l'algorithme d'alignement temporel proposé.

Après la phase de l'extraction des paramètres. L'algorithme prend les vecteurs caractéristiques (trames) de chaque séquence acoustique provenant de la base des données (Apprentissage, Test). A ces vecteurs, il sélectionne ensuite seulement les trames qui présentent le coefficient de kurtosis supérieur à 3. Dans ce cas, les distributions de la sélection sont plus pointues que celle de la loi normale et les queues des distributions sont plus épaisses, ce qui indique une plus forte probabilité des points extrêmes figure 4.5, puis il applique une comparaison de N avec N_k. Le résultat de la comparaison pousse l'algorithme de décider l'une des deux opérations suivantes. Soit il prend toute la sélection des trames, si

(Nk < N) et complète la différence (N-Nk) en insérant des trames vides de longueur Nc. Soit il garde seulement les N premiers échantillons de la sélection, si (Nk > N). Notez que le résultat de l'alignement des trames caractéristiques de l'énoncé d'entrée est chargé dans une matrice de dimension (N, Nc), tel que le couple (N, Nc) représente le nombre des (lignes, colonnes) respectivement. De cette façon on peut rendre des séquences de déférentes longueurs à des vecteurs de même taille, quelle que soit la longueur de son analyse, voir Algorithme 1 ci-dessous.

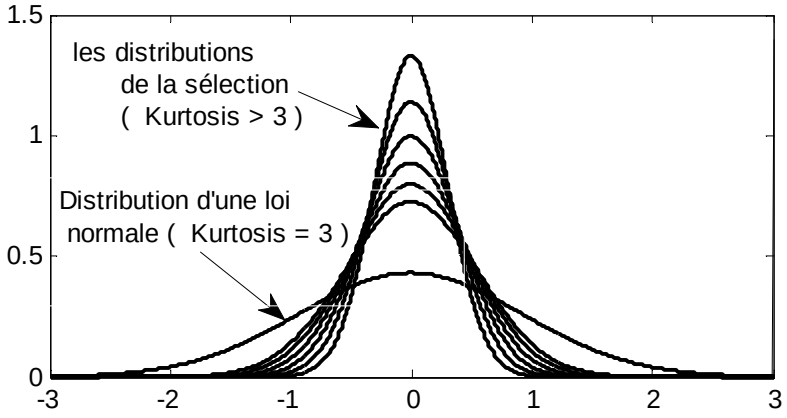


Figure 4.5 Les distributions des trames sélectionnées s'élèvent assez haut puis retombe assez brutalement et elles sont plus pointues que celle de la loi normale

4.3.4. L'apprentissage et le test

L'apprentissage et le test des SVM ont été réalisés à l'aide du logiciel LIBSVM dont les algorithmes sont décrits dans **[58].** La bibliothèque LIBSVM est développée dans le but de simplifier l'utilisation des SVM comme un outils, dans ce travail nous utilisant la 2.33 qui est présentée par le package Matlab libsvm-2.33. Nous avons choisi d'utiliser le C-SVM avec un noyau gaussien : $K(x, y) = \exp\left(-\|x - y\|^2 / 2\sigma^2\right)$

Les hyper paramètres σ et C ont été déterminés empiriquement en cherchant à minimiser le taux d'erreur sur la base de validation.

4.3.5. Les méthodes de fusion de scores

Après avoir introduit et définit les deux stratégies principales pour étendre la classification binaire au cas où l'on a M classes en chapitre 3, nous allons maintenant nous intéresser aux méthodes de fusion de scores. Les méthodes de fusion de scores, elles combinent les informations au niveau des scores issus de plusieurs systèmes comme indiqué sur la figure 4.6.

Un système de fusion est constitué de deux modules, un module de **fusion** et un module de **décision** (voir figure 4.6).

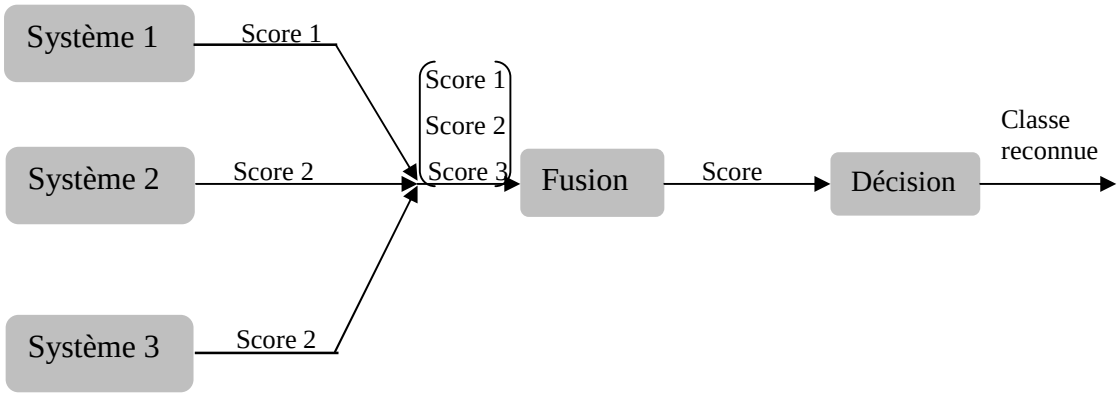


Figure 4.6 Schéma de la fusion de scores.

Il existe deux approches pour combiner les scores obtenus par différents systèmes. La première approche est de traiter le sujet comme un problème de **combinaison**, tandis que l'autre approche est de voir cela comme un problème de **classification**. Il est important de noter que ***Jain et al*** ont montrés que les approches par combinaison sont plus performantes que la plupart des méthodes de classification **[59,60].**

Dans **l'approche par combinaison**, les scores individuels sont combinés de manière à former un unique score qui est ensuite utilisé pour prendre la décision finale. Afin de s'assurer que la combinaison de scores provenant de différents systèmes soit cohérente, les scores doivent d'abord être transformés dans un domaine commun : on parle alors de ***normalisation de score*** **[61].**

4.3.5.1. Normalisation de scores

Les méthodes de normalisation de scores ont pour objectif de transformer individuellement chacun des scores issus des systèmes pour les rendre **homogènes** avant de les combiner. En effet, les scores issus de chaque système peuvent être de nature différente.

De plus chaque système peut avoir des intervalles de variations des scores différents, par exemple pour un système les scores varient entre 0 et 1 et pour un autre les scores varient entre 0 et 1000.

On comprend bien la nécessité de normaliser les scores avant de les combiner. Les différentes techniques de normalisation de scores sont :

- ✓ Normalisation par la méthode Min-Max
- ✓ Normalisation par la méthode Z-Score
- ✓ Normalisation par la méthode tangente hyperbolique "Tanh"
- ✓ Normalisation par la médiane et l'écart absolu médian (MAD)
- ✓ Normalisation par une fonction quadratique-linéaire-quadratique (QLQ)

- **Normalisation par la méthode Min-Max**

La technique de normalisation la plus simple est la **normalisation Min-Max**. Elle est la plus adaptée dans le cas où les bornes (valeurs minimales et maximales) des scores produits par des systèmes sont connues. Dans ce cas, on peut facilement translater les scores minimums et maximums respectivement vers 0 et 1. Cependant, même si les scores ne sont pas bornés, on peut estimer les valeurs minimales et maximales pour un jeu de scores donné et appliquer ensuite la normalisation Min-Max. Soit s_{ij} le j^{ème} score de sortie du i^{ème} système, où i = 1, 2, ...,N et j = 1, 2, ...,M (N est le nombre de systèmes et M le nombre de scores disponibles dans l'ensemble de données d'entraînement). Le score normalisé Min-Max pour le score de test s_{ik} est donné par (4.2):

$$s'_{ik} = \frac{s_{ik} - \min(\{s_{i.}\})}{\max(\{s_{i.}\}) - \min(\{s_{i.}\})} \qquad \textbf{(4.2)}$$

Où $\{s_{i.}\} = \{s_{i1}, s_{i2}, ..., s_{iM}\}$. Quand les valeurs minimales et maximales sont estimées à partir du jeu d'entraînement de scores donné, cette méthode n'est pas robuste (c'est à dire que cette méthode est fortement sensible aux valeurs aberrantes dans les données utilisées pour

l'estimation). La normalisation Min-Max conserve la distribution de scores originale à un facteur d'échelle près et transforme tous les scores dans l'intervalle [0, 1]. **[62, 63,64].**

- **Normalisation par la méthode Z-Score**

La technique de normalisation de score la plus employée est certainement la **Z-Score** qui utilise la **moyenne** arithmétique et **l'écart-type** des données. On peut s'attendre à ce que cette méthode fonctionne bien si on a une connaissance à priori du score moyen et des variations de score d'un système. Si on n'a pas de connaissance à priori sur la nature de l'algorithme de reconnaissance, nous devons alors estimer la moyenne et l'écart-type des scores à partir d'un jeu de scores donné. Les scores normalisés sont donnés par (4.3) :

$$s'_{ik} = \frac{s_{ik-\mu}}{\sigma} \quad , \qquad \textbf{(4.3)}$$

Où μ est la moyenne arithmétique et σ l'écart-type des données. Cependant, la moyenne et l'écart-type sont tous les deux sensibles aux valeurs aberrantes et donc cette méthode n'est pas robuste. De plus, la normalisation *Z-Score* ne garantit pas un intervalle commun pour les scores normalisés provenant de différents systèmes. Si la distribution des scores n'est pas gaussienne, la normalisation Z-Score ne conserve pas la distribution d'entrée en sortie. Cela est simplement dû au fait que la moyenne et l'écart-type sont les paramètres de position et d'échelle optimaux seulement pour une distribution gaussienne.

Pour une distribution arbitraire, la moyenne et l'écart-type sont respectivement des estimateurs raisonnables de position et d'échelle, mais ne sont pas optimaux **[62, 63,64].**

- **Normalisation par la méthode tangente hyperbolique "Tanh"**

Les scores normalisés sont donnés par (4.4):

$$s'_{ik} = \frac{1}{2}\left\{\tanh\left(0.001\,\frac{s_{ik-\mu}}{\sigma}\right)+1\right\} \quad , \qquad \textbf{(4.4)}$$

Où μ est la moyenne arithmétique, σ l'écart-type des données et tanh la tangente hyperbolique. La méthode tangente hyperbolique met chaque score normalisé dans l'intervalle [0, 1]. **[62, 63,65].**

- **Normalisation par la médiane et l'écart absolu médian (MAD)**

Sont insensibles aux valeurs aberrantes et aux points aux extrémités d'une distribution. Ainsi, une méthode de normalisation utilisant la médiane et la **MAD** (l'écart absolu médian) serait robuste et est donnée par (4.5):

$$s_{ik}^{'} = \frac{s_{ik-median}}{MAD} \quad \textbf{(4.5)}$$

Où: MAD = median ($\{|s_{i.} - \text{median}(\{s_{i.}\})|\}$)

Cependant, les estimateurs issus de la médiane et de la MAD ont une faible efficacité comparée aux estimateurs issus de la moyenne et de l'écart-type, c'est-à-dire que lorsque la distribution de score n'est pas gaussienne, la médiane et la MAD sont de pauvres estimateurs des paramètres de position et d'échelle. Ainsi, cette technique de normalisation ne conserve pas la distribution d'entrée et ne transforme pas les scores dans un intervalle commun **[62], [64].**

4.3.5.2. Approche par combinaison de scores

A. Méthode de combinaisons simples

Les méthodes de combinaisons de scores simples sont des méthodes très simples dont l'objectif est d'obtenir un score final ***s*** à partir des **N** scores disponibles $\mathbf{s_i}$ pour i = 1à N issus de N systèmes.

Les méthodes les plus utilisées sont la moyenne, le produit, le minimum, le maximum ou la médiane **[63,66].**

- Combiner les scores par la moyenne consiste à calculer ***s*** tel que

$$s = \frac{1}{N}\sum_{i=1}^{N} s_i \quad \textbf{(4.6)}$$

- Combiner les scores par le produit consiste à calculer ***s*** tel que

$$s = \prod_{i=1}^{N} s_i \quad \textbf{(4.7)}$$

- Combiner les scores par le minimum consiste à calculer ***s*** tel que

$$s = \min(s_i) \tag{4.8}$$

- Combiner les scores par le maximum consiste à calculer ***s*** tel que

$$s = \max(s_i) \tag{4.9}$$

- Combiner les scores par la médiane consiste à calculer ***s*** tel que

$$s = med(s_i) \tag{4.10}$$

- La somme pondérée c'est une méthode un peu plus évoluée qui nécessite une adaptation par le réglage de paramètres.

$$s = \sum_{i=1}^{N} w_i s_i \tag{4.11}$$

La somme pondérée permet de donner des poids différents w_i à chacun des systèmes en fonction de leur performance individuelle ou de leur intérêt dans le système multi-algorithmes.

B. Combinaison de scores par logique floue

La théorie de **la logique floue** (des sous-ensembles flous) a été introduite par Zadeh en 1965 **[67]** comme une extension de la logique binaire d'une part et une amélioration de la logique multivaluée (admettant plusieurs valeurs de vérité) d'autre part. L'importance de la logique floue réside dans le fait qu'elle s'approche du **raisonnement humain** par l'intégration et le traitement du caractère approximatif, vague, imprécis ou flou de la connaissance humaine. Les termes linguistiques tels que « **environ** », « **moyenne** », « **approximativement** » sont de nature à donner un caractère flou aux phrases énoncées. Par exemple, la règle « si le prix est inférieur à 6 000 DA, j'achète » sera intuitivement utilisable si le prix est de 6 002 DA, mais elle ne pourrait être exploitée en logique classique puisque le prix indiqué ne satisferait pas la prémisse.

B.1. Mesure floue

Un jeu de fonction g : $P(\boldsymbol{Y}) \rightarrow [0, 1]$ est appelée une **mesure floue** si les conditions suivantes sont remplies :

1. conditions aux limites: $g(\Phi) = 0$, $g(\boldsymbol{Y}) = 1$
2. monotonie : $g(A) \leq g(B)$, si $A \subset B$ et $A, B \in P(\boldsymbol{Y})$
3. continuité: $\lim_{i\to\infty} g(A_i) = g(\lim_{i\to\infty} A_i)$, si $\{A_i\}_{i=1}^{\infty}$ est une suite croissante d'ensembles mesurables.

A partir de cette définition, Sugeno **[68]** a introduit un soi-disant g_λ **mesure floue** qui est livré avec une propriété supplémentaire

$$g(A\cup B)=g(A)+g(B)+\lambda g(A)g(B) \qquad \textbf{(4.12)}$$

Pour tous les $A, B \subset \boldsymbol{Y}$ *et* $A \cap B = \Phi$, et pour certains $\lambda > -1$.

Évidemment quand λ=0, le g_λ mesure floue devient une mesure de probabilité standard.

En général, la valeur de λ peut être déterminée en raison de l'état limite de la mesure floue g_λ. Cette condition pour $g(\boldsymbol{Y}) = 1$. Par conséquent, la valeur de λ est déterminée par la résolution de ce qui suit:

$$g_\lambda(y)=\frac{1}{\lambda}\left(\prod_{i=1}^{n}(1+\lambda g^i)-1\right), \quad \lambda \neq 0 \qquad \textbf{(4.13)}$$

est l'équivalent de :

$$\lambda+1=\prod_{i=1}^{n}(1+\lambda g^i) \qquad \textbf{(4.14)}$$

Où $\lambda \in (-1, \infty)$, $\lambda \neq 0$, et g^i est la valeur de **la fonction de densité floue**. La solution peut être facilement obtenue; évidemment on s'intéresse à la racine supérieure à -1. **[69].**

B.2. Intégrale floue

L'intégrale floue de la fonction ***h*** calculée sur ***Y*** par rapport à une mesure floue ***g*** est définie sous la forme

$$\int_{\lambda} h(y) \, o \, g(\cdot) = \sup_{\alpha \in [0,1]} \left[\min \left[\alpha, g(\{ y \mid h(y) \geq \alpha \}) \right] \right] \quad \textbf{(4.15)}$$

1. Intégrale floue de Sugeno

Lorsque les valeurs des $h(.)$ sont classés dans l'ordre décroissant, $h(y1) \geq h(y2) \geq \ldots \geq h(y_n)$.

L'intégrale floue de **Sugeno [70, 71, 72,73]** est calculée comme suit:

$$\int_{\lambda} h(y) \, o \, g(\cdot) = \max_{i=1}^{n} \left[\min(h(y_i), g(A_i)) \right] \quad \textbf{(4.16)}$$

Où $A_i = \{y_1, y_2, \ldots, y_i\}$ désigne un sous-ensemble d'éléments. Les valeurs de $g(A_i)$ pris en charge par la mesure floue sur les sous-ensembles correspondant d'éléments peut être déterminée de manière récursive sous la forme :

$$g(A_1) = g(y_1) = g^1 \quad \textbf{(4.17)}$$

$$g(A_i) = g^i + g(A_{i-1}) + \lambda g^i g(A_{i-1}) , \quad \text{pour } 1 < i \leq n \quad \textbf{(4.18)}$$

Le calcul de la fonction de densité floue g^i sur la base des données sont assurées par la manière suivante :

$$\begin{cases} g^i = \beta p_i, & i = 1 \\ g^i = (1-\beta) p_i, & i = 2,3,4 \end{cases} \quad \textbf{(4.19)}$$

Où *pi* est le taux de classification dans l'intervalle [0, 1] pour chaque système.

$\beta \in [0, 1]$ est un facteur qui tente de mettre en place un certain équilibre entre les résultats de la classification.

2. Intégrante floue de Choquet

Il a été démontré que (4.15) n'est pas une extension correcte de l'intégration de **Lebesgue** habituelle. En d'autres termes, lorsque la mesure est additive l'expression ci-dessus ne retourne pas l'intégrale au sens de **Lebesgue**. Afin de remédier à cet inconvénient, Murofushi et Sugeno **[74]** ont proposés un soi-disant intégrante floue de Choquet **[70,71, 72,73]** calculé de la manière suivante :

$$\int_{\lambda} h(y)\, o\, \mathrm{g}(\cdot) = \sum_{i=1}^{n} \left[h(y_i) - h(y_{i+1}) \right] g(A_i), \qquad h(y_{n+1}) = 0 \tag{4.20}$$

4.4. Résumé de l'algorithme général de reconnaissance de chiffres parlés

Par exemple on a une base de données composée de deux corpus, l'un contient les données de l'apprentissage et l'autre pour le test. Notre système de reconnaissance se fonctionne de la façon suivante :

Etape 1

Une paramétrisation est effectuée à tous échantillons de la base (corpus d'apprentissage, corpus de test). Cette paramétrisation, consiste à un prétraitement, et une étape d'extraction des caractéristiques en utilisant le modèle MFCC, pour extraire les 14 premiers coefficients cepstraux.

Etape 2

Les segments caractéristiques qui viennent de cette étape, ont des longueurs différentes. Ce pondant, les techniques SVM exigent des vecteurs d'entrée de taille fixe et que l'utilisation directe des trames n'est pas très efficace. Pour lever cette difficulté, un alignement temporel est effectué après la phase d'analyse afin de garder une taille fixe pour le vecteur spectral, quelle que soit la longueur de son analyse.

Etape 3

Les deux stratégies de la classification binaire multi classe (SVM Multi-classe), un contre un, et un contre tous, sont entraînés et testés par les données d'entraînement et les données de teste respectivement. La diversité entre ces classifieurs est crée par changement des données d'apprentissage

Etape 3

Les taux de reconnaissance, obtenus par les systèmes et les scores issus des classifieurs (Voir le tableau ci-dessus), vont les utiliser dans la combinaison des classifieurs, afin de construire un système résultant espéré.

Stratégies	**un contre un**	**un contre tous**
Taux de reconnaissance	p_1	p_2
Score	S_1	S_2

Etape 4

Une normalisation de scores est effectuée, pour objectif de transformer individuellement chacun des scores (S_1, S_2) issus des deux systèmes pour les rendre homogènes. Nous avons choisi la normalisation de scores par la méthode tangente hyperbolique "*Tanh*". Cette méthode est choisie après plusieurs expériences. Nos épreuves ont montré qu'une normalisation utilisant la méthode *tangente hyperbolique* peut améliorer les performances de notre système de reconnaissance qu'une normalisation utilisant les autres méthodes qui ont été décrites dans la section 4.3.5. L'avantage de la *méthode tangente hyperbolique* est que ne modifie pas la forme des distributions et met chaque score normalisé dans l'intervalle [0, 1].

$$S_1^{Tanh} = \frac{1}{2}\left\{\tanh\left(0.001\frac{(S_1)_{ij}-\mu}{\sigma}\right)+1,\ i=1,...,N\ \ et\ \ j=1,...,k\right\}$$

$$S_2^{Tanh} = \frac{1}{2}\left\{\tanh\left(0.001\frac{(S_2)_{ij}-\mu}{\sigma}\right)+1,\ i=1,...,N\ \ et\ \ j=1,...,k\right\}$$

Où N le nombre de classes, k le nombre des échantillons du corpus de test

Etape 5

Les scores sont maintenant, disponibles pour les fusionner (combinaison de scores), par les méthodes suivantes :

Méthodes simples

{la moyenne ; le produit ; le minimum ; le maximum ; la médiane ; la somme}

Combinaison de scores par logique floue

L'algorithme suivant présente comment en fait la combinaison par les deux intégrales Sugeno et de Choquet :

a. Calcul la fonction de densité floue g^i

$$\begin{cases} g^i = \beta p_i, & i = 1 \\ g^i = (1-\beta) p_i, & i = 2 \end{cases}$$

Avec: p_i est le taux de classification dans l'intervalle [0, 1] pour chaque système. $\beta \in$ [0, 1] est un facteur, qui tente de mettre en place un certain équilibre entre les résultats de la classification, i l'indice de chaque système. Dans notre cas, nous avons deux systèmes.

b. Calcul λ par : $\lambda + 1 = \prod_{i=1}^{n} (1 + \lambda g^i)$, ou $\lambda \in (-1,\infty)$, $\lambda \neq 0$ et i=1,2.

c. Calcul la mesure floue $g(A_i)$ sur les sous-ensembles $A_i = \{S_1^{tanh}, S_2^{tanh}\}$ par :

$$g(A_1)=g(S_1^{Tanh})=g^1$$
$$g(A_i) = g^i + g(A_{i-1}) + \lambda g^i g(A_{i-1}) \text{ , pour } 1 < i \leq 2$$

b. Calcul l'intégrale floue de Sugeno par :

$$S^{fusion} = \int_{\lambda} h(y) \, o \, g(\cdot) = \max_{i=1}^{n} \left[\min(h(y_i), g(A_i)) \right]$$

$h(y_i)$ sont les scores et sont classés dans l'ordre décroissant , n =2 : $h(y_1) \geq h(y_2)$.

Ou calcul l'intégrale floue de Choquet par :

$$S^{fusion} = \int_{\lambda} h(y) \, o \, g(\cdot) = \sum_{i=1}^{n} \left[h(y_i) - h(y_{i+1}) \right] g(A_i),$$
$$h(y_{n+1}) = 0$$

Etape 6

La dernière étape est la décision, et sera calculer de la manière suivante:

$$classe(x) = \arg\max_{i=1,\ldots,N} S_i^{fusion} \qquad \textbf{(4.21)}$$

Après la fusion de scores. L'échantillon d'entrée x, est assigné à la classe qui a la plus grande valeur de la fonction de décision des N classifieurs antérieurs.

Où N est le nombre de classes, S_i^{fusion} est la sortie de l'i$^{\text{iéme}}$ SVM^{fusion}

4.5. Conclusion

Nous avons présenté dans ce chapitre les différentes étapes qui peuvent conduire à une conception convenable d'un système de reconnaissance de la parole pour la classification des chiffres parlés (digit recognition), spécifiques à la langue anglaise. Notre système utilise le mode multi locuteurs, on a choisi la méthode de codage MFCC pour la paramétrisation du signal acoustique et la méthode SVM pour la reconnaissance des mots isolés. La fusion des scores issus des deux stratégies de classification multiclasse, un contre un, et un contre tous, a été utilisé, pour but d'augmenter la fiabilité du système résultant.

CHAPITRE : 5

Expérience et Résultats

5.1. Chaînes de reconnaissance automatique de la parole

Les chaînes implémentées par le système de reconnaissance proposé s'inspirent des chaînes de reconnaissance de la parole décrites dans *Using Support Vector Machines for Spoken Digit Recognition* de **Issam Bazzi** et **Dina Katabi [47]**. Ce document traite dans une partie de la reconnaissance de chiffres. Pour mettre en œuvre mes chaînes, j'ai donc également utilisé une base de données de chiffres.

5.1.1. Using Support Vector Machines for Spoken Digit Recognition, Issam Bazzi et Dina Katabi

L'article de **Issam Bazzi** et **Dina Katabi** montre une adaptation de la classification par *SVM* à la reconnaissance automatique de chiffres prononcés en Anglais. Les *SVM* nécessitent des vecteurs de longueurs fixes. Ils proposent donc une méthode pour palier à ce problème. L'analyse des caractéristiques est réalisée par une analyse *MFCC* (Mel Frequency Cepstral Coefficients). Ils utilisent dans leurs systèmes 14 coefficients. Ensuite, ils suppriment des trames pour lesquels la différence entre deux trames successives est minimum. La suppression est effectuée tant que qu'un nombre fixe de trame n'est pas atteint. La différence entre deux trames est calculée par une distance Euclidienne. Ils concatènent ensuite l'ensemble des trames pour constituer un unique vecteur. Dans leurs travaux, ils ont réduit l'ensemble des trames issues de l'analyse *MFCC* à 30 trames. À la fin de l'analyse des caractéristiques, ils obtiennent donc un vecteur de 30x14 = 420 dimensions. Enfin, ils effectuent une analyse en composante principale pour réduire le vecteur à 45 dimensions.

Pour la classification multi classes, deux approches ont été adoptées : la première consiste à une classification 1 contre tous. Chaque classifieur binaire est entraîné pour reconnaître un chiffre particulier contre tous les autres chiffres. La classe choisie est donc la classe dont le test est positif. Cependant, plusieurs classes ou aucune peuvent être positives. Dans ce cas, on choisit la classe dont la distance entre le point de test et l'hyperplan est maximal. En effet, si plus d'une classe sont positives, on choisit la classe qui possède la plus grande confiance. Dans l'autre cas, si aucune classe n'a été choisit, toutes les distances sont négatives, on choisit la classe qui possède la plus petite erreur de classification.

La deuxième approche consiste à une classification 1 contre 1. Chaque classifieur binaire est construit pour chaque pair de chiffres. La classe choisie est donc la classe qui possède le plus grand nombre de votes. En cas d'égalité, la méthode de départage est la même que celle du 1 contre tous.

Ils finissent leur article en montrant leurs résultats. Ils ont utilisé la base de données *OGI* (**Oregon Graduate Institute**) **[75]** 1992 et 1994 avec le système de reconnaissance de la parole **MIT SUMMIT**. La base se compose de 1280 chiffres prononcés par 133 personnes. Chaque personne ayant enregistrée plusieurs chiffres entre 1 et 10. Ils ont utilisé 826 chiffres pour la phase d'entraînement et 454 chiffres pour la phase de test. Ce qui représente environ 83 points d'entrainement et 45 points de test pour chaque chiffre.

Leurs tests ont montrés qu'ils obtenaient de meilleurs résultats avec un noyau Gaussien avec une variance de 4 et avec une *ACP* avec 45 composantes.

Chaînes de reconnaissance automatique de la parole	**Taux de reconnaissance**
MFCC – Réduction – ACP – SVM RBF un contre tous	**94.90**

Cependant, ils indiquent que les systèmes actuels parviennent à des taux de 99% de reconnaissance. Mais dans leur système, ils ne prennent pas en compte la modélisation phonétique des chiffres prononcés. Chaque chiffre est traité pour avoir la même taille. De plus, il n'utilise aucune information de contexte comme des modèles de langage. Enfin, Il n'utilise que 826 points en phase d'apprentissage comparé aux 20 000 à 30 000 points utilisés dans la littérature.

5.2. Les bases de données

Le corpus de données constitue un élément important en reconnaissance de la parole. Il doit être suffisamment fourni pour permettre un apprentissage optimal. Il existe de nombreuses bases de données utilisées dans la recherche. Elles sont cependant très chères.

5.2.1. Les bases de données de reconnaissance de la parole

Pratiquement, tous les laboratoires faisant de la reconnaissance de la parole possèdent leur propre corpus de données. Néanmoins, je ne vais détailler que les principales utilisées dans la littérature.

L'une des plus connue et des plus utilisée est la base *DARPA TIMIT* **[76]** de **NTIS.** C'est un corpus de paroles lues en continue et en anglais américain. Il totalise 4h30 de paroles avec 630 orateurs différents.

POLYPHONE **[77]** est une base de son français et allemand. Elle se compose de mots isolés et de paroles continues. Elle regroupe des paroles de 5000 personnes différentes.

BDSONS **[78]** est une base distribuée par l'institut National Polytechnique de Grenoble. Elle se compose d'enregistrements constitués de chiffres et de lettres épelées en français. Elle compte 1339 fichiers sons.

ISOLET est une base qui est également très utilisée. Elle se constitue de lettres épelées en anglais. Elle recense 7800 lettres épelées prononcées par 150 orateurs.

Le corpus *VODIS* **[79]** est distribué par le laboratoire **LORIA**. Il se constitue de parole spontanée en français. Il inclut 6 sources de bruits. Les enregistrements sont prononcés par 100 hommes et 100 femmes.

5.2.2. La base de données utilisée

La base de donnée **TI DIGITS [80]** : est un corpus anglais Il est composé d'enregistrements de 208 locuteurs (114 femmes et 94 hommes), répartis en deux sous-ensembles : un ensemble d'apprentissage (95 locuteurs) et un de test (113 locuteurs). Chaque locuteur a prononcé deux fois chacun des onze chiffres (de 1 à 9, plus "oh" et "zéro"). Le sous-ensemble d'apprentissage (respectivement de test) contient donc 2068 occurrences de chiffres (respectivement, 2486 occurrences de chiffres). Tous les locuteurs sont adultes de divers états des USA. Le signal est échantillonné à fe = 8kHz, codé sur 16 bits. La parole n'est pas bruitée.

Nous avons combiné parallèlement des classifieurs de même type SVM multiclasse. La diversité entre ces classifieurs est crée par changement des données d'apprentissage. Il va donc falloir, répartir l'ensemble d'apprentissage (57 femmes et 38 hommes) en deux sous ensembles indépendants, l'un servant à l'apprentissage du classifieur ***SVM RBF un contre tous***, l'autre à l'entraînement du classifieur ***SVM RBF un contre un***.

5.3. Présentation des chaînes du système de reconnaissance proposé

J'ai implémenté diverses chaînes de reconnaissance de la parole faisant appel à diverses techniques de traitement du signal et de reconnaissance des formes. Chaque élément de la chaîne est considéré comme une boîte réalisant un traitement. Les paramètres des différents algorithmes sont essentiels. Un mauvais choix peut faire augmenter considérablement le temps d'apprentissage ou de donner des résultats très médiocres. Il n'existe pas vraiment de méthode pour trouver les paramètres. En effet, chaque traitement est effectué indépendamment mais les résultats d'un traitement influent sur les résultats du

traitement suivant. Le choix des paramètres est un des problèmes que j'ai rencontré pour le test de la plateforme.

5.3.1. Les chaînes à base de SVM

Nous avons présenté un système de reconnaissance automatique de la parole (RAP) indépendant du locuteur basé sur une combinaison parallèle des classifieurs SVM multiclasse. Ce système proposé utilise comme moteur de reconnaissance les deux Stratégie principales, *un contre un*, et *un contre tous* pour éviter des ambiguïtés.

Pour l'apprentissage, nous disposons de 46 locuteurs (18 hommes et 28 femmes), pour *SVM un contre un*, et 48 locuteurs différents (19 hommes et 29 femmes), pour le classifieur *SVM un contre tous*. Pour le test, nous Arrangeons de 113 locuteurs (56 hommes et 57 femmes). Ce qui représente environ 96 points d'entraînement et 226 points de test pour chaque chiffre.

5.3.1.1. Chaîne à base de SVM RBF un-contre-tous

On peut ensuite essayer de réaliser la chaîne présentée dans *Using Support Vector Machines for Spoken Digit Recognition* d'Issam **Bazzi** et **Dina Katabi**. Les tests montrent le résultat suivant :

Chaînes de reconnaissance automatique de la parole Chaîne		**Taux de reconnaissance**
	Nb axes caractéristiques	
Préaccentuation – MFCC – Réduction – ACP – SVM RBF un contre tous	**45**	**96.90**

On utilise dans notre système 14 coefficients. On a réduit l'ensemble des trames issues de l'analyse MFCC à 30 trames, en utilisant la méthode de réduction expliquée dans **[47].** À la fin de l'analyse des caractéristiques, on obtient donc un vecteur de 30x14 = 420 dimensions. Enfin, on effectue une analyse en composante principale pour réduire le vecteur à 45 dimensions. Les hyper paramètres γ (Gamma) et C ont été déterminés, empiriquement. Les valeurs retenues ($C = 10$ et $\gamma = 0.0038$) ont permis d'obtenir un taux d'erreurs de 0.00 % sur la base d'apprentissage (test par le corpus d'apprentissage) et un taux d'erreur de 3.10 % sur la base de test.

La matrice de confusion est la suivante :

	1	2	3	4	5	6	7	8	9	Oh	zéro	%
1	**96.90**	0.00	0.00	1.33	0.44	0.00	0.00	0.44	0.44	0.00	0.44	100
2	0.00	**96.90**	0.00	0.00	0.00	0.88	0.00	0.44	0.00	0.88	0.88	100
3	0.00	0.00	**98.23**	0.00	0.00	0.44	1.33	0.00	0.00	0.00	0.00	100
4	0.00	0.00	0.00	**97.35**	0.00	0.00	0.00	0.44	0.00	2.21	0.00	100
5	0.00	0.00	0.00	0.88	**96.46**	0.88	0.88	0.00	0.00	0.88	0.00	100
6	0.00	1.33	0.00	0.00	0.00	**93.36**	2.21	3.10	0.00	0.00	0.00	100
7	0.00	0.00	0.00	0.00	0.44	0.00	**97.79**	0.00	0.88	0.88	0.00	100
8	0.00	0.88	0.44	0.00	0.00	0.88	0.00	**97.79**	0.00	0.00	0.00	100
9	0.88	0.00	0.00	0.00	1.77	0.00	0.44	0.00	**96.90**	0.00	0.00	100
Oh	0.00	0.00	0.00	1.33	0.44	0.00	0.44	0.00	0.00	**97.79**	0.00	100
zéro	0.00	0.00	0.00	0.00	0.00	0.00	0.44	0.00	0.00	0.00	**99.56**	100

Les classes ont des taux de classifications homogènes.

L'ajout d'analyse discriminante linéaire (ADL) à la chaîne précédente donne les résultats suivants :

Chaînes de reconnaissance automatique de la parole Chaîne		**Taux de reconnaissance**
	Nb axes caractéristiques	
Préaccentuation – MFCC – Réduction – (ACP⊗ADL) – SVM RBF un contre tous	**40**	**95.82**

L'algorithme ACP réduit la dimensionnalité de l'espace en éliminant les valeurs propres les plus faibles, et l'ADL a pour objectif, de maximiser les variations inter-classes tout en minimisant les variations intra -classes.

La matrice de confusion pour la chaîne *Préaccentuation – MFCC – Réduction –* (*ACP⊗ADL*) *– SVM RBF un contre tous* est la suivante :

	1	2	3	4	5	6	7	8	9	Oh	zéro	%
1	**96.02**	0.00	0.44	1.33	0.88	0.00	0.00	1.33	0.00	0.00	0.00	100
2	0.00	**95.13**	0.88	0.00	0.00	2.21	0.00	1.33	0.00	0.44	0.00	100
3	0.00	0.88	**97.35**	0.00	0.00	0.00	1.33	0.44	0.00	0.00	0.00	100
4	0.00	0.00	0.00	**95.58**	0.00	0.00	0.44	0.88	0.00	3.10	0.00	100
5	0.44	0.00	0.00	1.33	**92.92**	3.10	0.88	0.00	0.00	1.33	0.00	100
6	0.00	1.33	0.44	0.00	0.00	**93.81**	0.88	3.10	0.00	0.00	0.44	100
7	0.00	0.00	0.00	0.00	0.44	1.33	**96.46**	0.00	0.44	1.33	0.00	100

8	0.00	1.77	0.88	0.00	1.33	0.88	0.44	**94.25**	0.00	0.44	0.00	100
9	0.44	0.00	0.00	0.00	2.21	0.00	0.88	0.00	**96.46**	0.00	0.00	100
Oh	0.00	0.00	0.00	0.00	0.00	0.44	0.88	0.00	0.00	**98.67**	0.00	100
zéro	0.00	0.88	0.00	0.00	0.00	0.88	0.00	0.00	0.88	0.00	**97.35**	100

Les classes ont des taux de classifications homogènes.

A. Normalisation des entrées par une analyse statistique

Si on utilise une analyse statistique à la place de l'analyse en composante principale, on obtient les résultats suivants :

Chaînes de reconnaissance automatique de la parole	**Taux de reconnaissance**
Préaccentuation – MFCC – Statistique – SVM RBF un contre tous	**78.12**

Les résultats présentent un taux de reconnaissance moyen bien inférieur que la précédente chaîne. En effet, le signal est fortement dégradé puisqu'il n'est représenté que par une Gaussienne. Sachant que, les valeurs retenues des hyper paramètres γ (Gamma) et C, sont ($C = 10$ et $\gamma = 0.048$)

La matrice de confusion est qui suit :

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**Oh**	**zéro**	**%**
1	**69.03**	0.00	0.44	6.64	0.44	0.00	7.96	0.44	3.98	4.87	6.19	100
2	0.44	**62.39**	24.78	0.00	0.00	3.54	2.21	1.33	0.44	0.88	3.98	100
3	0.00	14.60	**69.03**	0.00	0.00	1.33	5.31	2.21	0.44	0.00	7.08	100
4	5.31	0.00	0.44	**81.86**	1.33	0.44	0.88	0.88	0.44	7.96	0.44	100
5	0.00	0.00	0.00	0.00	**89.82**	0.00	0.00	0.00	4.87	4.87	0.44	100
6	0.00	2.21	0.00	0.00	0.00	**92.48**	2.21	3.10	0.00	0.00	0.00	100
7	0.00	1.33	0.00	0.00	0.88	7.96	**79.20**	0.00	1.77	0.88	7.96	100
8	0.00	0.88	0.44	0.00	0.00	5.31	0.00	**93.36**	0.00	0.00	0.00	100
9	5.31	0.44	0.00	0.44	9.29	2.21	14.60	0.00	**61.95**	0.00	5.75	100
Oh	0.44	0.44	0.44	10.62	3.54	0.44	0.88	0.44	0.00	**80.97**	1.77	100
zéro	2.21	1.77	7.52	0.00	0.00	0.00	6.64	0.00	1.33	1.33	**79.20**	100

A part les classes 1, 2, 3,9, on obtient des taux de classifications homogènes.

B. Normalisation des entrées par l'algorithme proposé

On peut essayer d'échanger la méthode de réduction par l'algorithme d'alignement proposé. On utilise dans notre système 14 coefficients. On a réduit l'ensemble des trames issues de l'analyse *MFCC* à 32 trames en utilisant notre algorithme d'alignement proposé. À

la fin de l'analyse des caractéristiques, on effectue une analyse en composante principale pour réduire le vecteur à 45 dimensions. Les hyper paramètres γ (Gamma) et C ont été déterminés, empiriquement. Les valeurs retenues (C = 10 et γ = 0.0017) ont permis d'obtenir un taux d'erreurs de 0.00 % sur la base d'apprentissage (*test par le corpus d'apprentissage*) et un taux d'erreur de 4.18% sur la base de test, comme indiqué dans le tableau suivant :

Chaînes de reconnaissance automatique de la parole		**Taux de reconnaissance**
	Nb axes caractéristiques	
Préaccentuation – MFCC – Normalisation des entrées – ACP – SVM RBF un contre tous	**45**	**95.82**

On remarque que cela n'améliore pas toujours ou cela améliore un peu. Mais cela à l'avantage de diminuer fortement la taille des vecteurs et de gagner ainsi en rapidité dans la phase de classification. De plus, les méthodes de l'alignement temporel ont le désavantage de supprimer de l'information utile

La matrice de confusion pour cette chaîne est la suivante :

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**Oh**	**zéro**	**%**
1	**97.35**	0.00	0.88	0.88	0.00	0.44	0.00	0.44	0.00	0.00	0.00	100
2	0.00	**95.13**	0.00	0.00	0.00	1.33	0.00	1.33	0.00	0.88	1.33	100
3	0.00	0.00	**97.79**	0.00	0.00	0.88	0.88	0.44	0.00	0.00	0.00	100
4	0.00	0.00	0.00	**96.02**	0.44	0.00	0.00	0.44	0.00	3.10	0.00	100
5	0.44	0.00	0.44	0.44	**96.02**	0.44	0.88	0.00	0.00	1.33	0.00	100
6	0.00	1.77	0.88	0.00	0.00	**89.82**	4.42	3.10	0.00	0.00	0.00	100
7	0.00	0.00	0.88	0.00	0.00	4.87	**92.48**	0.00	0.44	1.33	0.00	100
8	0.00	1.33	0.88	0.00	0.00	0.44	0.00	**97.35**	0.00	0.00	0.00	100
9	1.33	0.00	1.33	0.00	3.10	0.00	0.44	0.00	**92.92**	0.00	0.88	100
Oh	0.00	0.44	0.00	3.10	0.44	0.00	1.77	0.00	0.00	**94.25**	0.00	100
zéro	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.44	0.00	0.00	**99.56**	100

Les classes ont des taux de classifications homogènes.

Si on utilise la fusion de (ACP) et (ADL) dans la même chaîne avec l'algorithme d'alignement proposé. Les tests montrent le résultat suivant :

Chaînes de reconnaissance automatique de la parole Chaîne		**Taux de reconnaissance**
	Nb axes caractéristiques	
Préaccentuation – MFCC – Normalisation des entrées – (ACP⊗ADL) – SVM RBF un contre tous	**20**	**92.68**

Le taux de reconnaissance obtenu par cette chaîne est acceptable.

La matrice de confusion montre ces résultats :

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**Oh**	**zéro**	**%**
1	**94.25**	0.00	0.44	1.77	1.33	0.00	0.44	0.44	0.88	0.44	0.00	100
2	0.00	**93.81**	0.44	0.00	0.00	0.44	1.77	2.21	0.44	0.44	0.44	100
3	0.00	0.44	**96.90**	0.00	0.00	0.44	1.33	0.44	0.00	0.00	0.44	100
4	1.33	0.00	0.44	**89.82**	0.00	0.00	0.44	0.44	0.00	7.52	0.00	100
5	0.00	0.00	0.44	0.88	**93.36**	0.00	2.65	0.00	0.44	2.21	0.00	100
6	0.00	1.33	0.00	0.00	0.00	**86.73**	6.19	5.75	0.00	0.00	0.00	100
7	0.00	0.44	0.44	0.00	1.77	6.19	**85.40**	2.21	1.33	2.21	0.00	100
8	0.00	0.88	3.10	0.00	0.00	3.10	0.00	**92.04**	0.00	0.88	0.00	100
9	1.33	0.00	0.88	0.00	2.65	0.00	0.44	0.00	**93.36**	0.00	1.33	100
Oh	0.44	0.00	0.00	0.88	0.88	0.00	1.77	0.44	0.00	**95.58**	0.00	100
zéro	0.00	1.33	0.00	0.00	0.00	0.00	0.00	0.44	0.00	0.00	**98.23**	100

5.3.1.2. Chaîne à base de SVM RBF un-contre-un

Si on utilise le classifieur *SVM RBF un-contre-un* à la place du classifieur *SVM RBF un-contre-tous* avec un noyau Gaussien avec des hyper paramètres de ($C = 10$ et $\gamma = 0.0038$) et avec une ACP avec 45 composantes, on obtient les résultats suivants :

Chaînes de reconnaissance automatique de la parole Chaîne		**Taux de reconnaissance**
	Nb axes caractéristiques	
Préaccentuation – MFCC – Réduction – ACP – SVM RBF un contre un	**45**	**99.20**

Les résultats présentent un taux de reconnaissance moyen bien supérieur que la précédente chaîne, En effet, le classifieur *SVM RBF un contre un* beaucoup plus performant que le classifieur *SVM RBF un contre tous*. La matrice de confusion permet une meilleure compréhension du taux de reconnaissance moyen :

	1	2	3	4	5	6	7	8	9	Oh	zéro	%
1	**99.56**	0.00	0.00	0.00	0.00	0.00	0.44	0.00	0.00	0.00	0.00	100
2	0.00	**99.56**	0.00	0.00	0.00	0.00	0.44	0.00	0.00	0.00	0.00	100
3	0.00	0.00	**99.56**	0.00	0.44	0.00	0.00	0.00	0.00	0.00	0.00	100
4	0.88	0.00	0.00	**98.67**	0.00	0.00	0.00	0.00	0.00	0.44	0.00	100
5	0.00	0.00	0.00	0.00	**99.12**	0.00	0.44	0.44	0.00	0.00	0.00	100
6	0.00	0.00	0.00	0.00	0.00	**99.56**	0.00	0.44	0.00	0.00	0.00	100
7	0.00	0.00	0.00	0.00	0.00	0.00	**99.12**	0.00	0.44	0.44	0.00	100
8	0.44	0.44	0.00	0.00	0.00	0.00	0.00	**99.12**	0.00	0.00	0.00	100
9	0.44	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**99.56**	0.00	0.00	100
Oh	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**100**	0.00	100
zéro	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**100**	100

Si on fait la combinaison en série de l'ACP et l'ADL dans la précédente chaîne de reconnaissance. Les résultats sont reportés dans le tableau suivant :

Chaînes de reconnaissance automatique de la parole Chaîne		**Taux de reconnaissance**
	Nb axes caractéristiques	
Préaccentuation – MFCC – Réduction – (ACP⊗ADL) – SVM RBF un contre un	**40**	**98.51**

Le taux de reconnaissance fixé au tableau montre que la classification utilisant l'algorithme *SVM RBF un contre un* est plus performant que la classification utilisant l'algorithme *SVM RBF un contre tous*.

La matrice de confusion est la suivante :

	1	2	3	4	5	6	7	8	9	Oh	zéro	%
1	**99.56**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.44	0.00	0.00	100
2	0.44	**98.23**	0.44	0.00	0.00	0.88	0.00	0.00	0.00	0.00	0.00	100
3	0.00	0.00	**98.67**	0.00	0.00	0.44	0.44	0.00	0.00	0.00	0.44	100
4	1.33	0.00	0.00	**98.23**	0.44	0.00	0.00	0.00	0.00	0.00	0.00	100
5	0.00	0.00	0.00	0.44	**99.12**	0.00	0.44	0.00	0.00	0.00	0.00	100
6	0.00	0.88	0.44	0.88	0.00	**96.46**	0.88	0.44	0.00	0.00	0.00	100
7	0.00	0.88	0.00	0.00	0.44	0.00	**97.35**	0.44	0.44	0.44	0.00	100
8	0.44	0.00	0.00	0.00	0.00	0.00	0.88	**98.67**	0.00	0.00	0.00	100
9	0.00	0.00	0.00	0.00	0.00	0.44	0.00	0.00	**98.67**	0.44	0.44	100
Oh	0.00	0.00	0.00	0.44	0.00	0.00	0.00	0.44	0.00	**99.12**	0.00	100
zéro	0.00	0.00	0.00	0.00	0.00	0.00	0.44	0.00	0.00	0.00	**99.56**	100

A. normalisation des entrées par une analyse statistique

Chaînes de reconnaissance automatique de la parole	Taux de reconnaissance
Préaccentuation – MFCC – Statistique – SVM RBF un contre un	**86.04**

Les résultats présentent un taux de reconnaissance moyen bien inférieur que la précédente chaîne. En effet, le signal est fortement dégradé puisqu'il n'est représenté que par une Gaussienne. Les valeurs des hyper paramètres optimisées, sont : c=10 et Gamma=0.048.

La matrice de confusion pour la chaîne *Préaccentuation – MFCC – Statistique – SVM RBF un contre un* est la suivante :

	1	2	3	4	5	6	7	8	9	Oh	zéro	%
1	**86.28**	0.44	0.00	3.98	0.00	0.00	0.88	0.44	2.65	2.65	2.65	100
2	0.44	**75.66**	17.26	0.00	0.00	1.33	0.88	1.77	0.88	0.00	1.77	100
3	0.00	17.26	**71.68**	0.00	0.00	0.44	0.00	1.33	0.88	0.00	8.41	100
4	1.33	0.00	0.00	**92.04**	0.88	0.44	0.00	0.00	0.00	4.42	0.88	100
5	0.00	0.00	0.00	2.21	**95.13**	0.00	0.00	0.00	1.77	0.88	0.00	100
6	0.00	2.21	0.44	0.00	0.00	**94.25**	1.33	1.77	0.00	0.00	0.00	100
7	2.21	1.33	2.21	0.00	0.00	1.77	**78.76**	0.00	7.52	0.00	6.19	100
8	1.33	0.00	0.00	0.44	0.00	0.44	0.44	**96.90**	0.00	0.00	0.44	100
9	3.98	0.88	0.00	0.44	7.52	0.00	4.87	0.00	**81.42**	0.88	0.00	100
Oh	1.33	0.00	0.00	7.96	3.98	0.00	0.00	0.44	0.00	**86.28**	0.00	100
zéro	1.77	2.65	2.21	0.88	0.00	0.44	2.65	0.00	1.33	0.00	**88.05**	100

A part les classes 2 et 3, on obtient des taux de classifications homogènes.

B. Normalisation des entrées par l'algorithme proposé

La chaîne de traitement du signal est identique à celle utilisé pour *SVM RBF un contre tous*. J'obtiens au final les résultats figurant dans le tableau suivant :

Chaînes de reconnaissance automatique de la parole		Taux de reconnaissance
	Nb axes caractéristiques	
Préaccentuation – MFCC – Normalisation des entrées – ACP – SVM RBF un contre un	**45**	**98.75**

Par simple comparaison de nos résultats (**SVM RBF un contre tous** / **SVM RBF un contre un**) nous remarquons que la reconnaissance par un classifieur utilisant la stratégie *un*

contre un et beaucoup plus satisfaisante par rapport au classifieur employant la stratégie *un contre tous.* Les valeurs des hyper paramètres optimisées, sont : c=10 et Gamma=0.0017.

La matrice de confusion de la chaîne : *Préaccentuation – MFCC – Normalisation des entrées – ACP – SVM RBF un contre un* est la suivante :

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**Oh**	**zéro**	**%**
1	**99.56**	0.00	0.00	0.44	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100
2	0.00	**98.67**	0.00	0.00	0.00	0.00	0.00	0.44	0.00	0.00	0.88	100
3	0.00	0.00	**99.56**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.44	100
4	0.44	0.00	0.00	**98.23**	0.00	0.00	0.00	0.00	0.00	0.44	0.88	100
5	0.44	0.00	0.00	0.00	**98.67**	0.00	0.00	0.00	0.44	0.44	0.00	100
6	0.00	0.44	0.88	0.00	0.00	**97.35**	0.88	0.44	0.00	0.00	0.00	100
7	0.00	0.88	0.44	0.00	0.88	0.88	**96.02**	0.00	0.00	0.88	0.00	100
8	0.00	0.44	0.00	0.00	0.00	0.00	0.00	**99.56**	0.00	0.00	0.00	100
9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**99.56**	0.00	0.44	100
Oh	0.00	0.00	0.00	0.88	0.44	0.00	0.00	0.00	0.00	**98.67**	0.00	100
zéro	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.88	0.00	**99.12**	100

Si on emploi l'ACP et le ADL dans la même chaîne avec l'algorithme d'alignement proposé avec les même valeurs des hyper paramètres. Les résultats sont les suivant :

Chaînes de reconnaissance automatique de la parole		**Taux de reconnaissance**
	Nb axes caractéristiques	
Préaccentuation – MFCC – Normalisation des entrées – (ACP⊗ADL) – SVM RBF un contre un	**20**	**97.23**

Le taux obtenu est meilleur que celui obtenu par le *SVM RBF un contre tous* .La matrice de confusion permet une meilleure compréhension du taux de reconnaissance moyen :

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**Oh**	**zéro**	**%**
1	**98.23**	0.00	0.00	0.44	0.44	0.00	0.00	0.44	0.44	0.00	0.00	100
2	0.00	**96.90**	0.88	0.00	0.00	0.44	0.44	1.33	0.00	0.00	0.00	100
3	0.00	0.00	**99.56**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.44	100
4	0.44	0.44	0.00	**98.67**	0.00	0.00	0.00	0.00	0.00	0.00	0.44	100
5	0.00	0.00	0.00	1.33	**97.79**	0.44	0.00	0.00	0.44	0.00	0.00	100
6	0.44	1.77	0.44	0.00	0.00	**94.69**	1.33	1.33	0.00	0.00	0.00	100
7	0.00	3.54	0.44	0.44	1.33	1.33	**90.27**	0.44	0.00	1.77	0.44	100
8	0.44	0.44	0.44	0.00	0.00	0.88	0.00	**97.79**	0.00	0.00	0.00	100
9	0.00	0.00	0.00	0.00	0.00	0.00	0.44	0.00	**98.67**	0.00	0.88	100
Oh	0.00	0.00	0.00	0.88	0.88	0.00	0.00	0.44	0.00	**97.79**	0.00	100
zéro	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.88	0.00	**99.12**	100

5.3.1.3. Chaîne à base de combinaison de classifieurs : 'SVM RBF un-contre-tous' Et 'SVM RBF un-contre-un'

Nous avons utilisé l'intégrale floue de shoquet, pour fusionner les scores issus des systèmes *SVM RBF un contre un* et *SVM RBF un contre tous*, après plusieurs expériences sur cette méthode nous a démontré son efficacité dans l'augmentation des performances du système proposé comparant à d'autres méthodes de combinaison de scores. A la fin de ce chapitre nous démontrerons l'efficacité de l'intégrale floue de shoquet, en utilisant le système proposé, et en la comparant avec les méthodes de fusion de scores, qui ont été décrit dans la section (4.5.3). Les tests montrent le résultat suivant :

Chaînes de reconnaissance automatique de la parole		**Taux de reconnaissance**
Préaccentuation – MFCC – Réduction – ACP – SVM RBF un contre tous	---------->	**96.90**
Préaccentuation – MFCC – Réduction – ACP – SVM RBF un contre un	---------->	**99.20**
le classifieur proposé : ***(Fusion de scores issus des systèmes)***	**méthodes de fusion de scores**	
(*SVM RBF un contre tous*) ↘ ⊗ (*SVM RBF un contre un*) ↗	***Intégrale floue de Choquet***	**99.72**

Le résultat obtenu par rapport au classifieurs proposé est très satisfaisant. On peut donc dire que la combinaison de classifieurs peut améliorer les performances, sans augmenter la complexité du système. .La matrice de confusion permet une meilleure compréhension du taux de reconnaissance moyen :

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**Oh**	**zéro**	**%**
1	**100**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100
2	0.00	**100**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100
3	0.00	0.00	**99.12**	0.00	0.00	0.00	0.44	0.00	0.44	0.00	0.00	100
4	0.00	0.44	0.00	**99.12**	0.00	0.00	0.00	0.00	0.00	0.44	0.00	100
5	0.00	0.00	0.00	0.44	**99.56**	0.00	0.00	0.00	0.00	0.00	0.00	100
6	0.00	0.00	0.00	0.00	0.00	**99.56**	0.44	0.00	0.00	0.00	0.00	100
7	0.00	0.00	0.00	0.00	0.00	0.00	**100**	0.00	0.00	0.00	0.00	100
8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**100**	0.00	0.00	0.00	100
9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**100**	0.00	0.00	100
Oh	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**100**	0.00	100
zéro	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.44	0.00	**99.56**	100

A. normalisation des entrées par une analyse statistique

Les résultats sont fournis dans le tableau suivant :

Chaînes de reconnaissance automatique de la parole		**Taux de reconnaissance**
Préaccentuation – MFCC – Statistique – SVM RBF un contre tous	---------->	**78.12**
Préaccentuation – MFCC – Statistique – SVM RBF un contre un	---------->	**86.04**
le classifieur proposé : ***(Fusion de scores issus des systèmes)***	**méthodes de fusion de scores**	
(*SVM RBF un contre tous*) ⊗ (*SVM RBF un contre un*)	***Intégrale floue de Choquet***	**93.16**

Malgré, le signal est fortement dégradé puisqu'il n'est représenté que par une Gaussienne. Néanmoins, le classifieur proposé a été pu d'augmenter la performance de la classification. Les valeurs des hyper paramètres optimisées, sont : c=10 et Gamma=0.048.

La matrice de confusion est la suivante :

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**Oh**	**zéro**	**%**
1	**92.92**	0.44	0.44	2.65	0.44	0.00	0.00	0.00	0.44	0.88	1.77	100
2	0.00	**80.53**	18.14	0.00	0.00	0.00	0.00	0.00	0.88	0.00	0.44	100
3	0.00	14.60	**83.19**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2.21	100
4	0.44	0.00	0.00	**98.23**	0.44	0.00	0.00	0.00	0.00	0.88	0.00	100
5	0.00	0.00	0.00	0.00	**100**	0.00	0.00	0.00	0.00	0.00	0.00	100
6	0.00	0.00	0.44	0.00	0.00	**99.12**	0.00	0.44	0.00	0.00	0.00	100
7	0.00	0.00	0.00	0.00	0.00	0.44	**95.58**	0.00	3.10	0.00	0.88	100
8	0.00	0.88	0.44	0.00	0.00	0.44	0.44	**97.79**	0.00	0.00	0.00	100
9	0.88	0.00	0.00	0.00	2.65	0.44	5.75	0.00	**88.94**	0.00	1.33	100
Oh	0.00	0.00	0.00	3.98	0.88	0.00	0.00	0.44	0.00	**94.69**	0.00	100
zéro	0.88	1.33	2.65	0.00	0.00	0.00	0.44	0.00	0.44	0.44	**93.81**	100

On remarque un taux de classification assez bon pour la classe 5 et 6. On peut supposer qu'un apprentissage satisfaisant pour ces deux classes.

B. Normalisation des entrées par l'algorithme proposé

Chaînes de reconnaissance automatique de la parole		**Taux de reconnaissance**
Préaccentuation – MFCC – Normalisation des entrées – ACP – SVM RBF un contre tous	---------->	**95.82**
Préaccentuation – MFCC – Normalisation des entrées – ACP – SVM RBF un contre un	---------->	**98.75**
le classifieur proposé : ***(Fusion de scores issus des systèmes)***	**méthodes de fusion de scores**	
(*SVM RBF un contre tous*) ↘ ⊗ (*SVM RBF un contre un*) ↗	***Intégrale floue de Choquet***	***99.56***

Le système de combinaison de référence marque une amélioration du taux d'erreur de 0.44% tandis qu'une amélioration de 1.25% et 4.18% est obtenue respectivement par le classifieur SVM RBF un contre un et par le classifieur SVM RBF un contre tous.

La matrice de confusion donne les résultas suivants :

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**Oh**	**zéro**	**%**
1	**100**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100
2	0.00	**99.56**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.44	100
3	0.00	0.00	**99.56**	0.00	0.00	0.00	0.44	0.00	0.00	0.00	0.00	100
4	0.00	0.00	0.00	**99.56**	0.00	0.00	0.00	0.00	0.00	0.44	0.00	100
5	0.00	0.00	0.00	0.00	**99.56**	0.44	0.00	0.00	0.00	0.00	0.00	100
6	0.00	0.00	0.00	0.00	0.00	**99.56**	0.44	0.00	0.00	0.00	0.00	100
7	0.00	0.00	0.44	0.44	0.00	0.88	**98.23**	0.00	0.00	0.00	0.00	100
8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**100**	0.00	0.00	0.00	100
9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**99.56**	0.00	0.44	100
Oh	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**100**	0.00	100
zéro	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.44	0.00	0.00	**99.56**	100

Les classes ont des taux de classifications homogènes.

La figure 5.1 représente le taux de reconnaissance global en fonction du nombre de nombre de trames extraites (N) en utilisant, l'algorithme proposé de la normalisation des entrées. Le taux de reconnaissance atteint 99.56%, à partir de 32 trames extraites de chaque segment caractéristique. Des valeurs plus grandes (34 et 36 trames extraites) fournissent des résultats équivalents mais, il n'est pas nécessaire de les utiliser à cause du coût de temps d'apprentissage et de classification. *En effet, les systèmes de la reconnaissance de la parole sont crées pour une communication temps réel. La reconnaissance de la parole doit être*

rapide **[48]**. Nous remarquons que les performances de notre système se dégradent à partir de certain nombre de trames extraites (à partir de 38 trames).

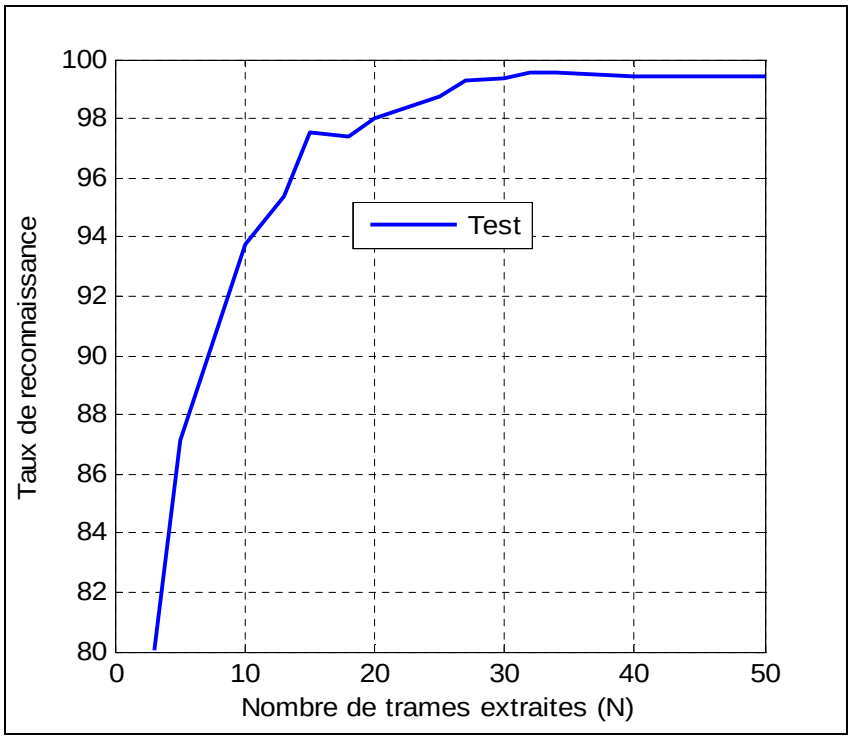


Figure 5.1. Influence du nombre de trames extraites sur les performances du système de RAP proposé.

Si on fusionne l'analyse en composantes principales (ACP) et l'analyse discriminante linéaire (ADL). Les valeurs des hyper paramètres optimisées, sont : Gamma = 0.0017 et C=10.

Chaînes de reconnaissance automatique de la parole		**Taux de reconnaissance**
Préaccentuation – MFCC – Normalisation des entrées – (ACP⊗LDA) – SVM RBF un contre tous	---------->	**92.68**
Préaccentuation – MFCC – Normalisation des entrées – (ACP⊗ADL) –SVM RBF un contre un	---------->	**97.23**
le classifieur proposé : ***(Fusion de scores issus des systèmes)***	**méthodes de fusion de scores**	
(*SVM RBF un contre tous*) ⊗ (*SVM RBF un contre un*)	***Intégrale floue de Choquet***	***99.28***

Le système de base marque une amélioration du taux de reconnaissance.

La matrice de confusion de la chaîne est :

	1	2	3	4	5	6	7	8	9	Oh	zéro	%
1	**99.56**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.44	0.00	0.00	100
2	0.00	**100**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100
3	0.00	0.00	**99.56**	0.00	0.00	0.00	0.44	0.00	0.00	0.00	0.00	100
4	0.00	0.00	0.00	**99.56**	0.00	0.00	0.00	0.00	0.00	0.44	0.00	100
5	0.00	0.00	0.00	0.00	**99.56**	0.00	0.00	0.00	0.00	0.44	0.00	100
6	0.00	0.00	0.00	0.00	0.00	**99.12**	0.44	0.44	0.00	0.00	0.00	100
7	0.00	0.00	0.00	0.00	0.44	0.44	**97.79**	0.44	0.88	0.00	0.00	100
8	0.00	0.00	0.00	0.00	0.00	0.88	0.00	**99.12**	0.00	0.00	0.00	100
9	0.44	0.00	0.00	0.00	0.00	0.00	0.44	0.00	**98.67**	0.00	0.44	100
Oh	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**100**	0.00	100
zéro	0.00	0.44	0.00	0.00	0.00	0.00	0.00	0.00	0.44	0.00	**99.12**	100

5.3.1.4. Comparaison ente les méthodes de fusion de scores

Pour comparer entre les différentes méthodes de fusion de scores nous avons utilisé plusieurs méthodes de réduction des entées, qui proviennent de phase d'extraction de paramètres, en utilisant le classifieur proposé comme technique de reconnaissance des formes.

Chaîne de reconnaissance de la parole	**Le système proposé : (Fusion de scores)**	**Méthodes de fusion de scores**	**Taux (fusion)**
MFCC – Réduction – ACP –SVM RBF un contre un	(***SVM RBF un contre tous***) ⊗ (***SVM RBF un contre un***)	La moyenne	**99.20**
		Le produit	**99.19**
		Le minimum	**99.67**
		Le maximum	**99.64**
		La médiane	**99.20**
MFCC – Réduction – ACP –SVM RBF un contre tous		La somme	**99.20**
		La logique floue	
		Intégrale floue de Sugeno	**99.64**
		Intégrante floue de Choquet	**99.72**

Chaîne de reconnaissance de la parole	Le système proposé : (Fusion de scores)	Méthodes de fusion de scores	Taux (fusion)
MFCC – Statistique – SVM RBF un contre un MFCC – Statistique – SVM RBF un contre tous	(***SVM RBF un contre tous***) ⊗ (***SVM RBF un contre un***)	La moyenne	**91.90**
		Le produit	**91.91**
		Le minimum	**99.67**
		le maximum	**92.96**
		La médiane	**91.92**
		La somme	**91.93**
		La logique floue	
		Intégrale floue de Sugeno	**92.96**
		Intégrante floue de Choquet	**93.16**

Chaîne de reconnaissance de la parole	Le système proposé : (Fusion de scores)	Méthodes de fusion de scores	Taux (fusion)
MFCC – Normalisation des entrées – ACP – SVM RBF un contre tous MFCC – Normalisation des entrées – ACP – SVM RBF un contre un	(***SVM RBF un contre tous***) ⊗ (***SVM RBF un contre un***)	La moyenne	**99.39**
		Le produit	**99.39**
		Le minimum	**99.55**
		le maximum	**97.14**
		la médiane	**99.40**
		La somme	**99.39**
		La logique floue	
		Intégrale floue de Sugeno	**99.55**
		Intégrante floue de Choquet	**99.56**

Chaque méthode de fusion des scores a ses avantages et inconvénients, les résultats dans les tableaux ci dessus, démontrent clairement l'utilité de la méthode de l'intégrale floue de shoquet, pour fusionner les scores issus des *SVM RBF un contre un* et *SVMRBF un contre tout*. L*'intégrale floue de shoquet* est plus adaptée dans notre cas, qui consiste à construire un nouveau classifieur utilisant la combinaison des classifieurs (deux) de type *SVM*. Les taux de reconnaissance obtenus par le classifieur proposé, qui ont été chargés dans les tableaux ci dessus spécifique à chaque méthode de fusion de scores, ont votés au profit de la méthode *l'intégrale floue de shoquet*.

5.4. Discussion des résultats obtenus

Les taux de reconnaissance obtenus par le processus de classement proposé, pour chacune de ces chaînes, sont encourageants. Cependant, ils peuvent s'expliquer facilement. Tout d'abord on ne prend pas en compte la modélisation phonétique des chiffres prononcés, on utilise la technique *SVM*, qui a la capacité à généraliser la classification. Ainsi, cette méthode est adaptée aux applications présentant une grande variation intra-classes, comme par exemple la parole.

En suite, la méthode de classification, utilisée, est basée sur la combinaison des classifieurs. Cette approche a montré son aptitude de concevoir des systèmes puissants et performants.

Troisièmement, les algorithmes de classification ont été modifiés pour être adaptés à la reconnaissance de la parole. La méthode de fusion de scores issus des classifieurs (*SVM RBF un contre tous*) et (*SVM RBF un contre un*), et la fonction noyau de Hilbert-Schmidt sont choisies après plusieurs expériences.

Enfin, les hyper-paramètres ont à chaque fois été optimisés. Car, il n'existe pas de techniques pour trouver les meilleurs paramètres. En effet, ils dépendent fortement du problème étudié.

Conclusion générale et perspective

Malgré les efforts et les travaux intensifs réalisés dans le domaine de la *Reconnaissance Automatique de la Parole*, aucun système RAP n'est jugé fiable à 100%. Mais au fur et à mesure les auteurs essayent d'améliorer les scores pour de meilleurs résultats.

A notre connaissance, tous les résultats présentés dans la littérature montrent clairement que la combinaison parallèle de classifieurs est une voie de recherche prometteuse. Elle permet d'améliorer les performances du système de reconnaissance et de s'adapter à une grande variété de situations par l'exploitation de connaissances à priori sur les comportements des classifieurs.

Les travaux effectués sur la combinaison montrent aussi la variété des méthodes de combinaison qui diffèrent par leur capacité d'apprentissage, le type de sortie des classifieurs et la manière de les choisir (sélection ou non). Toutefois, ces travaux qui peuvent être optimisés pour des problèmes donnés **[84]** sont difficilement généralisables en dehors d'un domaine applicatif donné. En effet, le développement d'un système de combinaison ne suit a priori aucune règle précise, et dépend étroitement de l'application que l'on veut traiter, de la façon dont on veut la traiter et des outils disponibles (bases de données, classifieurs et règles de combinaison). Ces contraintes expliquent bien les limites des systèmes développés du point de vue performance pour traiter d'autres applications plus complexes.

Dans ce travail nous avons développé un système de reconnaissance de chiffres parlés anglais, en mode indépendant du locuteur, basé sur une combinaison parallèle de classifieurs SVM multiclasse (Les deux approches classiques, one vs. one et one vs. all ont été mises en oeuvre pour éviter des ambiguïtés). Plus, nous avons présenté une méthode de normalisation des entrées qui a prouvé sa performance du point de vue taux de reconnaissance. Cependant, on peut dire, que l'algorithme proposé pour la normalisation des entrées, valable uniquement dans le cas du traitement des Signaux sonores (signaux voisés). Parce que, nous avons utilisé une base de données de sons et n'est pas d'autre chose comme (Traitement d'image). Certainement, pour généraliser l'application de cet algorithme nécessite d'autres genres de signaux afin de tester son efficacité.

Les résultats obtenus par rapport au classifieurs de base sont très satisfaisants. On peut donc conclure que la combinaison de classifieurs permet d'améliorer les performances, sans

augmenter la complexité du système. Le meilleur taux de reconnaissance moyen obtenu par le système proposé est de 99.72%. Cependant, plusieurs perspectives sont envisagées pour améliorer ce taux.

Comme perspective, il est préférable d'introduire dans la phase d'analyse l'***énergie*** et le ***TPZ*** (***ZCR*** en anglais) sont des paramètres qui peuvent améliorer les performances des systèmes de reconnaissance. L'*énergie* correspond à la puissance du signal. Elle est souvent évaluée sur plusieurs trames successives pour pouvoir mettre en évidence la non stationnarité du signal vocal. Le *TPZ* représente le nombre de fois où le signal, dans sa représentation amplitude/temps, passe par la valeur centrale de l'amplitude (généralement zéro). Il est fréquemment employé pour des algorithmes de détection de segment voisé/non voisé dans un signal de parole. En effet, du fait de sa nature aléatoire, le bruit possède généralement un *TPZ* supérieur à celui des parties voisées.

L'introduction du paramètre énergie dans la phase d'analyse permet d'évaluer le degré d'accentuation des sons **[85]**, le TPZ permet d'identifier les sons fricatifs parmi les non-fricatifs.

Références

[1] **Francis Collet,** « *Aide-mémoire TRAITEMENT DU SIGNAL* ». Dunod, paris 2000, ISBN 2 10 051687 Code 045168

[2] **Rais EL'hadi BEKKA** « *Fondement du traitement du signal* » .*3emm*e Edition- OPU .2005.

[3] Quantification (signal) Wikipédia.mht « Recherche www.google.fr » (11/05/2009)

[4] http://fr.wikipedia.org/w/index.php?title=Processus_stochastique&action=edit&section=4. (11/05/2009)

[5] Href=«http://fr.wikipedia.org/w/index.php?title=Bruit&action=edit". (11/06/2009)

[6] http://fr.wikipedia.org/w/index.php?title=Bruit&action=edit&section=4. (11/06/2009)

[7] **Le Manh Tuan** « *Analyse des voyelles spéciale du Vitnamien* ». Rapport final du tipe .Institut de la Francophonie pour l'Informatique En collaboration avec le Centre de Recherche MICA, Hanoi.2005

[8] **Thierry Dutoit**. « *Introduction au Traitement Automatique de la Parole* ». Faculté Polytechnique de Mons, Belgique, 2000

[9] **Dominique Genoud**. «*Reconnaissance et transformation de Locuteurs*». Thèse EPFL, no 1924 (1999).

[10] **René Boite et Murat Kunt**. « *Traitement de la parole* ». Presses Polytechniques Romandes, 1987

[11] **Guillaume Madre** « *Application de la transformée en nombres entiers à l'étude et au développement d'un codeur de parole pour transmission sur réseaux I*». Thèse de Doctorat, Université de Bretagne Occidentale - Ecole Doctorale SMIS. 2004

[12] **José Anibal ARIAS AGUILAR**, « *Méthodes à vecteurs de support et Indexation Sonore* ». Thèse de Doctorat. Laboratoire IRIT de **T**oulouse. Année 2003-2004.

[13] **LÊ Viet Bac**, « *Reconnaissance automatique de la parole pour des langues peu dotées* ».Thèse de doctorat en Informatique, université joseph fourier- GRENOBLE 1. juin 2006

[14] **Julien ALLEGRE**, « *Approche de la reconnaissance automatique de la parole* » Rapport cycle probatoire (CNAM, Centre Régional Languedoc-Roussillon) Année 2003

[15] http://membres.lycos.fr/guillaumerey/reconnaissance_principes.htm. (13/01/2011)

[16] http://perso.orange.fr/xcotton/electron/coursetdocs.htm. (13/01/2011)

[17] JOHN. Makhoul, «*Spectral linear prediction, properties and applications* » Journal: IEEE Transactions on Acoustics, Speech, and Signal Processing , vol. 23, no. 3, pp. 283-296, 1975.

[18] Tounsi bilal , *« Inférence d'identité dans le domaine forensique en utilisant un système de reconnaissance automatique du locuteur adapté au dialecte Algérien* ». Thèse de Doctorat, I.N.I, Oued-Smar -Alger, Année2007/2008

[19] Benjamin LECOUTEUX, « *Reconnaissance automatique de la parole guidée par des transcriptions a priori* » Thèse de Doctorat, Université d'Avignon et des Pays de Vaucluse, Spécialité : Informatique, École Doctorale 166 I2S, année 2008

[20] Hynek Hermansky et Louis Anthony Cox, « *Perceptual linear predictive (plp) analysis resynthesis technique* ». In IEEE ASSP Workshop on Applications of Signal Processing to Audio and Acoustics Final Program and Paper Summaries, pages 037–038. 1991.

[21] John Bridle MARKEL & A H GREY, « Linear prediction of speech, Communications and cybernetics». Vol. 12, Berlin, Springer. Ph MARTIN, 1981.

[22] Teddy DIDÉ, «*Réalisation d'un Framework Pour la reconnaissance de la parole* », Rapport de stage, Ecole Polytechnique, Université François Rabelais Tours, Années 2007

[23] Hynek Hermansky et al, «*Perceptually Based Linear Predictive Analysis of Speech* », paper appears in: Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP. 1985

[24] Hynek Hermansky, Nelson Morgan, Aruna Bayya, Phil Kohn. « *Compensation effect of the communication channel in auditory-like analysis of speech* ». Book, in Eurospeech, pages 1367-1370, September 1991.

[25] Haton, *«Contribution à l'analyse, la paramétrisation et la reconnaissance de* la parole », Thèse de Doctorat d'État, Université de Nancy I, 1974.

[26] Xuechuan Wang, « *Feature Extraction and Dimensionality Reduction in Pattern Recognition and their Application in Speech Recognition ».* Article (Refereed Article), 2003.

[27] Laurence R Rabiner, « *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition»*, Proceedings of the IEEE, 77(2):257--286, February 1989

[28] Robinson, « *A recurrent error propagation network speech recognition system»*, Journal: Computer Speech & Language, vol. 5, no. 3, pp. 259-274, 1991.

[29] Farid Melgani and Lorenzo Bruzzone, « *Classification of hyperspectral remote sensing images with support vector machines*». Paper appears in: IEEE Transactions on Geoscience and Remote Sensing, vol. 42, nº 8, pp. 1778-1790, Août 2004.

[30] Mercade Medina Sergio, «*Classification d'images hyperspectrales pour la caractérisation du milieu urbain par une approche multirésolution* » Rapport de Stage, ENSERG et Télécom – ENSIMAG, Grenoble France, Juin 2008

[31] Vladimir Vapnik, « *Statistical Learning Theory »*, Book .1998.

[32] Mohamadally Hasan, Fomani Boris, *« SVM machine a vecteurs de support ou séparateur a vaste marge »*. Article. BD Web, ISTY3, Versailles St Quentin, France, janvier 2006.

[33] Jeremy Mary, « *Méthodes d'apprentissage avancées* ». *Présentation ppt.* **C**entre **N**ational de la **R**echerche **S**cientifique. Université de Lille. janvier2006.

[34] Antoine Cornuéjols, « *Une nouvelle méthode d'apprentissage : les SVM. Séparateur à vaste marge* ». Article. Université de Paris-Sud, Orsay. Juin 2002.

[35] Jamal Kharroubi, « *Etude de Techniques de Classement_ Machines à Vecteurs Supports pour la Vérification Automatique du Locuteur* ». Thèse de doctorat. Ecole Nationale Supérieure des Télécommunications. Paris .juillet 2002

[36] Olivier Bousquet, « *Introduction aux Support Vector Machines* ». Présentation ppt. Centre de Mathématiques Appliquées, Ecole Polytechniques, Palaiseau. Novembre 2001.

[37] Bernhard Scholkôpf, « Support ***vector learning*** ». Thesis, R. Oldenbourg Verlag, Munich, 1997

[38] Jon Atli Benediktsson and Ioannis Kanellopoulos, **«** *Classification of multisource and hyperspectral data based on decision fusion"*. Paper appears in: IEEE Transactions on Geoscience and Remote Sensing, vol. 37, pp. 1367-1377, Mai 1999.

[39] Channussot J, « *Approches vectorielles ou marginales pour le traitement d'images multi-composantes* ».PhD thèse, Université de Savoie, France, 1998.

[40] John Shawe-Taylor & Nello Cristianini, **«** *Kernel methods for pattern analys*is ». Book. Cambridge University Press, 2004.

[41] Philippe Besse. *« Apprentissage Statistique & Data mining ». Cours.* INSA de Toulouse, France. Juillet 2009

[42] Mathieu Fauvel, « *Spectral and spatial methods for the classification of urban remote sensing data* ». PhD thèse, Institut National Polytechnique de Grenoble, France, 2007.

[43] Philip Clarkson, Pedro J. Moreno, *« On the use of support vector machines for phonetic classification»*. Article. In: *Proceedings of International Conference on Acoustics, Speech, Signal Processing*, pp. 585–588. 1999.

[44] Bernd Heisele, Purdy Ho, Jane Wu, and Tomaso Poggio, *« Face recognition: component-based versus global approaches »*, Article. Computer Vision and Image Understanding Volume 91, Issue 1-2 , Special issue on Face recognition, Pages: 6–21, ISSN: 1077-3142 . July 2003

[45] Chih-Wei Hsu et Chih-Jen Lin *A*, *« Comparison of Methods for Multi-class Support Vector* », Article In: IEEE Transactions on Neural Networks, Vol. 13, Nr. 2 (2002) , p. 415--425.

[46] John C. Platt « *Probabilities for sv machines* »**,** publication in Advances in Large Margin Classifiers, MIT Press, March 1999.

[47] Bazzi, Issam / Katabi, Dina (2000): "Using support vector machines for spoken digit recognition", In /ICSLP-2000, vol.1, 433-436.

[48] Woo-Yong Choi, Dosung Ahn, Sung Bum Pan, Kyo II Chung, Yongwha Chung, Sang-Hwa Chung. « *SVM-Based Speaker Verification System for Match-on-Card and Its Hardware Implementation* » ETRI Journal, Volume 28, Number 3, June 2006

[49] Nathan Smith, Mark Gales Gale « *Speech Recognition using SV* ». Article. Cambridge University , CB2 1PZ, U.K 2002.

[49] Xuechuan Wang, Kuldip K. Paliwal « *Feature extraction and dimensionality reduction algorithms and their applications in vowel recognition »*. Journal. School of Microelectronical Engineering, Nathan Campus, Griffith University, Brisbane, Qld 4111, Australia. 2002

[50] Roberto Brunelli, Daniele Falavigna. « *Person identification using multiple cues* ». journal. IEEE Transactions on Pattern Analysis andMachine Intelligence, Vol. 17, pp. 955–966, 1995

[51] Lawrence Rabiner and Biing-Hwang Juang. « Fundamentals of Speech Recognition ». Article. Prentice-Hall, 1993.

[52] ZOUARI BELTAïFA, *« Vers le Temps Réel en Transcription Automatique de la Parole Grand Vocabulaire »*. Thèse de doctorat, Spécial: Signal et Images. *Ecole Doctorale d'Informatique, Télécommunications et Electronique de Paris.* 2007*]*.

[53] Steven B Davis et Paul Mermelstein. « *Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences»*.Journal. IEEE Transactions on Acoustics, Speech and Signal Processing, vol. 28, no 4, pp 357-366, 1980.

[54] Mellor. B. A et Varga. A. P. « *Noise masking in a transform domain»*. Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, vol. 2, pp 87–90, 1993.

[55] Calliope, « *La parole et son traitement automatique* », Collection Technique et scientifique des télécommunication, CENT/ ENST, Ed. Masson, 1989.

[56] 'http://fr.wikipedia.org/wiki/Kurtosis'. (05/05/2011)

Le logiciel Praat 5201 : peut être téléchargé depuis le site « www.praat.org ». (01/20/2010)

[57] **Sabrina Khanniche,** « Mesurer le risque des hedge funds », rapport, Groupama Asset Management 'Economix', Université Paris X Nanterre .2007.

[58] CHANG C.-C., LIN C.-J, « LIBSVM: a library for support vector machines », rapport technique, national taiwan university, 2001. Logiciel disponible en ligne (http://www.csie.ntu.edu.tw/~cjlin/libsvm/). (12/05/2010).

[59] Alin Jain, Karthik Nandakumar and Arun Ross. « *Score normalization in multimodal biometric systems* ».Journal of Pattern Recognition, Vol. 38, No. 12, pp. 2270–2285, December 2005.

[60] ROSS Arun and JAIN Anil. « *Information fusion in biometrics* ». Appeared in pattern recognition Letters, Vol. 24, Issue 13, pp.2115-2125. USA. 2003.

[61] Roberto Brunelli, Daniele Falavigna. « *Person identification using multiple cues* ». Journal. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 17, pp. 955–966, 1995.

[62] Nicolas MORIZET, « *Reconnaissance Biométrique par Fusion Multimodale du Visage et de l'Iris* » ; Thèse de Doctorat, Ecole Nationale Supérieure des Télécommunications Spécialité : Signal et Images .2000

[63] Lorène ALLANO, « *La Biométrie multimodale : stratégies de fusion de scores et mesures de dépendance appliquées aux bases de personnes virtuelles* ».Thèse de Doctorat, Institut National Des Télécommunications .2009

[64] Sergey Tulyakov, Stefan Jaeger, and Venu Govindaraju, and David Doermann , « Review of Classifier Combination Methods », Studies in Computational Intelligence (SCI) 90, 361 386, Springer-Verlag Berlin Heidelberg 2008 .

[65] Arun A. Ross, Karthik Nandakumar, and Anil K. Jain. « *Handbook of Multibiometrics* ». Springer. 2006.

[66] Yong Li, Jianping Yin, En Zhu, Chunfeng Hu, and Hui Chen, «*Studies of Fingerprint Template Selection and Update* ». T.-h. Kim et al. (Eds.): FGCN 2008 Workshops and Symposia, CCIS 28, pp. 150–163, Springer-Verlag Berlin Heidelberg 2009

[67] **Lotfi A. Zadeh**. « Fuzzy sets. *Information Control* », Vol. 8, pp. 338–353, University of California, Berkeley .1965.

[68] Michio Sugeno, «Fuzzy measures and fuzzy integrals—A survey, » in *Fuzzy Automata and Decision Processes*, M. M. Gupta, G. N. Saridis, and B. R. Gaines, Eds. Amsterdam, The Netherlands: North Holland, 1977, pp. 89–102.

[69] Keun-Chang Kwak, **Witold Pedrycz**, «*Face Recognition Using Fuzzy Integral and Wavelet Decomposition M*ethod ». Article. IEEE, AUGUST 2004

[70] **Keun-Chang Kwak**, **Witold Pedrycz**, « *Face recognition: A study in information fusion using fuzzy integral»*, Journal Pattern Recognition Letters. Volume 26 Issue 6, 1 May 2005. Elsevier Science Inc. New York, NY, USA

[71] Sarbast Rasheed , Daniel W. Stashuk , **Mohamed S. Kamel,** « *Diversity-based combination of non-parametric classifiers for EMG signal decomposition»*, Springer-Verlag London Limited 2008 pp. :385–408,2008.

[72] Anupam Shukla et **al, Towards,** Chapter 18: « *Multimodal Biometric Systems*, Hybrid *and Adaptive computing»*, SCI 307, pp. 401–418, Springer-Verlag Berlin Heidelberg. 2010

[73] Anne M.P. Canuto and Marjory C.C. Abreu, « *Using Fuzzy, Neural and Fuzzy-Neural Combination Methods in Ensembles with Different Levels of Diversity»*, J. Marques de Sá et al. (Eds.): ICANN 2007, Part I, LNCS 4668, pp. 349–359, 2007. © Springer-Verlag Berlin Heidelberg .2007

[74] Murofushi T and Sugeno M, « An interpretation of fuzzy measures and the Choquet integral as an integral with respect to a fuzzy measure » *Fuzzy Sets Syst.*, vol. 29, pp. 201–227, 1988.

[75] Godfrey J, « *Multilingual speech databases at LDC* », ARPA HLT'94 Workshop, pp 23-26, Plainsboro, NJ, USA, 1994.

[76] Leonard R.G, « A database for speaker-independent digit recognition ». In proceedings of ICASSP, volume 3, San Diego, 1984.

[77] Chollet G, Cochard J.L, Constantinescu A, et Langlais Ph. Swiss French polyphone and polyvar: telephone speech databases to study intra and inter speker variability. Technical report, IDIAP, 1995.

[*78*] Carré R, Descout R, Eskénazi M, Mariani J, and Rossi M. *The french language database: defining, planning and recording a large database. In proceeding of ICASSP*, San Diego, 1984.

[79] Geutner P., Arevalo L. et Breuninger J., VODIS -voice-operated driver information systems: a usability study on advanced speech technologies for car environments, dans Proceedings of the 6th International Conference on Spoken Language Processing (ICSLP'2000), pages 378–382, Beijing, China, 2000.

[80] Disponible à l'adresse: Short course fundamentals speech recognition summer 2008. "http://cronos.rutgers.edu/~lrr/short%20course_fundamentals%20of%20speech%" (22/04/2011)

[81] John Pierce «*Whither Speech Recognition*». Journal of the Acoustical Society of America. 1969

[82] Klatt, « *ARPA Speech Understanding Project* ».1977.

[83] Duerr B, Haettich W, Tropf H & Winkler G, « *A combination of statistical and syntactical Pattern recognition applied to classification of unconstrained handwritten numerals* », Pattern Recognition, vol 12, pp. 189 199, 1980.

[84] Pham T & Yan H, « *Fusion of handwritten numeral classifiers based on fuzzy and genetic algorithms* », In : Proceedings of the 1997 Annual Meeting of the North American Fuzzy Information Processing Society, NA¬FIPS'97, New York, pp. 257-262, 1997.

[85] Yousfi A & Meziane A, « *Introduction de l'énergie dans un modèle de reconnaissance automatique de la parole* ». XXIVémes Journées d'Etude sur la Parole, Nancy (France), pp. 317-320, 24-27 juin, 2002.

Annexe

A : Analyse de données et sélection de caractéristiques

A.1. Analyse par Ondelettes
A.2. Quantification vectorielle

1. Analyse par Ondelettes

Nous avons vu que quelque soit le type de la transformée utilisée, il existe un problème de résolution en temps et en fréquence résultant d'un phénomène physique. Il est cependant possible d'analyser tout signal en utilisant une autre approche appelée analyse Multirésolution ou analyse par Ondelettes. Cette analyse comme son nom l'indique, analyse le signal à différentes fréquences avec des résolutions différentes.

L'analyse multirésolution fournit une bonne résolution temporelle et une moins bonne résolution en fréquence pour les hautes fréquences, et une bonne résolution en fréquence et une médiocre résolution en temps pour les basses fréquences. Cette approche permet une meilleure modélisation des signaux présentant des composantes hautes fréquences sur des courtes durées et des composantes de basses fréquences sur de longues durées. Le spectrogramme issu de l'analyse semble plus proche de la représentation du signal vocal par la cochlée.

On réalise une analyse en Ondelettes de la même manière que l'analyse par la transformée de Fourier : Le signal est multiplié par une fonction ondelette, semblable à la fonction de fenêtrage de la transformée de Fourier, puis la transformée est calculée séparément pour différents segments du domaine temporel. Par contre, les fréquences négatives ne sont pas calculées et la largeur de la fenêtre est modifiée à mesure qu'on calcule la transformée pour chacune des composantes spectrales individuellement. La transformée en ondelettes est définie par :

$$CWT^{\psi}_{x} = \Psi^{\psi}_{x}(\tau, s) = \frac{1}{\sqrt{|s|}} \int_{n=-\infty}^{\infty} x(t)\psi\left(\frac{t-\tau}{s}\right)dt$$

Par définition, Une ondelette **Ψ(t)** est donc une fonction à moyenne nulle qui peut être translatée en temps d'un paramètre de translation **τ** et dilatée par un paramètre d'échelle **s**. **ψ(t)** est la fonction de transformation, appelée l'ondelette mère. Un exemple de l'ondelette de *Morlet* :

$$\Psi(t) = e^{-\tau t^2} e^{i\omega t}$$

En revanche, l'intérêt des ondelettes pour la reconnaissance de la parole reste à démontrer. En effet, le choix de l'ondelette mère est primordial selon ce que l'on désire étudier.

2. Quantification vectorielle

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.

Par exemple, la quantification d'un vecteur **x** revient à le représenter par un vecteur proche $\mathbf{y_i}$ d'un dictionnaire fini **Y**. Le dictionnaire **Y** est obtenu par partition de l'espace d'origine en **M** classes. Cependant, la taille du dictionnaire joue un rôle très important dans l'erreur de quantification.

La phase de construction du dictionnaire peut être réalisée par l'algorithme des **Kmeans**. On peut résumer la construction d'un dictionnaire de la manière suivante :

- A partir d'un dictionnaire initial, calculer l'erreur moyenne qu'il introduit. Si elle est inférieure à un certain seuil, l'algorithme est terminé.
- Sinon, remplacer chaque centroïde par la moyenne de tous les vecteurs de la classe représentée par ce centroïde et recommencer avec le nouveau dictionnaire.

Cet algorithme ne permet d'avoir qu'un optimum local, le choix de l'initialisation est donc important.