SOMMAIRE

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES ABREVIATIONS

API	**:**	**Appliction Programming Interface**
TFD	**:**	**Transformée de Fourier Discrète**
GDI	**:**	**Graphical Device Interface**
EM	**:**	**Expectation – Maximisation**
FFT	**:**	**Fast Fourier Transform**
GMM	**:**	**Gaussian Mixture Model**
LSB	**:**	**Least Significant bit**
MFCC	**:**	**Mel – Frequency Cepstral Coefficient**

INTRODUCTION

Depuis toujours, la parole a été pour l'humanité le moyen de communication le plus utilisé. Grâce à son utilisation fréquente, elle peut être un moyen pour reconnaitre un individu. Pour réaliser, la reconnaissance se fait par l'enregistrement de la parole via un microphone, puis, traitement du signal de cette parole par les microordinateurs qui vont donner le résultat adéquate.

Ce présent rapport est donc axé sur la reconnaissance vocale. Il est intitulé « **identification par biométrie vocale** ».

Pour ce faire, ce présent travail a décomposé en quatre chapitres dont le premier est la généralité de la biométrie vocale englobant la technique de la reconnaissance.

Dans le deuxième chapitre est décrite l'opération de numérisation du signal qui se focalise sur l'échantillonnage, la quantification et le codage du signal quantifié.

Ensuite, dans le troisième chapitre qui est l'identification du locuteur proprement dite. Ici, on y trouve le procédure d'identification en utilisant les opérations de traitement du signal et des outils mathématiques.

Et le dernier chapitre sera la réalisation du logiciel **« RecSpeaker »** suivi d'une démonstration à propos de l'identification par biométrie vocale.

Chapitre 1 : GENERALITE SUR LA BIOMETRIE VOCALE

1.1 Voix

Définition

La voix est l'ensemble des sons caractérisés par deux fonctions mécaniques de base : la phonation qui consiste à la production d'un phénomène acoustique et l'articulation qui consiste à la modulation de ce dernier.

1.2 Parole

1.2.1 Caractéristiques

La parole est un moyen de communication naturel de l'humain avec une efficacité très importante.

Elle se distingue des autres sons par ses caractéristiques acoustiques qui ont leur origine dans les mécanismes de production. Elle apparait physiquement comme une vibration de pression de l'air causée par le système articulatoire. Les sons de la parole sont produits soit par des vibrations des cordes vocales (c'est la source de voisement), soit par une turbulence créée par l'air s'écoulant rapidement dans une constriction ou lors du relâchement d'une occlusion du conduit vocal (c'est la source de bruit).

1.2.2 Production de la parole

Essentiellement, il y a trois étapes pour le processus de la phonation :

- Premièrement, il faut avoir une énergie respiratoire suffisante pour mettre en mouvement les cordes vocales et générer des bruits.
- Les cordes vocales vibrant, provoquent la naissance des sons voisés.
- Enfin, une gestuelle articulatoire au niveau du conduit vocal et fosses nasales se réalise.

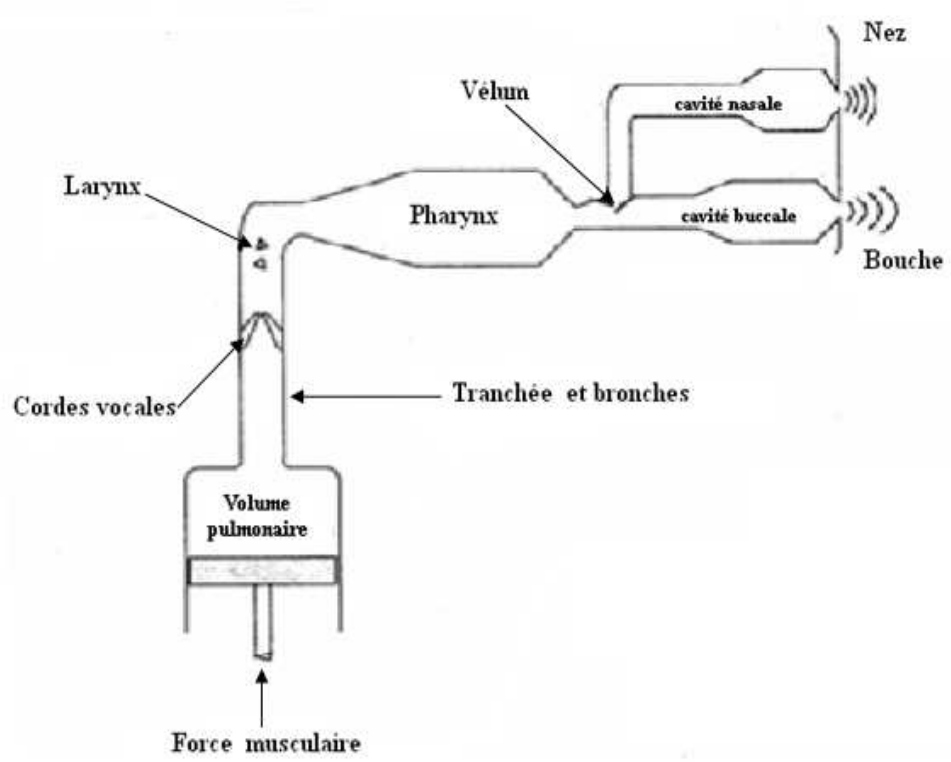


Figure 1.1 : **vue d'ensemble de la production de la parole [1]**

1.2.3 Signal de la parole

Le signal vocal est caractérisé par:

- Sa fréquence
- Son intensité (ou le niveau sonore)
- Son timbre (ou « la richesse » du signal)

a. Fréquence

La fréquence qui est l'inverse de la période **T**, est le nombre d'oscillation dans une seconde. On peut aussi avoir la fréquence **F** à partir de la formule suivante :

$$F = \frac{C}{\lambda}$$

, avec **C** la célérité ou la vitesse du son en **m/s** et **λ** la longueur d'onde en **m**.

- **Fréquence fondamentale**

Le signal de la parole comprend un son fondamental et des harmoniques dont les rapports de fréquences avec la fondamentale sont des quotients de nombres entiers. Toute vibration sonore peut être décomposée en une somme de fonctions sinusoïdales élémentaires dont les périodes plus courtes sont proportionnelles avec sa propre période (c'est la décomposition en « série de Fourier »).

- **Fréquence harmonique**

C'est la fréquence multiple de la fréquence fondamentale $\mathbf{F_0}$, c'est-à-dire $f_n = nF_0$ avec $n \in \{2,3,\ldots\}$. Une harmonique correspond à une fonction trigonométrique sinusoïdale dont la fréquence est un multiple de la fréquence de la fonction périodique décomposée. La somme de toutes les harmoniques d'une fonction périodique reconstitue la fonction.

Prenons par exemple un signal acoustique s(t) qui est la superposition de trois sinusoïdales pures dont la fréquence fondamentale est f = 440Hz et de fréquences harmoniques f_2 = 880Hz, f_3 = 1320Hz, d'équation :

$$s(t) = \sin 2\pi f_1 t + \sin 2\pi f_2 t + \sin 2\pi f_3 t$$

Les graphes de cette équation en fonction du temps et de la fréquence sont illustrés par la fig. 1.2.

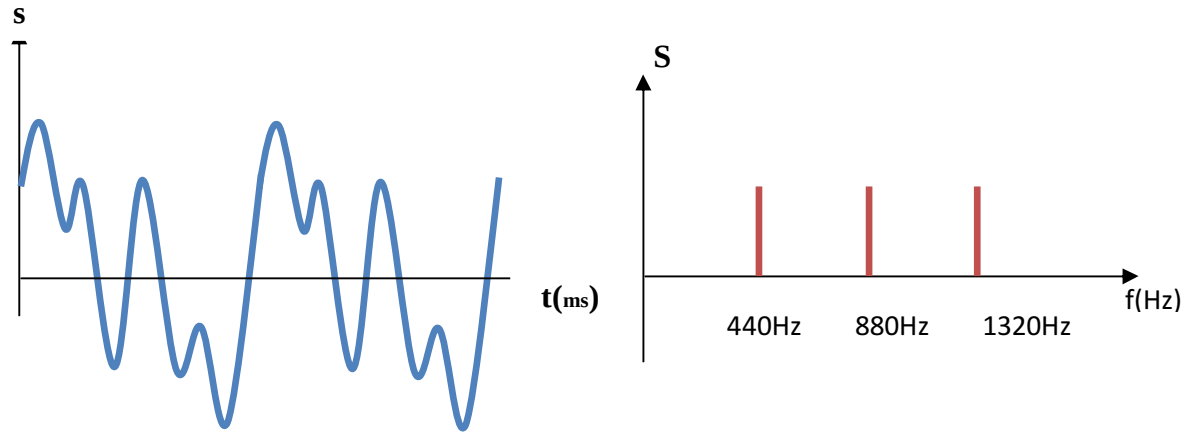


Figure 1.2 : **Présentation du signal acoustique, a : Présentation en fonction du temps, b : Présentation fréquentielle**

b. Intensité

Le son est une onde qui se propage de façon omnidirectionnelle. L'énergie acoustique **E** (en joule) produite par la source sonore est repartie sur une surface sphérique de plus en plus grande au fur et à mesure de la propagation de l'onde sonore. L'intensité notée **I**, est une fonction dépendante de la puissance **P** de l'émetteur et la distance notée **r**, qui sépare la source et le lieu d'écoute **[2]**.

L'intensité est donnée par la formule :

$$\mathbf{I = \frac{P}{4\pi r^2}}$$

, avec **P** est en **W** et r en **m** => I en **W/m²**.

Elle caractérise aussi le volume du signal mais dépend de l'amplitude du signal considéré.

En acoustique, elle s'exprime en décibel en raison de deux facteurs :

- Les valeurs obtenues sont faciles à manipuler (elles ne sont trop grandes ni trop petites)
- La perception humaine sur l'intensité sonore se fait de façon logarithmique.

L'intensité acoustique (en **dB**) est alors définie comme suit :

$$\mathbf{L = 10 \log I/I_0}$$

Où $\mathbf{I_0}$ le son le plus faible que l'on puisse entendre pour un signal de fréquence 1 kHz. Elle est appelée aussi intensité de référence.

$$\mathbf{I_0 = 10^{-12}\ W/m^2}$$

Pour que l'oreille perçoive un son dit audible, l'intensité sonore **I** doit être :

$$\mathbf{10^{-12}\ W/m^2 < I < 25\ W/m^2}$$

Où 10^{-12} **W/m²** est la limite de sensibilité de l'oreille et 25 **W/m²** borne supérieure de l'intensité sonore, correspond à une destruction de l'oreille.

c. Timbre

Il correspond à la richesse d'un signal sonore d'un instrument ou d'une voix d'une personne. Il est caractérisé par ses fréquences harmoniques, ses nombres et ses amplitudes. C'est pour cette raison qu'une même note ne produira pas le même son avec deux instruments différents. On dit qu'un son est riche lorsqu'il possède beaucoup d'harmoniques et pauvre lorsqu'il a moins d'harmoniques.

1.3 Phonème

Le phonème est la plus petite unité phonique distinctive, qui n'est défini sur une base acoustique, ni articulatoire, ni perceptuel, mais sur le plan fonctionnel. Ainsi, les phonèmes n'ont pas d'existence indépendante, c'est-à-dire, ils constituent un ensemble structuré dans lequel chaque élément est intentionnellement différent de tous les autres et la différence étant à chaque fois porteuse de sens **[3].**

Voici un exemple des phonèmes de la langue française. Dans la langue française, il existe 36 phonèmes qui sont représentés par Tab1.1.

Tableau 1.1 : **Les phonèmes de la langue française**

CONSONNES	VOYELLES
Paie, baie, mais, fait, vais, ouais, taie, dais, nez, sait, huer, lait,….	Lit, les, là, lin, lu, leu, leur, le, lent, loup, lot, lotte, long, ….

1.4 Prosodie

La prosodie est la façon de décrire ou de présenter formellement les éléments de l'expression orale à savoir les tons, les accents, l'intonation et la qualité. Ces éléments transmettent des informations sur la signification d'un énoncé. En d'autres termes, elle désigne les phonèmes liés à l'évolution dans le temps des paramètres de hauteur, d'intensité et de durée.

La perception de hauteur est essentiellement liée à la fréquence fondamentale qui correspond au niveau physiologique de la production et à la fréquence de vibration des cordes vocales.

La perception d'intensité est essentiellement liée à l'amplitude et à l'énergie du son, mais partiellement dépend aussi avec sa durée.

La perception de durée correspond à son temps d'émission et sa durée acoustique. A noter que le terme « durée » est utilisé pour désigner à la fois le paramètre perceptif et le paramètre acoustique et le terme « longueur » comme synonyme de durée perçue est utile quand la distinction est importante **[4]**.

- **Mélodie**

Elle est constituée par la variation dans le temps de la fréquence fondamentale, ou de la hauteur si l'on se place du point de vue perceptif. L'enchainement des durées relatives (y compris les durées des silences) constitue le rythme.

Avec le modèle d'intonation, il existe quatre niveaux d'intonation : basse, moyenne, haute et aiguë. Cette modélisation met en jeu les trois modalités suivantes : l'interrogation, l'exclamation et l'affirmation.

Les substitutions entre les intonations dans une phrase de même contenu, entrainent des changements de sens. Cela montre notamment que l'intonation joue un rôle très important pour la compréhension du message vocale.

- **Ton**

Le mot **ton** désigne le ou les niveaux de hauteurs observées dans une syllabe donnée. Le ton coïncide donc avec la partie de la courbe mélodique qui se rattache à une seule syllabe.

L'intonation d'un énoncé se présente comme une succession de ton. Et on distingue quatre niveaux de hauteur : haut, bas, infra-bas et suraigu.

- **Accent**

L'accent se situe par la manifestation d'intensité, de hauteur et/ou de durée, portant sur une syllabe.

L'équation suivante résume ce qui précède :

Prosodie = F_0 + énergie + durée (grandeur acoustique)

= hauteur + intensité +longueur (grandeur perçues)

=mélodie + rythme (structures) + accentuation

1.5 Notion de la biométrie

1.5.1 Identité

L'identité est une notion complexe, difficile à définir.

Du point de vue personnel, la caractérisation de l'identité prend en compte tout ce que l'individu considère comme faisant partie intégrante de lui et qui ne peut lui être enlevé.

Du point de vue externe, l'identité d'un individu est la façon dont il perçu par le monde qui l'entoure.

Pour identifier une personne, trois approches sont possibles:

- Utiliser un identifiant : ce que l'on possède (carte, badge, document).
- Utiliser une connaissance : ce que l'on sait (mot de passe).
- Utiliser une biométrie : ce que l'on est.

1.5.2 Biométrie

a. Définitions

C'est la science qui étudie, à l'aide des mathématiques (statistiques, probabilités), les variations biologiques à l'intérieur d'un groupe déterminé.

Autrement dit, c'est une méthode permet d'identifier ou de vérifier l'identité (authentification) d'une personne sur la base de données reconnaissable et vérifiable qui lui est propre.

b. Authentification biométrique

La biométrie permet l'authentification d'individus à partir de leurs caractéristiques physiologiques ou comportementales qui doivent être :

– universelles : présentes chez tous les individus.

– uniques : spécifiques à chaque individu.

– permanentes : pour permettre une authentification au cours du temps.

– mesurables : pour permettre l'enregistrement et les comparaisons futures.

Avantages

L'authentification biométrique présente de nombreux avantages :

- Elle permet de s'affranchir des intermédiaires que constituent les clefs, cartes et autres codes personnels susceptibles d'être oubliés, perdus ou volés.

- Elle supprime le risque qui peut être occasionné par le prêt d'une clef ou la communication d'un mot de passe à un tiers.
- L'utilisation de données intrinsèques à l'utilisateur lui permet, de plus, de recourir à la biométrie en tout lieu et à tout moment.

c. Différents types de biométrie

- **Biométrie morphologique**

Elle décrit les individus par des mesures de leurs caractéristiques biologiques ou physiologiques qui sont moins sujettes à l'influence du stress que la biométrie comportementale.

Exemples : empreintes digitales, le réseau veineux de la rétine, l'iris, l'empreinte, etc.

- **Biométrie comportementale**

La biométrie comportementale mesure et caractérise des éléments qui sont propres aux comportements d'un individu.

Exemple : signature dynamique.

- **Biométrie mixte**

Parfois on ne peut pas distinguer exactement la biométrie morphologique avec biométrie comportementale. D'où le nom biométrie mixte.

Exemple : la voix, qui est utilisée de façon naturelle par les êtres humains pour reconnaître un individu, est une modalité comportementale qui peut subir les influences d'une pathologie, du stress ou même d'un changement émotionnel.

1.6 Reconnaissance vocale ou la biométrie vocale

La reconnaissance de la voix (caractéristique propre pour chaque individu) ou la reconnaissance vocale est un terme générique regroupant les problèmes relatifs à la reconnaissance du locuteur, basé sur le contenu de l'information dans le signal acoustique de la parole qui est la faculté de communiquer la pensée par le moyen de sons articulés émis par les organes de la phonation, tandis que la voix, elle représente l'ensemble de sons produits par le système articulatoire et phonatoire.

Dans la reconnaissance vocale, on cherche à trouver ce qui caractérise le locuteur dans le signal acoustique. Ici, l'individualité est présente, parce que le locuteur peut être reconnu aussi bien que par son timbre vocale, que par la hauteur de sa voix, la particularité d'élocution, l'intonation,….

Les systèmes de reconnaissance vocale se concentrent sur les caractéristiques de voix qui sont uniques à la configuration de la parole d'une personne. Les configurations de la parole sont constituées par une combinaison des facteurs comportementaux et physiologiques. Les mouvements des organes de production de la parole engendrent des variations de pression acoustique instantanée qui peuvent être captées par un transducteur (microphone) et transformées en variations de tension électrique.

Un enregistrement de la parole n'est ni un prélèvement direct ni une trace laissée sur une surface au contact d'une partie de son corps, il ne s'agit que de la capture indirecte de mouvements articulatoires complexes faisant intervenir les cordes vocales, la langue, le voile du palais, la mâchoire et les lèvres. La reconnaissance vocale est considérée comme une des formes les moins intrusives de la technologie biométrique, car elle n'exige aucun contact physique avec le capteur (microphone) du système automatique de reconnaissance **[5].**

Chapitre II: NUMERISATION D'UN SIGNAL

2.1 Introduction

L'importance des systèmes numériques de traitement de l'information ne cesse de croître (téléphone, télévision, radio, instrumentation, ….). Ce choix est souvent justifié par des avantages techniques tels que la grande stabilité des paramètres, une excellente reproductibilité des résultats et des fonctionnalités accrues. Le monde extérieur étant par nature « analogique », une opération préliminaire de conversion analogique numérique est nécessaire .Cette conversion est la succession de trois effets sur le signal analogique qui n'est autre que le signal de départ :

- L'**échantillonnage** : qui rend le signal analogique en signal discret
- **La quantification** pour associer à chaque échantillon une valeur
- **Le codage** pour associer un code à chaque valeur.

2.2 Echantillonnage

Il consiste à prélever à des instants précis, le plus souvent équidistants, les valeurs instantanées d'un signal.

Soit s(t) un signal analogique, continue dans le temps, est représenté par un ensemble de valeurs discrètes $s_e(t)$ tel que :

$$\mathbf{s_e(t) = s(\ n.\ Te)}$$

, avec **n** : un entier et $\mathbf{T_e}$: période d'échantillonnage.

Théoriquement, l'opération d'échantillonneur est souvent symbolisée par un interrupteur. La Figure 2.1 montre cette opération.

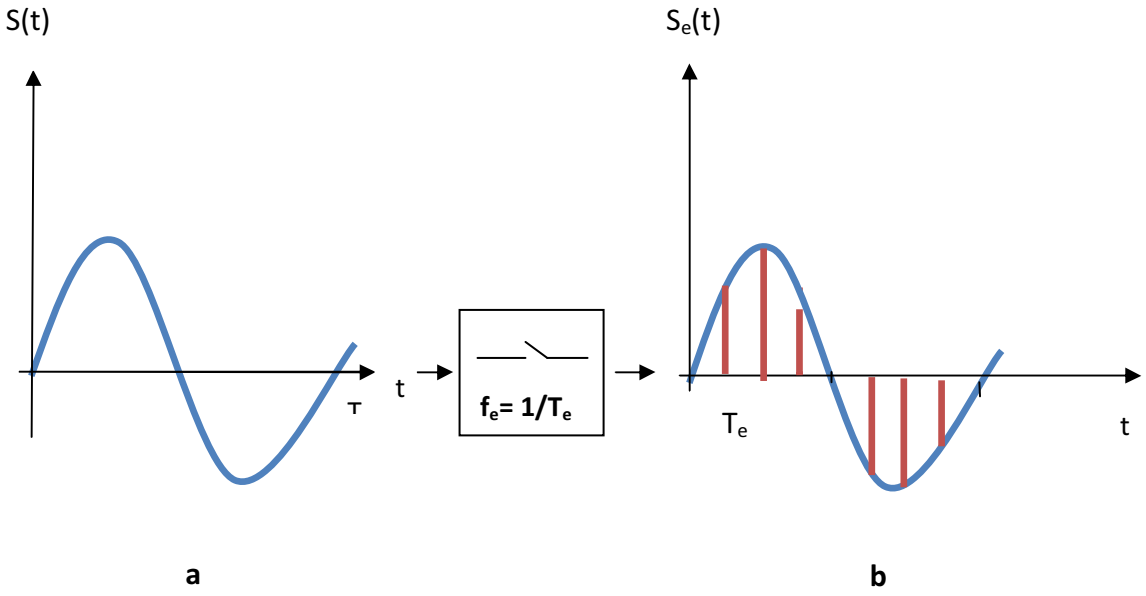


Figure 2.1: **Transformation du signal s(t) continu en signal échantillonné $s_e(t)$, a : signal analogique s(t) , b : signal échantillonné $s_e(t)$**

2.2.1 Echantillonnage idéal

L'échantillonnage idéal est modélisé par la multiplication du signal continu s(t) et d'un peigne de Dirac de période T_e ; c'est-à-dire :

$$S_e(t) = s(t).\ \delta_{Te}(t) = s(t) \sum_{n\to-\infty}^{+\infty} \delta(t - nTe) = s(nT_e) \sum_{n\to-\infty}^{+\infty} \delta(t - nTe)$$

Donc, le spectre du signal échantillonné est le suivant :

$$S_e(f) = \frac{1}{Te} \sum_{n\to-\infty}^{+\infty} S(f) * \delta(f - nfe) \longrightarrow \boxed{\mathbf{S_e(f) = \frac{1}{Te} \sum_{n\to-\infty}^{+\infty} S(f - nfe)}}$$

On obtient donc un spectre infini qui provient de la périodisation du spectre du signal d'origine autour des multiples de la fréquence d'échantillonnage **[6].**

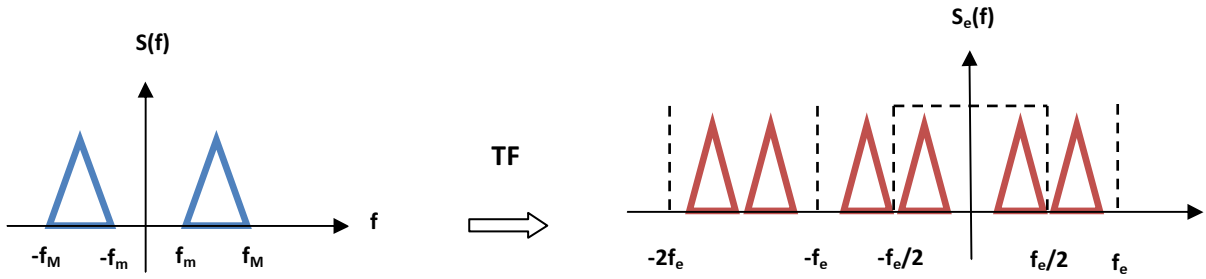


Figure2.2 : **Echantillonnage idéal**

Remarques :

- Sur le signal échantillonné, on voit qu'il est possible de restituer le signal original par un filtre passe-bas.
- Aussi, si $\mathbf{f_M} > \frac{fe}{2}$, la restitution de l'originale sera impossible car il va apparaître un recouvrement spectral lors de l'échantillonnage ;

  avec $\mathbf{f_M}$ est la fréquence maximale du spectre du signal à échantillonner.

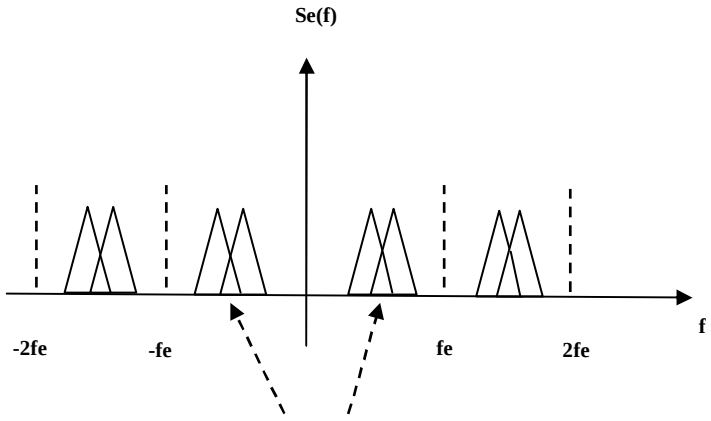


Figure 2.3 : **Recouvrement spectral**

Le théorème de **Shannon** montre que la reconstitution correcte d'un signal nécessite que la fréquence d'échantillonnage $\mathbf{f_e}$ soit au moins deux fois plus grande des fréquences $\mathbf{f_M}$ du spectre du signal :

fe >2f_M

2.2.2 Echantillonnage réel

L'échantillonnage réel est obtenu en commandant un interrupteur par un train d'impulsions étroites. Ce qui veut dire qu'il est impossible d'obtenir des échantillons de durée quasiment nulle. La modélisation de l'échantillonnage par un peigne de Dirac est donc erronée. En fait, chaque impulsion va avoir une durée très courte **λ**. L'échantillonnage peut donc être modélisé par la multiplication du signal par une suite de fonction rectangle ou porte de largeur λ **[6]**.

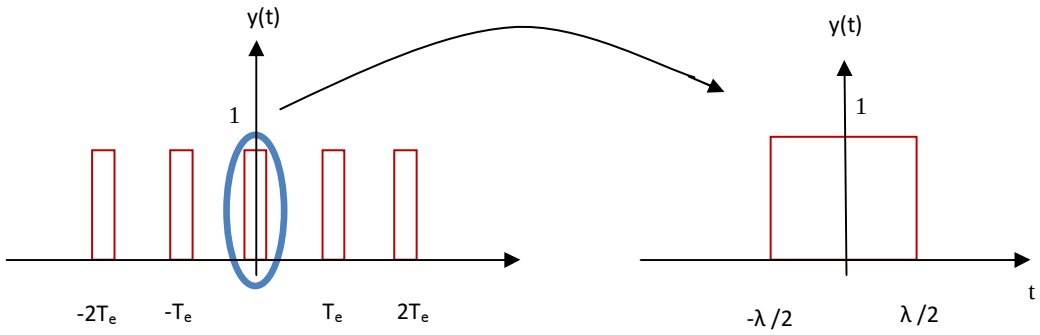


Figure 2.4 : **Signal porte de durée λ**

On a comme expression du signal d'échantillonnage :

$$y(t) = \sum_{k\to-\infty}^{+\infty} \text{rect}\ \frac{t-kTe}{\lambda} = \text{rect}\ (\frac{t}{\lambda}) * \sum_{k\to-\infty}^{+\infty}\ \delta(t-kTe)$$

Par conséquent, sa transformée de Fourier est égale à :

$$Y(f) = \lambda\ \text{sinc}(\lambda f)\ \frac{1}{Te} \sum_{k\to-\infty}^{+\infty}\ \delta(t-kfe)$$

Et comme l'expression du signal d'échantillonné est :

$$s_e(t) = s(t).y(t)$$

Sa transformée de Fourier devient :

$$S_e(f) = S(f) * Y(f) = S(f) * \frac{\lambda}{Te} \sum_{k\to-\infty}^{+\infty}\ \text{sinc}(\lambda f)\ .\delta(f-kfe)$$

$$S_e(f) = \frac{\lambda}{Te} \text{Sinc}(\lambda f) \sum_{k\to-\infty}^{+\infty} S(f - kfe)$$

On retrouve la même allure de spectre modulé en amplitude par une fonction en sinus cardinale.

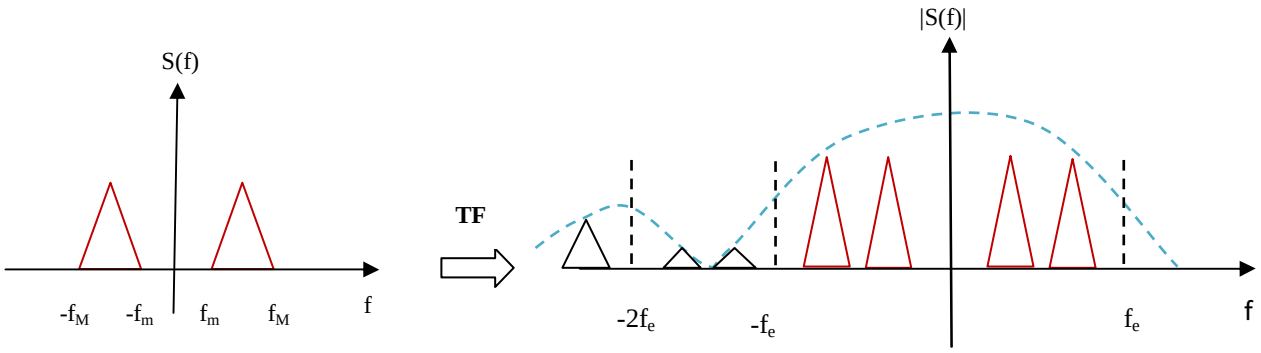


Figure 2.5 : **Spectre modulé en amplitude entrainant une fonction en sinus cardinale**

2.2.3 Echantillonnage-blocage

Pratiquement, on n'échantillonne pas un signal pour le reconstruire juste après. En effet, l'échantillonnage sert à prélever le signal à des instants multiples de T_e et ensuite convertir les échantillons sous forme binaire (8, 12, 16 bits, …) par l'intermédiaire d'un convertisseur analogique-numérique (**CAN**). Mais cette conversion n'est pas instantanée. Donc il est nécessaire de procéder au blocage du signal pour avoir une conversion sans erreur si le signal à convertir varie trop rapidement. C'est pourquoi on utilise un échantillonneur-bloqueur puisqu'il mémorise le signal à convertir et le maintient constant pendant toute la durée de conversion.

L'effet de blocage peut être modélisé par une fonction porte décalée de λ/2 :

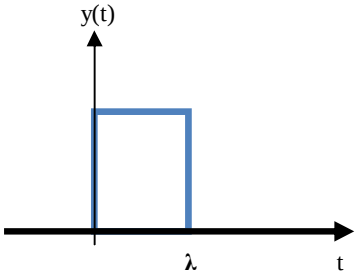


Figure 2.6 : **Fonction porte**

L'expression de cette fonction porte est définie comme suit :

$$y(t) = \sum_{k\to-\infty}^{+\infty} \text{rect}\left[\frac{t-\frac{\lambda}{2}-kTe}{\lambda}\right] = \text{rect}\left[\frac{t-\frac{\lambda}{2}}{\lambda}\right] * \sum_{k\to-\infty}^{+\infty} \delta(t-kTe)$$

L'échantillonnage-blocage consiste donc à la multiplication du signal par y(t). D'où la transformée de Fourier du signal échantillonné est défini de la manière suivante :

$$\mathbf{Se(f) = \frac{\lambda}{Te}\ sinc\ (\lambda f)\ \sum_{k\to-\infty}^{+\infty} S(f-kfe)\ e^{-j\pi f\lambda}}$$

2.3 Quantification

2.3.1 Définitions

La quantification consiste à associer à une valeur réelle **x** quelconque, une autre valeur $\mathbf{x_q}$ appartenant à un ensemble fini de valeurs et ce suivant une certaine loi : arrondi supérieur et arrondi le plus proche **[6]**.

L'écart entre chaque valeur $\mathbf{x_q}$ est appelé pas de quantification et le fait d'arrondir la valeur de départ entraine forcément une erreur de quantification que l'on appelle le bruit de quantification.

2.3.2 Quantification uniforme

La loi de quantification uniforme ou linéaire utilise un pas de quantification Δ constant entre chaque valeur $\mathbf{x_q}$.

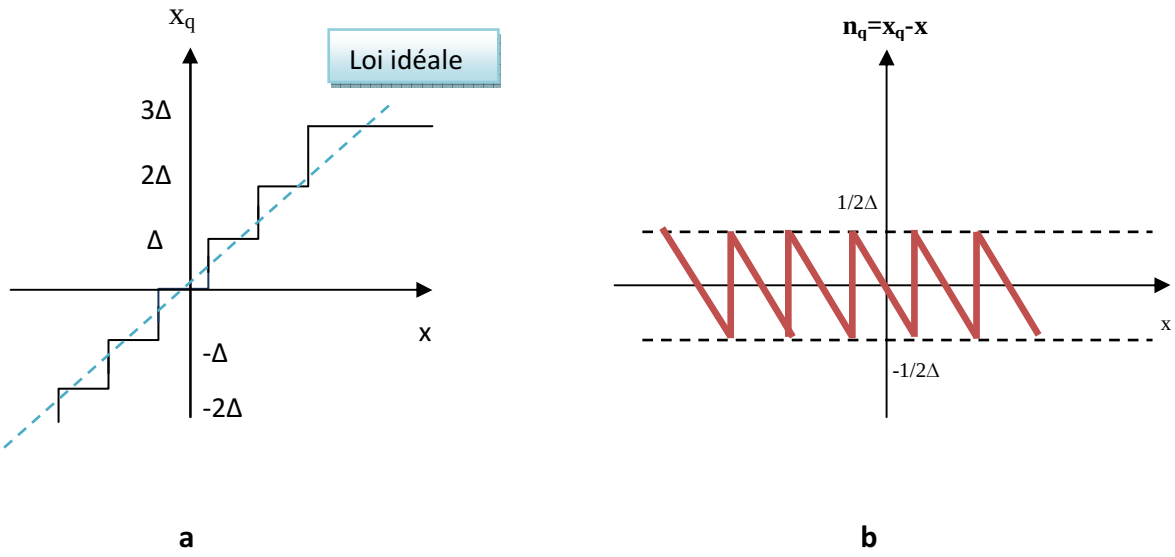


Figure 2.7 : **Quantification uniforme, a : signal quantifié, b :bruit de quantification**

L'évolution de bruit de quantification est une évolution en dent de scie avec une amplitude égale au quantum. Les caractéristiques du bruit de quantification varient en fonction de principe de quantification utilisée. Le bruit de quantification $\mathbf{n_q}$ est un signal aléatoire.

2.3.3 Quantification non linéaire

L'intérêt des techniques de quantification non linéaires réside dans le fait qu'elles permettent de coder de manière plus précise les valeurs qui apparaissent plus souvent. Dans la plupart des applications, on utilise la technique de quantification linéaire centrée mais dans certains cas particuliers, il peut s'avérer intéressant d'opter pour une technique plus adaptée.

2.4 Codage

Le codage consiste à associer à un ensemble de valeurs discrètes un code composé d'éléments binaires. Le codage est la dernière étape et un outil très important à la numérisation d'un signal analogique

Les codes les plus connus et les plus utilisés sont : code binaire naturel, code binaire décalé, code complément à 2, code DCB, code Gray.

2.4.1 Code binaire naturel ou DCBN

Il est défini de la manière suivante :

$$N = \sum a_i \ 2^i$$

Avec $a_i = \{0,1\}$ et $i \in IN$.

Ce code se prête parfaitement au traitement des opérations arithmétiques. Ses inconvénients sont les suivants :

- Il faut un grand nombre de bits pour exprimer un nombre dès que celui-ci est élevé.
- Ce code peut introduire des erreurs lors du codage des grandeurs variant de façon ordonnée. En effet entre deux mots successifs de ce code, plusieurs bits pourront être amenés à changer simultanément **[7]**.

2.4.2 Code Gray

Le passage d'un mot code au suivant se fait par un changement d'état par un seul bit à partir de **LSB** (Least Significant Bit) si possible **[7]**.

Il est obtenu à partir d'un code binaire :

$$G_i = B_{i+1} \oplus B_i$$

Caractéristiques

- C'est un code cyclique, en effet, lorsqu'on passe du dernier au permier le principe est toujours vérifié.
- Ce code a la même densité que le code binaire naturel.
- Il existe des symétries dans le contribution des mots du code, d'où le nom du code binaire réfléchis ou code reflexe.
- Le code Gray n'est pas un code pondéré.
- Il y a une correspondance entre le code Gray et la table de Karnaugh.

2.4.3 Code DCB

Chaque élément d'un nombre décimal (chiffre décimal) est représenté par son équivalent binaire naturel à 4 bits.

Exemple :

1 9 8 7 => 0001 1001 1000 0111

Caractéristiques

- C'est un code pondéré.
- Il conserve les avantages du système décimal et du code binaire pur.
- Les mots codes sont plus longs qu'en code BCDN **[7]**.

Chapitre III : RECONNAISSANCE DU LOCUTEUR / IDENTIFICATION DU LOCUTEUR

3.1 La reconnaissance du locuteur

3.1.1 généralité

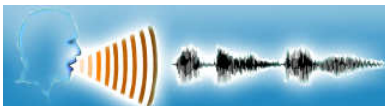
La reconnaissance d'un locuteur ou la reconnaissance sur la base de la voix est une motivation ancienne. Elle a pour objectif de déterminer l'identité d'un locuteur ou d'identifier une personne à partir de sa voix, plus précisément à partir d'un signal de la parole. Pour cela, on doit tester si la voix enregistrée provient vraiment d'un locuteur particulier ou non.

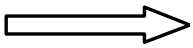
3.1.2 Conditions nécessaires :

Pour bien assurer une bonne qualité, acceptable et robuste du système de reconnaissance du locuteur, les caractéristiques suivantes sont nécessaires :

- Les locuteurs ne doivent pas déguiser leur voix, c'est-à-dire lors de l'enregistrement de la parole, ils doivent être en bonne santé et être concentrés.
- Pas de stress durant l'enregistrement.
- L'environnement doit être bien contrôlé, c'est-à-dire :
  - le micro d'enregistrement doit être le même pour tous les locuteurs.
  - L'endroit où l'enregistrement se déroule est obligatoirement un même local et bien calme.
- Des données de parole, enregistrées dans les mêmes conditions que le signal test, sont disponibles pour référencier un locuteur dans le système.
- Le contenu linguistique des messages inclut des mots connus du système, permettant à celui-ci de calculer une ressemblance entre voix en se basant sur des contenus comparables.
- L'usage d'un système de synthèse de la parole n'est pas autorisé.

3.1.3 Principe de base





Enregistrement **Extraction des paramètres**

Figure 3.1 : **Structure d'un système de reconnaissance du locuteur**

3.1.4 Les modes de reconnaissance du locuteur

En reconnaissance du locuteur, il existe 2 modes tels que : le mode dépendant du texte et le mode indépendant du texte.

a. Le mode dépendant du texte

En mode dépendant du texte, le texte prononcé par le locuteur durant l'enregistrement est le même que celui qu'il a prononcé lors de l'apprentissage ou de la vérification de sa voix. Il existe 3 niveaux de dépendance au texte et ils sont classés suivant les applications :

- système à texte libre
- système à texte suggérée
- système dépendant du vocabulaire ou système personnalisé dépendant du texte.

b. Le mode indépendant du texte

En mode indépendant du texte, le locuteur peut prononcer n'importe quelle phrase ou groupe de mots pour être reconnu. L'avantage dans ce mode c'est qu'on ne trouve aucune contrainte sur le message que le locuteur doit prononcer ni sur la langue qu'il peut utiliser.

3.2 L'identification du locuteur

Définition :

L'identification du locuteur consiste à confirmer ou infirmer par sa voix l'identité proclamée d'un individu. Elle doit établir à partir d'un échantillon de voix l'identité d'une personne parmi N personnes connues à l'avance par le système. Le système peut décider alors que l'échantillon de voix appartient à une des N personnes connues ou qu'il appartient à une personne inconnue.

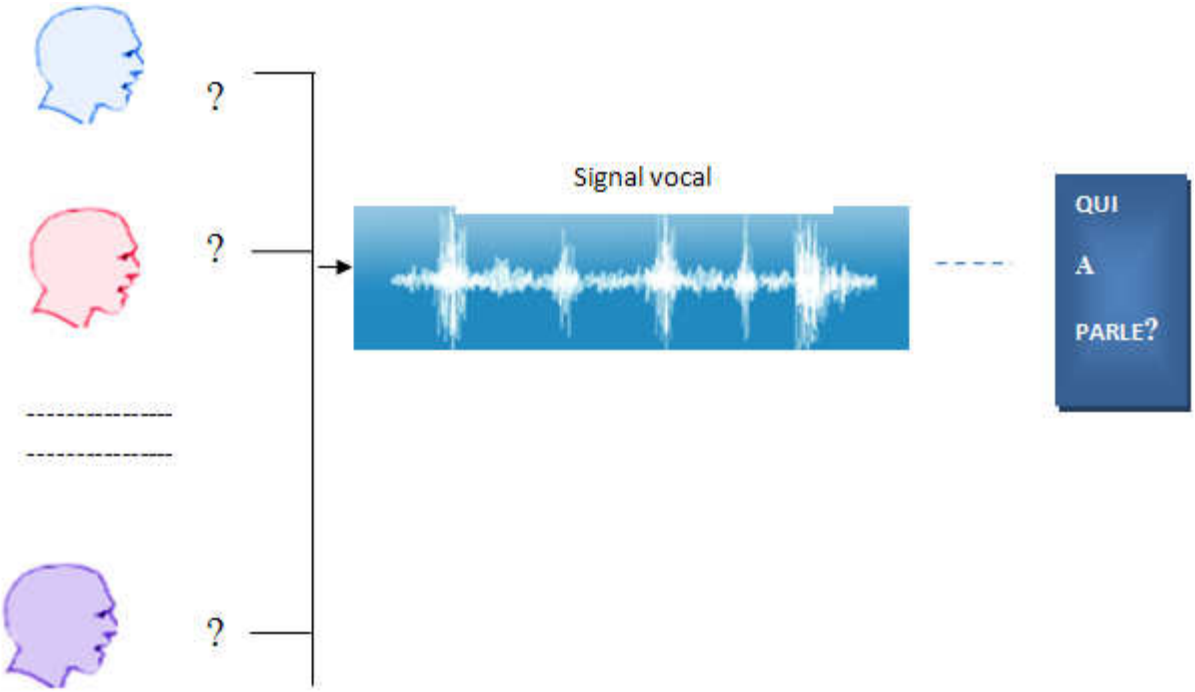


Figure 3.2 : **Principe d'identification du locuteur**

3.3 Procédure d'identification:

L'identification du locuteur est divisée en trois grandes principales phases :

- la phase de paramétrisation ou l'analyse acoustique,
- la phase de modélisation et la phase de décision.

Ici dans notre cas, les locuteurs ont prononcé le même message ou le même groupe de mots, c'est-à-dire que notre système de reconnaissance de locuteur est en mode dépendant du texte.

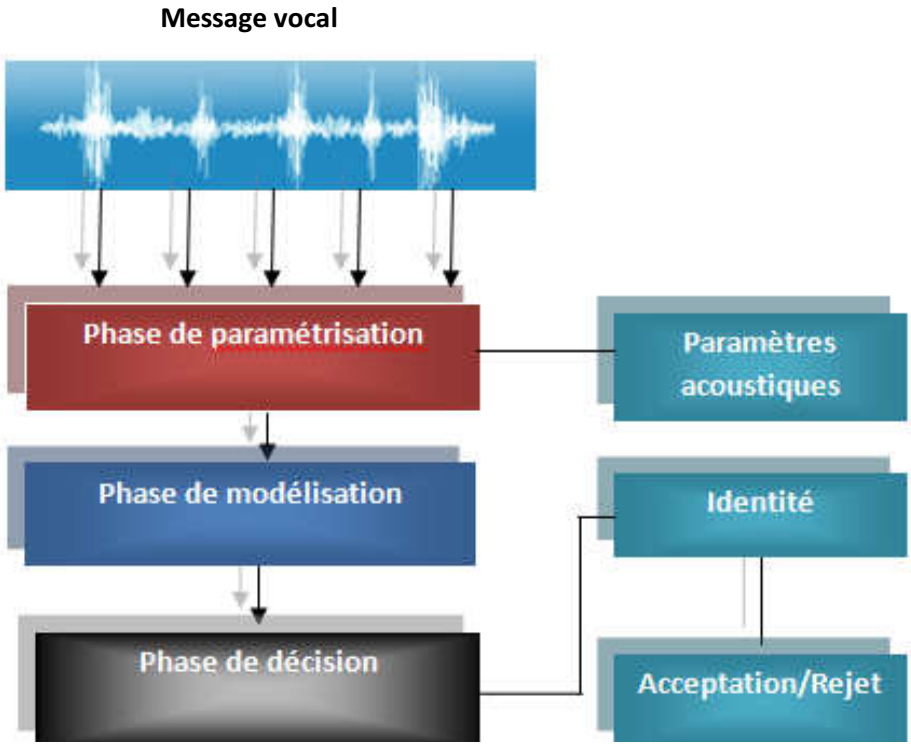


Figure 3.3 : **Différentes phases d'identification du locuteur**

3.3.1 La phase de paramétrisation

Les systèmes de reconnaissance du locuteur utilisent des représentations du signal de parole dans lesquelles le bruit et la redondance ont été réduits afin de ne conserver que les informations considérées comme utiles à la tâche spécifiée. Cette phase est appelée la phase de paramétrisation ou paramétrisation du signal de parole. C'est la phase la plus importante en reconnaissance du locuteur. Dans cette phase, les paramètres à identifier doivent être fréquents, facilement mesurables, pas trop sensibles à la variabilité intra-locuteur et robustes face aux imitateurs.

Pour pouvoir obtenir ces paramètres, la méthode la plus sûre et la plus utilisée est la méthode des **MFCC**s (Mel-frequency Cepstral Coefficients) qui seront calculés à l'aide de **TFD** (Transformée de Fourier discrète).

a. Transformée de Fourier discrète

La transformée de Fourier discrète ou **TFD** (Transformée de Fourier Discrète) permet de calculer la transformée de Fourier d'une suite d'échantillons au lieu d'une fonction continue **[8]**.
On définit la **TFD** comme suit :

$$X_K = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi jnk}{N}}$$

Où (x_n) est la suite des échantillons temporels, (X_k) la suite des échantillons fréquentiels pour k= [0, 1, 2,…, N-1], et N le nombre d'échantillons temporels.

b. **FFT**

La **FFT** (Fast Fourier Transform) est un algorithme rapide de **TFD** (Transformée de Fourier Discrète). Elle est très utilisée dans le traitement numérique du signal sonore. Grâce à elle on peut isoler les différentes fréquences qui composent tout son constitué par la superposition de plusieurs ondes sinusoïdales.

Le but de l'algorithme de Transformée de Fourier rapide est de calculer la **TFD** avec une complexité minimale ou d'évaluer les intégrales de Fourier en minimisant le nombre d'opération de façon à obtenir un algorithme rapide. L'algorithme le plus utilisé est celui de **Cooley-Tuckey [9]**.

Algorithme de Cooley-Tuckey

En posant

$$X[k] = \sum_{n=0}^{N-1} x[n] \,.\, W_N^{nk} \qquad \text{avec } W_N = e^{-\frac{2\pi j}{N}}$$

Propriétés du W_N

$$W_N^{2nk} = e^{-\frac{2\pi jnk}{N/2}} = W_{N/2}^{nk}$$

$$W_N^{nk+N/2} = e^{-\frac{2\pi j(nk+\frac{N}{2})}{N}} = -W_N^{nk}$$

Ces formules sont valables à condition qu'on ait un nombre d'échantillons puissance de 2, c'est-à-dire $N=2^m$ où m=1,2,3,…

Et les indices paires et impaires sont : $\begin{cases} x_1 = x[\ 2n\ ] \\ x_2 = x[2n+1] \end{cases}$

Ainsi l'algorithme de **Cooley-Tuckey** est défini comme suit :

$$\begin{cases} X[k] = x_1[k] + x_2[k].W_N^k \\ X\left[k+\frac{N}{2}\right] = x_1[k] - x_2[k].W_N^k \end{cases}$$

c. MFCC

Les coefficients Mel Cepstre ou **MFCC**s (Mel-frequency Cepstral Coefficients) servent à déterminer les paramètres de l'analyse spectrale. Ces coefficients font partie des paramètres les plus couramment utilisés en reconnaissance du locuteur.

Ils fournissent une représentation acoustique du signal vocal qui est adaptée à la reconnaissance vocale. Aussi, les MFCCs caractérisent bien la forme du spectre et permettent de séparer l'influence de la source du signal vocal de celle du conduit vocal **[10]**.

Les MFCCs sont définis de la manière suivante :

$$MFCC_i = \sum_{k=0}^{N} X[k].\cos\left[\frac{i\left(k-\frac{1}{2}\right)}{N}\right] \quad , i = [1, 2, \ldots\ldots, M]$$

Où **M** le nombre de coefficients **MFCC**, typiquement de l'ordre de 10 à 20 pour la voix, **X[k]** les énergies du signal analysé pour **k= [1,…, N]** et **N** est le nombre d'échantillonnage.

3.3.2 La phase de modélisation

La phase de modélisation est la modélisation acoustique du locuteur ou généralement la modélisation générative du locuteur dont le but est d'estimer la distribution qui a pu générer les vecteurs cepstraux du signal d'apprentissage.

C'est dans cette phase qu'on peut bien distinguer les caractéristiques spectrales des locuteurs. Généralement les systèmes de reconnaissance du locuteur utilisent pour la plupart des algorithmes de comparaison de motifs. La technique de modélisation la plus prometteuse est la technique à base du modèle **GMM (**Gaussian Mixture Model**)** ou Modèles à mélanges de gaussiens maximisant la vraisemblance des données d'apprentissage, car ce modèle est capable de capturer les points communs entre différentes représentations de motifs spectraux issus du même locuteur.

GMM

En reconnaissance du locuteur, on modélise souvent ce dernier comme une source pouvant avoir plusieurs comportements gaussiens. Le mélange de gaussiennes est un modèle basé sur l'algorithme **EM** (Expectation-Maximisation). L'objectif de ce modèle est d'estimer les densités de probabilités des classes acoustiques afin d'avoir une meilleure approximation de la distribution correspondant au locuteur. La densité de probabilité pour une mixture de gaussiennes à **N** composantes pour une variable aléatoire **x** s'exprime sous la forme suivante **[11]**:

$$P(X|\theta) = \sum_{i=1}^{N} \gamma_i \, N(x, \mu_i; \Sigma_i)$$

Où $N(x ; \mu_i, \Sigma_i)$ est la loi gaussienne de moyenne μ et de variance **Σ**, $\Theta = [\mu, \Sigma, \gamma]^T$ est le vecteur de paramètre global du **GMM** et γ est le vecteur de poids de la mixture.

Cette densité de probabilité d'une mixture de gaussiennes sert à calculer la vraisemblance qui est une méthode très utile à la reconnaissance des caractères acoustiques de chacun des locuteurs afin de les identifier. La vraisemblance pour un gaussien multidimensionnel est définie de la manière suivante :

$$l(x|\mu, \Sigma) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right]$$

Où **l(x|μ,Σ)=log p(x|μ,Σ)** et **d** la dimension de **x.**

Les paramètres de ce modèle **GMM** obtenus à partir de l'algorithme **EM** sont définis de la manière suivante :

- **La moyenne pondérée des données**

$$\mu_j = \left[\sum_{n=1}^{N} \gamma_j(x_n)\right] . \left[\sum_{n=1}^{N} \gamma_j(x_n)\right]^{-1}$$

- **Les covariances**

$$\Sigma_j=[\sum_{n=1}^{N}\gamma_j\,(x_n)\left(x_n-\mu_j\right)^{\mathsf{T}}].[\,\sum_{n=1}^{N}\gamma_j(x_n)]^{-1}$$

- **Le vecteur de poids de la mixture**

$$\gamma_j(x_n)=\ \pi_j N\left(x_n\middle|\mu_j,\Sigma_j\right)[\,\Sigma_k\pi_k N(x_n|\mu_k,\Sigma_k)]^{-1}$$

Avec $\pi_j=\frac{1}{N}\sum_{n=1}^{N}\gamma_i(x_n)$

3.3.3 La phase de décision

La phase de décision est la dernière étape en identification du locuteur qui désigne ce dernier finalement reconnu. Dans tous les systèmes de reconnaissance du locuteur il faut, à un moment ou à un autre, prendre la décision d'accepter ou de rejeter un segment de parole comme appartenant au client dont on cherche à vérifier l'identité. La stratégie mise en jeu de cette phase dépendra fortement de la phase de modélisation. Cette stratégie consiste, à partir d'un ensemble de locuteurs, à fournir l'identité correspondant au modèle qui a la plus forte vraisemblance sur les données dans la base **[12].**

C'est ici que l'étape de reconnaissance proprement dite s'effectue puisque le système émet une réponse : une identité pour l'identification, c'est-à-dire que si on considère un ensemble de locuteurs **$S=\{S_1,\ldots,S_n\}$,** l'identité **S_l** est acceptée si elle remplit la condition suivante :

$S_l=\max p(S_i|X)=\max p(X|S_i)$; avec X la phrase prononcée et i=1,…,n

, ou elle est rejetée dans le cas contraire.

Chapitre IV : REALISATION

4.1 Introduction

« **RecSpeaker**» est un logiciel conçu dans le but démonstratif. Il traite informatiquement la reconnaissance d'une voix humaine ou la reconnaissance du locuteur. En effet, il implémente le traitement du signal numérique dont l'objectif est de distinguer les caractéristiques vocales de chaque individu. De plus, ce logiciel intègre une application pour sécuriser vocalement n'importe quel type de fichier, et il sert aussi la visualisation des spectres et le traitement des fichiers audio.

4.2 Description du logiciel « RecSpeaker »

4.2.1 Programmation

Les programmes du logiciel « **RecSpeaker**» ont été écrits en langage C avec Visual Studio 2005. Il utilise :

- L'**API** Windows pour l'interface utilisateur.
- La base de données Access pour stocker les locuteurs.
- Fmod et Windows Multimedia pour l'interface avec les périphériques audio et les fichiers audio.
- L'algorithme de Cooley-Tukey pour le calcul de la **FFT**.
- L'algorithme de Gauss pour le calcul de **GMM**.
- Les syntaxes **Sql** pour la connexion à la base de données.

4.2.2 Présentations des interfaces

a. Interface principale

L'interface principale illustrée par la Fig4.1 a été réalisée avec l'**API Windows** en langage C. Elle a 6 boutons différents tels que : **A propos**, **Application**, **Visualisation**, **Enregistrer locuteur**, **RecSpeaker** et **Quitter**.

Figure 4.1 : **Interface principale du « RecSpeaker »**

b. La fonction des boutons

i. Bouton « A propos »

Ce bouton affiche la version du « **RecSpeaker** ».

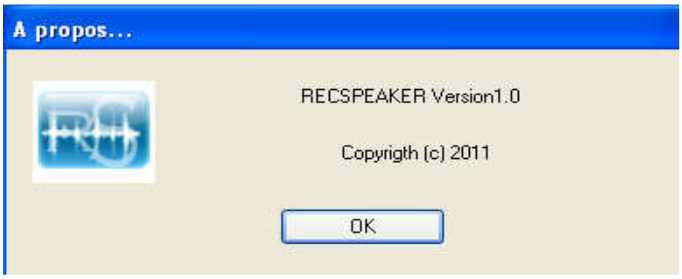


Figure 4.2 : **Effet du bouton « A propos»**

ii. Bouton « RecSpeaker »

C'est ici la principale tâche de l'identification par la biométrie vocale. Ce bouton **« RecSpeaker »** donne la boite de dialogue montrée par la Fig 4.3. Cette boite de dialogue contenant 8 boutons différents, sert à identifier une voix d'un individu au format **wave** seulement**.** Mais, peut aussi lire des fichiers audio comme **mp3** et **wma**.

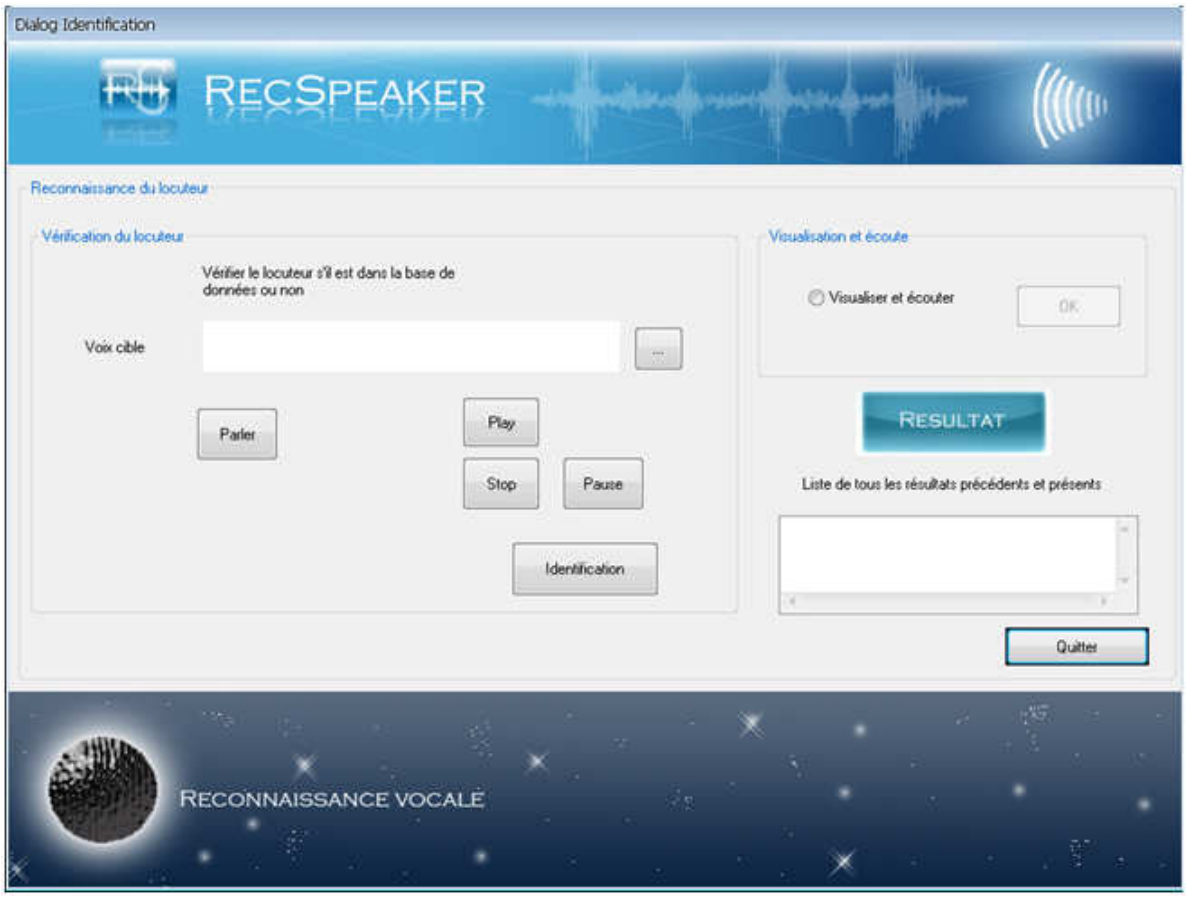


Figure 4.3 : **Boite de dialogue pour l'identification du locuteur**

– ***Bouton « parcourir »*** 

Apres avoir cliqué sur ce bouton, la boite de dialogue de navigation permettant de choisir le fichier **wave, mp3** et **wma** de la Fig 4.4 apparait. Si on veut identifier un locuteur, il faudra choisir un fichier **wave** seulement. Mais pour la lecture on peut choisir un fichier **mp3** ou **wave** ou **wma**.

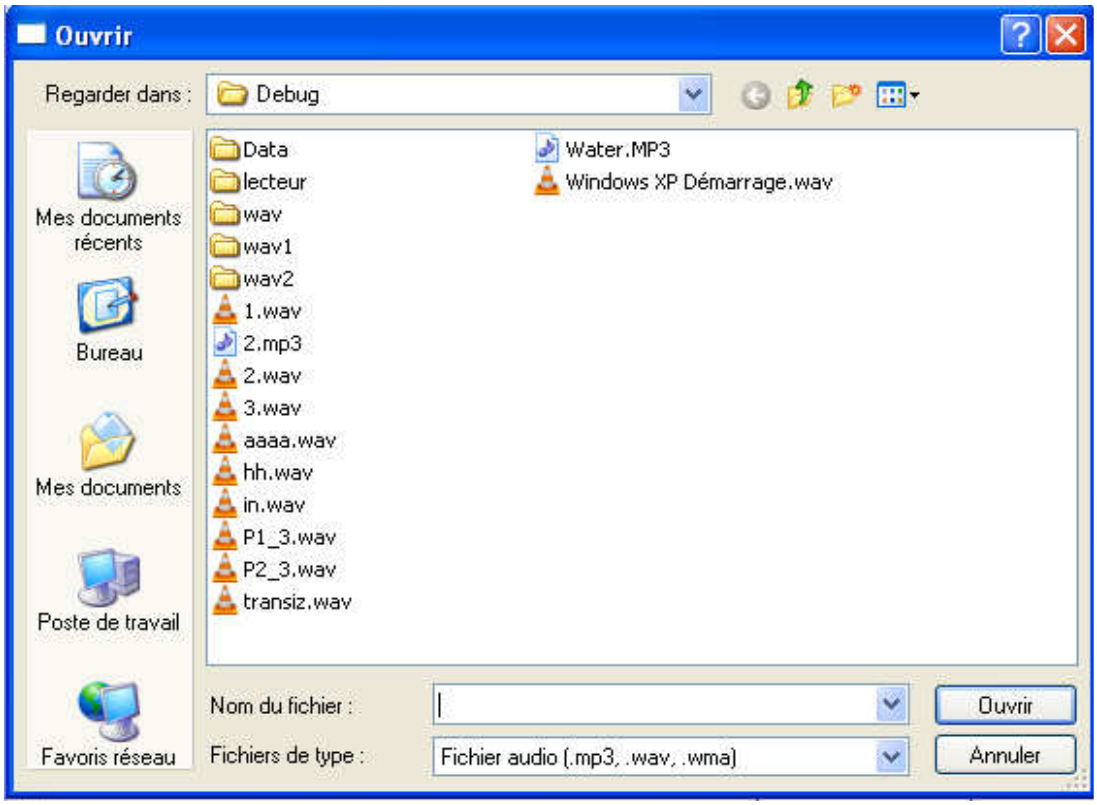
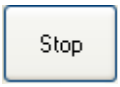


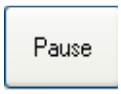
Figure 4.4 : **Boite de dialogue de navigation**

- ***Bouton « Play »*** 

  Ce bouton sert pour la lecture des fichiers audio aux formats **wave**, **mp3** et **wma**.

- ***Bouton « Stop »*** 

  Si on veut arrêter la lecture, on clique sur ce bouton.

- ***Bouton « Pause »*** 

  Ce bouton permet de faire la pause d'une lecture encours.

- ***Bouton « Parler »*** 

  Ce bouton ouvre une autre boite dialogue illustrée par la Fig 4.5 pour l'enregistreur vocal.

Figure 4.5 : **Magnétophone**

– ***Bouton « Identification »*** 

Après avoir ouvert un fichier au format **wave** contenant les caractéristiques vocales d'un locuteur dans la boite de dialogue de navigation, la zone de texte de la Figure 4.3 contient le chemin complet du fichier. Ensuite, il suffit d'appuyer sur le bouton ***« Identification »*** pour vérifier si ces caractéristiques ressemblent à celles qui sont déjà préenregistrées.

– ***Bouton « Quitter »*** 

Pour quitter l'application, on clique sur ce bouton.

– ***Bouton radio « Visualiser et écouter »*** 

Si on veut visualiser des fichiers audio tels que **mp3** et **wav**, il suffit d'appuyer sur ce bouton radio et le bouton ***« ok »*** sera activé.

iii. Bouton « Visualisation »

On peut visualiser à l'aide de la boite de dialogue de la Fig 4.6 le spectre d'un signal sonore. Elle est composée de 3 boutons tels que : ***« WaveEditor »***, ***« VisualSpectro »*** et ***« WavMaker ».***

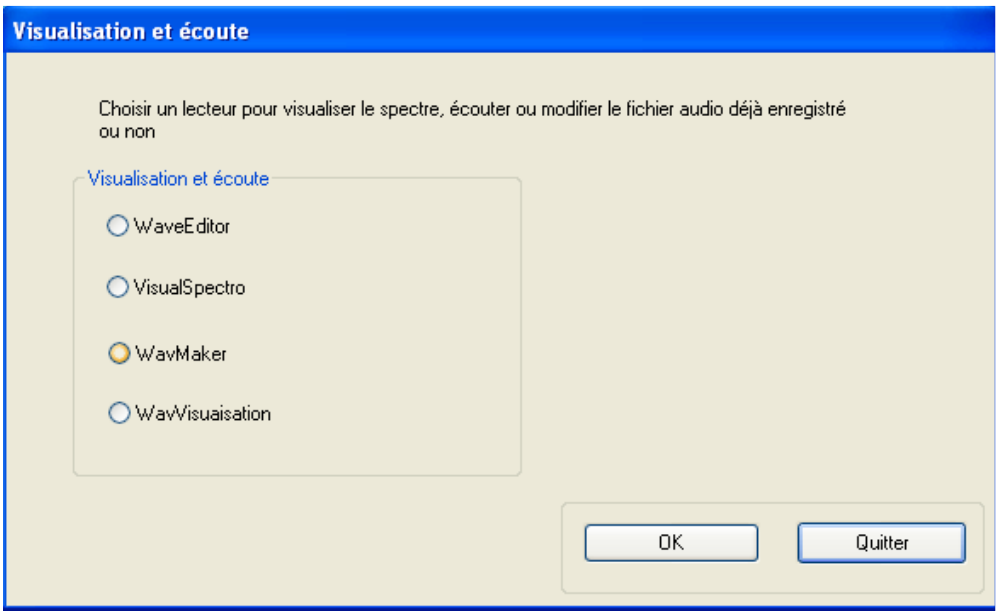


Figure 4.6 : **Boite de dialogue de visualisation spectrale**

- ***Bouton radio « WaveEditor »***

RecSpeaker fait aussi appel à ***WaveEditor1.0*** pour la visualisation d'un signal en fonction du temps. Le fichier doit être au format **wave**.

Sphinx WaveEditor version1.0 est un logiciel à opensource.

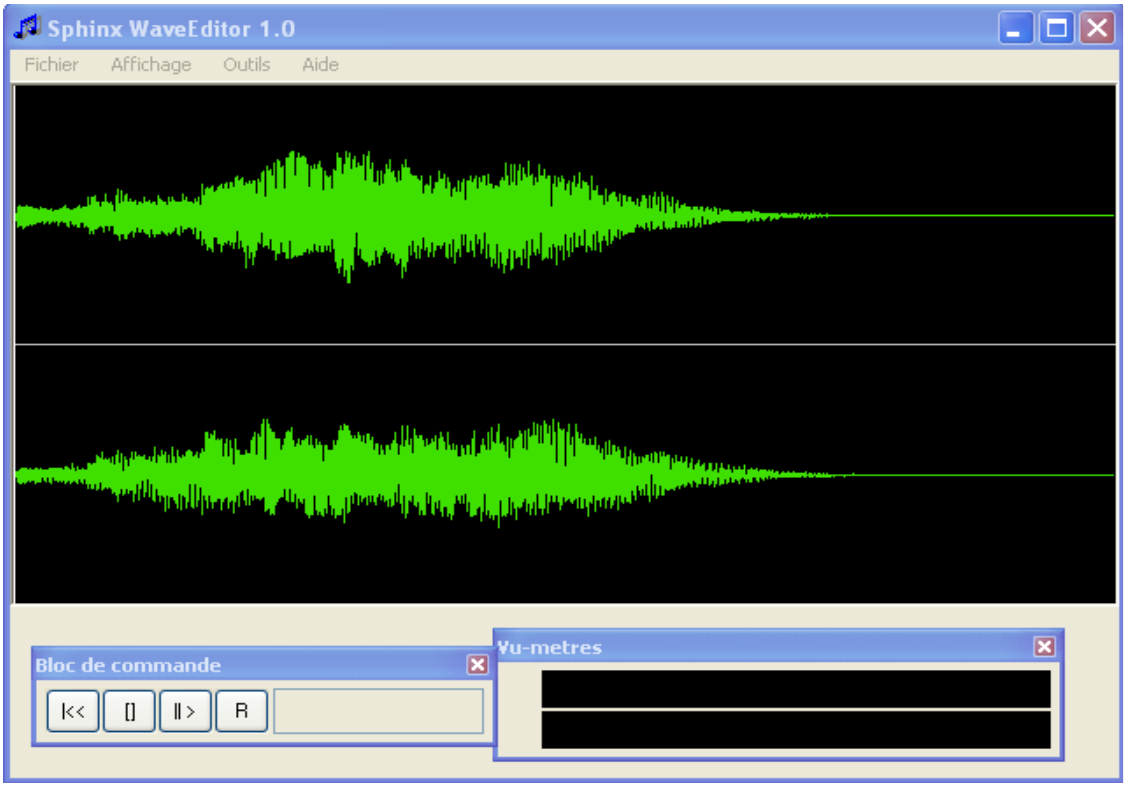


Figure 4.7 : **Sphinx WaveEditor 1.0**

Après avoir cliqué sur le bouton ***« VisualSpectro »*** et ensuite sur ***« ok »***, l'interface de « ***VisualSspectro »*** de la Fig 4.8 apparait. Il est, non seulement un outil de visualisation spectrale des signaux vocaux, mais aussi un lecteur des fichiers audio (**mp3,wma et wave**).

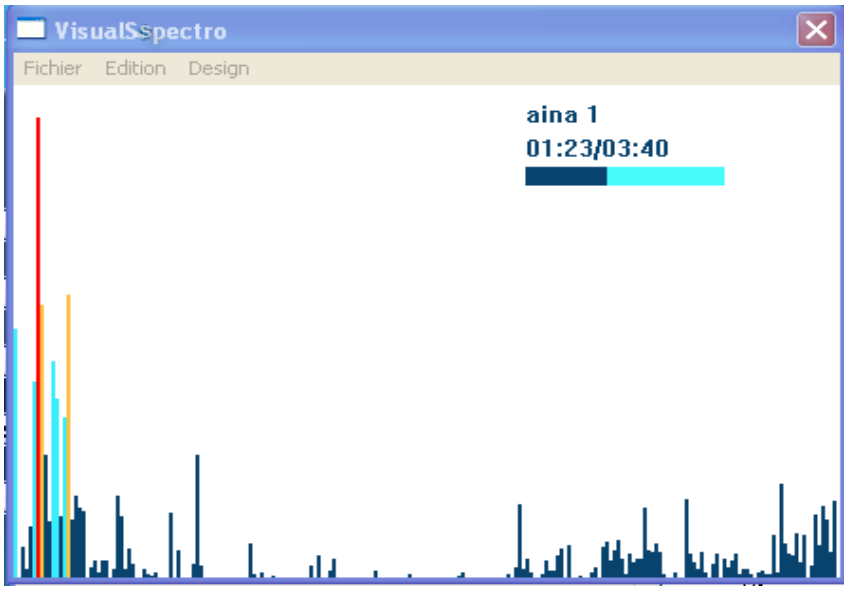


Figure 4.8 : **VisualSspectro**

– ***Bouton radio « Wavemaker »***

Ce bouton fait apparaitre l'éditeur de fichier wave ***« Makewav »*** permettant de changer les caractéristiques de ce dernier, illustré par la Fig 4.9.

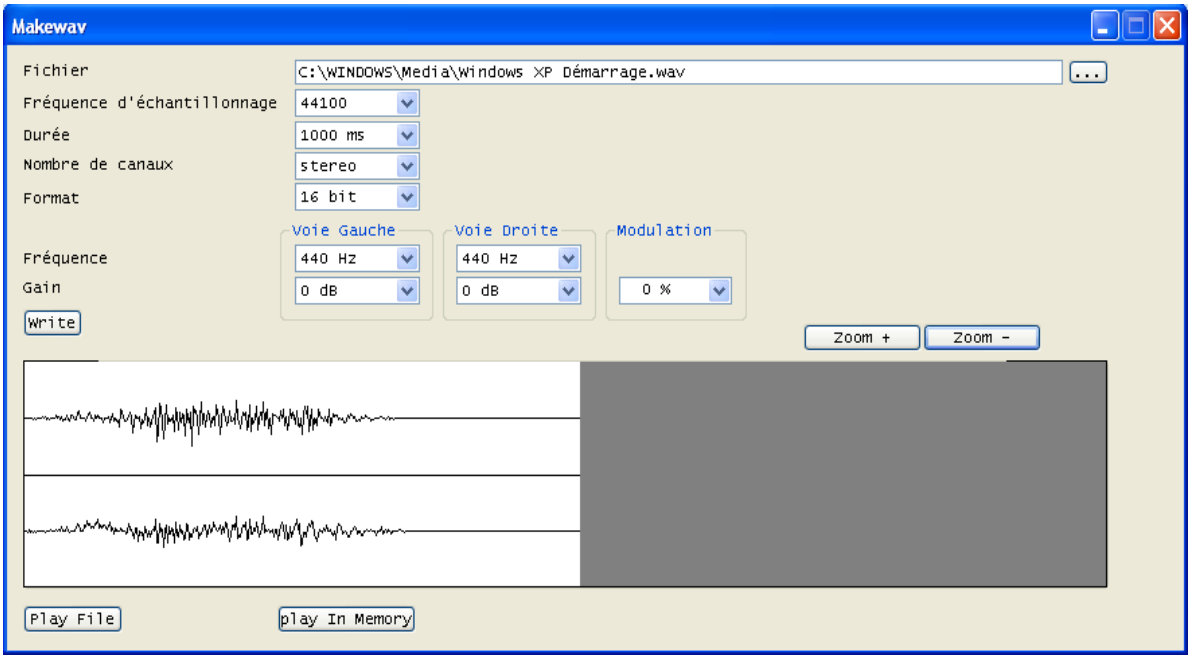


Figure 4.9 : **Makewav**

– ***Bouton radio « WavVisualisation »***

Ce bouton fait apparaitre la boîte de dialogue de visualisation d'un fichier wav permettant de changer les caractéristiques de ce dernier, illustré par la Fig 4.10.

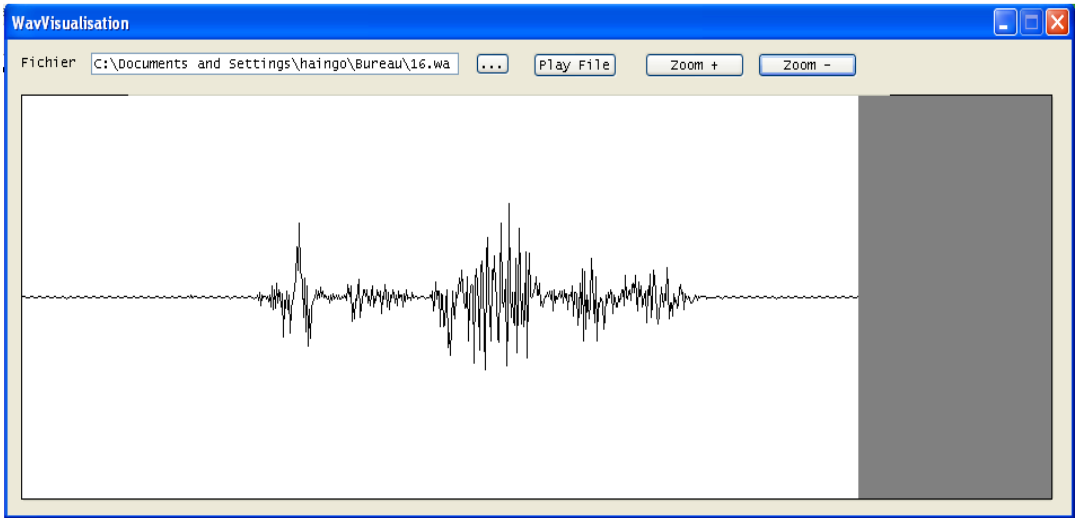


Figure 4.10 : **Visualisation d'un fichier wav**

iv. Bouton « Enregistrer locuteur »

C'est ici qu'on devra faire l'enregistrement, la suppression et la modification des locuteurs. La Fig 4.11 nous montre la boite de dialogue de la mise à jour des données.

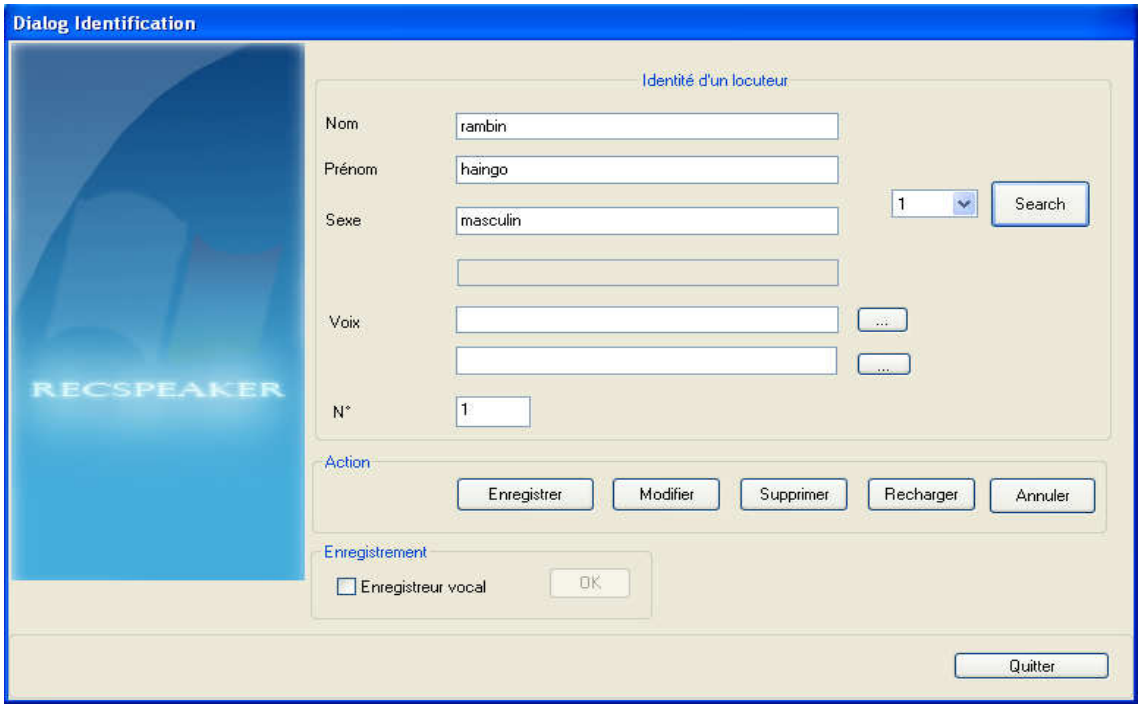


Figure 4.11 : **Mise à jour des données**

v. Bouton « Application »

La boite de dialogue illustrée par la Fig 4.12 nous montre l'application de la reconnaissance du locuteur. Cette application consiste à crypter un fichier quelconque du locuteur déjà enregistré. Et lors du décryptage, il suffit d'entrer la voix clé du locuteur.

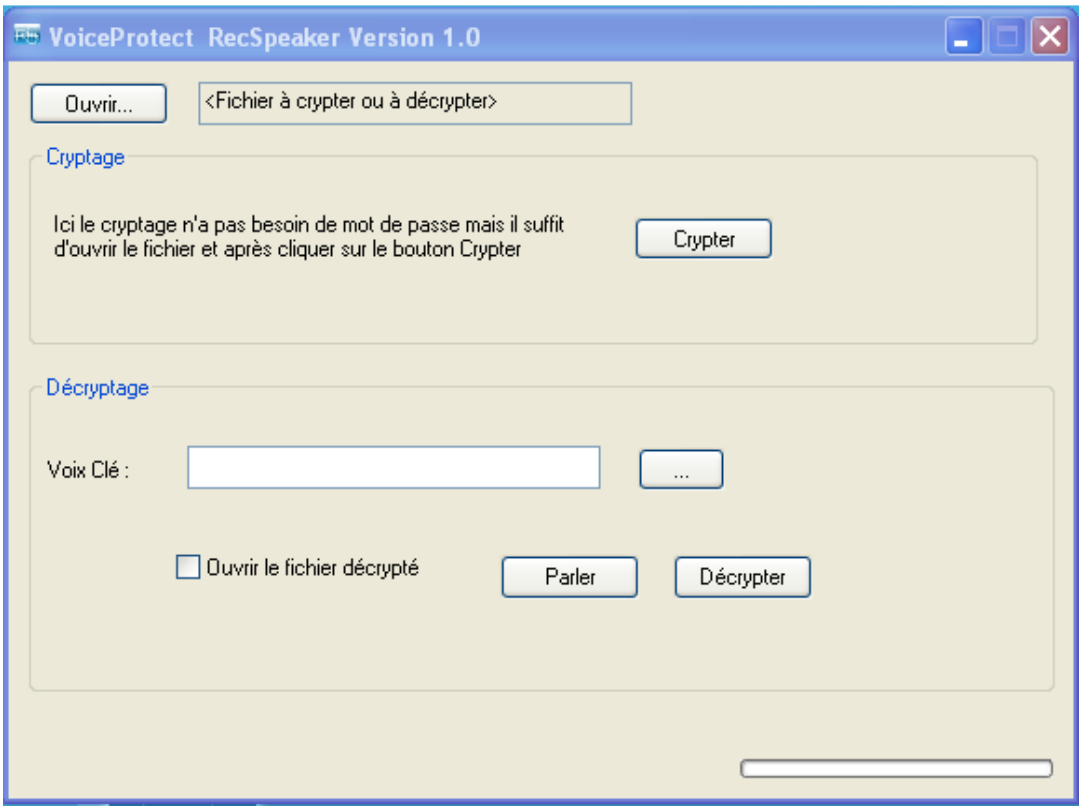


Figure 4.12 : **Effet du bouton « RecSpeaker »**

Conclusion

Ce travail est axé sur le traitement vocal qui a pour but de reconnaitre une personne à partir de sa voix. Par ailleurs, On a pu constater que chacun a ses propres caractéristiques vocales, ce qui nous amène dans le sujet.

Pour ce faire, on a vu dans ce rapport de mémoire, en premier lieu, la numérisation d'un signal précédé de la généralité de la biométrie vocale, ainsi que l'identification du locuteur grâce à l'évolution du calcul mathématique en but d'extraire les différents paramètres comme le MFCC, le GMM. Enfin, la réalisation du logiciel qui s'intitule « **RecSpeaker** », qu'on peut identifier le locuteur à partir de sa voix.

La biométrie vocale est parmi les techniques biométriques la plus utilisée et convenable pour tout le monde. Cependant en réalisant ce travail, on a pu constater qu'une difficulté de la biométrie vocale vient du fait que suivant le locuteur (variation intra-locuteur), les matériels utilisés (l'intelligibilité des matériels), la nature de l'environnement, ce qui amène toujours une certaine incertitude sur la décision du système malgré tout effort.

Annexes

Annexe A : LA PROGRAMMATION SOUS WINDOWS

A.1 Fenêtre principale

a. Introduction

Windows est un système d'exploitation proposant une interface graphique. Dans ce premier article, nous allons créer une application composée seulement d'une fenêtre principale, dont le but et de montrer l'architecture d'un programme Windows.

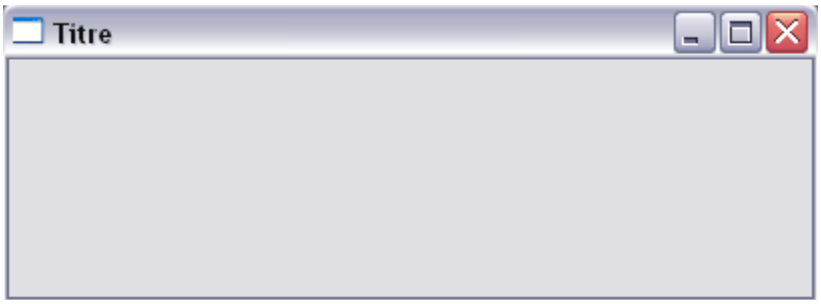


Figure A.1 : **Fenêtre principale**

b. Fonction WinMain

Le point d'entrée d'une application Windows est la fonction **WinMain**. C'est l'équivalent de la fonction main des applications classiques. Elle est appelée par le système d'exploitation au lancement du programme. Il lui fournit 4 paramètres.

```
int WINAPI WinMain(HINSTANCE hinstance, HINSTANCE hPrevInstance,
                                                LPSTR lpCmdLine, int nCmdShow);
```

Le premier paramètre est le handle d'instance de l'application. C'est un numéro unique attribué par le système d'exploitation qui lui permet de l'identifier.
Le second paramètre est toujours **NULL** pour les applications Win32.
Le troisième paramètre est un pointeur sur la ligne de commande.

c. Création de la fenêtre

La fonction qui permet de créer une fenêtre se nomme **CreateWindow** :

```
HWND CreateWindow(
    LPCTSTR lpClassName,  // Pointeur sur une classe de fenêtre.
    LPCTSTR lpWindowName,  // Pointeur sur le texte de la fenêtre.
    DWORD dwStyle,  // Style de la fenêtre.
    int x,  // Position horizontale de la fenêtre.
    int y,  // Position verticale de la fenêtre.
    int nWidth,  // Largeur de la fenêtre.
    int nHeight,  // Hauteur de la fenêtre.
    HWND hWndParent,  // Handle de la fenêtre parent.
    HMENU hMenu,  // Handle de menu ou ID de contrôle.
    HANDLE hInstance,  // Handle d'instance de l'application.
    LPVOID lpParam   // Pointeur sur des données passées à WM_CREATE.
   );
```

Le premier paramètre qu'elle reçoit est un pointeur sur une chaîne de caractères identifiant la classe de fenêtre. Les classes de fenêtre sont des modèles pour construire les fenêtres (le terme de classe n'a rien à voir avec les classes du C++). Si pour les contrôles standard nous avons des classes de fenêtres prédéfinies et globale, nous devrons en créer une pour la fenêtre principale. Nous devons pour cela remplir une structure de type WNDCLASS (définie dans winuser.h).

```
WNDCLASS wc;

wc.style = 0;
wc.lpfnWndProc = MainWndProc;
wc.cbClsExtra = 0;
wc.cbWndExtra = 0;
wc.hInstance = hinstance;
wc.hIcon = LoadIcon(NULL, IDI_APPLICATION);
wc.hCursor = LoadCursor(NULL, IDC_ARROW);
wc.hbrBackground = (HBRUSH)(1 + COLOR_BTNFACE);
wc.lpszMenuName =  NULL;
wc.lpszClassName = "MaWinClass";
```

d. Boucle de message

Pour communiquer avec une application ou ses diverses fenêtres, Windows leurs envoie des messages. Par exemple, si vous cliquez sur le bouton fermeture (en haut à droite de la fenêtre), Windows va créer un message approprié qu'il va envoyer dans la file d'attente de l'application. La file d'attente est un tampon où sont stockés les messages en attente de traitement. C'est à nous de coder l'extraction des messages de la file d'attente :

```
    MSG msg;

    while (GetMessage(&msg, NULL, 0, 0))
    {
        TranslateMessage(&msg);
        DispatchMessage(&msg);
    }
```

e. La procédure d'une fenêtre

Voici la procédure de fenêtre de notre fenêtre principale dont nous avions passé un pointeur à la classe de fenêtre au début de ce document (on passe ce pointeur à la classe de fenêtre car c'est nous qui la créons, mais le système d'exploitation qui l'appelle, il doit donc la localiser).

```
LRESULT CALLBACK MainWndProc(HWND, UINT, WPARAM, LPARAM);


LRESULT CALLBACK MainWndProc(HWND hwnd, UINT uMsg, WPARAM wParam, LPARAM lParam)
{
    switch (uMsg)
    {
        case WM_CREATE:

            return 0;

        case WM_DESTROY:
            PostQuitMessage(0);
            return 0;

        default:
            return DefWindowProc(hwnd, uMsg, wParam, lParam);
    }
}
```

Les paramètres qu'elle reçoit sont les données du message en cours de traitement. Celui qui a été envoyé par DispatchMessage. C'est à nous, développeurs, d'implémenter des actions en fonction du message que la procédure de fenêtre reçoit. Dans cet exemple, un seul message est traité : WM_DESTROY. Il nous indique que la fenêtre est fermée et a été détruite. Comme il s'agit de la fenêtre principale de l'application, il est d'usage de fermer l'application. Ce que nous ferons en appelant la fonction **PostQuitMessage** qui a pour fonction de poster un message WM_QUIT dans la file d'attente. Qui comme nous l'avons vu plus haut dans ce document, met fin à la boucle de messages et donc à l'application. Les messages non traités doivent l'être par la fonction **DefWindowProc**. Fonction qui implémente le comportement par défaut d'une fenêtre.

A.2 La boite de dialogue

a. Le script de ressources

Figure A.2 : **Boîte de dialogue personnalisée**

Les boîtes de dialogue personnalisées sont créées à partir de ressources :

```
DIALOG1 DIALOG
     60, 60, 160, 80
          STYLE WS_POPUP | WS_VISIBLE | WS_CAPTION | WS_SYSMENU
                                                    CAPTION "A propos"
BEGIN
    DEFPUSHBUTTON "Ok", IDOK, 56, 50, 42, 12
    ICON 2, -1, 20, 15, 32, 32
    LTEXT "Mon beau programme !", -1, 60, 18, 80, 10
END
```

La ressource est composée de son identificateur littéral, suivi de son type (DIALOG), suivie de ses propriétés (position, dimensions, style et titre), suivi de son contenu balisé par les mots BEGIN et END. Il est en général constitué de contrôles. Chaque contrôle est décrit par son type, suivi de son identification visuelle (texte pour le bouton et le contrôle texte, identificateur de ressource pour l'icône), suivi de sa constante numérique d'identification, puis de sa position dans la boîte de dialogue et enfin de ses dimensions (les largeurs et hauteurs des contrôles ne sont pas en pixel, mais dépendent de la police de caractères utilisée). L'icône et le le contrôle texte ont un identificateur à -1 car ils ne sont là que pour la décoration, mais il faut tout de même leur mettre un identificateur. Celui du bouton est à IDOK que nous n'avons pourtant pas défini. IDOK est défini dans Windows, il est envoyé à la procédure de fenêtre de la boîte de dialogue quand on appuie sur la touche "Entrée". Est défini aussi l'identificateur IDCANCEL qui lui est envoyé quand on appuie sur la touche "Echap" ou que l'on tente de fermer la boîte de dialogue.

b. Appel de la boîte de dialogue

La boîte de dialogue étant maintenant dans les ressources, nous pouvons l'appeler afin de l'ouvrir. C'est le rôle de la fonction **DialogBox**. Son premier paramètre est le handle d'instance de l'application, le second, un pointeur sur la chaîne de caractères identifiant la ressource, le troisième est le handle de la fenêtre parent et le dernier un pointeur sur sa procédure de fenêtre.

```
    switch (uMsg)
    {
        case WM_COMMAND:
            if(LOWORD(wParam) == IDM_ABOUT)
                      DialogBox(hinst, "DIALOG1" , hwnd, (DLGPROC)Dialog1Proc);
```

c. Procédure de fenêtre de la boîte de dialogue

Comme pour la procédure de fenêtre de la fenêtre principale c'est à vous de la définir. C'est Windows qui l'appellera quand elle aura un message.

```
LRESULT CALLBACK Dialog1Proc(HWND, UINT, WPARAM, LPARAM);

BOOL APIENTRY Dialog1Proc(HWND hDlg,UINT uMsg,WPARAM wParam,LPARAM lParam)
{
    switch (uMsg)
    {
      case WM_INITDIALOG:

         return TRUE;

      case WM_COMMAND:
         if (LOWORD(wParam) == IDCANCEL || LOWORD(wParam) == IDOK)
                {
                   EndDialog(hDlg,0);
                   return TRUE;
                }

      default:
         return FALSE;
    }
}
```

Elle est fort semblable à celle de la fenêtre principale. Si le message est traité, elle doit renvoyer TRUE sinon elle doit renvoyer FALSE. WM_INITDIALOG doit être intercepté et renvoyer TRUE afin que les commandes IDOK et IDCANCEL soit envoyées lors des appuis sur les touches correspondantes. Le message WM_INITDIALOG est envoyé par Windows après la création de la boîte de dialogue, mais avant qu'elle soit visible **[13]**.

Annexe B : GENERALITE SUR LE TRAITEMENT DU SIGNAL

B1. Généralité sur les signaux

Un signal est toute manifestation sous forme d'une grandeur physiquement observable d'un phénomène le plus souvent physique, acoustique, optique ou électronique. Un signal est aussi la représentation physique de l'information qui est transportée d'une source vers la destination [**8].**

a. Notion de base

— **Energie**

L'énergie d'un signal x(t) est donnée par :

$$\mathbf{E} = \int_{-\infty}^{+\infty} |x(t)|^2 \, \mathbf{dt}$$

On dit que le signal est énergie finie si :

$$\mathbf{E} = \int_{-\infty}^{+\infty} |x(t)|^2 \, \mathbf{dt} < \infty$$

— **Théorème de conservation d'énergie**

Soit x(t) un signal à énergie finie et X(f) son spectre :

$$\int_{-\infty}^{+\infty} |x(t)|^2 \, \mathbf{dt} = \int_{-\infty}^{+\infty} |X(f)|^2 \, \mathbf{dt}$$

— **Produit de convolution**

Le produit de convolution de deux signaux x(t) et y(t) noté x(t)*y(t) est

$$\mathbf{x(t)*y(t)} = \int_{-\infty}^{+\infty} \mathbf{x(t).y(t-\tau)\,dt} = \int_{-\infty}^{+\infty} \mathbf{x(t-\tau).y(t)\,dt}$$

Le produit de convolution a les propriétés suivants :

$$\mathbf{x(t)*y(t) = y(t)*x(t)}$$

$$\mathbf{x(t)*\delta(t-\tau) = x(t-\tau)}$$

b. Les signaux

- **Les différents types des signaux**
  - ✓ **Les signaux à temps continus (ou analogiques)**

Les signaux à temps sont des signaux dont l'amplitude peut varier librement sur l intervalle de temps sur lequel il est défini (Fig. B.1).

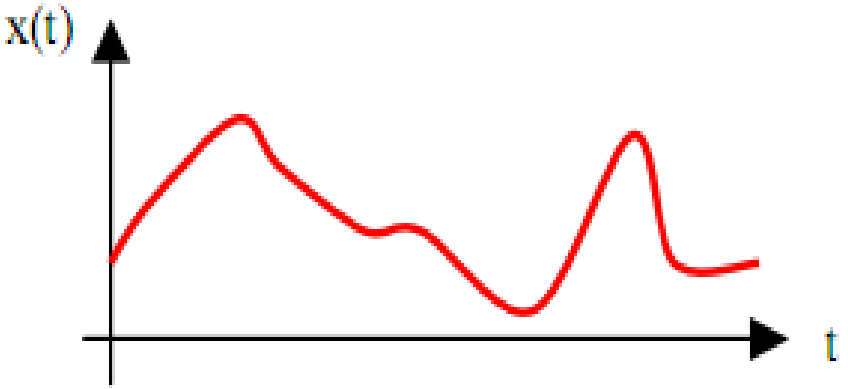


Figure B.1 : **Signal analogique**

✓ **<u>Les signaux à temps discrets</u> (ou échantillonnés)**

Les signaux à temps discrets sont des signaux qui ne sont définis qu'à des instants précis [t_k]. C'est une suite d'impulsions infiniment brèves (Fig. B.2)

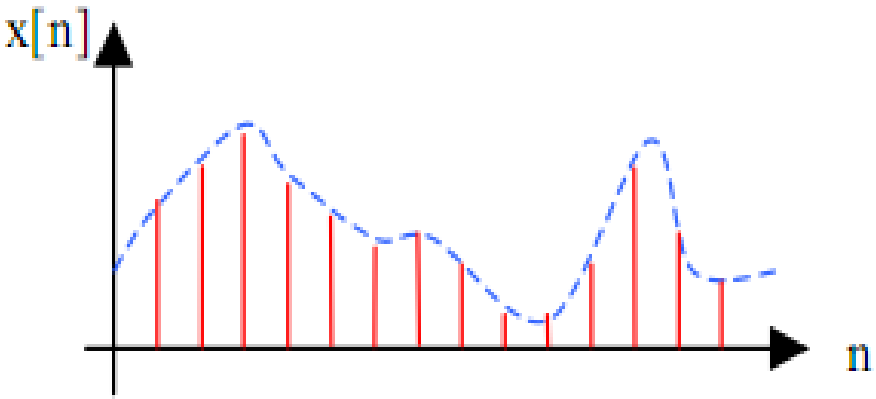


Fig. B.2 : **Signal échantillonné**

✓ **<u>Les signaux quantifiés</u>**

Les signaux quantifiés $x_T(t)$ sont des signaux dont l'amplitude ne prendre que des valeurs précises (Fig. B.3).

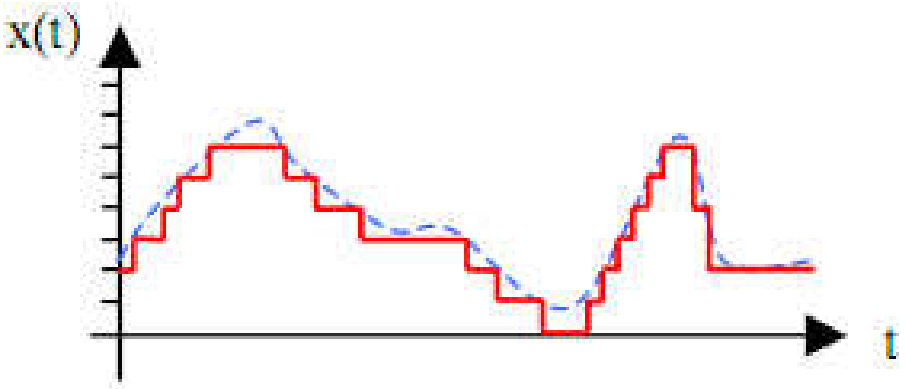


Fig. B. 3 : **Signal quantifié**

- **Signaux particuliers**

Afin de simplifier les opérations ainsi que les formules obtenues, certains signaux fréquemment rencontrés en traitement du signal dispose d'une modélisation propre **[6].**

✓ **Fonction signe**

$$\mathbf{Sgn(t)} = \begin{cases} -\mathbf{1} & \mathbf{si\ t} < 0 \\ \mathbf{1} & \mathbf{si\ t} > 0 \end{cases}$$

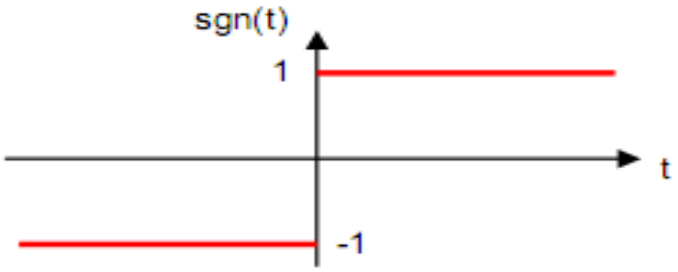


Fig. B. 4 : **Fonction signe**

Par convention, on admet pour valeur à l'origine : sgn (t) =0 pour t=0.

✓ **Fonction échelon**

$$\mathbf{Sgn(t)} = \begin{cases} \mathbf{0} & \mathbf{si\ t} < 0 \\ \mathbf{1} & \mathbf{si\ t} > 0 \end{cases}$$

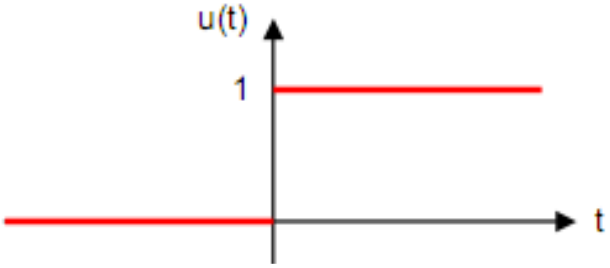


Fig. B. 5 : **Fonction échelon**

Par convention, on admet pour valeur à l'origine: u (t) = ½ pour t=0.

Dans certains, il sera préférable de lui donner la valeur 1.

- ✓ **Fonction rampe**

r(t) = t . u(t)

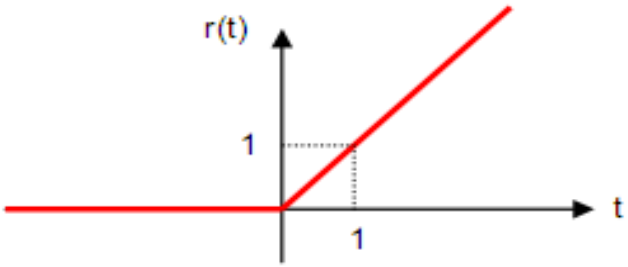


Fig. B. 6 : **Fonction rampe**

- ✓ **Fonction rectangulaire**

$$\mathbf{rect(t)} = \begin{cases} \mathbf{1} & \mathbf{si} -\mathbf{1/2} < t/T < 1/2 \\ \mathbf{0} & \mathbf{ailleurs} \end{cases}$$

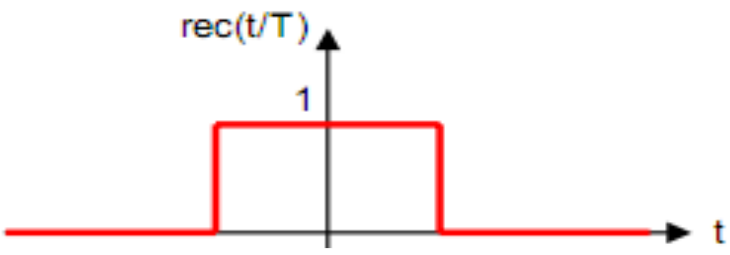


Fig. B. 7 : **Fonction rectangulaire**

On l'appelle aussi fonction porte. Elle sert de fonction de fenêtrage élémentaire.

✓ **Impulsion de Dirac**

L'impulsion de Dirac correspond à une fonction porte dont la largeur T tendrait vers 0 et dont l'aire est égale à 1.

$$\delta(t) = \begin{cases} \infty & pour\ t = 0 \\ 0 & pour\ \neq 0 \end{cases}$$

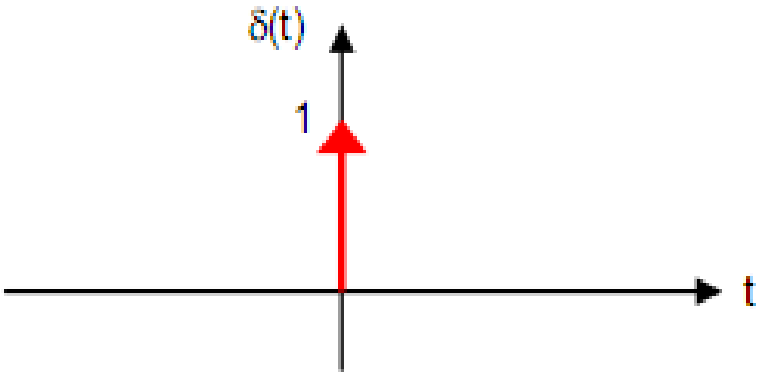


Fig. B. 8 : **Fonction Impulsion de Dirac**

On peut encore considérer $\delta(t)$ comme la dérivée de la fonction échelon : **δ(t) =** $\frac{du}{dt}$

✓ **Peigne de Dirac**

On appelle peigne de Dirac une succession périodique d'impulsions de Dirac.

$$\delta_T(t) = \sum_{k\to-\infty}^{+\infty} \delta(t - kT)$$

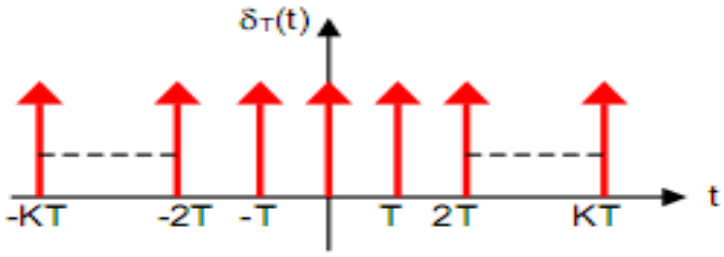


Fig. B. 9 : **Fonction Peigne de Dirac**

B.2. Transformée de Fourier

C'est une généralisation de la décomposition de série de Fourier à tous les signaux déterministes. Elle permet d'obtenir une représentation en fréquence (représentation spectrale) de

ces signaux. Elle exprime la répartition fréquentielle de l'amplitude, de la phase et de l'énergie (ou de la puissance) des signaux considérés.

NB : Les signaux déterministes ou signaux certains, leur évolution en fonction du temps peut être parfaitement modélisé par une fonction mathématique. On retrouve dans cette classe les signaux périodiques, les signaux transitoires, les signaux pseudo-aléatoires, etc.…

a. Définition

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

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

Si cette transformée existe, la transformée de Fourier inverse est donnée par :

S(t) = TF^{-1} [S(f)] = $\int_{-\infty}^{+\infty} \boldsymbol{S(f)}\ \mathbf{e^{j2\pi ft}}$ **df**

- **La transformée de Fourier d'un peigne de Dirac**

Comme TF[$e^{-j2\pi f_0 t}$] = $\delta(f + f_0)$
Alors **TF[$\boldsymbol{\delta_T(t)}$] =** $\frac{1}{T_e}\sum_{k\to-\infty}^{+\infty} \boldsymbol{\delta(f - kf_e)}$

- **Transformée de Fourier de la signal porte**

On a : TF[rect($\frac{t}{T}$)] = $\int_{-\infty}^{+\infty}$ rect($\frac{t}{T}$) $e^{-j2\pi ft}$dt = $\int_{-T/2}^{T/2} e^{-j2\pi ft}$dt

D'où **TF[rect($\frac{t}{T}$)] = T sinc(Tf)**

b. Notion de Filtrage

Le filtrage est une forme de traitement de signal qui modifie le spectre de fréquence et/ou la phase du signal présent en entrée du filtre et donc par conséquent sa forme temporelle. Il peut s'agir soit :

- ✓ d'éliminer ou d'affaiblir des fréquences parasites indésirables
- ✓ d'isoler dans un signal complexe la ou les bandes de fréquences utiles.

On classe les filtres en deux grandes familles :

- ✓ les filtres numériques réalisés à partir de structure intégrée microprogrammable
- ✓ es filtres analogiques réalisés à partir de composants passifs (résistance, inductance, condensateur) ou actifs.

B3. Transformée de Fourier d'un signal discret

a. Définition

Un signal discret est défini par une suite d'échantillons espacés entre eux d'une période T_e. La transformée de Fourier appliquée à un signal discret x[n] devient donc :

$$\boldsymbol{X(f) = \sum_{n\to-\infty}^{+\infty} X[n].e^{-2j\pi\frac{nf}{Fe}}}$$

Si cette série converge, la transformée de Fourier inverse est définie par :

$$\boldsymbol{X[n] = \frac{1}{F_e}\int_{-F_e/2}^{F_e/2} X[f].e^{2j\pi\frac{nf}{Fe}}}\ \textbf{df}$$

b. Propriétés

Le tableau B.1 montre quelques propriétés de la Transformée de Fourier.

	x[n]	**X(F)**
Linéarité	α.x(n) + β.y(n)	α.X(f) + β.Y(f)
Translation	x(n-k)	$X(f)\ e^{-2j\pi\frac{kf}{F_e}}$
	$x(n)\ e^{-2j\pi nf_0}$	X(f-f_0)
Convolution	x(n)* y(n)	S(f).R(f)
	s(t).r(t)	S(f) *R(f)

Tableau B.1 : **Propriétés de la Transformée de Fourier**

B.4. Transformée de Fourier discrète

a. Fenêtrage

Avec un ordinateur, il est impossible de calculer la transformée de Fourier d'un signal discret. En effet il faudrait un temps et une mémoire infinie Pour ces raisons, on est toujours amené à travailler avec un nombre fini de points N. Cela revient à dire que les signaux exploités numériquement sont toujours une troncation de signaux réels.

On construira donc un signal tronqué $x_T[n]$. Il résulte de la multiplication des échantillons de x[n] par une fenêtre d'analyse (ou encore fenêtre de troncature) qui limitera $x_T[n]$ à N échantillons. En pratique, on calcule donc :

$$\boldsymbol{X_{T(f)} = \sum_{n=0}^{N-1} X_T\ [n].e^{-2\pi j\frac{nf}{F_e}}}$$

La fenêtre d'analyse est définie par une suite d'échantillons y[n] tels que

$$\begin{cases} \mathbf{X_T[n] = Y[n].X[n]\ \ pour\ \ 0 \le n \le N-1} \\ \mathbf{X_T[n] = 0\ \ pour\ \ n\ } < 0\ et\ n\ >\ N-1 \end{cases}$$

Mais le fait de tronquer un signal peut notablement affecter son spectre.

b. <u>Echantillonnage en fréquence</u>

- En fait, lorsque l'on veut pouvoir représenter le spectre $X_T(f)$, il faut calculer $X_T(f)$ pour toutes les valeurs de f (f est une variable continue). Ceci est impossible avec un ordinateur qui ne peut traiter que des valeurs de f discrètes. Comme $X_T(f)$ est périodique de période F_e, on découpe donc cet intervalle en M parties égales et on ne calcule $X_T(f)$ que pour les multiples de F_e/M : on effectue un **échantillonnage fréquentiel** de pas $\Delta f = F_e/M$.

En remplaçant f par Δf, le calcul de la transformée de Fourier devient :

$X_T(k) = \sum_{n=0}^{N-1} X_T[n].\, e^{-2\pi j \frac{n.k.\Delta f}{Fe}}$ Avec k= [0……M-1]

Puisque $\Delta f = F_e/M$, on a :

$X_T(k) = \sum_{n=0}^{N-1} X_T[n].\, e^{-2j\pi \frac{n.k}{M}}$, Avec k= [0……M-1]

On vient ainsi d'introduire la **transformée de Fourier discrète**.

Le problème réside dans le choix du pas d'échantillonnage en fréquence et donc du choix de M. En effet, le fait d'échantillonner en fré TF^{-1} revient à périodiser dans le domaine temporel la partie du signal qui a été tronquée :

$X_T(k) = \sum_{k\to-\infty}^{+\infty} X_T(f).\, \delta(t - k.\Delta f)$ $\Longrightarrow$ $X_T(k) = \sum_{k\to-\infty}^{+\infty} X_T(t).\, \delta(t - \frac{r}{\Delta f})$

Ainsi, suivant le choix de Δf, plusieurs cas peuvent se présenter lors de la reconstitution du signal dans le domaine temporel à partir de son spectre échantillonné :

<u>Δf > 1/T</u> : La résolution spectrale Δf est trop grande. On a un recouvrement dans le domaine temporel. C'est un peu Shannon à l'envers : si on choisit une résolution spectrale trop grande, on ne peut pas reconstituer le signal dans le domaine temporel correctement.

<u>Δf < 1/T</u> : Il n'y aura plus de repliement temporel, mais des intervalles durant lesquels le signal dont on calcule le spectre sera nul.

<u>Δf = 1/T</u> : On a un signal périodique idéal. On périodise la fenêtre temporelle choisie avant le calcul spectral.

En pratique, on choisira donc toujours Δf de telle sorte à avoir Δf = 1/T.

Comme $T = N.T_e$ et $\Delta f = \frac{Fe}{M}$ $\Longrightarrow$ $\frac{Fe}{M} = \frac{1}{N.\mathrm{Te}}$

D'où N=M

Ainsi, la définition de la **transformée de Fourier discrète** devient :

$\mathbf{X_T(k) = \sum_{n=0}^{N-1} X_T[n].e^{-2j\pi\frac{n.k}{N}}}$, Avec k= [0……N-1]

B.5. Notion de transformée de Fourier rapide

Pour obtenir une valeur particulière de $X_T(k)$, il faut varier n, par exemple :

Pour n= 0 :

$X_T(k) = X_T[0].e^{-2j\pi.0}$ = ($X_T[0]$. cos (0) - $X_T[0]$. j sin (0)) ; on a 2 produits complexes et 1 somme complexes

Pour n = 1 :

$X_T(k)$ = ($X_T[0]$. cos (0) - $X_T[0]$. j sin (0)) +[$X_T[1]$. cos ($\frac{2\pi k}{N}$) - $X_T[1]$. j sin ($\frac{2\pi k}{N}$)]

Ici, on a 4 produits complexes et 3 sommes complexes.

Pour n = N-1 :

On a 2N produits complexes et 2(N-1) sommes complexes :

Ainsi, pour obtenir les N valeurs de $X_T[k]$, il faut donc 2N multiplications et 2(N-1)N additions. Alors on arrive très vite à des temps de calcul très longs. Si ces durées ne sont pas gênantes pour des traitements en temps différé, il n'en est pas de même en temps réel. En effet, plus le temps de calcul sera important et plus la fréquence maximale du signal à analyser sera réduite (Théorème de Shannon).

Pour pouvoir utiliser la transformée de Fourier discrète en temps réel, on dispose d'algorithmes de calcul permettant d'obtenir les résultats beaucoup plus rapidement sous certaines conditions. Ces algorithmes sont connus sous le nom de Transformée de Fourier Rapide (TFR) ou Fast Fourier Transform (FFT). L'algorithme le plus connu est celui de **Cooley-Tuckey**.

Présentation à l'algorithme de Cooley-Tuckey

On pose :$X[k] = \sum_{n=0}^{N-1} X[n].W_N^{nk}$ avec $W_N = e^{\frac{-2j\pi}{N}}$

Propriétés de W_N

- $W_N^{2nk} = e^{\frac{-2j\pi nk}{N/2}} = W_{N/2}^{nk}$
- $W_N^{nk+N/2} = e^{\frac{-2j\pi(nk+\frac{N}{2})}{N}} = W_N^{nk}$

La condition d'utilisation est d'avoir un nombre d'échantillons puissance de 2 qi est $N=2^m$.

Si on effectue un dédoublement temporel en séparant les indices pairs et impairs **[6]**:

$$\begin{cases} X_1[n] = X[2n] \\ X_2[n] = X[2n+1] \end{cases}$$

Annexe C : NOTION A LA CRYPTOGRAPHIE

C.1 Définitions

- **Cryptologie**

C'est la science des messages secrets, elle se décompose en deux disciplines qui sont la cryptographie et la cryptanalyse.

- Cryptographie

C'est l'art de transformer un message clair en un message inintelligible pour celui qui ne possède pas les clés de déchiffrement.

En d'autre terme, c'est une science qui utilise les mathématiques pour le cryptage et le décryptage des donnés.

- Cryptanalyse

Cryptanalyse est l'art d'analyser les messages chiffrés afin de le décrypter. La cryptanalyse classique implique combinaison de raisonnement analytique, d'application d'outil mathématique, de recherche de modèle, de patience, de détermination et de chance.

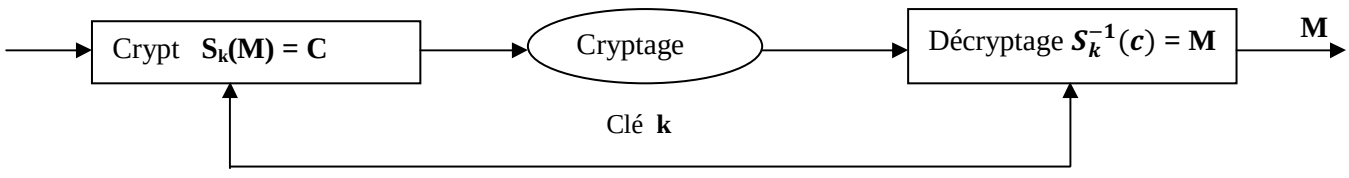
C.2 La cryptographie d'une manière générale

Le message M à crypter (texte clair) est transformé grâce à une fonction paramétrée $\mathbf{S_k(x)}$ par une clé k. A la sortie du processus du cryptage, on obtient un texte chiffré appelé **cryptogramme**.

$$\mathbf{C = S_k(n)}$$

Ce message est transmit à travers le réseau de transmission. A la réception cryptogramme subit le processus de décryptage par la fonction inverse $\boldsymbol{S_k^{-1}(x)}$ par une clé **k** et on obtient le texte clair envoyé

$$\mathbf{M = } \boldsymbol{S_k^{-1}(c)}$$



La cryptanalyse est la science de la reconstruction du texte en clair sans connaitre la clé. Elle peut aussi mettre en évidence les faiblesses d'un cryptosystème. Il existe 4 types génériques d'attaque cryptanalyse.

a. Attaque à texte chiffré

Le cryptanalyse dispose un texte chiffré de plusieurs messages tous ayant été chiffrés avec le même algorithme. Le but est ici de retrouver le texte en clair du plus grand nombre possible ou de retrouver la ou les clés qui ont été utilisées , ce qui permettrait de déchiffrer d'autres messages avec ce même clé.

b. Attaque à texte en clair connu

Le but est ici de retrouver la ou les clés utilisées ou un algorithme qui permet de déchiffrer n'importe quel message chiffré avec le même clé.

c. Attaque à texte en clair choisi

Le cryptanalyste a l'accès au texte chiffré et au texte en clair et en plus il peut choisir les textes en clair à chiffrer (il peut choisir des textes en clair spécifique qui donneront plus d'informations sur la clé).

d. Attaque en texte en clair choisi adaptive

Le cryptanalyste peut choisir les textes en clair et il peut également adapter à ces choix en fonction du texte chiffré précédent. Il choisit un bloc initial plus petit, ensuit, il choisir un bloc en fonction du résultat du premier et ainsi de suite.

C.3 Modèle de communication OSI et cryptographie

Théoriquement, le chiffrement peut être effectué dans n'importe quelle couche OSI de module de communication.

En pratique, le chiffrement a lieu soit dans les basses couches, soit dans les hautes couches :

- S'il s'effectue dans les basses couches (qui sont : la couche application, présentation, session), il est appelé chiffrement lien par lien, dans ce cas, tout ce qui passe par un lien de donné particulier est chiffré ;
- Il est appelé chiffrement bout en bout si le chiffrement s'effectue dans les couches plus élevées (qui sont : la couche transport, réseau, liaison, physique).

a. Chiffrement lien par lien

L'endroit le plus simple pour effectuer le chiffrement se trouve dans la couche physique. La gestion de clé est simple car seul les deux extrémités de lignes doivent partager une clé, et elles peuvent changer leur clé indépendamment du reste du réseau.

b. Chiffrement de bout en bout

Le dispositif chiffre seulement les unités des donnés de transport qui sont recombinées avec les informations de routage non chiffrés et envoyés au couches inferieurs du protocole. Les données restent chiffrées jusqu'à ce qu'elles attiennent leur destination finale. La gestion de clé est plus difficile car les utilisateurs doivent s'assurer qu'ils ont bien les clés communes.

c. Combinaison des deux méthodes

C'est le moyen le plus efficace de rendre sûr un réseau. Le chiffrement des tous liens physiques rend toute analyse des informations de routage impossible tandis que le chiffrement de bout en bout réduit le danger des données non chiffrés dans différents réseaux.
Les gestions des clés peuvent être complètement séparées, les gestionnaires peuvent s'occuper du chiffrement au niveau physique tandis que les utilisateurs peuvent avoir la responsabilité du chiffrement de bout en bout.

C.4 Notion sur le protocole

Un protocole est une série d'étape impliquant de plusieurs participants (correspondants) conçu pour accomplir une tâche (ayant une début et une fin) .Chaque étape doit être exécutée à son tour et aucune autre étape ne peut être exécutée avant que la précédente ne soit achevée.

Les caractéristiques des protocoles sont :

- Chaque participant doit connaitre les protocoles et toutes les étapes à suivre d'avance.
- Chaque participant doit être d'accord pour adhérer au Protocole.
- Le protocole doit être non ambigüe, chaque étape doit être bien définie et elle ne doit pas y avoir de possibilité de mésentente.
- Un protocole doit être complet, il doit y avoir une action spécifique pour chaque situation possible.

L'exécution d'un protocole parcours linéairement ces étapes sauf s'il y a les instructions de branchement indiquant qu'il faut poursuivre par une autre étape.

C.5 Les algorithmes cryptographiques

a. Algorithme à clé unique

La cryptographie moderne utilise une clé k qui peut prendre une valeur parmi un grand nombre de valeurs posible.

$$\begin{cases} \boldsymbol{C = E_k(M)} & \text{pour le chiffrement} \\ \boldsymbol{M = D_k(C)} & \text{pour le dechiffrement} \end{cases}$$

Exemples

- **Méthode par substitution**

Elle consiste à remplacer chaque lettre ou chaque groupe de lettres par une autre lettre ou groupe de lettres en déplaçant l'alphabet de **k** lettres (k est appelé clé à permutation circulaire).

Deux méthodes sont nées à partir de la méthode de substitution :

- Cryptage par substitution monoalphabetique, qui consiste à remplacer chaque lettre du texte en clair par une autre lettre quelconque ou d'autre symbole.
- Cryptage par substitution polyalphabetique, qui consiste à utiliser une matrice carrée contenant les 26 alphabets ; la première ligne contient ABC….XYZ, la deuxième ligne c'est-à-dire la ligne B contient BCD……XYZA, ligne C qui est CDE….XYZAB et ainsi de suite. La clé est un mot ou une phrase. On répète cette clé autant de fois qu'il le faut au dessus du texte en clair. La lettre de la clé qui est au dessus de lettre du texte en clair détermine la ligne de la matrice à utiliser.

- **Méthode par transposition**

Elle consiste à changer l'ordre des caractères et utilise une clé qui est un mot ou une phrase ne contenant aucune lettre répétée. Elle sert à numéroter les colonnes. Le texte en clair est écrit horizontalement tandis que le texte chiffré est lu en colonne en commençant par le colonne du plus petit numéro.

b. Algorithme de cryptage par bloc

Il est fondé sur les principes de confusions et de diffusions de Shannon. La confusion sert à cacher les relations entre le texte en clair, le texte chiffré et la clé. La diffusion répond l'influence d'un bit particulier du texte en clair ou de clé aussi loin que possible dans le texte chiffré (camoufle les relations statistiques).

c. Algorithme de chiffrement produit

Il consiste à mélanger d'une manière répétitive la confusion et la diffusion avec un seul algorithme de chiffrement dans différente combinaison.

d. Réseau de Feistal

C'est un algorithme de chiffrement par bloc itératif où la i-ème ronde est déterminé d'après la sortie de la ronde précédente $(i-1)^{-ème}$.

Pour un bloc n pair divisé en 2, on aura 2 moitiers de longuer n/2.

C.6 Mode opératoire cryptographique

a. ECB (Electronic Code Back)

Chaque bloc de texte en clair est chiffré indépendamment, on n'est pas obligé de chiffrer un fichier linéairement c'est-à-dire on peut d'abord chiffrer les dix blocs du milieu ensuite les blocs de la fin puis ceux de début.

b. CBC (Cipher Bloc Chaining)

Le chainage utilise une méthode de rétroaction car les résultats du chiffrement de bloc précédent sont réutilisés comme entrée par le chiffrement de bloc courant. Chaque bloc chiffré dépend non seulement du bloc du texte qui l'a engendré mais aussi de taux de bloc de texte en clair qui le précède. En mode DCB, le texte en clair est combiné par « ou exclusif » avec le bloc précédent avant d'être chiffré. Après qu'un boc de texte en clair a été chiffré, le texte chiffré correspondant est stocké dans un registre de rétroaction. Avant que le bloc suivant de texte en clair soit chiffré, il est

combiné par « ou exclusif » avec le registre de rétroaction pour devenir la nouvelle entrée de l'algorithme de chiffrement.

$$\begin{cases} C_i = E_k\,(M_i \oplus C_{i-1}) \\ M_i = C_{i-1} \oplus D_k\,(C_i) \end{cases}$$

Où $\mathbf{M_i}$: message en clair

C_i : texte chiffré d'i^{ème} bloc

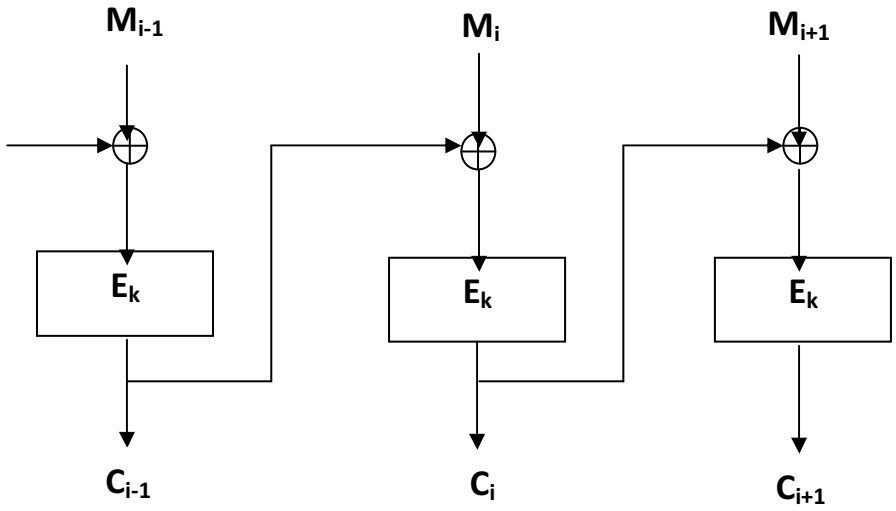


Figure C.1 : **mode de cryptage en CBC**

Et inversement pour le décryptage :

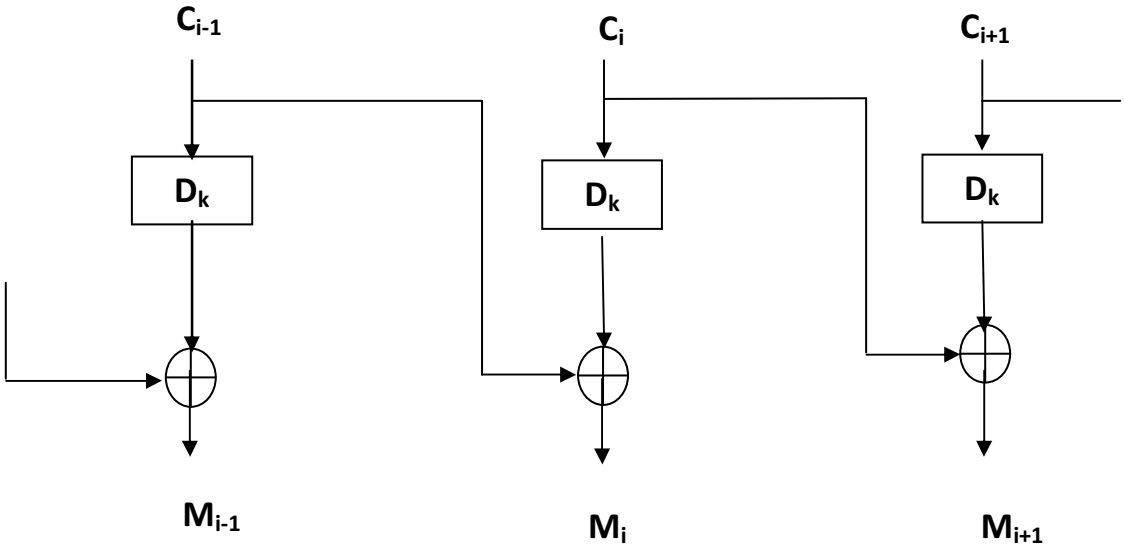


Figure C.2 : **mode de décryptage en CBC**

c. CFB (Cipher Feed Back)

Le mode de CFB manipule à une file d'attente de la taille d'un bloc d'entrée, initialement ,la file d'attente est chiffrée et les 8 bits les plus à gauche du résultat sont combinés à « ou exclusif » avec le 1er caractère de 8 bits de texte en clair pour devenir les 8 1er bits du texte chiffré. Les 8 bits sont placés dans les 8 bits le plus à droit de la file d'attente et tout les autres 8 bits sont décalés de 8 positions vers la gauche. Les 8 bits le plus à gauche sont ignorés, le caractère suivant est alors chiffré de la même façon (le déchiffrement est le processus inverse).

d. OFB (Output Feed Back)

Le mode CFB est une mode similaire à CFB sauf que n bits du bloc de sortie précédente sont mis dans la position la plus à droite de la file d'attente.

$$\begin{cases} \boldsymbol{C_i = M_i \oplus S_i} \\ \boldsymbol{M_i = C_i \oplus S_i} \end{cases}$$

$S_i = E_k(\boldsymbol{S_{i-1}})$: état qui est indépendant du texte chiffré et du texte clair [14].

REFERENCES

[1] http://www.laboratoiredelavoix.com/wp-content/uploads/lavoix.pdf

[2] RAZAFIARISON Zo Manankasina, Livre de mémoire de fin d'étude intitulé : « Réalisation d'une suite complète d'acquisition, de génération et de traitement du son », novembre 2009.

[3] http://tcts.fpms.ac.be/cours/1005-08/speech

[4] Vu Minh Quang, Thèse « Exploitation de la Prosodie pour la Segmentation et l'Analyse Automatique de Signaux de Parole », soutenue le 20 septembre 2007.

[5] http://scgwww.epfl.ch/courses/Traitement_de_la_parole-2005-2006-pdf/Drygajlo-Reconnaissance-du-locuteur.pdf

[6] http://www.geea.org/IMG/pdf/Cours_TS.pdf

[7] Cours Logique combinatoire, 2ème Année, Département Electronique, ESPA 2007.

[8] E531 Cours Traitement Numérique du signal, 5ème Année, Département Electronique, ESPA 2010

[9] http://en.wikipedia.org/wiki/Cooley-Tukey_FFT_algorithm

[10] Mohamed CHETOUANI, Thèse de DOCTORAT « Codage neuro-prédictif pour l'extraction de caractéristiques de signaux de parole »

[11] Anthony LARCHER, Thèse « Modèles acoustiques à structure temporelle renforcée pour la vérification du locuteur »

[12] http://w3.u-grenoble3.fr/idl/IMG/protege/form17/ConfBONASTRE.pdf

Nicolas Scheffer Thèse, « Structuration de l'espace acoustique par le modèle générique pour la vérification du locuteur »

[13] http://www.developpeez.com

[14] EIA 558 Cours Cryptographie, 5ème Année, Département Electronique, ESPA 2010

Auteur : RAMBININTSOARIMASY Haingo Tiana

Titre : IDENTIFICATION PAR BIOMETRIE VOCALE

Nombre de pages : 60

Nombres de figures : 37

Nombre de tableaux : 02

RESUME

A l'heure actuelle, la technologie biométrique ne cesse de s'évoluer, en particulier la biométrie vocale. Cette dernière devenant le plus utilisé et répandu dans le monde en vue de sa simplicité.

Le principal but de ce présent rapport est focalisé sur le traitement du signal vocal. Les deux applications qui sont « **RecSpeaker** » permettant de reconnaitre vocalement l'identité d'un locuteur, et « **Application** » qui est uniquement pour but démonstratif du RecSpeaker. Elles sont écrites en langage C et programmées sur l'environnement de Visual C++ de Microsoft.

Mots clés : son, voix, timbre, phonème, Prosodie, ton, Accent, wave, reconnaissance, locuteur

Rapporteur : Monsieur ANDRIAMANANTSOA Guy Danielson