TABLE DES MATIÈRES

Page

LISTE DES TABLEAUX

Page

LISTE DES FIGURES

Page

LISTE DES ABRÉVIATIONS ET SIGLES

LPC	Linear Predictive Coding
MFCC	Mel Frequency Cepstral Coefficients
FFT	Transformée de Fourier rapide
DCT	Transformée discrète en cosinus
DTW	Dynamic Time Warping
HMM	Hidden Markov Model
TF	Transformée de Fourier
STFT	Transformée de Fourier à fenêtre glissante
TO	Transformée en ondelettes
CWT	Transformée en ondelettes continue
DWT	Transformée en ondelette discrète
AWP	Paquets d'ondelettes admissibles
DSP	Digital Signal Processor
CCS	Code Composer Studio
AWGN	Bruit blanc gaussien ajouté
ψ	Ondelette mère
$\psi_{a,b}$	Base d'ondellettes
a	Facteur d'échelle (dilatation)
b	Paramètre de translation

INTRODUCTION

Dans la dernière décennie, avec l'avènement des nouvelles technologies, l'industrie des télécommunications a connue un progrès considérable, notamment dans le domaine de la reconnaissance automatique de la parole qui fait l'objet de ce mémoire. Ce domaine trouve ses applications dans la téléphonie, dans la commande, dans la reconnaissance du locuteur...etc. La reconnaissance des chiffres isolés est l'un des axes de la reconnaissance de la parole. Cependant, son problème majeur est de réaliser des systèmes de reconnaissances rapides, non complexes et capables d'être implémentés dans des processeurs limités par la mémoire et par la rapidité comme le DSP [41-43]. Donc, notre objectif est de proposer une méthode de reconnaissance de chiffres isolés qui permet de remédier à ce problème.

Avant l'extraction des paramètres de la reconnaissance, le chiffre isolé est subdivisé en segments de durées fixes sur lesquels on pourra supposer que le signal est stationnaire. Vu que durant la prononciation, la durée du chiffre varie selon la vitesse d'élocution, la subdivision donnera des nombres de segments différents d'un chiffre à un autre. Pour remédier à ce problème, on utilise l'algorithme d'alignement dynamique (DTW) [9] durant l'étape de la reconnaissance. Toutefois, l'utilisation de cette méthode est complexe et affecte la rapidité du système de reconnaissance de chiffres isolés.

Les méthodes d'extractions de paramètres les plus connues sont: la méthode de codage linéaire prédictif (LPC) basée sur le système de production de la parole auto régressif et la méthode MFCC (Mel Frequency Cepstral Coefficient) où les paramètres sont extraits à partir des filtres triangulaires réparties selon l'échelle de Mel [7]. La venue de la théorie des ondelettes a ouvert un nouveau champ dans le domaine de la reconnaissance de la parole. Elle est utilisée dans l'étape d'extraction de paramètres avec efficacité. Les coefficients sont obtenus à partir des sous bandes de la décomposition en paquets

d'ondelettes ou dyadique. Par rapport à la transformée de Fourier, la transformée en ondelettes a l'avantage d'utiliser des fenêtres d'analyses de tailles variables pouvant détecter les signaux de hautes et de basses fréquences. En plus, elle a l'avantage d'utiliser un algorithme pyramidal rapide pour la décomposition.

Dans ce mémoire, nous présenterons une méthode de reconnaissance des chiffres isolés dont l'originalité consiste en l'application d'une méthode de segmentation du signal de la parole en un nombre fixe de segments dont la taille diffère d'un chiffre à un autre ce qui permet d'éviter l'utilisation de l'algorithme complexe DTW. Pour l'extraction des paramètres, nous avons utilisé la méthode de décomposition en paquets d'ondelettes admissibles espacées selon l'échelle de Mel. L'algorithme de classification Fuzzy C-Means sera ensuite utilisé pour la réalisation de notre dictionnaire de référence. Dans l'étape de la reconnaissance, la distance euclidienne sera utilisée. Le débruitage avec les ondelettes sera aussi ajouté au début du système afin d'effectuer une reconnaissance robuste de chiffres isolés affectés par un bruit. La méthode proposée a été testée à l'aide du logiciel de simulation MATLAB® en utilisant la base de données TIDIGITS de Texas Instrument avant d'être implémentée sur le DSP TMS320C6711.

Dans le premier chapitre, nous décrirons brièvement le système de production et de perception de la parole. Nous présenterons également les différentes étapes classiques utilisées pour la réalisation d'un système de reconnaissances de chiffres isolés et de quelques méthodes utilisées.

Dans le deuxième chapitre, nous présenterons la transformée en ondelettes et de son l'analyse multirésolution. Une méthode de reconnaissance de la parole qui utilise la décomposition en paquets d'ondelettes similaire à l'échelle de Mel pour l'extraction de paramètres et d'une méthode de débruitage utilisant aussi les ondelettes seront également présentées.

Dans le troisième chapitre, nous décrirons le processeur dédié au traitement des signaux (DSP) TMS320C6711 de Texas Instrument et de la carte DSK6711 qui contient le DSP. Nous présenterons aussi le logiciel Code Composer Studio qui permet de développer le programme.

Le dernier chapitre, consiste en la présentation de l'approche choisie pour la reconnaissance de la parole où les différents blocs qui le constituent seront développés. Nous présenterons ensuite les différents résultats de simulation obtenus à l'aide du logiciel MATLAB®. Nous terminerons par les résultats de l'implémentation sur le DSP TMS320C6711.

CHAPITRE 1

RECONNAISSANCE DE LA PAROLE

1.1 Introduction

Dans ce chapitre, nous décrivons les systèmes de production et de perception de la parole. Nous abordons ensuite les étapes nécessaires pour la reconnaissance des mots isolés. Après le prétraitement, l'étape suivante consiste à extraire les paramètres. Elle est suivie par l'étape de reconnaissance où nous présenterons les méthodes nécessaires à sa mise en œuvre. Nous terminerons par un exemple d'algorithme de classification qui est utilisé pour créer un dictionnaire de référence dans le cas d'une reconnaissance multilocuteur.

1.2 Production de la parole

La production de la parole est due à l'action des systèmes respiratoire et masticatoire sous l'effet du contrôle du système nerveux central [1]. Pour produire de la parole, le larynx, constitué de cartilages et de muscles, reçoit une quantité d'air des poumons. Ensuite, avec l'ouverture et la fermeture du larynx à l'aide des cordes vocales, il fait varier le flux d'air avant d'être envoyé à la région vocale qui est constituée d'une cavité buccale et nasale. Le système de production de la parole est illustré par la figure 1.

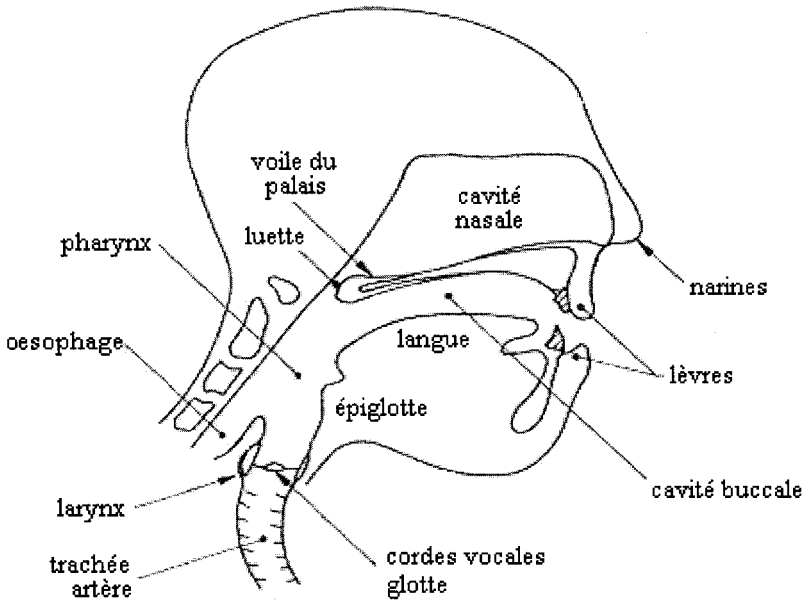

Figure 1 Appareil phonatoire [1]

Le processus de production de la voix consiste en deux modes : celui qui donne des sons voisés, par la pression de l'air qui fait vibrer les cordes vocales du larynx et celui qui donne des sons non voisés, ceci est dû à un flux d'air turbulent dans le conduit vocal [2]. Une vue du larynx est donnée par la figure 2:

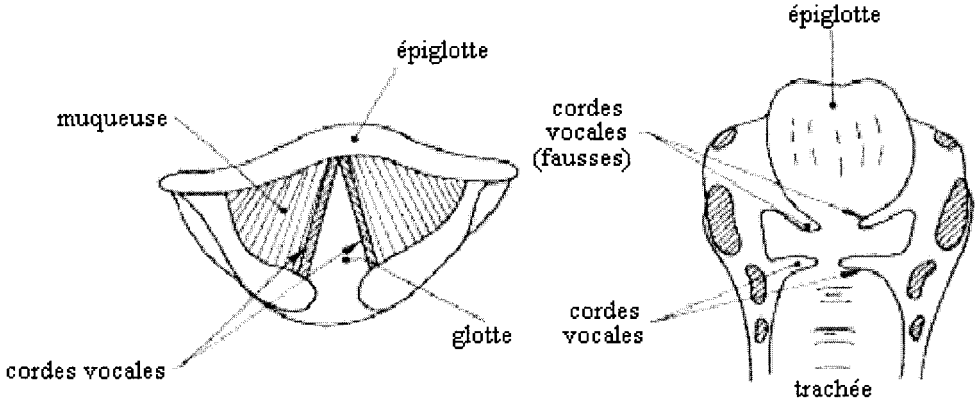

Figure 2 Vue du larynx [1]

1.3 Perception de la parole

La perception de la parole est effectuée par l'appareil auditif (oreille) qui est constitué de l'oreille externe, l'oreille moyenne et l'oreille interne. La perception de l'appareil auditif humain a une bande de fréquences qui s'étend entre 800 Hz et 8 KHz et au maximum entre 20 Hz et 20 KHZ [1]. La figure 3 illustre l'appareil auditif :

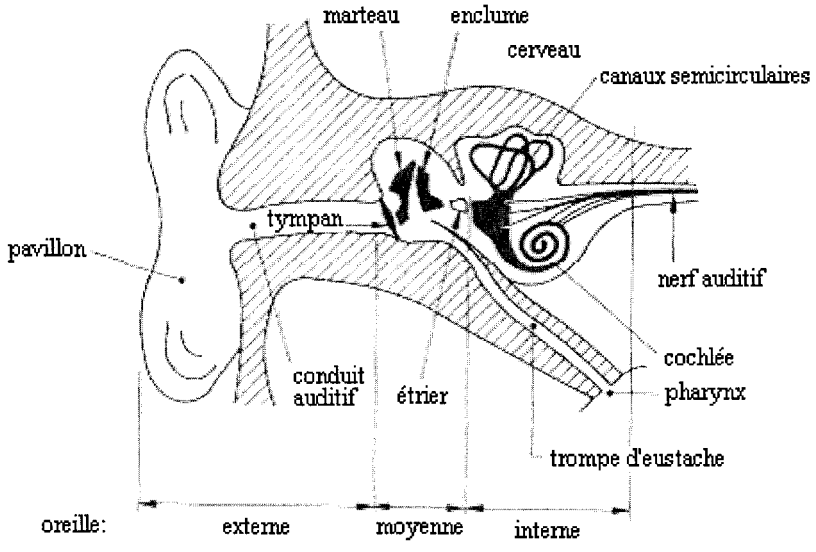

Figure 3 Appareil auditif humain [1]

1.3.1 Oreille externe

Le pavillon qui est la grande partie de l'oreille externe, protège l'oreille contre les corps étrangers et permet aussi une localisation du son qui est transmis au tympan à travers le conduit auditif. Ce dernier est un tube acoustique qui a sa première fréquence de résonance autour de 3 kHz et par conséquence, la sensibilité de l'appareil auditif est élevée dans cette gamme de fréquences [1].

1.3.2 Oreille moyenne

L'oreille moyenne est une cavité d'air qui est constituée du tympan et des osselets (le marteau, l'enclume et l'étrier). Ces derniers ont pour rôle de transmettre les vibrations reçus par le tympan au milieu liquide de l'oreille interne. L'oreille moyenne permet aussi de protéger l'oreille interne des sons très forts [3]. La trompe d'eustache, qui est reliée à la gorge, a pour rôle de régler la pression d'air des deux faces du tympan.

1.3.3 Oreille interne

L'oreille interne est formée d'un milieu liquide. Elle contient la cochlée qui comprend la membrane basilaire. Quand cette dernière reçoit des vibrations, les cellules ciliées, des milliers de cellules, de l'organe de corti situé sur la membrane basilaire déclenchent des influx nerveux au nerf auditif [4]. Une coupe transversale de la cochlée est donnée par la figure 4:

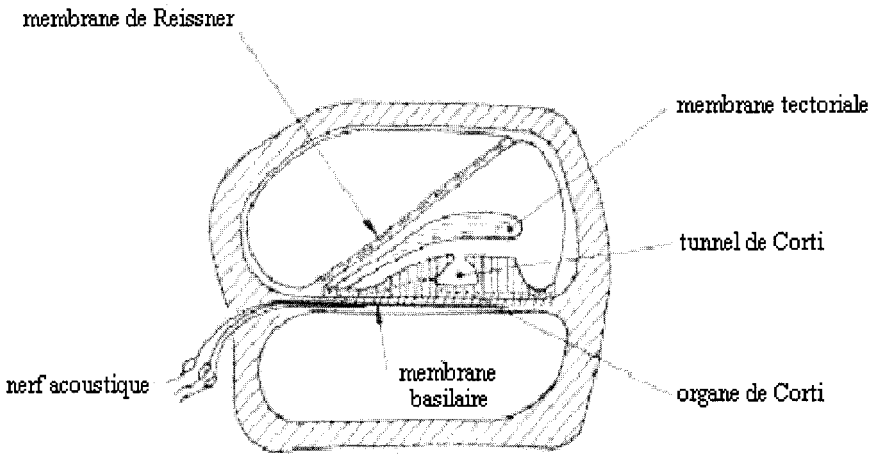

Figure 4 Coupe transversale de la cochlée [1]

1.4 Reconnaissance de mots isolés

Le principe de base d'un système de reconnaissance de mots isolés est de donner une image acoustique à chacun des mots à reconnaître [4]. Il existe deux méthodes de reconnaissance de mots isolés. La première méthode dite globale consiste à comparer le mot à reconnaître en entier avec le mot de référence. Par contre, dans la deuxième méthode dite analytique le mot à reconnaître est subdivisé en composantes élémentaires (phonèmes ou syllabes etc.) qui sont comparées avec les composantes élémentaires de la référence. Dans notre cas, nous utiliserons la première méthode. Les différentes étapes de la reconnaissance des mots isolés de la méthode globale sont illustrées par l'organigramme donné par la figure suivante :

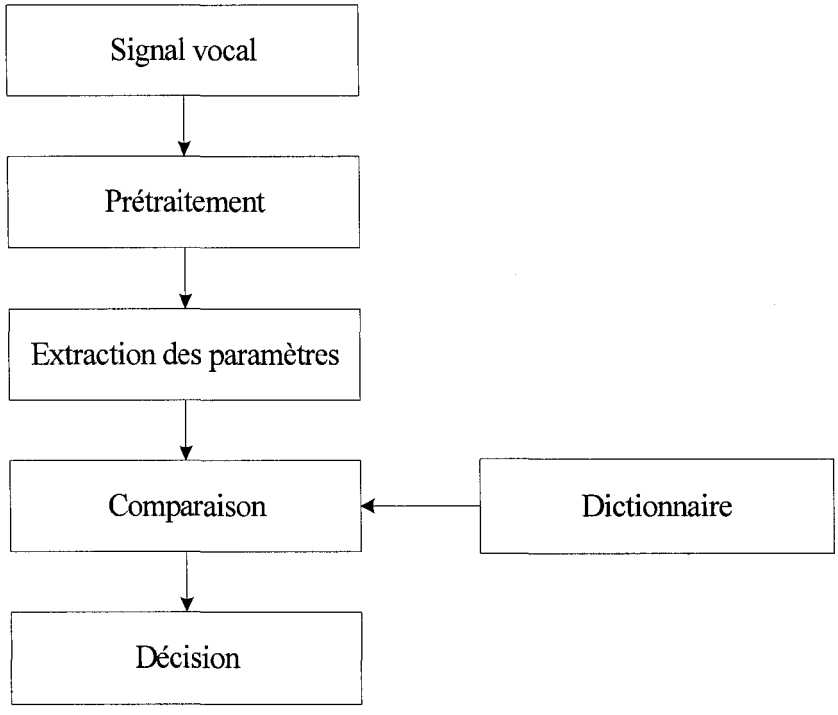

Figure 5 Reconnaissance des mots isolés

Les étapes de l'organigramme sont définies comme suit:

- **Prétraitement :** après avoir séparé le mot du silence, on effectue la préaccentuation et on divise le mot en différents segments.

- **Extraction de paramètres :** on extrait les paramètres pour chaque segment.

- **Dictionnaire :** on crée des modèles de références pour chaque mot.

- **Comparaison et décision:** les paramètres du mot à reconnaître sont comparés avec ceux des modèles du dictionnaire. Le modèle qui a les paramètres proches, va être choisi comme mot reconnu.

1.4.1 Détection du mot

Durant l'enregistrement, un moment de silence est présent au début et à la fin du mot. Pour réaliser une reconnaissance robuste du mot, il faut extraire le mot du silence. Une cause majeure de mauvaise reconnaissance automatique de la parole est la mauvaise détection des bornes du mot. La figure suivante illustre le signal correspondant au chiffre "one" enregistré entre deux périodes de silence :

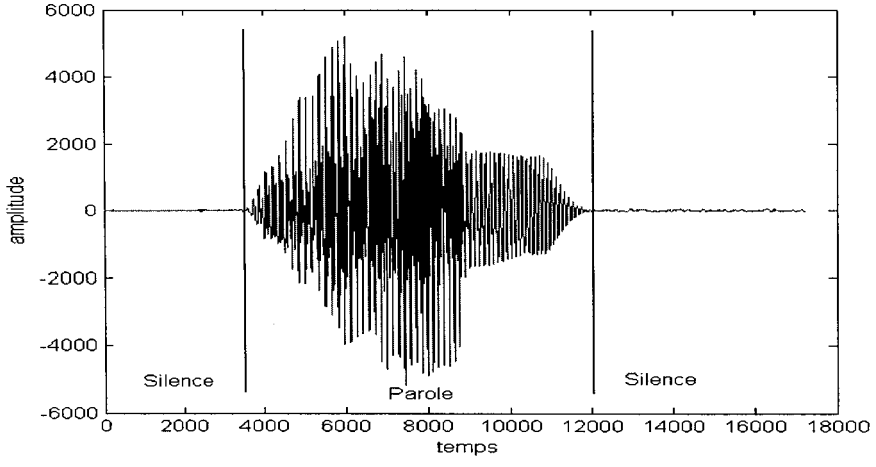

Figure 6 Chiffre "one" enregistré avec le silence

1.4.2 Préaccentuation

La préaccentuation est une opération de filtrage d'un signal de parole $s(n)$ avec un filtre dont la fonction de transfert $H(z)$ est donnée par:

$$H(z) = 1 - \mu z^{-1} \tag{1.1}$$

La valeur la plus utilisée pour le paramètre μ est 0.95.

Si $s_p(n)$ est le signal de la parole préaccentué alors:

$$s_p(n) = s(n) - 0.95\, s(n-1) \tag{1.2}$$

Cette opération permet d'accentuer les hautes fréquences du signal.

1.4.3 Segmentation et fenêtrage

Le signal de la parole est de nature non stationnaire. Il est donc nécessaire, avant d'extraire les paramètres de la reconnaissance, de le subdiviser en segments. Cette étape permet d'obtenir pour chaque segment de parole un signal quasiment stationnaire [5]. La figure suivante 7 illustre l'opération de segmentation d'un chiffre avec recouvrement.

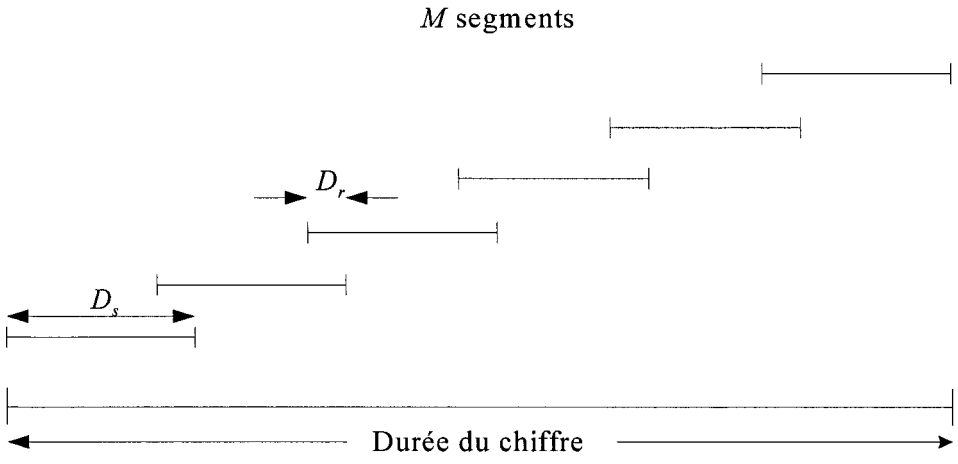

Figure 7 Segmentation en M segments avec recouvrement

Dans la figure 7, D_s est la durée du segment et D_r est la durée du recouvrement.

Les discontinuités aux extrémités des segments peuvent être amoindries en multipliant chaque segment par une fenêtre de Hamming [6]. La fenêtre de Hamming est donnée par l'équation suivante :

$$w(n) = \begin{cases} 0.54 - 0.46\cos(\frac{2\pi n}{N-1}), & 0 \le n \le N-1 \\ 0, & ailleurs \end{cases} \qquad (1.3)$$

où N est le nombre d'échantillons du segment.

1.4.4 Extractions de paramètres

Dans cette étape, nous survolerons les deux méthodes les plus utilisées d'extraction des paramètres. La première est basée sur le principe de production de la parole et la deuxième sur le principe de perception de la parole.

1.4.4.1 Codage linéaire prédictif

Le codage linéaire prédictif (Linear Predictive Coding LPC) [5] est une méthode d'extraction des paramètres basée sur le principe qu'un échantillon du signal peut être estimé à l'aide d'une composition linéaire de p échantillons passés. Elle est fondée sur un modèle auto régressif. Ce modèle est représenté par la figure 8:

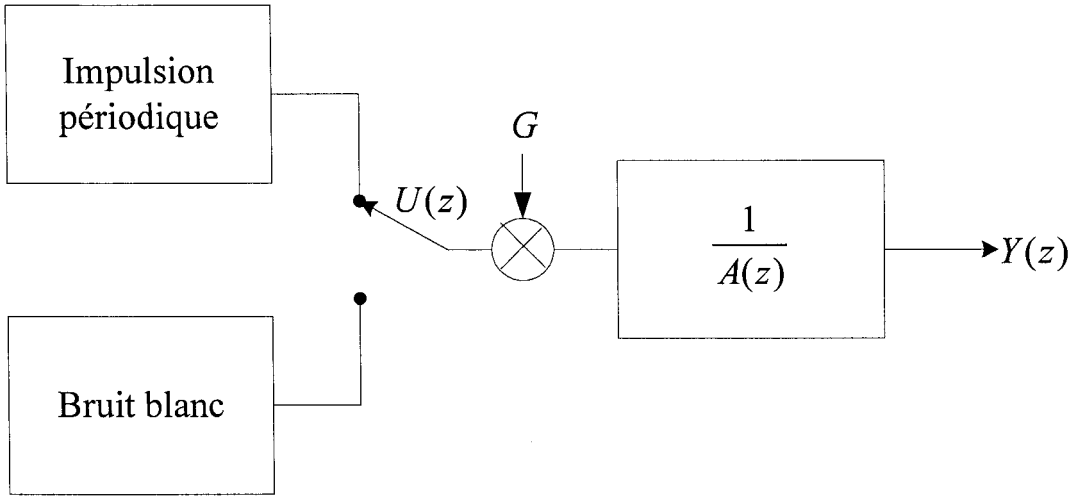

Figure 8 Modèle auto régressif de production de la parole

Comme le conduit vocal peut être assimilé à un filtre récursif, à pôles seulement défini par:

$$Y(z) = \frac{G}{A(z)} U(z) \qquad (1.4.a)$$

$$\text{avec} \quad A(z) = 1 - \sum_{i=1}^{p} a_i z^{-i} \qquad (1.4.b)$$

où, G est le gain, $\frac{G}{A(z)}$ est la fonction de transfert du filtre, les paramètres a_i sont les coefficients de prédiction, $Y(z)$ est le signal de sortie du filtre et $U(z)$ représente le signal d'excitation, qui est formé d'impulsions périodiques pour le son voisé et de bruit blanc pour le son non voisé. L'équation (1.4) donne:

$$G\,U(z) = Y(z) - Y(z)\sum_{i=1}^{p} a_i z^{-i} \tag{1.5}$$

Dans le domaine temporel, l'équation (1.5) devient :

$$G\,u(n) = y(n) - \sum_{i=1}^{p} a_i\, y(n-i) = y(n) - \hat{y}(n) \tag{1.6}$$

où $\hat{y}$ est le signal estimé à partir de la composition linéaire des p échantillons passés.

L'une des méthodes d'estimation des coefficients a_i est la méthode d'autocorrélation [5] où les paramètres sont estimés à partir des segments de la parole prétendus de durée limitée de N échantillons. On définit alors le signal de la parole segmenté $y_n(m)$ à l'instant n:

$$y_n(m) = \begin{cases} y(m+n)\,w(m), & 0 \le m \le N-1 \\ 0, & \text{ailleurs.} \end{cases} \tag{1.7}$$

Pour l'estimation des paramètres a_i, on doit minimiser l'erreur quadratique suivante :

$$E_n = \sum_{m=0}^{N-1+p} \left[y_n(m) - \hat{y}_n(m) \right]^2 \tag{1.8}$$

Pour que E_n soit minimale, il faut que :

$$\frac{\partial E_n}{\partial a_k} = 0, \quad 1 \le k \le p \tag{1.9}$$

On obtient l'équation (1.10) à résoudre :

$$\sum_{k=1}^{p} a_k \sum_{m=0}^{N-1+p} y_n(m-i)\, y_n(m-k) = \sum_{m=0}^{N-1+p} y_n(m-i)\, y_n(m) \tag{1.10}$$
$$avec \quad 1 \le i \le p$$

L'équation (1.10) peut être exprimée à l'aide de la fonction d'autocorréaltion suivante :

$$r_n(i-k) = \sum_{m=0}^{N-(i-k)-1} y_n(m)\, y_n(m+i-k) \tag{1.11}$$
$$\text{avec} \quad 1 \le i \le p, 0 \le k \le p$$

Sous la forme matricielle de l'équation (1.10):

$$\begin{bmatrix} r_n(0) & r_n(1) & r_n(2) & \dots & r_n(p-1) \\ r_n(1) & r_n(2) & r_n(3) & \dots & r_n(p-2) \\ r_n(2) & r_n(1) & r_n(0) & \dots & r_n(p-3) \\ . & . & . & . & . \\ r_n(p-1) & r_n(p-2) & r_n(p-3) & \dots & r_n(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ . \\ a_p \end{bmatrix} = \begin{bmatrix} r_n(1) \\ r_n(2) \\ r_n(3) \\ . \\ r_n(p) \end{bmatrix} \tag{1.12}$$

sachant que $r_n(k) = r_n(-k)$.

La matrice d'autocorrélation est une matrice de Toelpitz. Cette propriété permet de résoudre l'équation à l'aide de l'algorithme Levinson-Durbin comme suit [5] :

Initialisation : $E^{(0)} = r(0)$

Boucle:
$$k_i = \frac{r(i) - \sum_{j=1}^{i-1} \alpha_j^{(i-1)} r(|i-j|)}{E^{i-1}} \quad , 1 \le i \le p$$
$$\alpha_i^{(i)} = k_i$$
$$\alpha_j^{(i)} = \alpha_j^{(i-1)} - k_i \alpha_{i-j}^{(i-1)}$$
$$E^{(i)} = (1 - k_i^2) E^{(i-1)}$$

Coefficients : $a_j = \alpha_j^p \ , \quad 1 \le j \le p$

Il existe une autre méthode d'estimation des paramètres de prédiction qui est appelé la méthode de covariance. Les coefficients sont obtenus par la résolution d'un système d'équations matricielles en utilisant la décomposition de Cholesky [1].

1.4.4.2 MFCC

La méthode MFCC (Mel Frequency Cepstral Coefficients) [7] est une méthode d'extraction des paramètres selon l'échelle de Mel. En effet, la perception de la parole par le système auditif humain est fondée sur une échelle fréquentielle semblable à l'échelle de Mel [8]. Cette échelle est linéaire aux basses fréquences et logarithmique en hautes fréquences et elle est donnée par l'équation suivante:

$$Mel(f) = 2595 \log_{10}\left(1 + \frac{f}{700}\right) \tag{1.13}$$

Après une préaccentuation et une subdivision en différents segments avec fenêtrage du signal de la parole, on applique la méthode MFCC qui consiste à calculer la transformée de Fourier de chaque segment. Puis à utiliser des filtres triangulaires, espacés suivant

l'échelle de Mel, pour filtrer cette transformée et obtenir les énergies à partir du module au carré de la transformée de Fourier. Les filtres triangulaires sont illustrés par la figure 9 :

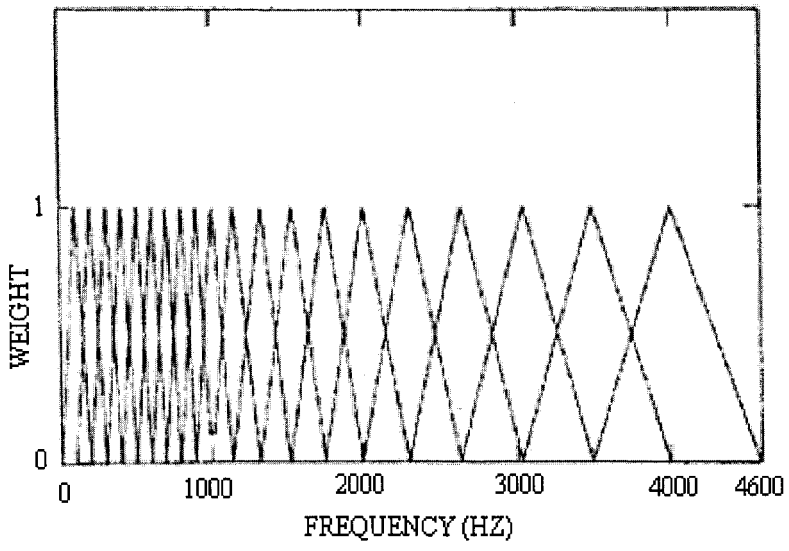

Figure 9 Filtres triangulaires espacés suivant l'échelle de Mel [7]

Finalement, on calcule la transformée discrète en cosinus (DCT) des logarithmes des énergies obtenues par les filtres triangulaires afin d'extraire les coefficients MFCC utilisés pour la reconnaissance. Ces coefficients sont donnés par l'équation suivante [7]:

$$MFCC(i) = \sum_{k=1}^{D} G_k \cos(i(k - \frac{1}{2})\frac{\pi}{D}), \quad i = 1, 2, ..., L \tag{1.14}$$

où G_k est le logarithme de l'énergie obtenue avec le filtre triangulaire k, D est le nombre de filtres triangulaires et L nombre de coefficients MFCC.

L'organigramme de la figure 10, illustre le principe d'extraction de paramètres MFCC :

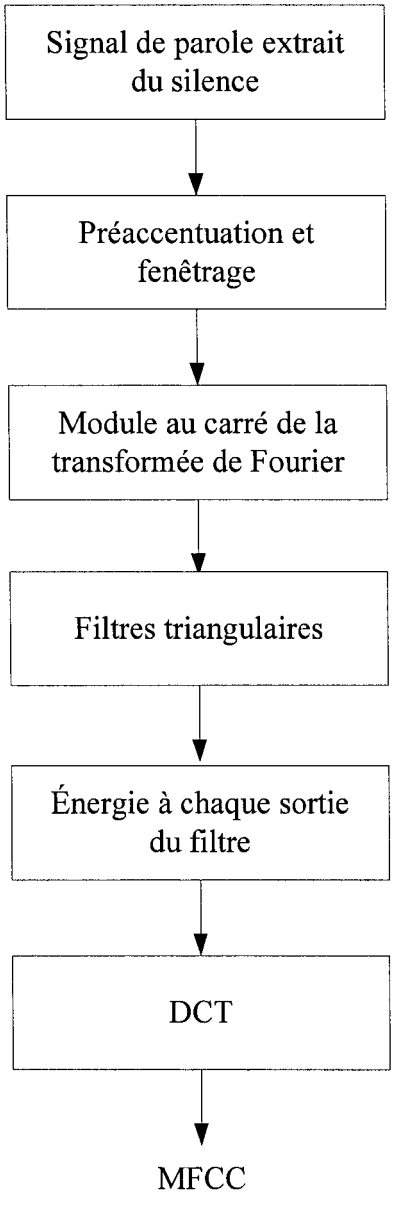

Figure 10 Extraction de paramètres MFCC

1.4.5 Quelques méthodes de reconnaissance des mots isolés

Après l'extraction des paramètres, on passe à l'étape de la reconnaissance. Dans cette partie nous présenterons les deux méthodes de reconnaissance les plus utilisées. La première est basée sur l'algorithme d'alignement temporel dynamique et la deuxième sur un modèle stochastique.

1.4.5.1 Distance

Durant l'étape de la reconnaissance du mot, une distance est utilisée pour mesurer la ressemblance entre le mot à reconnaître et le mot du dictionnaire de référence. Cette distance doit être [1] :

- " Significative sur le plan acoustique.
- Formalisable d'une façon efficace sur le plan mathématique.
- Définie dans un espace de paramètres judicieusement choisi. "

Pour le cas de la mesure spectrale, la distance la plus utilisée est [4]:

$$d_n(x,y) = \left(\sum_{k=1}^{p} |x_k - y_k|^n \right)^{\frac{1}{n}} \tag{1.15}$$

où x et y sont deux vecteurs ayant chacun p composants. Les distances souvent utilisées sont d_1 et la distance euclidienne d_2.

La distance cepstrale euclidienne, souvent utilisée, est aussi utilisée avec pondération [3]:

$$d_{cep} = [\sum_{n=1}^{L} (w(n)(c_y(n) - c_x(n)))^2]^{1/2} \tag{1.16}$$

où $w(n)$ est la fonction de pondération.

Il existe plusieurs types de fonctions de pondérations. Par exemple on a:

- Pondération par indexe n.

$$w(n) = n \tag{1.17}$$

- Pondération par filtrage

$$w(n) = \begin{cases} 1 + h\sin(\frac{n\pi}{L}), & 1 \le n \le L \\ 0, & ailleurs \end{cases} \tag{1.18}$$

La distance d'Itakura est fréquemment utilisée pour les coefficients de prédiction linéaire. Elle est donnée par l'équation suivante [4] :

$$d_I(x, y) = \log(\frac{xRx^T}{yRy^T}) \tag{1.19}$$

où x et y sont les vecteurs des coefficients de prédiction linéaire, R est la matrice d'autoccorélation des coefficients du vecteur y. x^T et y^T sont respectivement les vecteurs transposés des vecteurs x et y.

1.4.5.2 Alignement temporel dynamique

Le même mot peut être prononcé avec différents rythmes et différentes vitesses, ceci entraîne des modifications de l'échelle temporelle. On distingue deux types de ce genre de modifications [4] :

- Les modifications de la vitesse de prononciation donnent une transformation linéaire de l'échelle temporelle.

- Les modifications du rythme de prononciation qui entraîne une transformation non linéaire de l'échelle temporelle.

Pour la reconnaissance des mots isolés, l'évaluation de la mesure de ressemblance entre le mot à reconnaître et le mot de référence qui ont des échelles temporelles différentes est un problème. Pour remédier à ce problème, on utilise un algorithme d'alignement temporel dynamique (DTW Dynamic Time Warping). Le DTW est un algorithme non linéaire qui consiste à dilater ou compresser les axes de temps des mots à comparer.

Soit à comparer le mot T avec le mot de référence R. Ils sont tous deux représentés par des vecteurs de nombre différent à cause de la déformation de l'échelle temporelle. Chaque vecteur représente les coefficients d'un segment du mot extraits pour la reconnaissance, on a donc :

$$\begin{aligned} T &= T_1, T_2, ..., T_i, ...T_I \\ R &= R_1, R_2, ..., R_j, ...R_J \end{aligned} \tag{1.20}$$

L'algorithme de programmation dynamique procède en associant un vecteur du mot à tester avec le vecteur du mot de référence afin de trouver un chemin optimal qui

minimisera l'écart global entre les deux mots. L'algorithme DTW est illustré par la figure 11 suivante:

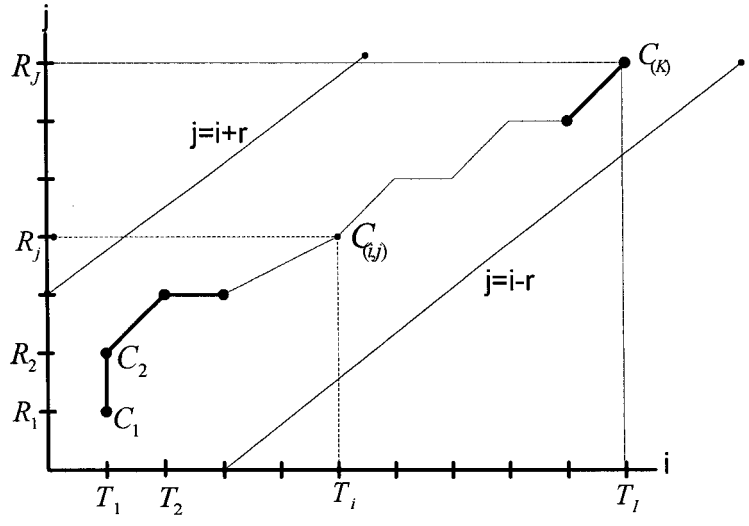

Figure 11 Algorithme d'alignement temporel dynamique [9]

Dans cette figure, C_k, $1<k<K$, représente les associations des vecteurs du mot à reconnaître $T_{i(k)}$ avec les vecteurs du mot de référence $R_{j(k)}$.

La mesure de différence entre deux vecteurs T_i *et* R_j est notée comme une distance locale $d(.)$. Soit $D(.)$ une distance globale qui normalisera les échelles temporelles des mots T *et* R. Si $w(k)$ est un coefficient de pondération, on a :

$$D(T,R) = \frac{\sum_{k=1}^{K} d(T_{i(k)}, R_{j(k)}).w(k)}{\sum_{k=1}^{K} w(k)} \qquad (1.21)$$

Le chemin optimal entre les vecteurs des deux mots T *et* R est la distance minimale globale $D(T,R)$. Pour l'obtenir l'algorithme DTW doit satisfaire les contraintes suivantes [9]:

- Contrainte monotone :

$$i(k-1) \leq i(k) \; et \; j(k-1) \leq j(k) \tag{1.22}$$

.

- Contrainte de continuité qui assure un déplacement à travers le chemin optimal:

$$C(k-1) = \begin{cases} C(i(k), j(k)-1) \\ ou \; C(i(k)-1, j(k)-1) \\ ou \; C(i(k)-1, j(k)) \end{cases} \tag{1.23}$$

- Contrainte de limite qui assure que le début du chemin soit l'association du début des deux mots et que la fin du chemin soit l'association de la fin des deux mots.

$$\begin{aligned} C(1) &= C(i(1)=1, j(1)=1) \\ C(K) &= C(i(K)=I, j(K)=J) \end{aligned} \tag{1.24}$$

- Contrainte de la fenêtre ajustée qui assure à limiter la dilation ou la compression de l'échelle temporelle. Pour r positif on a :

$$|i(k) - j(k)| \leq r \tag{1.25}$$

À toutes ces contraintes s'ajoute la contrainte locale. Exemple de contraintes locales [9] :

- Contrainte de type symétrique :

$$g(i,j)=\min\begin{cases} g(i-1,j-2)+2d(i.j-1)+d(i,j) \\ g(i-1,j-1)+2d(i,j) \\ g(i-2,j-1)+2d(i-1,j)+d(i,j) \end{cases} \tag{1.26}$$

- Contrainte de type asymétrique :

$$g(i,j)=\min\begin{cases} g(i-1,j-2)+[d(i.j-1)+d(i,j)]/2 \\ g(i-1,j-1)+d(i,j) \\ g(i-2,j-1)+d(i-1,j)+d(i,j) \end{cases} \tag{1.27}$$

Avec

$$D(T,R)=\frac{g(I,J)}{N} \tag{1.28}$$

g est la distance cumulée et $N=\sum_{k=1}^{K} w(k)$ est le coefficient de normalisation, il est égale à I+J pour le contrainte symétrique et égale à I ou J pour les contraintes asymétrique.

1.4.5.3 Modèle de Markov cachés

Les modèles de Markov cachés (Hidden Markov Model HMM) sont des approches stochastiques qui utilisent la probabilité à la place de la distance où le signal de la parole est représenté par une séquence d'états d'observations. Le principe de reconnaissance d'un mot avec HMM consiste à trouver un modèle qui reconstitue le mot avec une grande probabilité [10].

Soit $\lambda=(A,B,\pi)$ un modèle de Markov caché. IL est déterminé par [11]:

- Un ensemble $S = \{s_1, s_2, ...s_{N-1}, s_N\}$ de N états où un état est défini par $q_t \in S$ à l'instant t.

- Un ensemble $V = \{v_1, v_2, ...v_M\}$ qui contient M symboles d'observations. L'observation d'un symbole est notée $o_t \in V$ à l'instant t.

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

$$\begin{aligned} &a_{ij} = P(q_{t+1} = s_j / q_t = s_i) \\ &\text{avec } 1 \le i, j \le N \end{aligned} \tag{1.29}$$

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

$$\begin{aligned} &b_j(k) = P(O_t = v_k / q_t = s_j) \\ &\text{avec } 1 \le j \le N, 1 \le k \le M \end{aligned} \tag{1.30}$$

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

$$\begin{aligned} &\pi_i = P(q_1 = s_i) \\ &\text{avec } 1 \le i \le N \end{aligned} \tag{1.31}$$

Un exemple d'un modèle de Markov caché à 5 états est représenté par la figure 12:

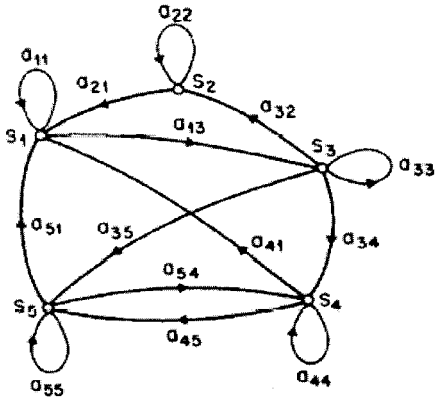

Figure 12 Modèle de Markov caché à 5 états [5]

Pour utiliser la méthode des HMM, il faut résoudre les problèmes suivants [11]:

- On doit calculer la probabilité d'observation $P(O/\lambda)$.

- On doit obtenir une séquence d'états $Q = q_1, q_2, ...q_T$ d'un modèle λ qui va être optimale pour une observation O.

- On doit varier les paramètres du modèle afin que $P(O/\lambda)$ soit optimale, ce qui se traduit à une opération d'apprentissage.

La solution des problèmes mentionnés ci-dessus est donnée par [5, 11].

1.4.6 Dictionnaire de références

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

performance de la reconnaissance. Il existe deux types de dictionnaire: le dictionnaire monolocuteur et le dictionnaire multilocuteur.

L'apprentissage monolocuteur est plus facile à réaliser, mais il comporte aussi des difficultés à cause de la variabilité intra-locuteur [4] (vitesse d'élocution, rhume ...etc.). Parmi les méthodes utilisées pour réaliser le dictionnaire monolocuteur [4], il y'a la méthode à références variés qui prend la totalité des prononciations d'un mot par le locuteur comme des références du mot, et la méthode par moyennage où le mot de référence est obtenu à partir des prononciations moyennées après les avoir ramenées sur une échelle temporelle identique avec l'algorithme DTW.

L'apprentissage multilocuteur est plus difficile à réaliser car on a en plus la variabilité inter locuteur [4] (particularités anatomiques, accents régionaux, ...etc). Pour réaliser un dictionnaire de référence multilocuteur, on utilise généralement un algorithme de classification qui divise les prononciations d'un mot en classes d'occurrences. Chaque centre de classe d'occurrence va être une référence du mot.

L'un des algorithmes le plus utilisé pour la classification est l'algorithme K-means, décrit dans le paragraphe suivant.

1.4.6.1 Algorithme de classification (K-means)

K-means est un algorithme de classification populaire utilisé dans divers applications. Il permet la partition de N vecteurs acoustiques en M classes qui sont représentés par des centres $c_i \quad 1 \leq i \leq M$.

Si l'on a un ensemble de N vecteurs $X = \{x_1, x_2, ..., x_N\}$, M classes $\omega_i \quad 1 \le i \le M$ où chaque classe ω_i est représenté par un centre c_i et d une distance. L'algorithme K-means procède comme suit [12]:

On initialise les centres de classes $c_i \in X \quad 1 \le i \le M$ et par itération successive :

1. Chaque vecteur $x_j \in X$ est affecté à une classe qui a un centre plus proche au vecteur tel que :

$$x_j \in w_i \quad si \quad \forall k \; d(x_j, c_i) \le d(x_j, c_k) \quad 1 \le k \le M \; .$$

2. Pour chaque classe, on calcule son nouveau centre $c_i \quad 1 \le i \le M$:

$$c_i = x_{j \in w_i} \text{ tel que } \max_k \left\{ d(x_{j \in w_i}, x_{k \in w_i}) \right\} \text{ est minimum}$$

On répète les deux étapes jusqu'à ce que les nouveaux centres de classes ne changent pas par rapport aux anciens.

1.5 Conclusion

Dans ce chapitre, nous avons donné un bref aperçu de systèmes de production et de perception de la parole. Nous avons présenté le principe de réalisation d'un système de reconnaissance de mots isolés par la méthode dite globale, dont le principe est de comparer les paramètres acoustiques du mot à reconnaître avec ceux des mots du dictionnaire de référence obtenu avec une classification pour une reconnaissance multilocuteur.

Après la segmentation du mot, on extrait les paramètres acoustiques, en utilisant la méthode basé sur le système de production de la parole (LPC) ou la méthode basé sur la perception de l'oreille (MFCC). Ensuite la reconnaissance est obtenue en utilisant l'algorithme d'alignement temporel dynamique (DTW) ou l'algorithme basé la méthode stochastique (HMM).

Dans le prochain chapitre, nous décrirons les transformées en ondellettes qui seront utilisées dans l'étape d'extraction de paramètres de notre système de reconnaissance des mots isolés.

CHAPITRE 2

TRANSFORMÉE EN ONDELETTES

2.1 Introduction

La transformée en ondelette est utilisée dans plusieurs domaines. L'un de ces domaines est la reconnaissance automatique de la parole où elle est utilisée avec efficacité pour l'extraction de paramètres du signal de la parole qui est non stationnaire. Dans ce chapitre, nous introduirons la transformée de Fourrier et la transformée en ondelettes. Cette dernière présente l'avantage d'utiliser des fenêtres de longueurs variables pour l'analyse du signal. Nous aborderons ensuite la notion d'analyse multirésolution et l'utilisation des filtres pour la décomposition en ondelettes. Nous décrirons aussi brièvement la décomposition dyadique et la décomposition en paquets d'ondelettes. Nous terminerons ce chapitre par la présentation d'une méthode de reconnaissance de la parole qui fait usage pour l'étape d'extraction des paramètres d'une décomposition en paquets d'ondelettes similaire à l'échelle de Mel et par la présentation d'une méthode de débruitage qui utilise aussi les ondelettes.

2.2 Transformée de Fourier

La transformée de Fourier (TF) permet d'obtenir la composition fréquentielle $X(f)$ d'un signal $x(t)$ qui varie avec le temps. La transformée de Fourier d'un signal est donnée par l'équation suivante :

$$X(f) = \int_{-\infty}^{+\infty} x(t)\, e^{-j2\pi ft}\, dt \tag{2.1}$$

La transformée de Fourier $X(f)$, qui est habituellement une fonction complexe, décompose le signal en éléments de bases formées par des cosinus et des sinus de durée illimitée. Elle ne possède pas de localisation temporelle. Cela est dû à son incapacité de définir temporellement les différentes valeurs des composantes fréquentielles. Elle est donc limitée aux signaux stationnaires.

Pour remédier à ce problème, on utilise la transformée de Fourier à fenêtre glissante (Short Time Fourier Transform STFT). Cette dernière permet de calculer la transformée de Fourier d'un signal sur des segments temporels de durée finie. Ces segments du signal $x(t)$ sont extraits à l'aide d'une fenêtre glissante $g(t-k)$ de taille fixe. La fenêtre glissante peut être par exemple une fenêtre de Hamming ou une fenêtre gaussienne. La transformée de Fourier à fenêtre glissante est donnée par l'équation suivante :

$$X_{STFT}(f,k) = \int_{-\infty}^{+\infty} x(t)\, g(t-k)\, e^{-j2\pi ft}\, dt \qquad (2.2)$$

L'utilisation d'une fonction fenêtre de longueur fixe dans le calcul de la transformée de Fourier à fenêtre glissante cause certains problèmes. Si la fenêtre est courte, on détecte facilement les pics ou les discontinuités par contre il est plus difficile de détecter les basses fréquences du signal. Si la fenêtre est longue, on a le cas inverse [13].

Contrairement à la transformée de Fourier à fenêtre glissante la transformée en ondelettes (TO) se sert d'une fonction fenêtre de taille variable. Des fenêtres analysantes courtes sont utilisées pour détecter les hautes fréquences de longues fenêtres le sont pour détecter les basses fréquences. La TO peut donc représenter le signal de la parole selon différentes résolutions. Le plan temps fréquence de la TO est illustré par la figure 13 :

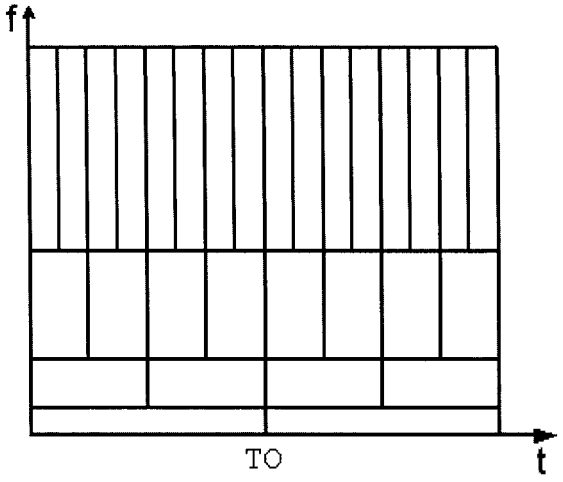

Figure 13 Plan temps fréquence de la transformée en ondelettes [14]

2.3 Transformée en ondelettes

Les ondelettes sont formées par des translations et des dilatations de l'ondelette mère ψ selon l'équation suivante :

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi(\frac{t-b}{a}) \qquad a > 0, b \in R. \tag{2.3}$$

où a et b sont respectivement le facteur d'échelle (dilatation) et le paramètre de translation. Pour que ψ soit admissible comme ondelette, elle doit satisfaire la condition suivante [15] :

$$\int_{-\infty}^{+\infty} \psi(t)\, dt = 0 \tag{2.4}$$

Cette condition montre que l'intégrale de la fonction mère doit être nulle.

La transformée en ondelettes continue (Continuous Wavelet Transform CWT) d'un signal $x(t)$ consiste à mesurer sa similarité avec des bases d'ondelettes $\psi_{a,b}$. Elle est donnée par l'équation suivante :

$$WT_x(a,b) = \int_{-\infty}^{+\infty} x(t)\, \psi^*_{a,b}(t)\, dt \tag{2.5}$$

L'inverse de la transformée en ondelette continue est donnée par l'équation suivante :

$$x(t) = \frac{1}{C_\psi} \int_{-\infty}^{+\infty} \int_{0}^{+\infty} \frac{1}{a^2} WT_x(a,b)\, \psi_{a,b}(t)\, dbda \tag{2.6}$$

$C_\psi < \infty$ est le coefficient calculé par [15] :

$$C_\psi = \int_{-\infty}^{+\infty} |\psi(w)|^2 \frac{dw}{|w|} \tag{2.7}$$

On ne peut pas avoir une bonne localisation temporelle et fréquentielle en même temps. Car le principe de Heisenberg exige que si le signal est focalisé sur un intervalle de temps réduit, sa gamme de fréquence sera grande et vice-versa [13].

2.4 Transformée en ondelettes dyadique

Avec le facteur de dilatation $a = a_0^j$ et le paramètre de translation $b = k\, a_0^j$ discrétisés, on obtient la transformée en ondelette discrète (Discrete Wavelet Transform DWT) qui est donnée par l'équation suivante :

$$DWT_x(j,k) = d_x(j,k) = \frac{1}{\sqrt{a_0^j}} \int_{-\infty}^{+\infty} x(t)\, \psi^*(a_0^{-j}t - k)dt \tag{2.8}$$

Dans la majorité des cas, on utilise $a_0 = 2$ [16] et l'on obtient la transformée en ondelette discrète dyadique. Le signal original $x(t)$ peut être reconstruit à partir des coefficients obtenus par la transformée en ondelette dyadique et il est donné par l'équation suivante:

$$x(t) = \sum_{j=-\infty}^{+\infty} \sum_{k=-\infty}^{+\infty} d_x(j,k) \frac{1}{\sqrt{2^j}} \psi(2^{-j}t - k) \tag{2.9}$$

2.5 Concept d'analyse multirésolution

L'analyse multirésolution de $L^2(R)$, développée par Mallat [14, 17] et Meyer [18] , permet la représentation d'un signal par des approximations dans une suite d'échelles 2^{-j}. Ces échelles sont définies par des espaces imbriqués $\{V_j\}_{j\in\mathbb{Z}}$ qui sont soumis aux conditions suivantes [14]:

$$\forall (j,k) \in \mathbb{Z}^2 \,,\, f(t) \in V_j \Leftrightarrow f(t - 2^j k) \in V_j \tag{2.10}$$

La condition (2.10) indique la stabilité de l'espace V_j pour n'importe quelle translation de l'ordre de $2^j k$.

$$\forall j \in \mathbb{Z},\ V_{j+1} \subset V_j \tag{2.11}$$

La condition (2.11) exprime que l'espace du niveau de résolution $j+1$ découle de l'espace du niveau de résolution j .

$$\forall j \in \mathbb{Z}, f(t) \in V_j \Leftrightarrow f(\frac{t}{2}) \in V_{j+1} \tag{2.12}$$

La condition (2.12) indique l'approximation de la fonction f dans l'espace V_{j+1} est obtenue à partir de la dilatation par 2 de son approximation dans l'espace précédant V_j.

$$\lim_{j \to +\infty} V_j = \bigcap_{-\infty}^{+\infty} V_j = \{0\} \tag{2.13}$$

La condition (2.13) exprime qu'on n'a pas d'approximation du signal quand le niveau de la résolution j tend vers $+\infty$.

$$\lim_{j \to -\infty} V_j = \overline{\bigcup_{-\infty}^{+\infty} V_j} = L^2(R) \tag{2.14}$$

La condition (2.14) exprime que l'approximation du signal reproduit le signal original quand le niveau de la résolution j tend vers $-\infty$.

$$\exists \varphi \in V_0 \text{ tel que } \{\varphi(t-n)\}_{n \in \mathbb{Z}} \text{ est une base orthonormée de } V_0 \tag{2.15}$$

La condition (2.15) indique qu'il y'a une fonction de l'espace V_0 qui forme une base orthonormée de cette espace. Cette fonction est nommée fonction d'échelle φ. D'où :

$$\varphi_{j,n}(t) = \frac{1}{\sqrt{2^j}} \varphi(\frac{1}{2^j} t - n) \tag{2.16}$$

Si $Pv_j(f)$ est la projection orthogonale de la fonction $f \in L^2(R)$ sur l'espace V_j, ce qui représente les coefficients d'approximation, on aura :

$$Pv_j(f) = \sum_{-\infty}^{+\infty} \langle f, \varphi_{j,n}(t) \rangle \varphi_{j,n}(t) \tag{2.17}$$

Lors du passage de l'espace V_j vers l'espace V_{j+1}, des détails du signal sont perdus. Ces coefficients détails peuvent être calculés en faisant une projection orthogonale de l'approximation du signal sur un espace W_{j+1} qui est le complément de l'espace V_{j+1} dans l'espace V_j, on a :

$$V_j = V_{j+1} \oplus W_{j+1} \tag{2.18}$$

Selon (2.15), on peut construire une fonction appelé ondelette ψ qui sera une base orthonormée de l'espace W_j. telle que:

$$\psi_{j,n}(t) = \frac{1}{\sqrt{2^j}} \psi(\frac{1}{2^j} t - n) \tag{2.19}$$

Donc, si $Pw_j(f)$ est la projection orthogonale de la fonction $f \in L^2(R)$ sur l'espace W_j, ce qui représente les coefficients détails, on aura :

$$Pw_j(f) = \sum_{-\infty}^{+\infty} \langle f, \psi_{j,n}(t) \rangle \psi_{j,n}(t) \tag{2.20}$$

2.6 Décomposition par banc de filtres

La décomposition en ondelettes orthogonales peut être effectuée par une opération de filtrage en cascade qui est appelée algorithme pyramidal [14]. Ainsi, les coefficients de détails d_{j+1} sont obtenus par filtrage avec un filtre passe haut g complété par une

décimation par 2 et les coefficients approximations a_{j+1} avec un filtre passe bas h complété par une décimation par 2. L'opération de la décomposition est illustrée par la figure 14 :

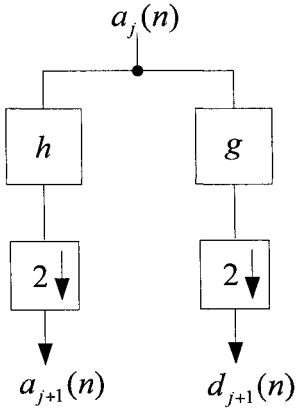

Figure 14 Algorithme pyramidal de la décomposition

Les coefficients de chaque niveau sont calculés à l'aide des coefficients du niveau précédent qui sont données par les relations suivantes [14] :

$$\begin{aligned} a_{j+1}(n) &= a_j(n) * h(2n) = \sum_{l=-\infty}^{+\infty} a_j(l)\, h(l-2n) \\ d_{j+1}(n) &= a_j(n) * g(2n) = \sum_{l=-\infty}^{+\infty} a_j(l)\, g(l-2n) \end{aligned} \tag{2.21}$$

où * est le produit de convolution. La relation entre les deux filtres est donnée par :

$$h(L-1-n) = (-1)^n g(n) \qquad n = 0,1,....,L-1. \tag{2.22}$$

où L est la taille du filtre.

À chaque décomposition, les nouveaux coefficients d'approximation caractériseront la première moitié de la largeur de bande des coefficients précédents et la deuxième moitié sera occupée par les nouveaux coefficients de détails comme le montre la figure 15 :

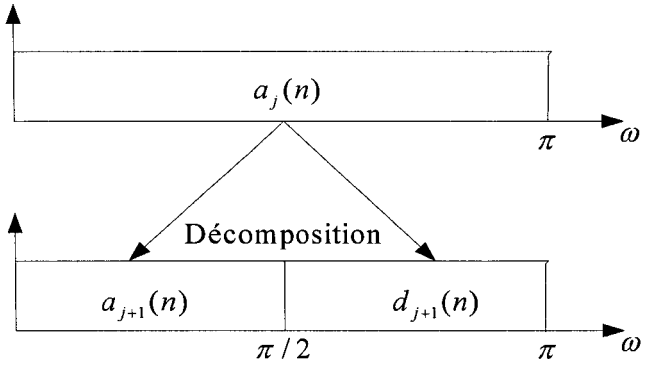

Figure 15 Répartition fréquentielle des coefficients de la décomposition

L'algorithme pyramidal rapide est inversible [14], c'est à dire que l'on peut obtenir la reconstruction du signal par interpolation par 2 suivi d'un filtrage. Ce dernier est réalisé à l'aide des filtres miroir en quadrature. L'opération de la reconstruction est donnée par la figure 16 :

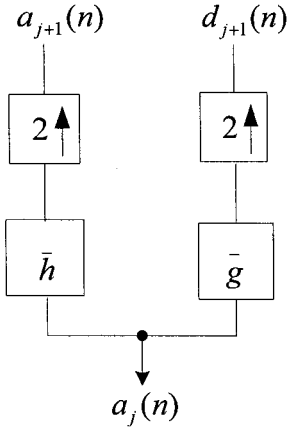

Figure 16 Algorithme pyramidal de la reconstruction

La relation entre les filtres de la décomposition et ceux de la reconstruction est donnée par :

$$g(n) = (-1)^{1-n}\,\bar{h}(n)$$
$$\bar{g}(n) = (-1)^{1-n}\,h(n) \tag{2.23}$$

2.7 Décomposition dyadique

La décomposition dyadique [14] est obtenue en décomposant le signal avec des bancs de filtres en coefficients d'approximations et en coefficients de détails pour le premier niveau. Pour les autres niveaux, les décompositions sont obtenues en décomposant de nouveau les coefficients d'approximations des niveaux précédents en coefficients d'approximations x_0^{j+1} et en coefficients de détails x_1^{j+1}. Les coefficients de détails ne sont pas décomposés dans la décomposition dyadique. La figure 17 illustre une décomposition dyadique :

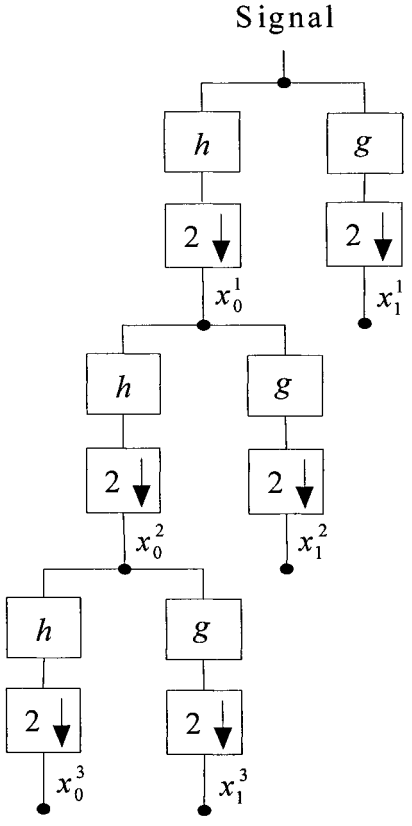

Figure 17 La décomposition dyadique à 3 niveaux

La répartition fréquentielle de la décomposition dyadique à 3 niveaux est donnée par la figure 18 :

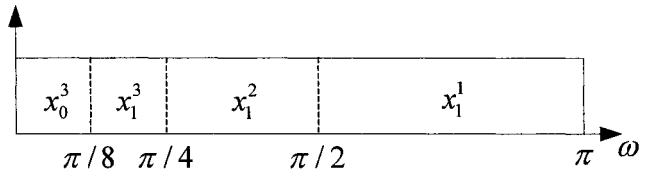

Figure 18 Répartition fréquentielle de la décomposition dyadique

2.8 Décomposition en paquet d'ondelettes

Les paquets d'ondelettes sont une transformation en ondelettes qui généralise l'analyse multirésolution [14]. Contrairement à la décomposition en ondelettes dyadique, les coefficients de détails de chaque niveau sont aussi décomposés. En effet, Les coefficients d'approximations et de détails de chaque niveau sont obtenus à partir des coefficients d'approximations et de détails du niveau précédent et ils sont exprimés par les relations suivantes [19]:

$$\begin{aligned} x_{2p}^{j+1}(n) &= \sum_l h(l)\, x_p^j(n-2^j l) \\ x_{2p+1}^{j+1}(n) &= \sum_l g(l)\, x_p^j(n-2^j l) \end{aligned} \tag{2.24}$$

où x_{2p}^{j+1} et x_{2p+1}^{j+1} sont respectivement les coefficients d'approximations et de détails de la décomposition du niveau $j+1$ *et* x_p^j sont les coefficients d'approximations ou de détails décomposés.

La figure 19 illustre la structure d'une décomposition en paquet d'ondelettes à trois niveaux:

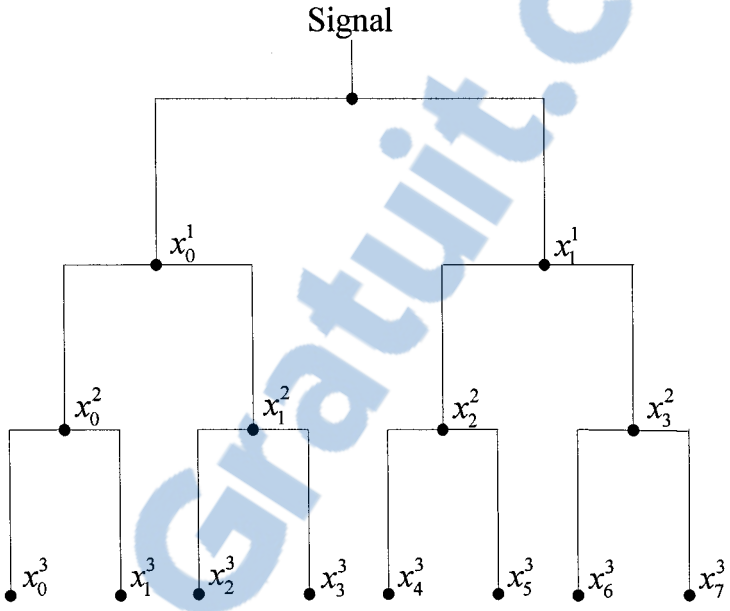

Figure 19 La décomposition en paquets d'ondelettes à 3 niveaux

La répartition fréquentielle de la décomposition en paquets d'ondelettes à 3 niveaux est donnée par la figure 20 :

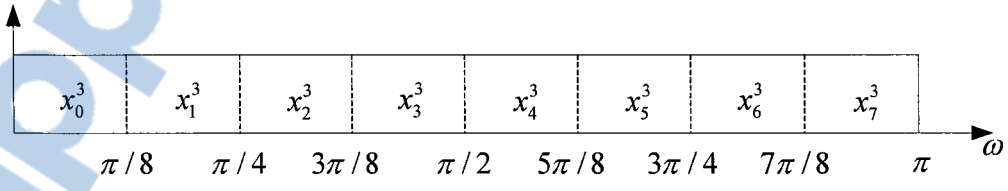

Figure 20 Répartition fréquentielle de la décomposition en paquets d'ondelettes

2.9 Reconnaissances de la parole avec les ondelettes

Dans la littérature, il existe plusieurs méthodes [20-22] de reconnaissance automatique de la parole qui utilisent la transformée en ondelette. La transformée en ondelette a été utilisée avec succès durant l'étape d'extraction des paramètres. Ces derniers sont obtenus à partir des coefficients d'approximations et de détails obtenus durant la décomposition en ondelettes. Les différentes méthodes utilisent diverses structures de décomposition en ondelettes. Farooq et Datta [19] ont présenté une structure de décomposition en paquets d'ondelettes admissibles (AWP Admissible Wavelet Packet) qui est illustrée par la figure 21 :

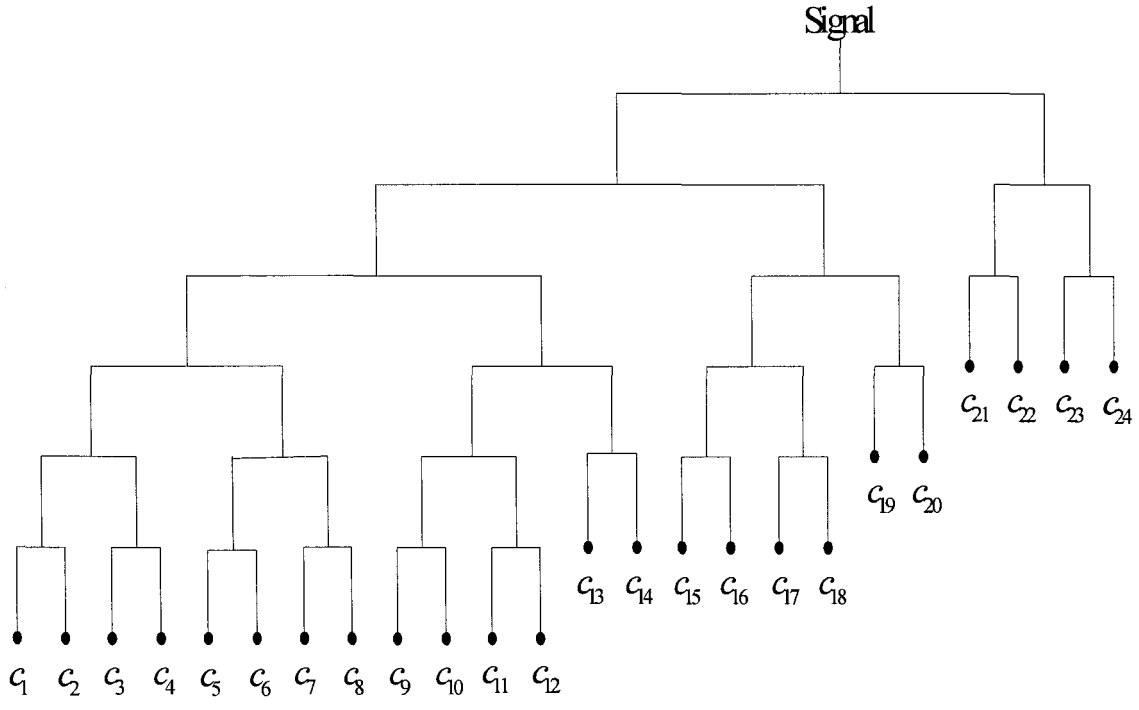

Figure 21 Paquets d'ondelettes admissibles

Cette structure donne une décomposition fréquentielle en sous bandes qui est similaire à l'échelle de Mel.

La décomposition en paquet d'ondelettes de la structure est obtenue avec l'algorithme pyramidal rapide. Elle est effectuée jusqu'au niveau 6 et seulement quelques coefficients seront utilisés. Farooq et Datta ont utilisé une base de données échantillonnée à 16 Khz qui donne donc une largeur de bande de 8 Khz. La décomposition fréquentielles des coefficients de la structure est donnée par le tableau I.

L'utilisation de la méthode MFCC pour la reconnaissance de la parole détecte mal les variations brusques dans le signal de la parole. Cela est due à l'utilisation de la transformée Fourier à fenêtre glissante des fenêtres d'analyses de taille fixe [19]. Par contre, la transformée en ondelette permet de bien détecter ces variations grâce à ses fenêtres d'analyses de taille variable.

Farooq et Datta ont obtenu les paramètres de la reconnaissance à partir de la transformée discrète en cosinus des logarithmes des énergies de chaque sous bande de la structure. Seulement les 13 premiers coefficients de la DCT ont été choisis. Le processus d'extraction des paramètres est donné par la figure 22.

Tableau I

Décomposition fréquentielle des coefficients
selon la méthode de Farooq et Datta [19]

Coefficients c_n	Bande occupé en Hz	Largeur de bande en Hz
1	0-125	125
2	125-250	125
3	250-375	125
4	375-500	125
5	500-625	125
6	625-750	125
7	750-875	125
8	875-1000	125
9	1000-1125	125
10	1125-1250	125
11	1250-1375	125
12	1375-1500	125
13	1500-1750	250
14	1750-2000	250
15	2000-2250	250
16	2250-2500	250
17	2500-2750	250
18	2750-3000	250
19	3000-3500	500
20	3500-4000	500
21	4000-5000	1000
22	5000-6000	1000
23	6000-7000	1000
24	7000-8000	1000

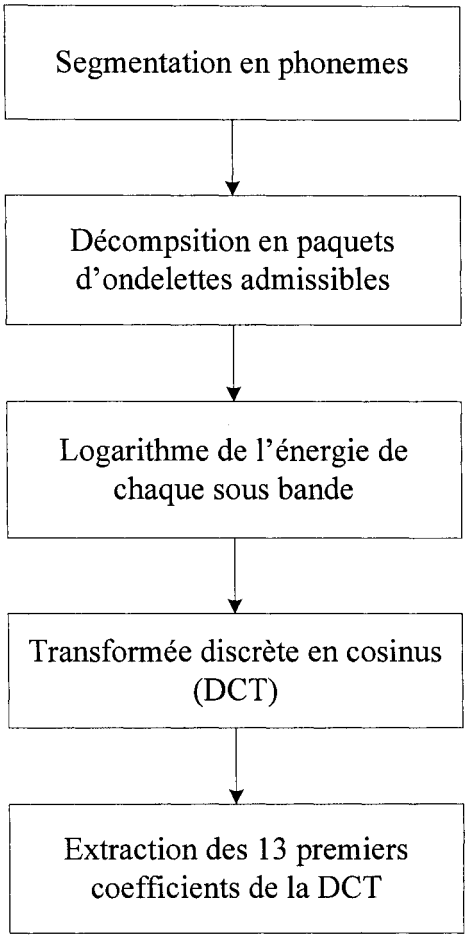

Figure 22 Paquets d'ondelettes admissible pour l'extraction de paramètres selon la méthode de Farooq et Datta

2.10 Débruitage à l'aide de la transformée en ondelettes

Le signal de la parole capté par les systèmes de reconnaissance est généralement affecté par le bruit du capteur et celui de l'environnement. Pour cette raison, il faut effectuer un débruitage afin de réaliser une reconnaissance robuste. Parmi les méthodes de débruitage existantes, il y a celles qui utilisent les ondelettes. Le principe consiste à estimer un seuil à partir des coefficients obtenus par la décomposition en ondelettes. Après un seuillage

de ces coefficients, on effectue la reconstruction en ondelettes pour obtenir le signal débruité. L'algorithme du débruitage avec les ondelettes est représenté par la figure 23:

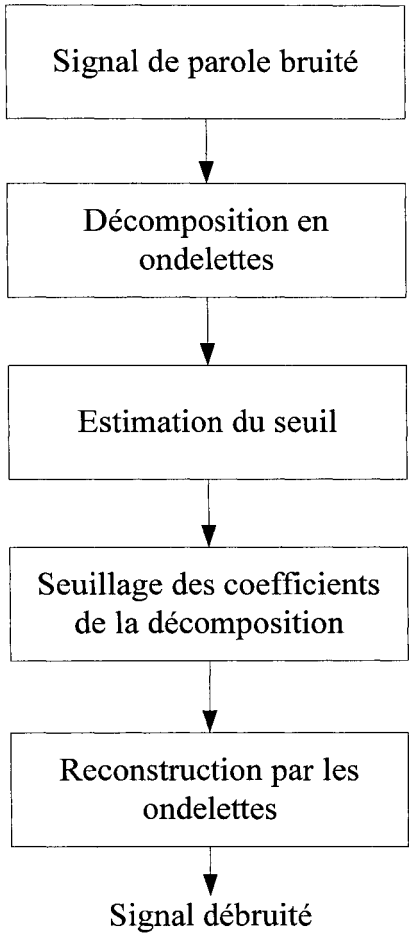

Figure 23 Débruitage par ondelettes

2.10.1 Estimation du seuil

La méthode la plus connue d'estimation du seuil est celle proposée par Donoho et Johnston [23]. Ce seuil est appelé le seuil universel et il est donné par l'équation suivante:

$$T = \sigma\sqrt{2\log(N)} \tag{2.25}$$

Avec

$$\sigma = \frac{\text{mediane}(|d_{jk}|)}{0.6745} \tag{2.26}$$

où N est le nombre d'échantillons du signal de la parole et d_{jk} sont les coefficients de détails de la décomposition en ondelette du niveau 1.

Pour améliorer la reconnaissance dans le cas d'un signal bruité avec un bruit corrélé, Johnston et Silverman [24] ont estimé σ à partir des coefficients de détails de la haute résolution du dernier niveau de la décomposition. L'estimation du σ à partir des coefficients dépendant du nœud [25] a été aussi utilisé pour effectuer une reconnaissance robuste dans le cas d'un bruit coloré:

L'équation (2.25) est utilisée dans le cas de la décomposition dyadique. Pour la décomposition en paquets d'ondelettes, l'équation (2.25) doit être modifiée comme suit [25]:

$$T = \sigma\sqrt{2\log(N\log_2(N))} \tag{2.27}$$

2.10.2 Seuillages des coefficients de la décomposition

Après avoir calculé le seuil T, on passe à l'étape du seuillage. Parmi les méthodes de seuillages les plus connus sont le seuillage mou (soft) et le seuillage dur (hard) [26, 27]:

- Seuillage mou

$$\text{Seuillage}_{\text{mou}} = \begin{cases} \text{sign}(d_{jk})\,(|d_{jk}| - T) & |d_{jk}| > T \\ 0 & |d_{jk}| \le T \end{cases} \tag{2.28}$$

d_{jk} est un coefficient de la décomposition en ondelettes et T est le seuil estimé.

- Seuillage dur

$$\text{Seuillage}_{\text{dur}} = \begin{cases} d_{jk} & |d_{jk}| > T \\ 0 & |d_{jk}| \le T \end{cases} \tag{2.29}$$

Chang et al. [25] ont présenté une méthode de seuillage dur modifié en utilisant la "loi de μ" pour le seuillage des coefficients inférieur au seuil T et elle est donnée par l'équation suivante :

$$\text{Seuillage}_{\text{modifié}} = \begin{cases} d_{jk} & |d_{jk}| > T \\ \dfrac{1}{\mu}\,\text{sign}(d_{jk})\,T\,((1+\mu)^{\left|\frac{d_{jk}}{T}\right|} - 1) & |d_{jk}| \le T \end{cases} \tag{2.30}$$

2.11 Conclusion

Dans ce chapitre, nous avons brièvement présenté la transformée de Fourier et la transformée en ondelettes ainsi que les avantages de cette dernière pour l'analyse du signal avec des fenêtres de taille variable. Nous avons présenté aussi l'analyse multirésolusion qui est la notion de base pour la construction des ondelettes

orthogonales. Nous avons vu que la décomposition et la reconstruction en ondelettes peuvent être obtenues à l'aide de l'algorithme pyramidal rapide. Un aperçu a été donné sur la décomposition en ondelette dyadique et en paquets d'ondelettes. Nous avons présenté ensuite une méthode de reconnaissance de phonèmes qui utilise une décomposition en paquets d'ondelettes similaire à l'échelle de Mel et ce chapitre s'est terminé par la présentation d'une méthode de débruitage par les ondelettes.

CHAPITRE 3

PROCESSEUR DÉDIÉ AU TRAITEMENT NUMÉRIQUE DES SIGNAUX

3.1 Introduction

Le processeur dédié au traitement numérique des signaux ou DSP (Digital Signal Processor) a une architecture distincte des autres processeurs. Il est adapté aux applications de traitement du signal. Il permet d'effectuer des applications en temps réel et des opérations de calculs complexes avec une grande vitesse. Les avantages du DSP sont les suivants [28]:

- **Facilité** : Simplicité de programmation.
- **Flexibilité** : On peut le programmer ou le reprogrammer pour différentes applications.
- **Économie** : À cause de son faible coût et de sa consommation d'énergie minimale.

Ces avantages ont permis au DSP de concurrencer les autres processeurs sur le marché. Sa place par rapport aux autres processeurs est donnée par la figure 24 :

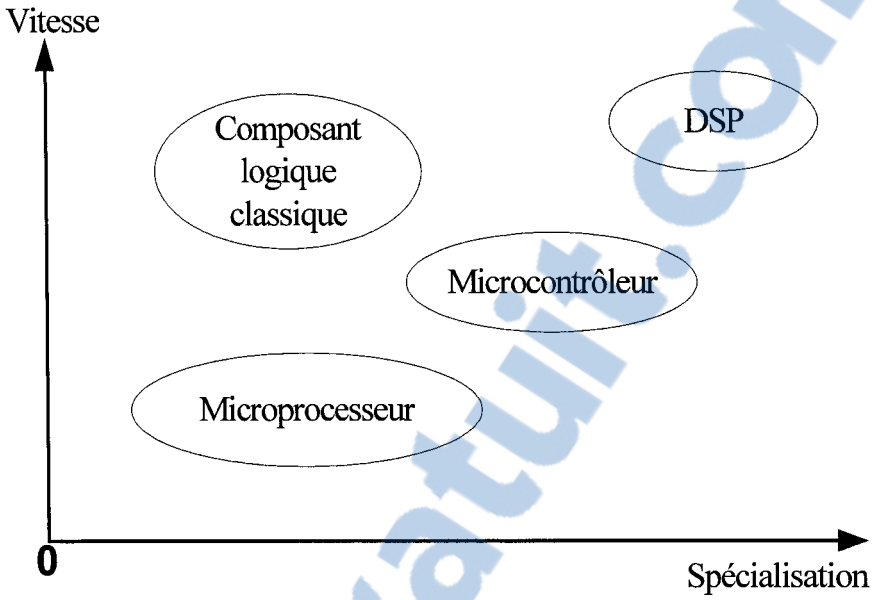

Figure 24 Place du DSP vis-à-vis aux autres processeurs [29].

Il existe différents domaines d'applications du DSP, tels que [30] : système de contrôle, traitement de la parole, télécommunication, militaire, médicale, traitement d'image … etc.

Dans ce chapitre, nous présenterons le logiciel Code Composer Studio (CCS) qui offre un environnement de développement intégré (IDE) et qui permet de créer le fichier exécutable nécessaire pour le fonctionnement du DSP. Nous décrirons ensuite le DSP TMS320C6711 de Texas Instrument, qui sera utilisé pour notre projet et qui a l'avantage de permettre l'acquisition du signal de la parole à une fréquence de 8 KHz à l'aide des convertisseurs disponibles sur la carte DSK. Finalement, nous présenterons le kit DSK qui contient le DSP et qui est connecté à un ordinateur de type PC à l'aide d'un port parallèle.

3.2 Code Composer Studio

Le logiciel Code Composer Studio (CCS) est un environnement de programmation utilisable pour programmer le DSP. Il permet de générer des fichiers exécutables qui seront chargés dans le DPS afin d'assurer son fonctionnement. Il contient aussi un simulateur qui permet de simuler l'exécution des programmes sur DSP. En plus, Le CCS permet de visualiser les données et de vérifier le contenu de la mémoire et des registres durant le fonctionnement du DSP. L'environnement CCS est illustré par la figure 25:

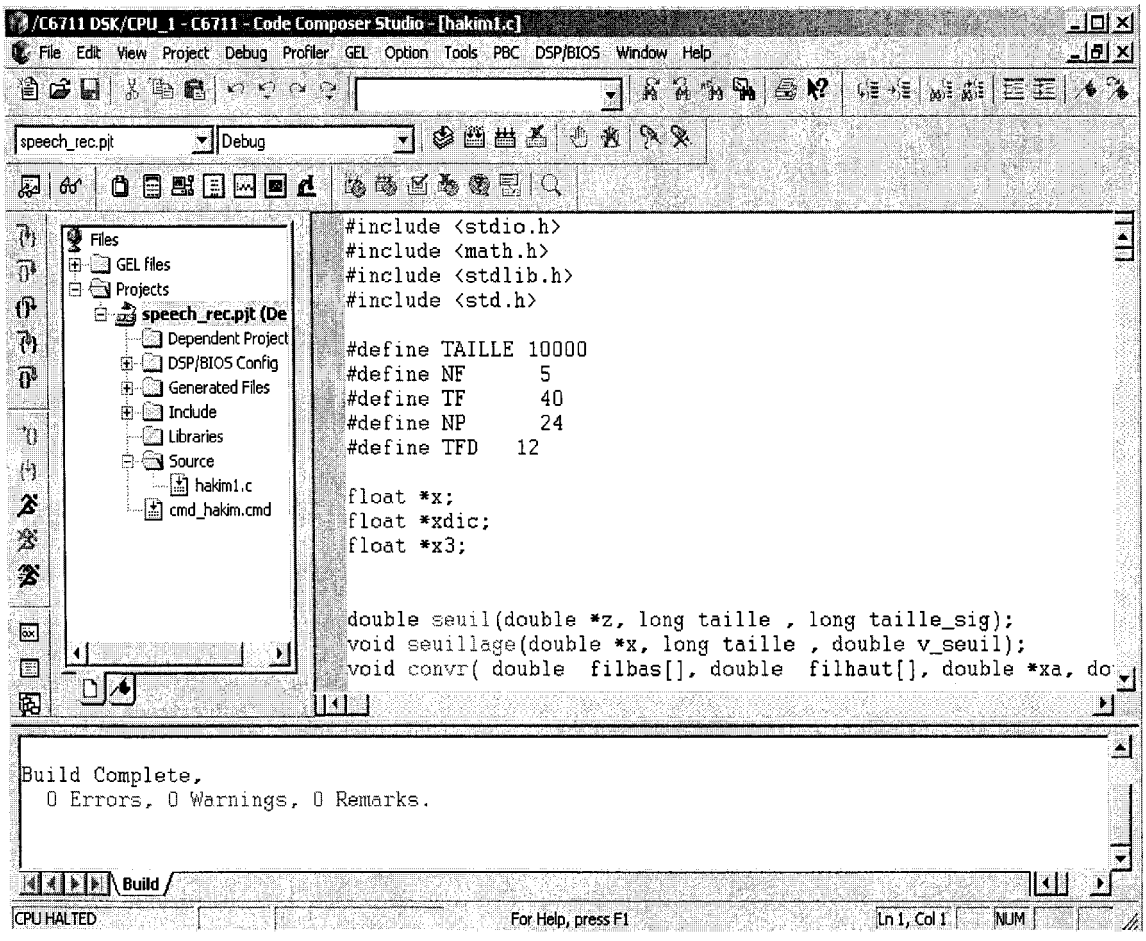

Figure 25 Environnement du code composer studio

3.2.1 Les composantes du CCS

Le Code Composer Studio offre un environnement qui permet d'écrire des programmes en langage C ou en Assembleur, de les compiler et de générer le fichier exécutable à partir du programme édité et des fichiers ajoutés qui sont nécessaires pour le bon fonctionnement du programme principal.

La génération du fichier exécutable passe par les étapes suivantes [28]:

- Le compilateur C génère un fichier en langage assembleur qui a l'extension .asm à partir du fichier édité.
- L'assembleur transforme le fichier assembleur .asm en un fichier objet .obj.
- L'éditeur de lien (linker) rassemble les fichiers .obj en un seul fichier exécutable .out.

3.2.2 Création du fichier exécutable

Pour créer le fichier exécutable qui sera chargé sur le DSP, il faut passer par plusieurs étapes. La première étape consiste à créer le projet (Project →New). Ensuite, il faut écrire le programme principal en C ou en Assembleur. Aussi, il faut joindre tous les fichiers indispensables pour le bon fonctionnement du programme (Project → Add Files to Project) ainsi que les fichiers .cdb et .cmd qui sont créés avec la commande (File→New→DSP/BIOS Configuration …). La figure 26 illustre une fenêtre du projet CCS avec ses fichiers.

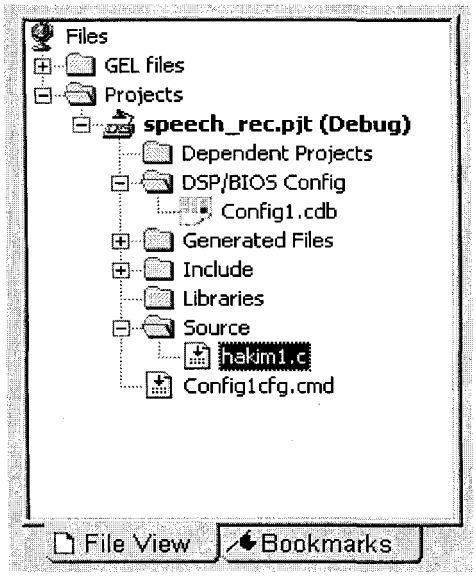

Figure 26 Fenêtre du projet CCS avec ses fichiers

Avant de générer le code, il faut configurer les options de compilation et de lien (Project → Build Options) tel qu'illustré par la figure 27:

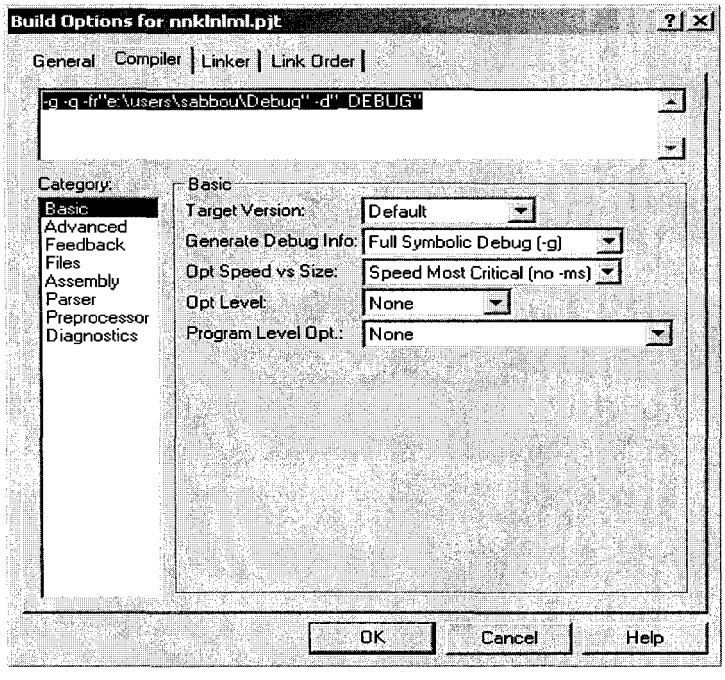

Figure 27 Fenêtre de configuration des options de compilation et de lien

Après l'étape de configuration, on construit le projet (Project→Rebuild All) qui génère le fichier exécutable avec l'extension .out. Ce dernier est ensuite chargé sur le DSP (File →Load). Finalement, il ne reste qu'à exécuter le programme (Debug→Run).

3.2.3 Temps réel avec CCS

Des applications en temps réel sont effectuées à l'aide du RTDX (Real-time data exchange) qui permet des échanges bidirectionnelles entre le PC et le DSP. Les données sont échangées durant l'exécution du DSP et elle s'effectue à travers l'interface JTAG (Joint team action group) [31].

Une analyse du DSP en temps réel peut être réalisée avec le DSP/BIOS qui offre la possibilité de suivre et de surveiller une application de DSP pendant son fonctionnement sans que l'exécution en temps réel soit perturbée [31].

3.3 DSP TMS320C6711

Le TMS320C6711 est un DSP de Texas instrument de la famille TMS320. On distingue trois groupes principaux de DSP de la famille TMS320 [32]: les DSP (C1x, C2x, C2xx, C5x, C54x et C62xx) à point fixe, les DSP (C3x, C4x et C67xx) à virgule flottante et les DSP Multi-corps C8x qui contient plusieurs DSP's. L'évolution de la famille de DSP TMS320 est donnée par la figure 28:

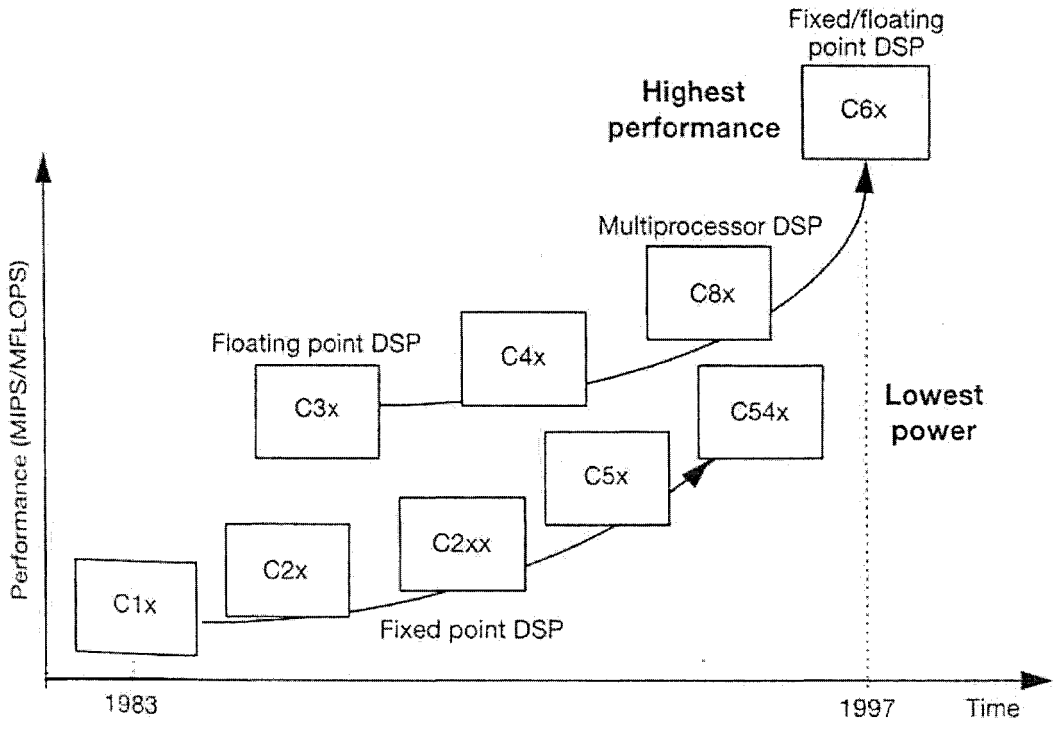

Figure 28 Évolution de la famille TMS320 [32]

Le TMS32010 est la première génération de DSP [28] introduit en 1982 par Texas instrument (TI). Le TMS320C6711 est un DSP de la famille C6x à virgule flottante qui a une architecture VLIW (Very Long Instruction Word) améliorée [30]. Il peut chercher 8 instructions de 32 bits en un seul cycle avec une fréquence d'horloge de 150 MHZ. Son architecture est donnée par la figure 29:

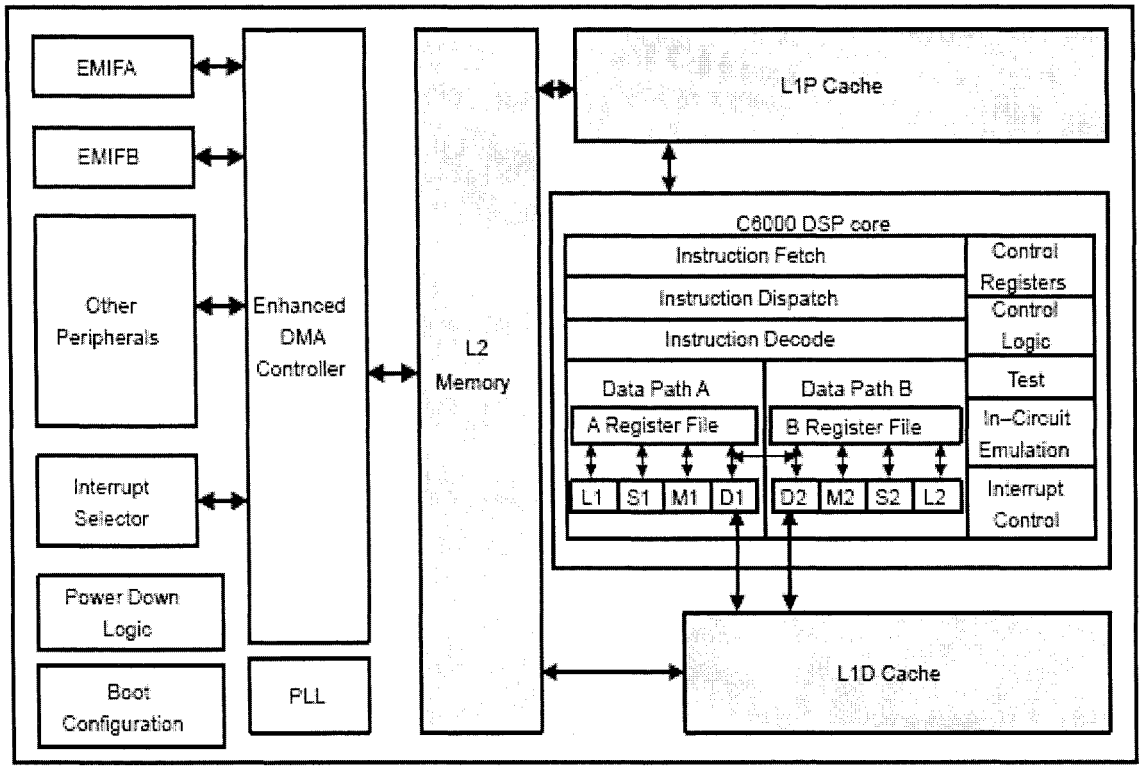

Figure 29 Architecture du DSP TMS320C6711 [33]

Le TMS320C6711 est constitué de :

- L'unité centrale de traitement CPU.
- Périphériques.
- Mémoire.

3.3.1 L'unité centrale de traitement (CPU)

Le CPU est constitué d'une unité de contrôle de programme, de deux unités fonctionnelles, de deux blocs de 16 registres de 32bits, de contrôleurs d'interruptions et d'autres éléments.

3.3.1.1 Unité de contrôle de programme

Elle est constituée des éléments suivants [32]:

- Unité "fetch" programme : Elle a pour rôle récupérer les programmes. Cette opération se déroule en quatre phases :

    1. Phase PG : L'adresse du code est générée.
    2. Phase PS : L'adresse est envoyée à la mémoire.
    3. Phase PW : Attente de lecture du code de la mémoire.
    4. Phases PR : Lecture du code.

- Unité "dispatche" de l'instruction : Le code récupéré de le mémoire est affecté à l'unité fonctionnelles associée.

- Unité de décodage de l'instruction : Elle a pour rôle de décoder l'instruction.

3.3.1.2 Unités fonctionnelles

Le CPU contient huit unités fonctionnelles divisées en deux parties 1 et 2. Leurs fonctions sont les suivants:

- Unités .M1 et .M2 : Ces unités sont dédiées à la multiplication.
- Unités .L1 et .L2 : Ces unités sont dédiées à l'arithmétique et la logique.
- Unités .D1 et .D2 : Ces unités sont dédiées au chargement, la sauvegarde et calcul d'adresse.
- Unités .S1 et .S2 : Ces unités sont dédiées pour le décalage de bit, l'arithmétique, la logique et le branchement.

3.3.1.3 Registres

Le CPU contient 32 registres de 32 bits divisé en deux blocs égaux : registre fichier A (A0-A15) et registre fichier B (B0-B15) et leurs fonctions sont réparties comme suit [28] :

- Les registres A0-A1 et B0-B2 : Utilisés comme registres conditionnels.
- Les registres A4-A7 et B4-B7 : Utilisés pour adressage circulaire.
- Les registres A0-A9, B0-B2 et B4-B9 : Utilisés comme registres temporaires.
- Les registres A10-A15 et B10-B15 : Utilisés pour la sauvegarde et la restitution de données d'un sous-programme.

À ces 32 registres, s'ajoutent les registres de contrôles et d'interruptions.

3.3.2 Les périphériques du TMS320C6711

Le TMS320C6711 a plusieurs périphériques qui sont [33, 34] :

- Le contrôleur DMA. Il permet sans l'aide du CPU de transférer des données entre les espaces mémoire (interne, externe et des périphériques). Il a quatre canaux programmables et un autre canal auxiliaire.

- Le contrôleur EDMA. Il permet de transférer des données entre les espaces mémoire comme le DMA. Il a 16 canaux programmables

- L'interface port hôte HPI. Il donne au processeur hôte un contrôle total pour un accès direct de l'espace mémoire du CPU et à la cartographie de la mémoire des périphériques du DSP.

- Deux McBSP qui sont des ports séries multi-Canaux protégés . Ils permettent la communication avec les périphériques externes. Ils ont la même structure. Ils supportent une communication full-duplex.

- L'interface de mémoire externe EMIF. Il permet l'interface avec plusieurs éléments (mémoires) externes.

- Les compteurs. Le DSP possède deux compteurs qui peuvent être synchronisés par une source interne ou externe et ils sont utilisés comme générateurs de pulsations, compteurs d'événements externes, interrupteur du CPU après l'exécution de tâches et déclencheur du DMA/EDMA.

- Les interruptions : l'ensemble des périphériques contient jusqu'à 32 sources d'interruptions.

3.3.3 La structure de la mémoire

Le TMS320C6711 utilise une mémoire externe et une mémoire interne. La mémoire externe occupe les espaces CE0, CE1, CE2, CE3 avec une capacité de 256 MOctets chaque. La mémoire interne a une taille de 72 KOctets qui est décomposée en deux niveaux [30] : Le niveau (L1) est constitué de deux mémoires caches de 4KB chacune, (L1P) qui est utilisée pour les programmes et (L1D) qui est utilisée pour les données. Le Niveau (L2) est composé de 64KB de mémoire RAM ou de mémoire cache qui est utilisée pour les données et les programmes.

L'organisation de la mémoire interne est illustrée par la figure 30:

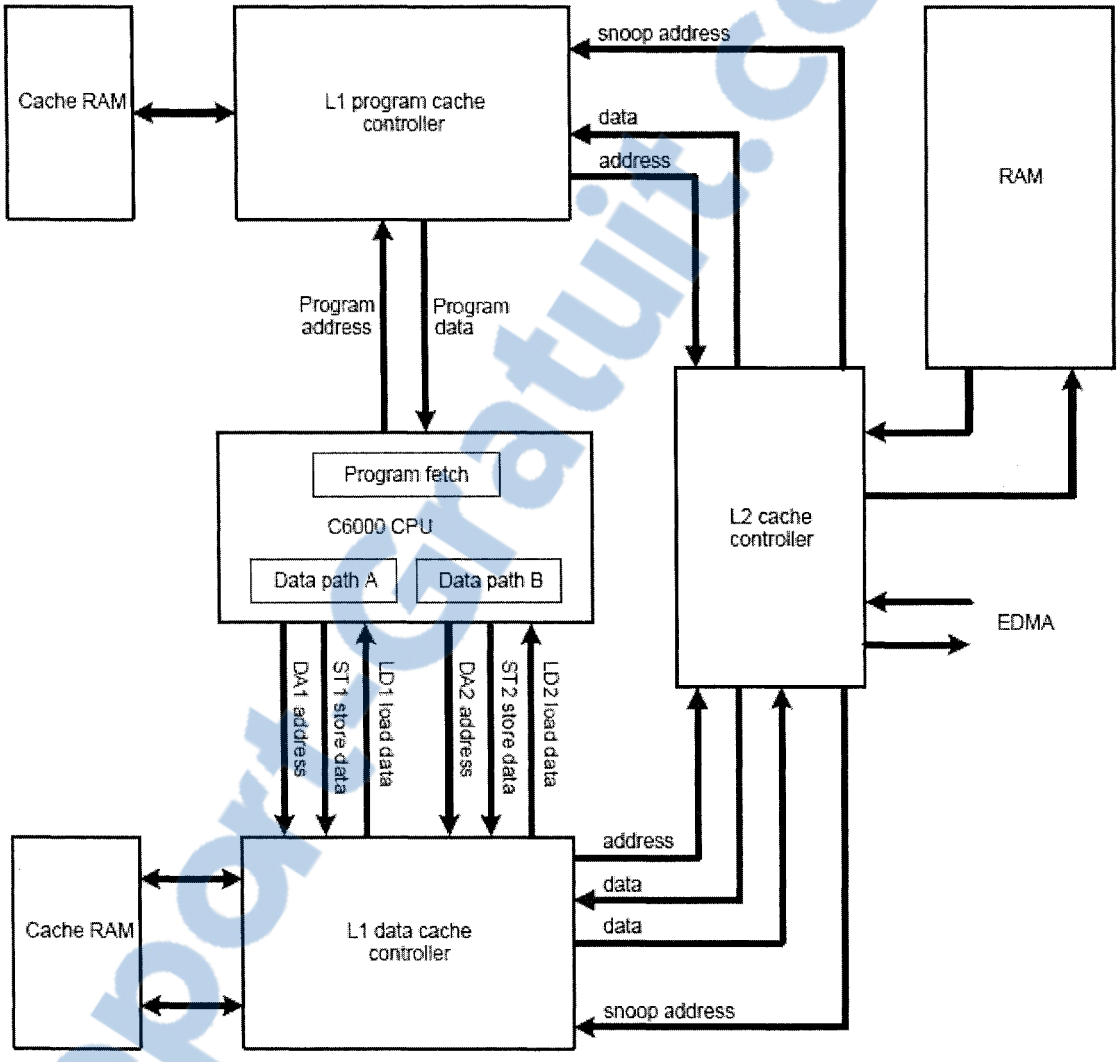

Figure 30 Organisation de la mémoire interne [33]

Un espace mémoire d'une taille totale de 4GB peut être adressé par le DSP TMS320C6711.

3.4 Carte DSK6711

La carte DSK6711 est une carte qui contient le DSP et les périphériques nécessaires à son fonctionnement. Elle permet des applications en temps réel. La carte DSK est constituée des éléments principaux suivants [28]:

- Le DSP TMS320C6711 qui peut atteindre jusqu'à 900 millions floating-point opérations par second (MFLOPS).

- Un port parallèle pour la connexion à un PC à l'aide d'un câble DB25.

- 16 MOctets de mémoire synchrone dynamique RAM (SDRAM) et de 128 KOctets de mémoire flash ROM.

- Un codec TLC320AD535 qui est configuré avec une entrée et une sortie audio mono. Il peut acquérir un signal de la parole avec une fréquence d'échantillonnage de 8KHz et utilise la conversion analogique-numérique et la conversion numérique-analogique sigma-delta. Le système de conversion élimine les signaux d'entrées erronés à l'aide d'un filtre d'entrée antialiasing et contient aussi un filtre de sortie qui permet le lissage du signal de sortie traité [28].

- L'alimentation pour les composants.

La figure 31 illustre la carte DSK6711 :

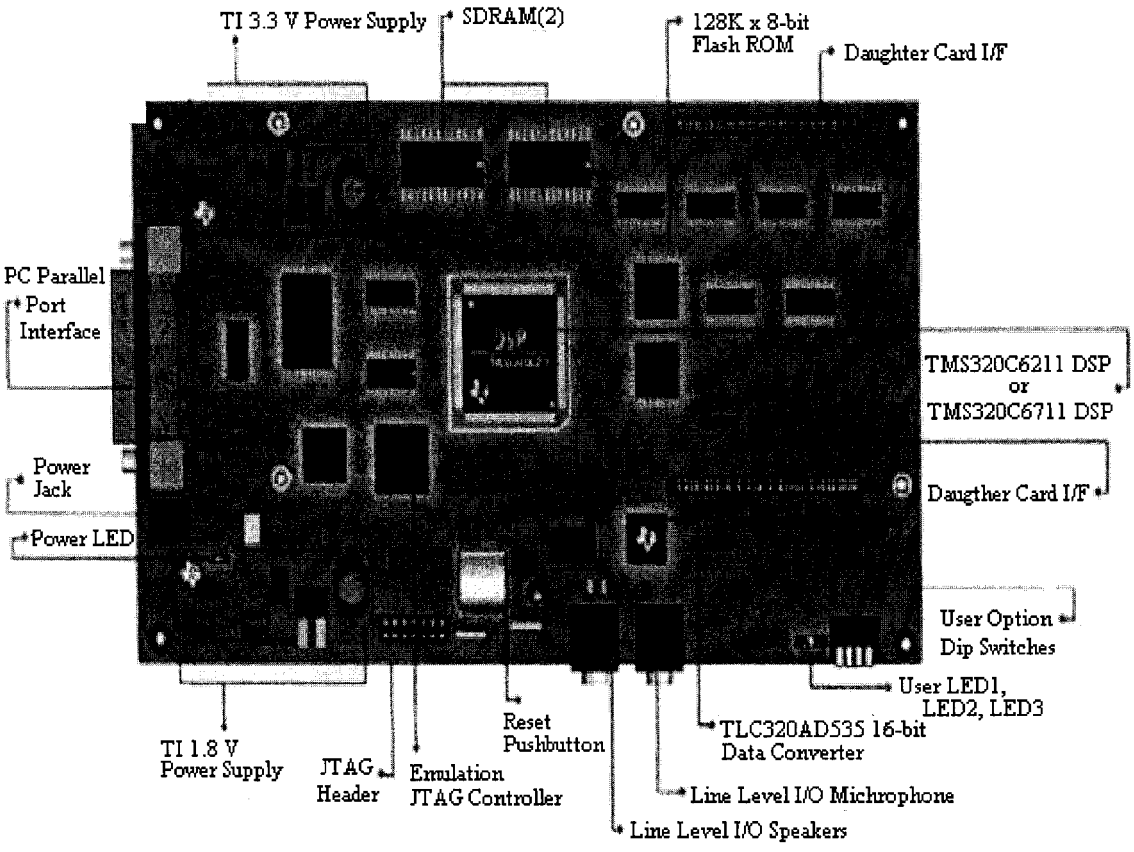

Figure 31 Carte DSK6711 [28]

3.5 Conclusion

Dans ce chapitre, nous avons présenté quelques un des avantages du DSP ainsi que quelques un de ses domaines d'applications. Nous avons ensuite introduit le logiciel Code Composer Studio qui permet de générer les fichiers exécutables nécessaires au fonctionnement du DSP. Nous avons aussi mentionné la capacité du Code Composer Studio à générer des applications en temps réel et sa possibilité de suivre le fonctionnement du DSP. Nous avons aussi donné un aperçu de l'architecture du DSP

TMS320C6711. Finalement, nous avons brièvement décrit la carte DSK6711 qui contient le DSP et les différents outils nécessaires pour mettre en oeuvre une application.

CHAPITRE 4

MÉTHODOLOGIE ET SIMULATIONS

4.1 Introduction

Après avoir globalement décrit les différentes méthodes classiques de réalisation d'un système de reconnaissance de la parole et avoir donné un aperçu sur la transformée en ondelettes et de l'analyse multirésolusion, nous décrirons dans ce chapitre notre système de reconnaissance des chiffres isolés. Le système est basé sur une méthode de segmentation du chiffre en un nombre fixe de segments, ce qui donne des segments de tailles différentes d'un chiffre à un autre. L'extraction de paramètres à partir de ces segments sera effectuée par la méthode proposée par Farooq et Datta [19], qui est décrite au chapitre 2. La méthode utilise une structure de décomposition en paquets d'ondelettes similaire à l'échelle de Mel.

Pour ce faire, nous décrirons d'abord les différentes étapes de la réalisation de notre système de reconnaissance de chiffres isolés. Dans la deuxième partie, nous présenterons et commenterons les différents résultats de la simulation effectuée à l'aide de l'environnement MATLAB® qui nous a permis de gagner beaucoup de temps au niveau de la programmation grâce aux fonctions qui y sont incorporées. Finalement, nous présenterons les résultats de l'implémentation de notre système de reconnaissance des chiffres isolés à l'aide du DSP TMS320C6711 et de la carte DSK6711.

4.2 Méthodologie du système de reconnaissance

Les différentes étapes du système de reconnaissance des chiffres isolés proposé sont illustrées par la figure 32 :

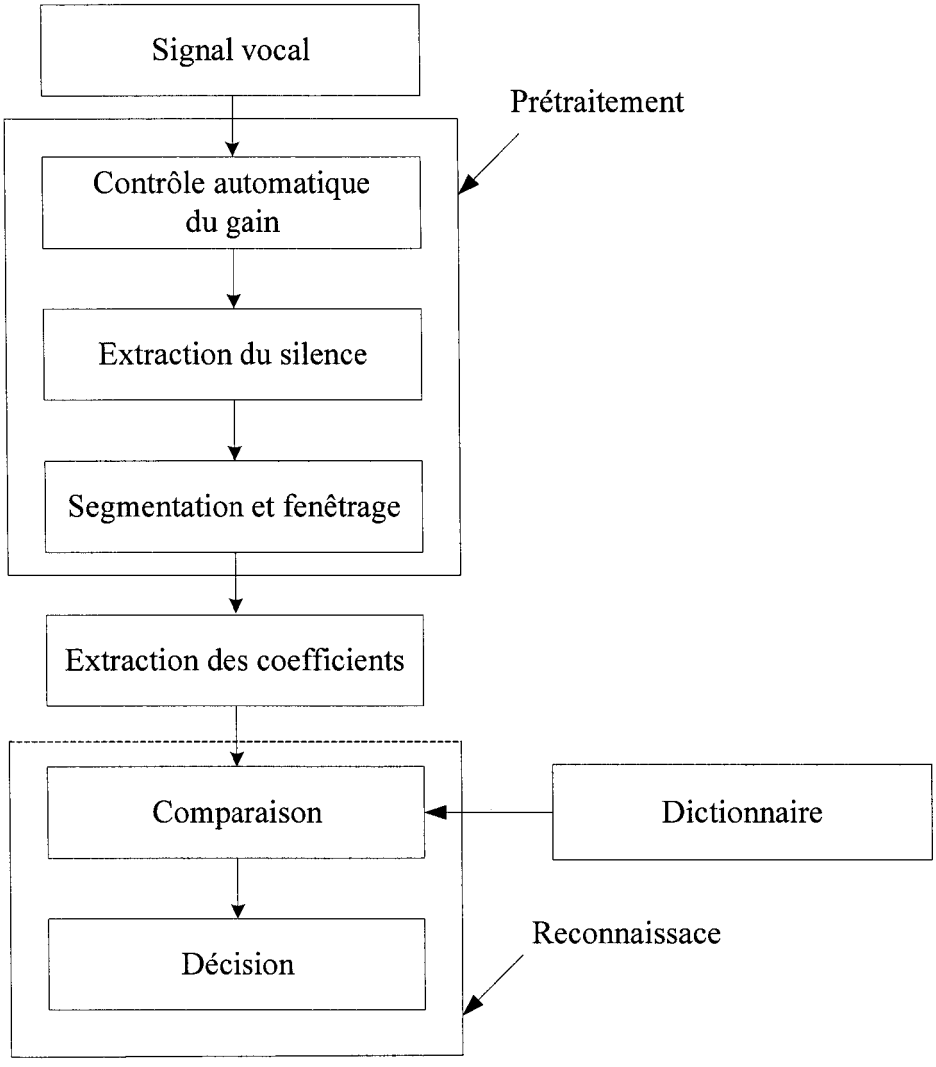

Figure 32 Système de reconnaissance des chiffres isolés

Après avoir effectué la normalisation du gain, on extrait la parole du silence. Ensuite, on passe à l'étape de la segmentation et du fenêtrage. À partir des segments obtenus, on

extrait les paramètres de chaque sous bande de la décomposition en paquet d'ondelettes. Finalement, on compare les paramètres du chiffre à reconnaître avec ceux du dictionnaire de référence créé avec un algorithme de classification. Le chiffre reconnu sera le chiffre du dictionnaire de référence qui a la plus petite différence des paramètres.

4.3 Prétraitement

Avant d'effectuer l'extraction des paramètres qui seront utilisés pour la reconnaissance, on passe d'abord par les étapes de prétraitement suivant:

- Contrôle automatique du gain.
- Isolation du mot du silence.
- Segmentation et fenêtrage.

4.3.1 Contrôle automatique du gain

Durant la prononciation du chiffre, l'amplitude du signal diffère d'un chiffre à un autre à cause de l'intensité de locution. Pour des voix fortes, on aura des grandes amplitudes et pour des voix faibles, on en aura de plus petites. Pour remédier à ce problème, on effectue une normalisation des amplitudes, c'est à dire un contrôle automatique du gain où l'amplitude maximale de chaque chiffre est fixée à une constante. Une normalisation de deux différents chiffres "one" à une amplitude max =1000 est illustrée par la figure 33:

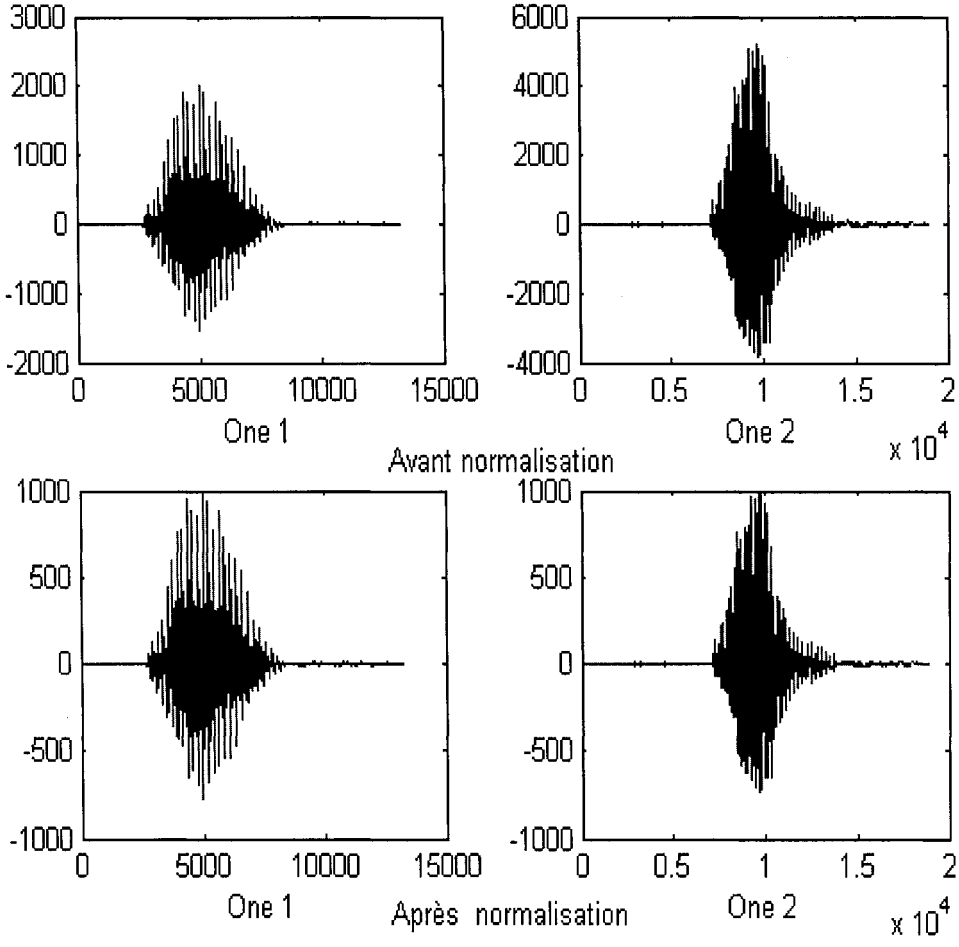

Figure 33 Contrôle automatique du gain

4.3.2 Isolation du mot du silence

Dans cette partie, nous allons utiliser une méthode d'isolation de la parole du silence qui a été proposée par Rabiner et Sumbur [35]. La méthode est illustrée par l'organigramme de la figure 34:

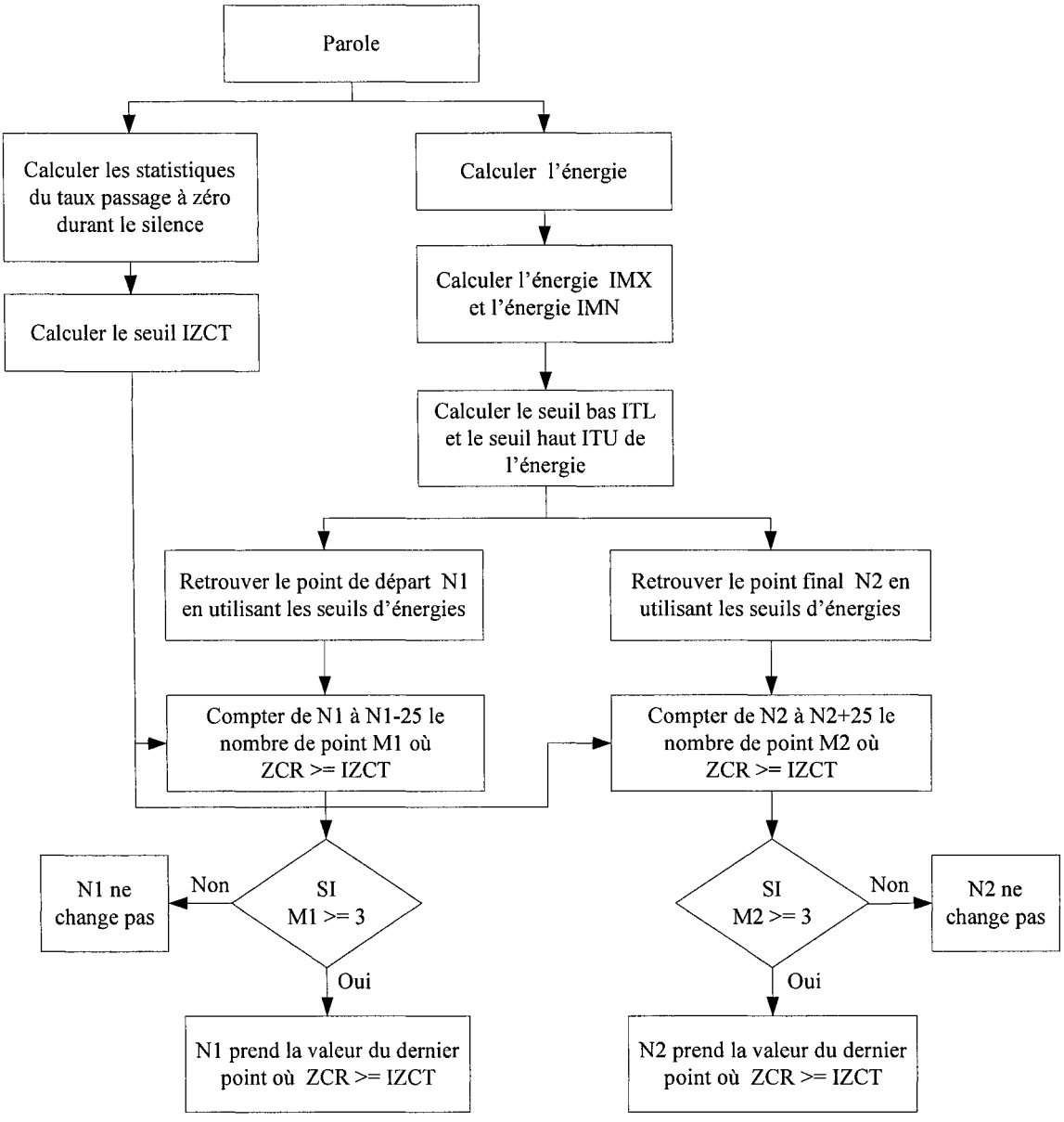

Figure 34 Algorithme de recherche du début et la fin du mot [35]

Le principe de la méthode utilise l'énergie et le taux de passage par zéro. Pour calculer l'énergie $E(n)$, Rabiner et Sambur l'ont déterminée comme la somme des amplitudes du signal $s(n)$ sur des intervalles de temps de 10 ms. Pour une fréquence d'échantillonnage de 20 kHz, l'énergie est donnée par la formule suivante :

$$E(n) = \sum_{i=-100}^{+100} |s(n+i)| \tag{4.1}$$

Le taux de passage par zéro est aussi calculé sur des intervalles de temps de 10 ms et il est donné par la formule suivante :

$$IZC(n) = \sum_{i=-100}^{+100} (\operatorname{sgn} s(n+i) - \operatorname{sgn} s(n+i-1)) \tag{4.2}$$

où sgn est la fonction signe.

L'algorithme de détection du début et de la fin de la parole commence par calculer les statistiques du silence, qui sont évaluées durant le premier intervalle du temps de l'enregistrement de la parole de 100 ms. On suppose que durant cet intervalle la parole n'existe pas (silence). Dans cette étape, on calcul le seuil $IZCT$ du taux de passage par zéro qui est donné par l'équation suivante :

$$IZCT = \min(IF, \overline{IZC} + 2\sigma_{IZC}) \tag{4.3}$$

où $\overline{IZC}$ est la moyenne de la somme du taux de passage par zéro et IF est égale à 15 (10 ms). On calcule aussi le seuil bas ITL et le seuil haut ITU de l'énergie:

$$\begin{aligned} I1 &= 0.03 * (IMX - IMN) + IMN \\ I2 &= 4 * IMN \\ ITL &= 1.75 * \min(I1, I2) \\ ITU &= 4 * ITL \end{aligned} \tag{4.4}$$

où IMN est la moyenne de l'énergie du silence (les premières 100 ms) et IMX est l'énergie (crête) maximale du signal enregistré. Les valeurs de IF , ITL et ITU ont été modifiées par rapport à celles de la référence [35] pour une fréquence d'échantillonnage de 8 kHz.

L'étape suivante de l'algorithme consiste à calculer le début et la fin du mot en utilisant l'énergie. Pour trouver le début du mot, la méthode procède en cherchant à partir du début du signal de la parole, le point où l'énergie est supérieure au seuil bas *ITL*. Si à partir de ce point le seuil haut *ITU* est dépassé sans tomber sous le seuil *ITL*, alors le point sera considéré comme étant le début du mot. Sinon, on recherche de nouveau le point qui va satisfaire ces conditions. L'algorithme de la recherche du début du mot avec l'énergie est illustré par la figure 35 :

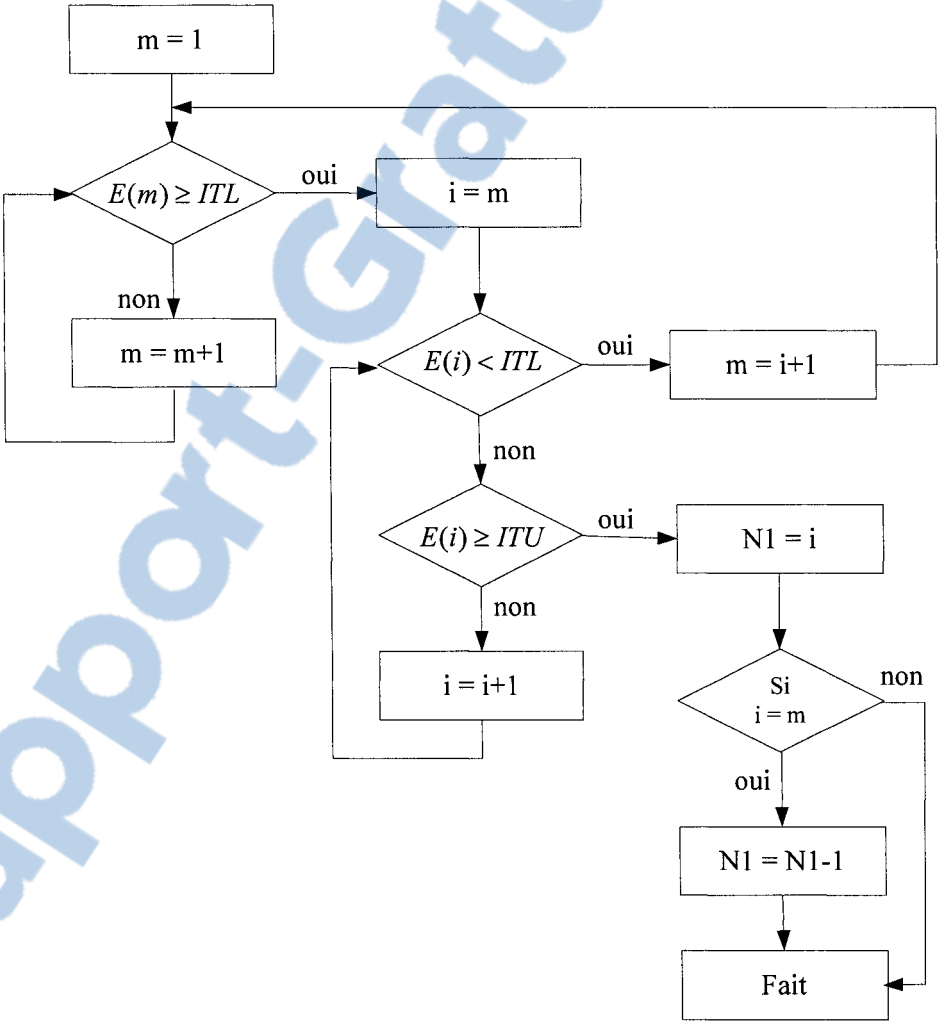

Figure 35 Détection du début du mot avec l'énergie [35]

Dans la base de données utilisée, nous avons constaté la présence de bruit comme le montre la figure ci-dessous. Ce bruit affecte la détection du début du mot. Pour remédier à ce problème, on vérifie à partir du début du mot trouvé jusqu'à 5 intervalles qui suivent si l'énergie ne retombe pas au dessus du seuil *ITL* . Si l'énergie ne retombe pas, le point sera conservé comme début du mot ; sinon on recalcule de nouveau le début du mot avec l'algorithme qui utilise l'énergie jusqu'à ce que la vérification soit satisfaite.

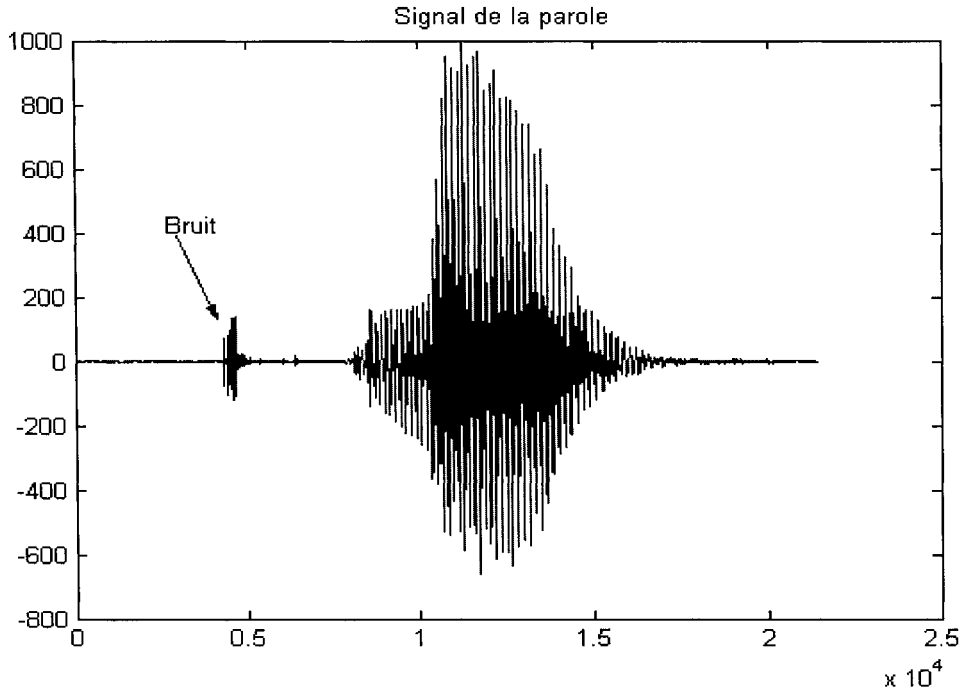

Figure 36 Mauvaise détection du début du mot à cause du bruit

Pour trouver la fin du mot, l'algorithme utilise le même principe que celui qui est utilisé pour trouver le début du mot. Mais seulement la recherche débute à partir de la fin du signal. L'algorithme de la recherche de la fin du mot en utilisant l'énergie est illustré par la figure 37:

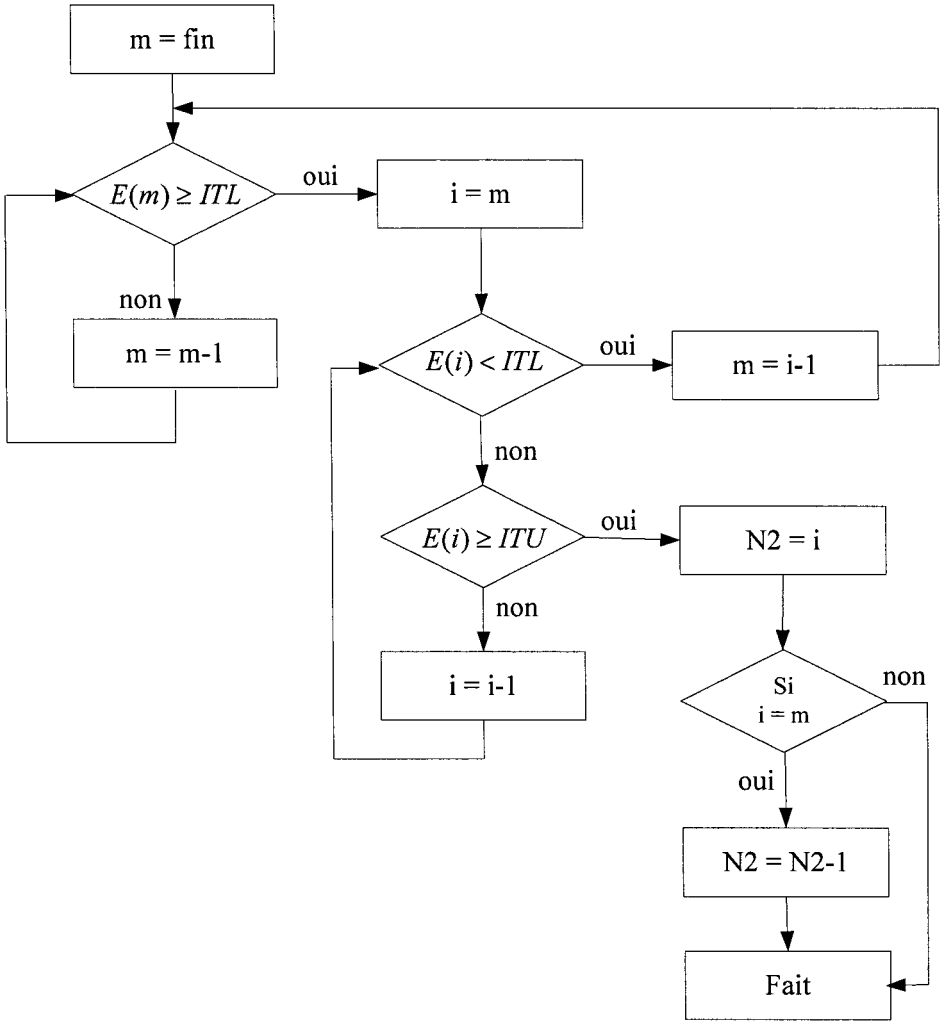

Figure 37 Détection de la fin du mot avec l'énergie [35]

La dernière étape de l'algorithme [35] consiste à valider le début du mot N1 et la fin du mot N2 trouvés dans l'étape précédente. Pour le début du mot, la méthode procède en vérifiant à partir N1 jusqu'à 25 trames passés combien de fois le taux de passage à zéro a dépassé le seuil *IZCT*. Si l'on n'a pas plus de deux dépassements, on garde l'ancien point. Sinon, le dernier point qui a le taux de passage à zéro dépassant le seuil *IZCT*, deviendra le début du mot. Pour la fin du mot, le contrôle sera réalisée à partir N2 jusqu'à 25 trames prochaines.

4.3.3 Segmentation et fenêtrage

Afin d'extraire les paramètres acoustiques de la parole, il faut subdiviser le signal en plusieurs segments. De façon à assurer la stationnarité du signal de chaque segment. Pour cela, nous avons choisi une méthode de segmentation [36] qui subdivise le chiffre isolé en un nombre fixe de segments. La taille d'un segment diffère d'un chiffre à un autre. Le choix de cette méthode, nous permet d'éviter l'utilisation de l'algorithme complexe DTW. La segmentation est réalisée avec un recouvrement proportionnel à la taille du segment. La figure 38 illustre une segmentation d'un chiffre isolé en un nombre fixe de segments:

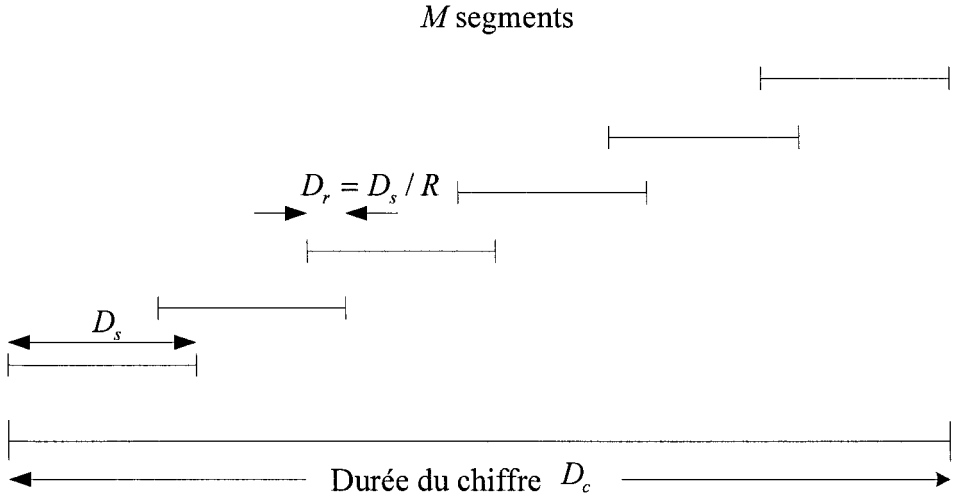

Figure 38 Segmentation du chiffre isolé en M segments [36]

Dans cette figure, D_c est la durée du chiffre isolé, D_s est la durée du segment, D_r est la durée du recouvrement et R est le rapport de recouvrement.

Si M est le nombre de segments, la durée du segment est donnée par la formule suivante:

$$D_s = \frac{R\, D_c}{M\, R - M + 1} \qquad \text{avec} \quad R = \frac{D_s}{D_r} \tag{4.5}$$

4.4 Extraction de paramètres

Après avoir subdivisé le chiffre isolé en segments, on passe à l'étape d'extraction de paramètres. La méthode choisie est celle proposée par Farooq et Datta [19] qui est décrite dans le deuxième chapitre. Cette méthode utilise une décomposition en paquet d'ondelettes similaire à l'échelle de Mel. Les étapes de l'extraction de paramètres sont décrites par l'organigramme suivant:

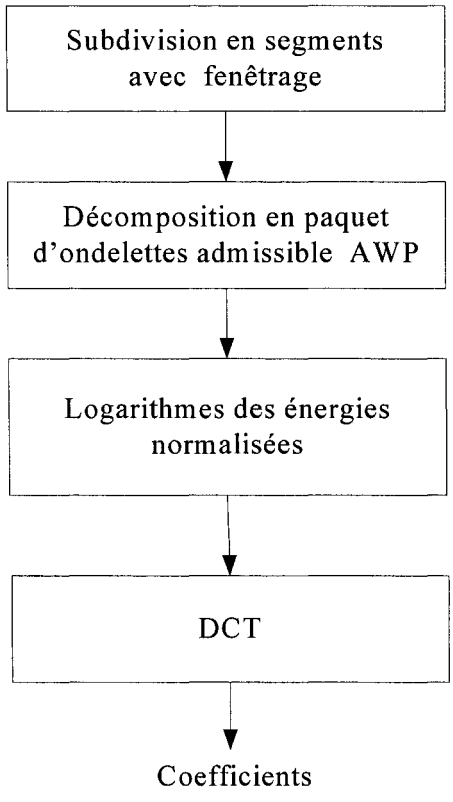

Figure 39 Extraction de paramètres

Après avoir appliqué à chaque segment une fenêtre de Hamming, on fait une décomposition en paquets d'ondelettes admissibles pour extraire les énergies à partir de chaque sous bande. Ces énergies seront normalisées par rapport au nombre de coefficients de la sous bande [37] avant que leurs logarithmes ne soient appliqués à la transformée en cosinus discrète. On obtient ainsi les coefficients de la DCT.

4.5 Dictionnaire de référence

Dans cette étape, nous proposons une méthode de création d'un dictionnaire de références en utilisant la théorie des ensembles flous, dont l'idée se trouve dans l'application de l'algorithme Fuzzy-C-Means (FCM) [38] pour la classification. Cet algorithme est basé sur la minimisation sous contrainte, d'une fonction coût qui est donnée par l'équation suivante :

$$J_m(U,V) = \sum_{k=1}^{n} \sum_{i=1}^{c} u_{ik}^m \, d_{ik}^2 \tag{4.6}$$

et la contrainte est donnée par :

$$\sum_{i=1}^{c} u_{ik} = 1 \tag{4.7}$$

Si $X = \{x_1, x_2, ..., x_k, ..., x_n\}$ est un ensemble de n éléments à classifier, $V = \{v_1, v_2, ..., v_i, ..., v_c\}$ est un ensemble de c centres de classes, alors d_{ik} est la distance entre l'élément x_k et le centre de classe v_i, $u_{ik} \in [0,1]$ est le degré d'appartenance de l'élément x_k à la classe i et $m \in [1,\infty[$ est le facteur flou.

La minimisation de la fonction coût $J_m(U,V)$ est obtenue en appliquant l'algorithme de classification suivant :

Après avoir fixé le nombre de classes c et le facteur flou m, on calcul par itération successives:

- Les ensembles suivants:

$$\begin{aligned} I_k &= \{\, i, 1 \le i \le c \,/\, d_{ik} = \|x_k - v_i\| = 0 \,\}; \\ J_k &= \{\, 1, 2, ..., c \,\} - I_k. \end{aligned} \tag{4.8}$$

- Les degrés d'appartenances des paramètres avec la relation :

$$I_k = \varnothing \quad \Rightarrow \quad u_{ik} = \frac{1}{\sum_{j=1}^{c} \left(\frac{d_{ik}}{d_{jk}} \right)^{2/m-1}}$$

$$ou \quad I_k \neq \varnothing \quad alors \quad u_{ik} = 0 \quad \forall i \in J_k \quad avec \quad \sum_{k=1}^{n} u_{ik} = 1 \tag{4.9}$$

- Et les nouveaux centres de classes avec la relation:

$$v_i = \frac{\sum_{k=1}^{n} (u_{ik})^m \, x_k}{\sum_{k=1}^{n} (u_{ik})^m} \quad \forall i \tag{4.10}$$

On répète ces étapes jusqu'à ce que l'écart entre les nouveaux centres de classes et les anciens centres de classes successifs soit inférieur à un seuil de convergence ε. À l'itération b, le critère de convergence est donné par :

$$\max\left| v_i^{b-1} - v_i^b \right| < \varepsilon, \quad 1 \le i \le c \tag{4.11}$$

Dans notre application, l'initialisation des centres de classes v_i sera faite de manière automatique, c'est-à-dire avec des centres quelconques.

4.6 Reconnaissance

Pour réaliser une reconnaissance d'un chiffre isolé, les coefficients extraits sont comparés à ceux du dictionnaire de référence, où chaque chiffre est représenté par des modèles de coefficients. La comparaison sera effectuée en calculant la distance euclidienne qui est donnée par la formule suivante:

$$d = [\sum_{n=1}^{p} (c_r(n) - c_d(n))^2]^{1/2} \tag{4.12}$$

où c_r représente les coefficients du chiffre à reconnaître, c_d représente les coefficients du chiffre du dictionnaire et p est le nombre de coefficients du chiffre.

Le chiffre reconnu est le chiffre qui présente la plus petite distance par rapport aux autres chiffres.

4.7 Débruitage

Un bloc de débruitage du signal de la parole est également ajouté à notre système de reconnaissance afin de nous permettre de réaliser une reconnaissance robuste de chiffres affectés par le bruit. La méthode de débruitage choisie est celle qui utilise les ondelletes , elle est décrite dans le chapitre 2 et elle est appliquée avant l'étape de reconnaissance. Le processus de reconnaissance de chiffres bruités est illustré par la figure 40.

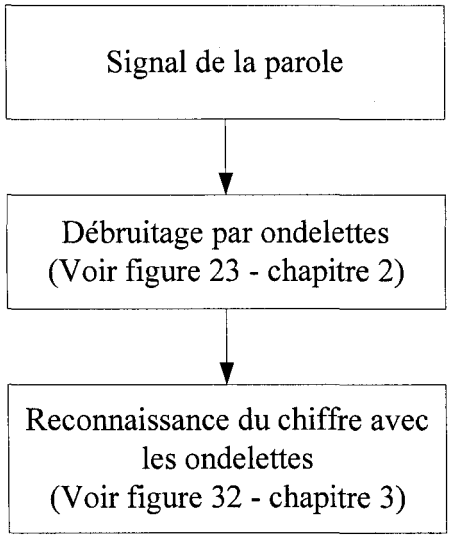

Figure 40 Reconnaissance du chiffre avec débruitage

4.8 Simulation et résultats

Afin de tester la méthode proposée, nous avons utilisé une base de données de Texas Instrument TIDIGITS [39] qui est échantillonnée à 20 KHz, ce qui donne une largeur de bande de 10 KHz. On a utilisé 55 locuteurs chacun d'entre eux prononce en anglais deux fois (groupe A et B) onze chiffres isolés. Les chiffres prononcés sont "one" jusqu'à "nine", "zéro" et "oh".

Pour les simulations dont les résultats sont donnés aux tableaux II à VII, ont été réalisées sans débruitage sur des chiffres non bruités. Tout d'abord, l'isolation du chiffre du silence a été effectuée manuellement, par la suite nous l'avons effectuée automatiquement (tableau VII). Nous avons ensuite testé notre méthode de reconnaissance avec débruitage sur des chiffres bruités par un bruit blanc gaussien (AGWN)

Pour les deux premières simulations dont les résultats sont données aux tableaux II et III, sont une reconnaissance dépendante du locuteur. nous avons varié le nombre de segments et le rapport de recouvrement. nous avons utilisé différents ordres d'ondelettes de Daubechies [40]. Ces dernières sont à support compact.

Le tableau II montre les résultats de la reconnaissance monolocuteur où pour chaque locuteur les prononciations A sont utilisés comme chiffres à reconnaître et les prononciations B comme chiffres de références et vice-versa. Nous avons utilisé l'ondelette de Daubechies d'ordre 4 (db4) et nous avons varié le nombre de segments $M = 5$, 10 et 15 et le rapport de recouvrement R de 2 à 6.

Tableau II

Taux de reconnaissance monolocuteur avec db4

Segments N	Rapport de recouvrement Dr	DCT				
		13	15	18	20	24
5	2	98,84	98,84	98,93	98,93	99,26
	3	99,17	99,17	99,42	99,5	99,67
	4	98,76	98,93	98,76	99,17	99,34
	5	99,50	99,34	99,34	99,67	99,83
	6	99,17	99,17	99,42	99,67	99,67
10	2	99,5	99,5	99,5	99,59	99,67
	3	99,26	99,17	99,09	99,17	99,59
	4	98,84	99,01	99,01	99,09	99,09
	5	99,34	99,17	99,17	99,17	99,34
	6	99,09	99,17	9926	99,26	99,26
15	2	99,09	98,93	99,26	99,34	99,5
	3	99,09	99,17	99,09	99,01	99,34
	4	99,01	99,17	99,09	99,09	99,26
	5	98,76	98,76	99,01	99,01	99,17
	6	98,76	99,01	99,17	99,09	99,26

Nous remarquons que pour un nombre de coefficients DCT égale à 24, les taux de reconnaissance monolocuteur dépassent 99 %. Le plus haut taux atteint (99,83%) est obtenu avec un nombre de segments égal à 5 et un rapport de recouvrement égal à 5.

Nous avons ensuite fixé le nombre de segments à 5 et le rapport de recouvrement à 5 et nous avons varié l'ordre de l'ondelette de Daubechies. Les résultats obtenus sont présentés au tableau III.

Tableau III

Taux de reconnaissance monolocuteur
avec différents ordres d'ondelettes

Ondelettes	DCT				
	13	15	18	20	24
db2	98,60	98,68	98,84	99,01	99,42
db4	99,50	99,34	99,34	99,67	99,83
db6	99,42	99,50	99,42	99,67	99,67
db8	99,42	99,50	99,34	99,42	99,59
db20	99,50	99,50	99,67	99,67	99,67

Nous constatons (Tableau III) que le taux de reconnaissance dépasse 98.50 % et dans tout les cas ce même taux dépasse les 99 % pour un nombre de coefficients DCT allant de 20 à 24.

Pour la reconnaissance multilocuteur, nous avons utilisé les chiffres prononcés des 30 premiers locuteurs pour la création du dictionnaire de référence et les chiffres des 25 locuteurs restant pour la reconnaissance. Nous avons fixé le nombre de segments utilisés et le rapport de recouvrement à 5 et nous avons varié l'ordre d'ondelettes. Le nombre de coefficients DCT utilisés est égal à 24. Pour la création du dictionnaire de référence, nous avons fixé la valeur du facteur flou m à 1,2. Nous avons obtenu les résultats avec différents nombres de classes de modèle qui sont donnés par le tableau IV:

Tableau IV

Reconnaissance multilocuteur avec 24 coefficients DCT

Ondelettes	Nombre de classes					
	2	3	4	5	7	10
db2	98	98,36	97,45	97,64	97,27	97,27
db4	97,09	97,09	97,64	97,64	97,64	97,82
db6	97,27	97,64	96,73	95,64	96,18	96,36
db8	97,09	97,64	97,45	97,27	97,45	96,18
db20	97,09	96,73	95,64	95,09	96,18	95,09

Nous remarquons (Tableau IV) que pour un nombre de modèles (classes) de chiffres égale à deux, le taux de reconnaissance est toujours supérieur à 97% et ce quelque soit l'ordre de l'ondelette.

Nous avons répété les simulations du tableau IV en utilisant la distance euclidienne pondérée pour la comparaison. Cette distance est donnée par la formule suivante:

$$d = [\sum_{n=1}^{p} n\,(c_r(n) - c_d(n))^2]^{1/2} \qquad (4.13)$$

Les résultats avec pondération de la distance euclidienne, sont donnés par le tableau V ci-dessous:

Tableau V

Reconnaissance multilocuteur avec pondération

Ondelettes	Nombre de classes					
	2	3	4	5	7	10
db2	98,73	98,55	98	97,82	97,64	98,36
db4	98,55	97,64	98,36	97,82	98	97,82
db6	99,27	98	97,27	97,64	97,09	97,09
db8	97,82	98,91	97,82	97,64	97,45	97,09
db20	98,55	97,45	97,82	97,27	96,91	96,18

Nous remarquons que les résultats de la reconnaissance avec pondération sont meilleurs que ceux obtenus sans la pondération. Nous constatons aussi qu'en général le taux de reconnaissance est meilleur avec un nombre de classe égale à deux et que le taux de reconnaissance maximal atteint est égal à 99.27 % pour l'ondelette de Daubechies db6.

Pour les simulations suivantes, nous avons transformé la fréquence d'échantillonnage de la base de donnés de 20 KHz à 8 KHz pour se conformer à la fréquence d'échantillonnage de la carte DSK6711 pour l'acquisition du signal. La transformation est effectuée par une opération d'interpolation par 2 suivis par une décimation par 5. Nous avons répété les simulations du tableau IV avec seulement un nombre de classes égale à 2 et nous avons obtenu les résultats suivants:

Tableau VI

Reconnaissance multilocuteur
avec la base de données
échantillonnée à 8 KHz
(isolation du chiffre manuelle)

Ondelettes	taux de reconnaissance
db2	95,82
db4	97,09
db6	98,18
db8	97,82
db20	98,73

Nous remarquons qu'en réduisant la fréquence d'échantillonnage de 20 KHz à 8 Khz, le signal de la parole perd de sa qualité, ce qui affecte sur les résultats des taux de reconnaissance

Nous avons répété les simulations du tableau VI avec une isolation automatique du chiffre du silence et nous avon obtenu les résultats qui sont donnés par le tableau VII:

Tableau VII

Reconnaissance multilocuteur
avec isolation
automatique du chiffre

ondelette	taux de reconnaissance
db2	94.18
db4	95.45
db6	96.00
db8	97.27
db20	98.18

Nous constatons une diminution des taux de reconnaissance par rapport à la méthode d'isolation du chiffre de façon manuelle. Le taux maximal atteint est de 98.18 % pour l'ondelette db20.

Afin de tester la méthode de reconnaissance proposée sur des chiffres bruités, nous avons ajouté au système un bloc de débruitage qui utilise aussi les ondelettes. La décomposition en paquets d'ondelettes s'effectue jusqu'au niveau 3 avec l'ondelette de daubechies db6. Pour calculer le seuil, nous avons utilisé la formule (2.27) et l'estimation de σ est obtenu avec les coefficients de détails de la haute résolution du niveau 3 (d_3^8).

Pour le seuillage, nous avons choisi les méthodes de seuillage mou "soft", dur "hard" et dur modifier "modified hard" [25]. Dans le cas des deux premières méthodes, nous avons effectué des modifications afin de l'adapter à notre système de reconnaissance.

Pour les deux méthodes mou et dur, nous avons remarqué que choisir les valeurs des coefficients $d_{jk}=0$ dans le cas où $|d_{jk}|\leq T$ cause des problèmes pour détecter le début et la fin du mot. En effet, la mise à zéros des coefficients donne des valeurs nulles au début du signal enregistré ce qui est un inconvénient pour calculer les statistiques du silence nécessaires pour la méthode de détection du début et la fin du mot. Pour remédier à ce problème, au lieu de mettre les coefficients à zéro en les divise par 20 c'est-à-dire $d_{jk}=d_{jk}/20$.

Pour la méthode du seuillage mou, la soustraction de la valeur du seuil T des coefficients de la décomposition en ondelettes $|d_{jk}|-T$ dans le cas où $|d_{jk}|>T$ cause aussi des problèmes de détection du début et la fin du mot. En effet, la valeur de l'énergie ITU=4*ITL nécessaire pour la détection du début ou de la fin du mot ne sera jamais atteint pour certains cas. Pour remédier à ce problème, on soustrait seulement 40% de la valeur du seuil.

Les modifications proposées sur les méthodes de seuillage mou et seuillage dur sont exprimées par les équations suivantes :

- Seuillage mou avec modification proposée

$$\text{Seuillage}_{\text{soft}}=\begin{cases}\text{sign}(d_{jk})\,(|d_{jk}|-0.4*T) & |d_{jk}|>T\\ d_{jk}/20 & |d_{jk}|\leq T\end{cases} \tag{4.14}$$

- Seuillage dur avec modification proposée

$$\text{Seuillage}_{\text{hard}}=\begin{cases}d_{jk} & |d_{jk}|>T\\ d_{jk}/20 & |d_{jk}|\leq T\end{cases} \tag{4.15}$$

Afin de tester notre système de reconnaissance avec le débruitage, nous avons ajouté du bruit blanc gaussien (AGWN) aux chiffres à reconnaître. L'ondelette db20, avec celle que nous avons obtenu le meilleur taux de reconnaissance (Tableau VII), a été utilisé pour la décomposition afin d'obtenir les coefficients de la reconnaissance. Les résultats de simulation avec différentes méthodes de seuillages mou (4.14), dur (4.15) et dur modifié (2.30) [25] sont donnés par le tableau VIII.

Tableau VIII

Taux de reconnaissance multilocuteur
avec différentes méthodes de seuillages

Bruit	modifié μ=255	hard	soft
20 db	96,18	95,09	95,09
15 db	93,64	91,45	90,73
10 db	85,27	79,82	77,27
5 db	69,82	60,18	57,64
0 db	51,64	42,36	43,64

Nous remarquons que la méthode de débruitage qui utilise le seuillage dur modifié avec la loi de μ, proposée par Chang et al. [25], donne des meilleurs résultats par rapport aux méthodes hard et soft avec modifications que nous avons proposées.

La figure 41 illustre les résultats de simulation du tableau VIII.

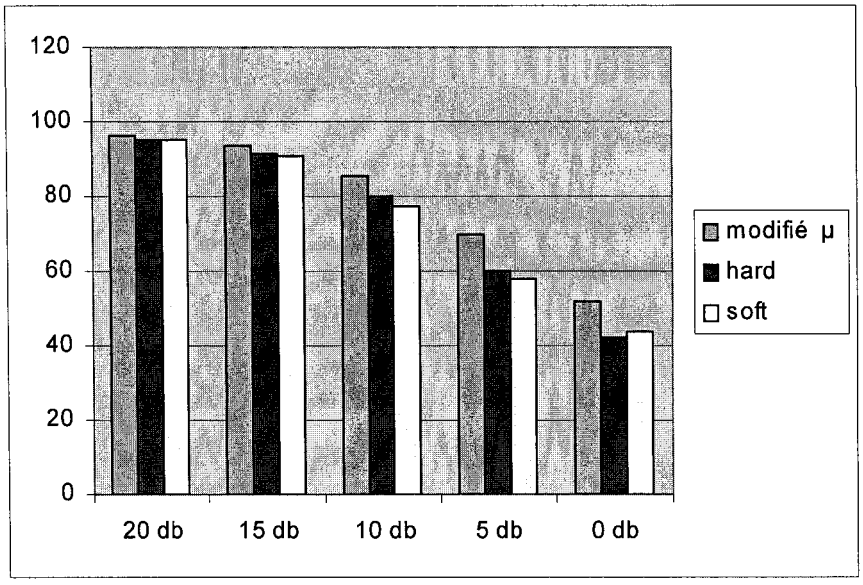

Figure 41 Résultats de reconnaissance avec le bruit

4.9 Implémentation sur DSP

Après avoir effectué les simulations à l'aide du logiciel MATLAB®, nous avons réalisé notre système de reconnaissance des chiffres isolés sur le DSP. Les paramètres choisis sont ceux qui ont donné en simulation le meilleur taux de reconnaissance. C'est à dire, qu'on a utilisé l'ondelette de Daubechies d'ordre 20 (db20) pour obtenir les coefficients de reconnaissance et un nombre de segments égal à 5 avec un rapport de recouvrement égal à 5 aussi. Pour le débruitage, nous avons utilisé la méthode de seuillage hard modifié avec μ =255 et nous avons utilisé l'ondelette de Daubechie (db6) pour la décomposition et la reconstruction. Le programme est écrit en langage C. les paramètres des filtres des ondelettes db20 et db6 sont obtenus avec Matlab®. Le chargement du signal de la parole et du dictionnaire de référence avec la fonction "fread" prend beaucoup de temps. Pour remédier a ce problème, on a utilisé la sonde Probe Point pour le chargement rapide des données dans la mémoire. Pour cela, le programme a été modifié en ajoutant la fonction dataIO( ) en deux fois. L'une pour le chiffre à

reconnaître et l'autre pour le dictionnaire de référence. Avant la compilation du programme, il faut placer un point d'arrêt sur chacun des deux fonctions dataIO( ) en utilisant le bouton Toggle Probe Point. Il faut ensuite utiliser la fenêtre File I/O obtenue avec la commande (File → File I/O) pour le chargement du chiffre isolé à reconnaître et du dictionnaire de référence. Durant le chargement, chaque donnée est affectée au pointeur spécifique dans le programme avec la sonde Probe Points. La figure suivante illustre la fenêtre de chargement des données.

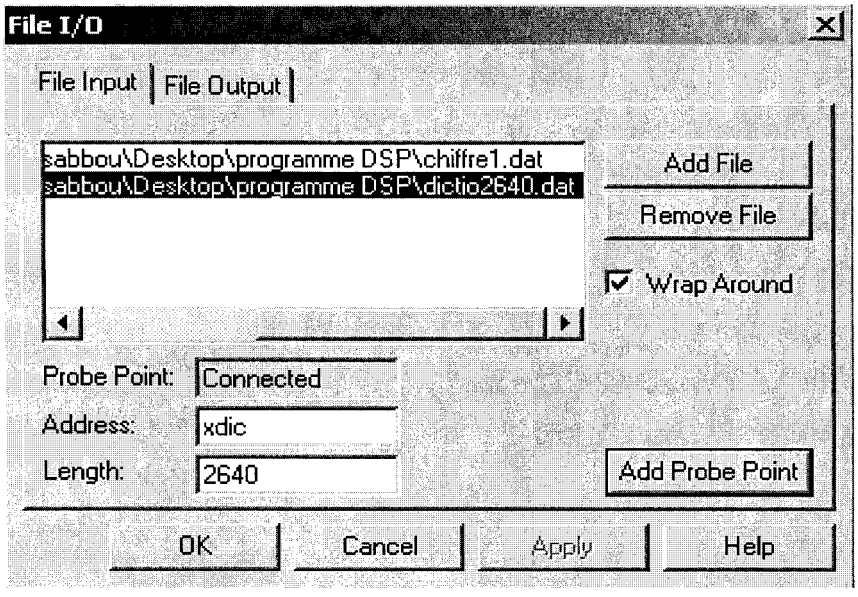

Figure 42 Sonde probe point

Le CCS permet aussi de visualiser le signal de la parole chargé sur un graphe. Pour cela, il faut configurer les propriétés de la fenêtre obtenue avec la commande (View → Graph → Time/Frequency).

Les différentes étapes de la méthode programmée sur le dsp sont illustrées par la figure 43.

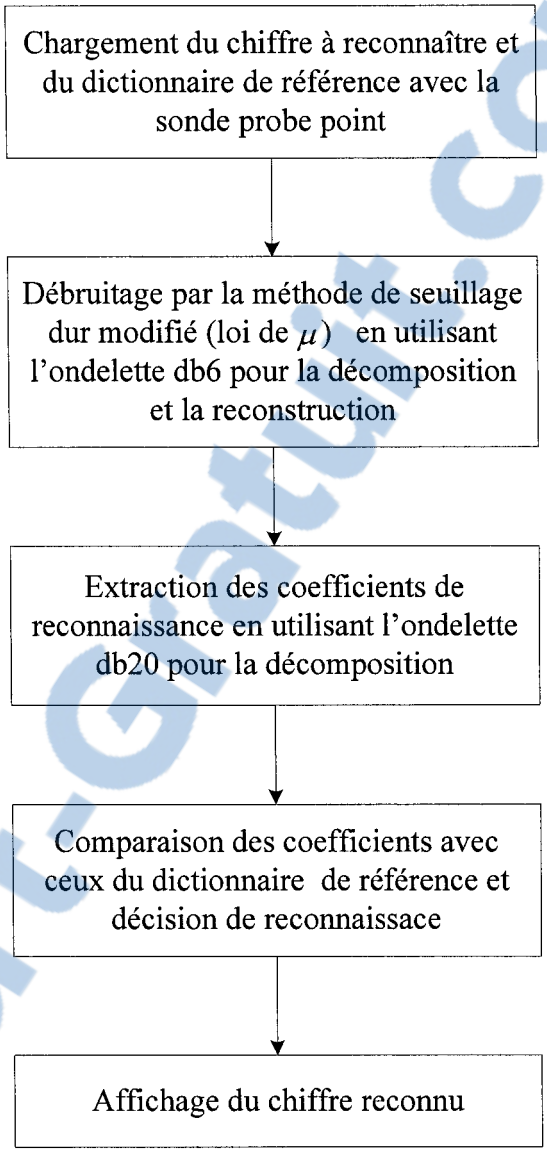

Figure 43 Les étapes du système de reconnaissance implémentées sur le DSP

La figure suivante illustre le résultat de la reconnaissance du chiffre isolé "One" implémenté sur DSP.

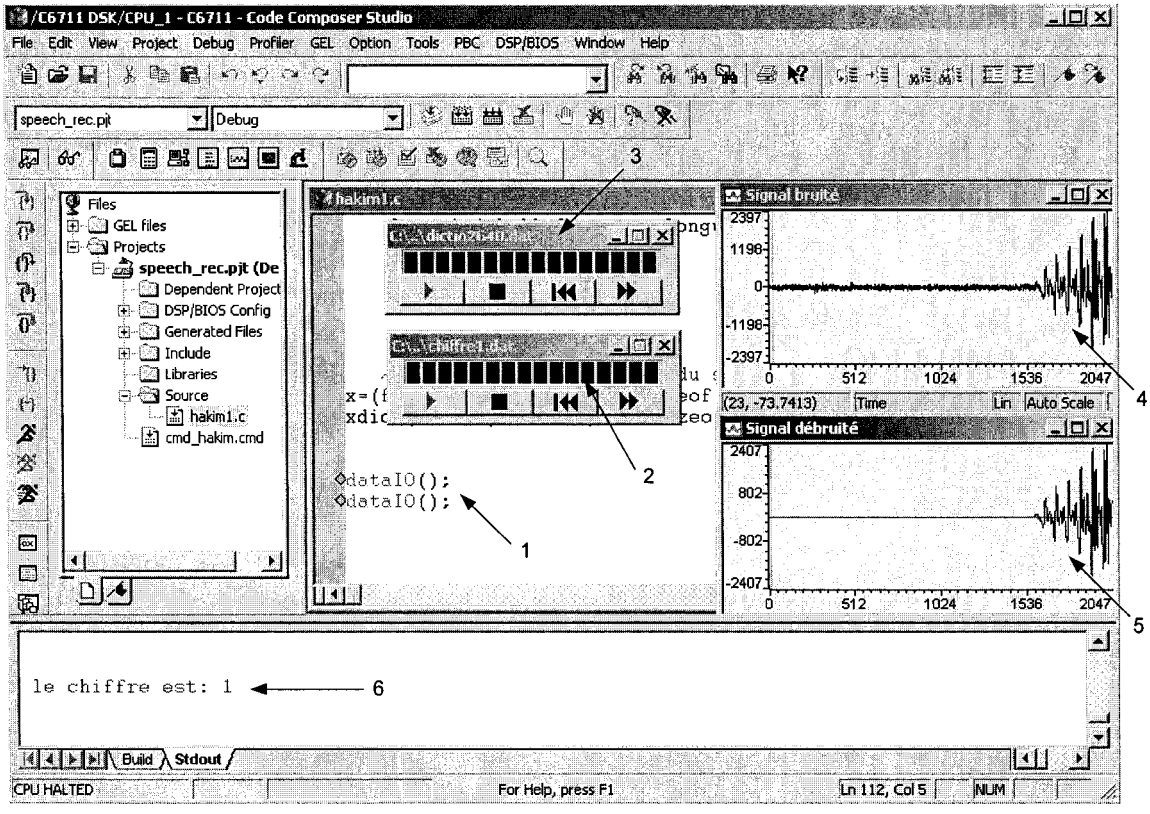

Figure 44 Reconnaissance du chiffre "one" avec DSP

Les éléments numérotés de la figure 44 sont décrits ci-dessous :

1. les fonctions dataIO( ) avec point d'arrêt nécessaires pour le chargement du chiffre à reconnaître et du dictionnaire de référence avec la sonde probe point.
2. barre d'état du chargement du chiffre à reconnaître.
3. barre d'état du chargement du dictionnaire de référence.
4. visualisation du signal du chiffre à reconnaître bruité.
5. visualisation du signal du chiffre à reconnaître débruité.
6. affichage du chiffre reconnu.

4.10 Conclusion

Dans ce chapitre, nous avons présenté les différentes étapes de réalisation d'un système de reconnaissance des chiffres isolés. Les différents blocs du système ont été programmés avec le logiciel MATLAB®. Nous avons testé la méthode et analysé les résultats des simulations, obtenues avec différents ordres d'ondelettes. Nous avons obtenu des résultats intéressant qui peuvent être améliorés.

Dans un premier temps, on a exploré l'effet du nombre de segments et le rapport de recouvrement sur la qualité de la reconnaissance. Il a été constaté qu'un nombre de segments égal à cinq et qu'un rapport de recouvrement égal aussi à cinq représentent des valeurs optimales en terme de taux de reconnaissance (Tableau II). Dans un deuxième temps, nous avons vérifié l'effet de l'ordre de l'ondelette de daubechies sur le taux de reconnaissance. Les résultats obtenus, ont révélé qu'un ordre d'ondelette égal à vingt, donne les meilleurs performances en terme de taux de reconnaissance multilocuteur (Tableau V). En ajoutant un bloc de débruitage avec les ondelettes, nous avons aussi obtenu de bons résultats de reconnaissance sur des chiffres bruités avec un bruit blanc gaussien (Tableau VIII). Les résultats obtenus à l'aide des différentes méthodes proposés dans la littérature, sont difficiles à comparer avec ceux que nous avons obtenu. La raison est qu'ils utilisent différents types de bases de données et différents nombres de locuteurs de tests. En comparant d'une façon générale, nous avons obtenu de bons résultats. Une fois ces paramètres réglés, les algorithmes ont été implémentés sur DSP.

Pour l'implémentation, nous avons utilisé la carte DSK qui contient le processeur DSP TMS320C6711 ainsi que le logiciel Code Composer Studio. L'exécution du programme a été effectuée sans aucun problème avec une rapidité comparable à celle obtenue avec le logiciel MATLAB®.

CONCLUSION

L'objectif de ce mémoire est de réaliser un système de reconnaissance de chiffres isolés rapide et simple à implémenter sur un processeur dédié au traitement des signaux (DSP). En effet De nos jours, de nombreuses applications utilisent les DSP à cause de leurs faibles coûts et de leurs flexibilités. Parmi ces applications, on note la téléphonie à main libre.

Nous avons proposé une méthode simple de subdivision du chiffre isolé en un nombre fixe de segments avec recouvrement, qui donne des segments de tailles différentes d'un chiffre à un autre. Cette méthode de segmentation nous a permis d'utiliser la distance euclidienne durant l'étape de la reconnaissance au lieu de l'algorithme complexe d'alignement temporel DTW. La méthode proposée a permis de réduire considérablement le temps de calcul.

La transformée en ondellettes a été choisie pour l'extraction de paramètres en utilisant la méthode de décomposition en paquets d'ondelettes admissibles selon l'échelle de Mel et qui permet une représentation du signal de la parole sans redondance. L'utilisation de la transformée en ondellettes permet de mieux représenter le signal de la parole. Avec ses fenêtres d'analyses variables, elles permettent la détection des hautes fréquences et des basses fréquences. Elle est aussi utilisée pour le débruitage. L'implémentation de la décomposition en paquet d'ondelettes est rapide à cause de son utilisation de l'algorithme pyramidal.

Les résultats des simulations obtenus sur MATLAB® sont satisfaisants. Nous avons obtenu un pourcentage de reconnaissance multilocuteur de 98.18%, en utilisant l'ondellette de Daubechies d'ordre vingt (db20) dans le cas des chiffres non bruités. En plus, un nombre de segments de la subdivision du chiffre égale à 5 et un nombre de

modèle de chaque chiffre utilisé dans le dictionnaire de référence, qui est obtenu avec l'algorithme de classification floue FCM, égale à 2 ont permis de réduire de l'espace mémoire utilisé. Un bloc de débruitage avec les ondelettes a été aussi ajouté pour donner une reconnaissance robuste des chiffres bruités et nous avons aussi obtenu des résultats satisfaisants. Avec ces caractéristiques, la méthode de la reconnaissance des chiffres isolés avec débruitage a été implémentée sans difficulté sur le DSP TMS320C6711 de Texas instrument.

Les résultats obtenus par l'approche que nous avons proposés peuvent toutefois être améliorés dans le cadre de travaux futurs, en utilisant un dictionnaire de référence conçu avec une base de données de fréquence d'échantillonnage de 8 KHz, en essayant d'autres types d'ondelettes, et en utilisant d'autres méthodes d'isolation du chiffre du silence ainsi que d'autres méthodes de débruitage.

BIBLIOGRAPHIE

[1] R. Boite and M. Kunt, Traitement de la parole, Presses Polytechnique Romandes, Lausanne, 1987.

[2] C. Rowden, Speech processing, McGRAW-HILL Book Company Europe, London, 1992.

[3] J. C. Junqua and J. P. Haton, Robustness in automatic speech recognition, Kluwer Academic Publisher, Massachusetts, 1996.

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

[5] L. Rabiner and B. H. Juang, Fundamentals of speech recognition, Prentice Hall, New Jersey, 1993.

[6] N. Badri, Utilisation de la transformée de Fourier et de la transformée en ondelettes pour la reconnissace du locuteur, ÉTS, Montréal (Qc), 2002.

[7] S. Davis and P. Mermelstein, " Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," IEEE Transaction On Acoustics, Speech and Signal Processing, vol28, pp. 357-366, August 1980.

[8] J. W. Picone, "Signal Modeling Techniques in Speech Recognition," Proceeding of the IEEE, vol. 81, pp. 1215-1247, 1993.

[9] H. Sakoe and S. Chiba, "Dynamic Programming Algorithm Optimization for Spoken Word Recognition," IEEE Transaction On Acoustics, Speech and Signal Processing, vol. 26, pp. 43-49, February 1978.

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

[11] A. Belaid and Y. Bealid, Reconnaissace des formes - Méthodes et applications, InterEdition, Paris, 1992.

[12] S. Levinson, L. Rabiner, A. Rosenberg and J. Wilpon, "Interactive clustering techniques for selecting speaker-independent reference templates for isolated

word recognition," IEEE Transaction On Acoustics, Speech and Signal Processing, vol. 27, pp. 134-141, April 1979.

[13] B. B. Hubbard, Ondes et ondelettes, la saga d'un outil mathématique, Pour la Science, diffusion Belin, Paris, 1995.

[14] S. Mallat, A wavelet tour of signal processing, Academic Press, New York 1998.

[15] C. K. Chui, An Introduction to Wavelets, Academic Press, INC, Boston, 1992.

[16] O. Rioul and M. Vetteri, "Wavelets and signal processing," IEEE Signal Processing Magazine, vol. 8, pp. 14-38, 1991.

[17] S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation" IEEE Transaction On Pattern Analysis and Machine Intelligence, vol. 11, pp. 674-693, July 1989.

[18] Y. Meyer, Ondelettes et opérateurs, Hermann, Paris, 1990.

[19] O. Farooq and S. Datta, "Mel filter-like admissible wavelet packet structure for speech recognition," IEEE Signal Processing Letters, vol. 8, pp. 196-198, July 2001.

[20] O. Farooq and S. Datta, "Phoneme recognition using wavelet based feature," Information Sciences, pp. 5-15, 2003.

[21] J. R. Karam, W. J. Phillips and W. Robertson, "New low rate wavelet models for the recognition of single spoken digits," Canadian Conference on Electrical and Computer Engineering, vol.1, pp. 331-334, 2000

[22] O. Farooq and S. Datta, "Robust features for speech recognition based on admissible wavelet packets," Electronics Letters, vol. 37, pp. 1554-1556, 2001.

[23] D. L. Donoho and M. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," Biometrika, vol. 81, pp. 425-455, 1995.

[24] I. M. Johnstone and B. W. Silverman, "Wavelet threshold estimators for data with correlated noise.", J. ROY. Statist. Soc. B, vol. 59, pp. 319-351, 1997.

[25] S. Chang, Y. Kwon, S. Yang, and I. Kim, "Speech enhancement for non-stationary noise environment by adaptive wavelet packet." Acoustics, Speech, and Signal Processing,. (ICASSP '02). IEEE International Conference on, vol. 1, pp. I-561 - I-564, 2002

[26] D. L. Donoho and I. M. Johnstone, "Threshold selection for wavelet shrinkage of noisy data." Proceedings of the 16th Annual International Conference of the IEEE, vol. 1, pp. A24 - A25, 1994.

[27] O. Farooq and S. Datta, "Wavelet-based denoising for robust feature extraction for speech recognition," Electronics Letters, vol. 39, pp. 163 - 165, 2003.

[28] R. Chassaing, DSP applications using C and the TMS320C6x DSK, J. Wiley and Sons, New York, 2002.

[29] M. Pinard, Les DSP, famille ADSP218x, principes et applications, Dunod, Paris, 2000.

[30] S. A. Tretter, Communication system design using DSP algorithms: with laboratory experiments for the TMS320C6701 and TMS320C6711. Kluwer Academic / Plenum Publishers, New York, 2003.

[31] TMS320C6000 Code Composer Studio Tutorial, SPRU301C, Texas Instruments, Dallas, Texas, 2000.

[32] N. Dahnoun, Digital signal processing implementation using the TMS320C6000 DSP platform, Prentice-Hall, London, 2000.

[33] TMS320C6000 Peripherals Reference Guide, SPRU190D, Texas Instruments, Dallas, Texas, 2001.

[34] TMS320C6000 CPU and Instruction Set Reference Guide, SPRU189F, Texas Instruments, Dallas, Texas, 2000.

[35] L. R. Rabiner and M. R. Sumbur, "An Algorithm for determining the endpoints of isolated utterances," The Bell System Technical Journal, vol. 54, pp. 279-315, 1975.

[36] S. H. Abbou, M. Gabrea, and C.S.Gargour, "Extraction of characteristics for the recognition of isolated words using the wavelet packet method," Electrical and Computer Engineering, Canadian Conference, IEEE., vol. 1, pp. 523 - 526, May 2004

[37] S. Chang, Y. Know and S. Yang, "Speech feature extracted from adaptive wavelet for speech recognition," Electronics Letters, vol. 34, pp. 2211-2213, Nov. 1998.

[38] J. C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, New york, 1981.

[39] R. G. Leonard, "A Database for Speaker-independent Digit Recognition," Proc ICASSP 84, vol. 3, 1984.

[40] I. Daubechies, Ten lectures on wavelets, Society for Industrial and Applied Mathematics, Philadelphia, Pa, 1992.

[41] M. Yuan, T. Lee, and P. C. Ching, "Speech recognition on DSP: issues on computational efficiency and performance analysis," Communications, Circuits and Systems, 2005. Proceedings IEEE., vol. 2, pp. 852-856, May. 2005.

[42] G. Hui, K. -C. Ho, and Z. Goh, "A robust speaker-independent speech recognizer on ADSP2181 fixed-point DSP," Signal Processing Proceedings IEEE, vol. 1, pp. 694 - 697, Oct. 1998.

[43] Q. He, J. Liu, and R. Liu, "Discriminative training for discrete HMM of a fixed-point DSP Mandarin digits recognition system," Signal Processing, 2002 6th International Conference on. IEEE, vol. 1, pp. 532 - 535, Aug. 2002.