N° d'ordre : 10/2005-M/EL

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université des Sciences et de la Technologie Houari BOUMEDIENE
FACULTÉ D'ÉLECTRONIQUE & D'INFORMATIQUE

Mémoire

Présenté pour l'obtention du diplôme de MAGISTER

En : ELECTRONIQUE

Spécialité Communication Parlée

Par

M. BOUKHARI Aïssa

Thème

Quantification des Coefficients LSF Application au codage CELP

Soutenu le 06 juillet 2005, devant le jury composé de :

M. HOUACINE	*Amrane*	*Professeur*	*USTHB*	*Président*
Mme. BOUDRAA	*Malika*	*Maître de Conférences*	*USTHB*	*Directeur de Thèse*
Mme. TOUHAMI	*Rachida*	*Maître de Conférences*	*USTHB*	*Examinateur*
M. HADDAD	*Boualem*	*Maître de Conférences*	*USTHB*	*Examinateur*
M. TEFFAHI	*Hocine*	*Docteur d'état*	*USTHB*	*Examinateur*

A mes parents

Remerciements

Louange et amples remerciements au Dieu le tout puissant

Ce travail a été réalisé au sein du laboratoire de communication parlée et de traitement du signal de la faculté d'électronique et d'informatique (LCPTS).

J'adresse mes vifs remerciements à Monsieur HOUACINE, professeur à l'USTHB, pour l'honneur qu'il me fait en acceptant de présider le jury.

J'exprime ma profonde reconnaissance à Madame BOUDRAA, maître de conférences à l'USTHB, pour les nombreux conseils qu'elle n'a cessé de me prodiguer et qui m'ont été très précieux tout au long de la réalisation de ce travail.

Je remercie vivement Madame TOUHAMI et Monsieur HADDAD, maîtres de conférences à l'USTHB, ainsi que Monsieur TEFFAHI docteur à la faculté d'électronique et d'informatique, d'avoir accepté de faire partie du jury.

Je remercie également Monsieur DJAMAH Mouloud du laboratoire LCPTS pour son aide et ses conseils, ainsi que Monsieur BOUDRAA pour sa contribution à la correction de ce mémoire.

Mes recherches ont été amplement facilitées par l'ambiance d'amitié et de coopération, je serais gré à tous mes collègues du laboratoire de communication parlée et de traitement du signal.

Toute ma reconnaissance aux membres de ma famille pour l'assistance qu'elle a su m'apporter tout au long de ce travail.

Abréviations

CELP	Code Excited Linear Prediction¨
FS1016	Federal Standard 1016
GLA	Generalized Lloyd Algorithm
HSVQ/MSVQ	Quantification vectorielle hybride SVQ et MSVQ
LSF	Line spectrum frequencies (ligne spectrale de fréquences)
MCMSVQ	Multi codes books MSVQ
MCSVQ	Multi codes books SVQ
MELP	Mixed Excited Linear Prediction
MSVQ	Multistage vector quantization (quantification vectorielle à multi étages)
QV	Quantification vectorielle
SD	Distorsion spectrale
SVQ	Split vector quantization (quantification vectorielle divisée)

Sommaire

Chapitre III
La quantification vectorielle

Chapitre IV
Mise en œuvre de cinq méthodes de Quantification vectorielle des coefficients LSF

Liste des figures

Liste des tableaux

Introduction

Le moyen de communication que l'homme utilise le plus est bien la parole; il est simple, et les développements des communications lui ont donné un aspect privilégié. L'usage du téléphone aussi bien le fixe que le mobile de dernière génération en est une bonne illustration. Ainsi, pour permettre à un grand nombre d'usagers de communiquer, il faut que les équipements qui véhiculent ces énormes flux d'informations ne soient pas trop encombrés et saturés, d'où l'utilité de compresser l'information avant de la transmettre.

En effet, dans les premiers systèmes de téléphonie numérique, la parole est numérisée à 64 Kbps. De nombreux algorithmes ont été proposés pour diminuer ce débit tout en essayant de conserver une qualité subjective donnée en fonction des exigences de l'application à laquelle le codeur est destiné. Un système de codage de la parole comprend deux parties : le codeur et le décodeur. Le codeur analyse le signal pour en extraire un nombre réduit de paramètres pertinents qui sont représentés par un nombre restreint de bits pour l'archivage ou la transmission. Le décodeur utilise ces paramètres pour reconstruire un signal de parole synthétique.

La plupart des algorithmes de codage mettent à profit un modèle linéaire simple de production de parole. Le signal de parole n'étant pas stationnaire, les codeurs le découpent généralement en trames quasi-stationnaires de durée entre 5 et 30 ms pour extraire les paramètres représentant l'enveloppe spectrale du signal. Les paramètres souvent utilisés sont les paires de raies spectrales ou coefficients LSF (Line Spectral Frequencies) qui sont déduites des coefficients de prédiction linéaire et qui possèdent de bonnes propriétés pour la quantification.

Le standard américain FS1016 est un codeur de parole à 4800 bits/s, basé sur la technique CELP (Code Excited Linear Prediction). Ce standard a été développé par le département de la défense des Etats Unis d'Amérique (DoD) et le laboratoire AT&T Bell. Dans ce système de transmission, 10 coefficients LSF représentant les coefficients de prédiction sont codés sur 34 bits pour chaque trame de parole de 30 ms. Ce codage est effectué après une quantification scalaire.

L'objet de ce travail est d'améliorer ce codeur en proposant de nouvelles méthodes de quantification plus efficaces et ceci dans le but de réduire le débit alloué aux coefficients LSF. Notre étude a porté sur une méthode de quantification en treillis et plusieurs méthodes de quantification vectorielle.

Dans une première étape nous avons entrepris une étude bibliographique orientée vers les travaux les plus récents traitant de la quantification des coefficients LSF. Ceux-ci sont actuellement les plus utilisés dans les codeurs bas débits et qui sont l'un des centres d'intérêt de recherche de notre laboratoire de codage de la parole.

Dans une deuxième étape, nous avons effectué une analyse fine sur les combinaisons des dictionnaires de la quantification scalaire utilisée dans la norme FS1016. Cette analyse a montré qu'il y'avait un nombre très important de vecteurs LSF qui rendaient le filtre de synthèse instable. Aussi, nous avons proposé une méthode d'énumération utilisant un système en treillis et réduisant considérablement le nombre de bits nécessaires (30 bits au lieu de 34 bits), sans changer les dictionnaires de la quantification scalaire suscitée.

Dans une troisième étape, cinq méthodes de la quantification vectorielle ont été mises au point. Il s'agit de la quantification divisée SVQ (Split Vector Quantization), la quantification vectorielle à multi-étages MSVQ (Multi Stage Vector Qauntization), la qualification vectorielle hybride H-SVQ/MSVQ (Hybrid SVQ/MSVQ) ainsi que la technique des multi-dictionnaires appliquée au SVQ et MSVQ (MCSVQ : Multi code books SVQ et MCMSVQ : Multi code books MSVQ). Les trois dernières méthodes sont les variantes que nous proposons en vue de diminuer la distorsion induite par les deux premières. A cet effet, une étude comparative entre les différentes méthodes a été élaborée. Pour cela, deux bases de données références : américaine TIMIT (institut de technologie de Massachusetts"MIT", institut de recherche de Stanford "SRI" et Texas Instruments "TI") et arabe standard PAPE (Phrases Arabes Phonétiquement Equilibrées) ont été utilisées lors de l'apprentissage et le test. Avec TIMIT, nos tests ont porté aussi bien sur des locuteurs masculins que féminins issus de huit régions différentes.

Ainsi notre mémoire se présentera comme suit:

Dans le premier chapitre on exposera le principe du codeur CELP et on présentera les propriétés des coefficients LSF. La méthode de Kabal pour l'extraction des coefficients a été utilisée.

Au deuxième chapitre, une conception d'un système d'énumération en treillis sera exposée. On montrera notre contribution par celui-ci à la diminution du nombre de bits nécessaires au codage des indices de la quantification scalaire de la norme FS1016.

Le troisième chapitre sera consacré aux différentes méthodologies de conception de la quantification vectorielle.

Le chapitre quatre sera consacré à cinq méthodes de quantification vectorielle que nous avons élaborées et testées avec une discussion sur les résultats obtenus. Une comparaison sera faite entre ces méthodes. En annexe de ce chapitre, nous donnerons les résultats obtenus, aussi bien pour les locuteurs masculins que les locuteurs féminins pour la base de données arabe et pour les cinq méthodes.

Enfin nous donnons une conclusion générale sur le travail accompli ainsi que les perspectives futures qui peuvent l'enrichir.

Chapitre I

Le standard FS1016 et le calcul des coefficients LSF

Les techniques de codage à bas débit tendent à réduire le débit d'information, lorsqu'on transmet ou lorsqu'on mémorise le signal de parole. Actuellement la technique la plus utilisée est le codage par prédiction linéaire à excitation par codes ou CELP (Code Exited Linear Prediction)[1,2], où l'on peut atteindre des débits de transmission aussi bas que 4 800 bits/s.

Le codeur CELP (basé sur le standard FS1016) utilisé dans notre étude a été développé au sein du laboratoire de communication parlée et de traitement du signal (LCPTS) de l'université des sciences et de technologie Houari Boumediene (USTHB).

I.1 La prédiction linéaire

La prédiction linéaire est l'un des plus important outil de l'analyse de la parole. Cela exploite l'avantage de la redondance dans le signal de parole. Cette redondance permet de dire que les échantillons de ce signal sont corrélés. Ceux-ci peuvent donc être estimés en fonction des échantillons précédents avec une combinaison linéaire comme suit : [4]

$$\hat{s}(n) = -\sum_{k=1}^{P} a_k s(n-k) \qquad \text{(I.1)}$$

a_k sont les coefficients de prédiction et $\hat{s}(n)$ est l'échantillon prédit à partir des k échantillons précédents $s(n-k)$, et P est l'ordre de prédiction. On appelle résiduel la différence entre le signal et sa prédiction:

$$r(n) = s(n) - \hat{s}(n) \tag{I.2}$$

On peut donc en posant $a_0 = 1$ définir un filtre dit filtre LPC qui soit tel que :

$$r(n) = \sum_{k=0}^{P} a_k s(n-k) \tag{I.3}$$

Une transformée en z de l'équation (I.3) donne :

$$R(z) = S(z) + S(z)\sum_{k=1}^{p} a_k z^{-k} \tag{I.4}$$

D'où le modèle de filtrage suivant :

$$S(z) = \frac{R(z)}{A(z)} \tag{I.5}$$

Avec
$$A(z) = 1 + \sum_{k=1}^{p} a_k z^{-k} \tag{I.6}$$

$A(z)$ est appelé le filtre inverse. On arrive ainsi à interpréter la prédiction linéaire comme un processus de filtrage sur le signal d'erreur (résiduel) de prédiction (figure I.1).

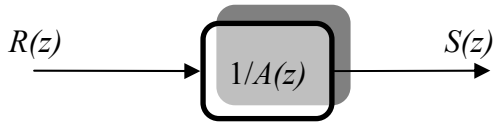


Figure I.1 Model de filtrage

L'interprétation de la prédiction linéaire comme un filtrage revient à substituer la fonction de transfert exacte de conduit vocal par la fonction de transfert $H(z) = \frac{1}{A(z)}$.

Le problème consiste maintenant à la détermination d'une façon optimale des coefficients a_k du filtre *H(z)* d'une manière à approcher au mieux le signal de parole original. Pour des raisons de simplicité, le critère d'optimisation utilisé est la minimisation de l'erreur quadratique moyenne *E*.

$$E = \sum_{n=-\infty}^{+\infty} r(n)^2 = \sum_{n=-\infty}^{+\infty} \left(s(n) + \sum_{k=1}^{p} a_k s(n-k) \right)^2 \qquad \text{(I.7)}$$

Les coefficients a_k sont calculés en minimisant *E* (figure I.2). Il y'a deux méthodes pour l'estimation des coefficients de prédiction [4].

- La méthode d'auto corrélation
- La méthode de covariance

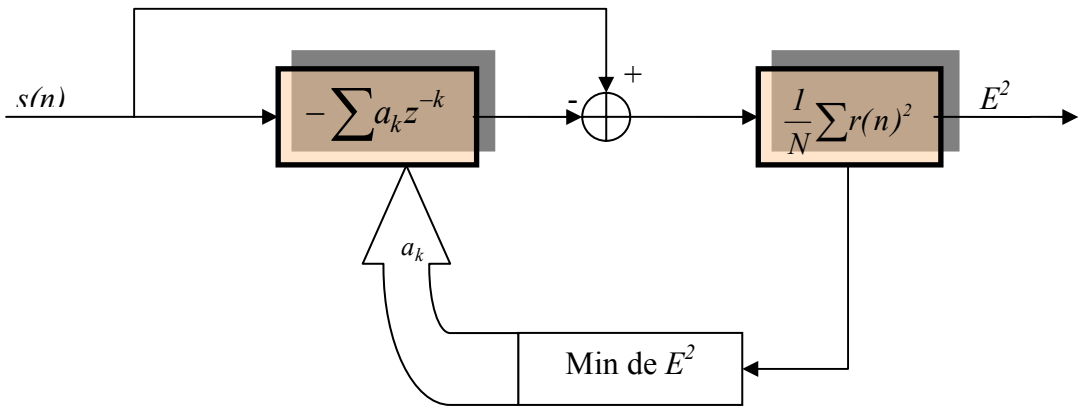


Figure I.2 Schéma fonctionnel de la prédiction linéaire

I.2 Principe de l'analyse par synthèse

La figure I.3 montre le schéma de principe de la technique d'analyse par synthèse où le signal de parole est estimé pendant la phase d'analyse. Dans ce genre de codage, le signal de parole original est analysé avec la prédiction linéaire, qui s'effectue à chaque trame pour produire les P coefficients du filtre de synthèse. Un générateur d'excitation produit à chaque sous-trame entre 40 à 80 échantillons qui excitent ce filtre pour générer un signal de parole synthétique $\hat{s}(n)$. Ce signal résultat sera lui-même comparé au signal original afin de donner le signal résiduel (ou erreur de prédiction) $r(n)$. Enfin, le signal d'excitation qui donne le minimum d'énergie pour cette erreur sera sélectionné pour être transmis au décodeur qui utilisera le même filtre LPC pour la reconstruction du signal parole.

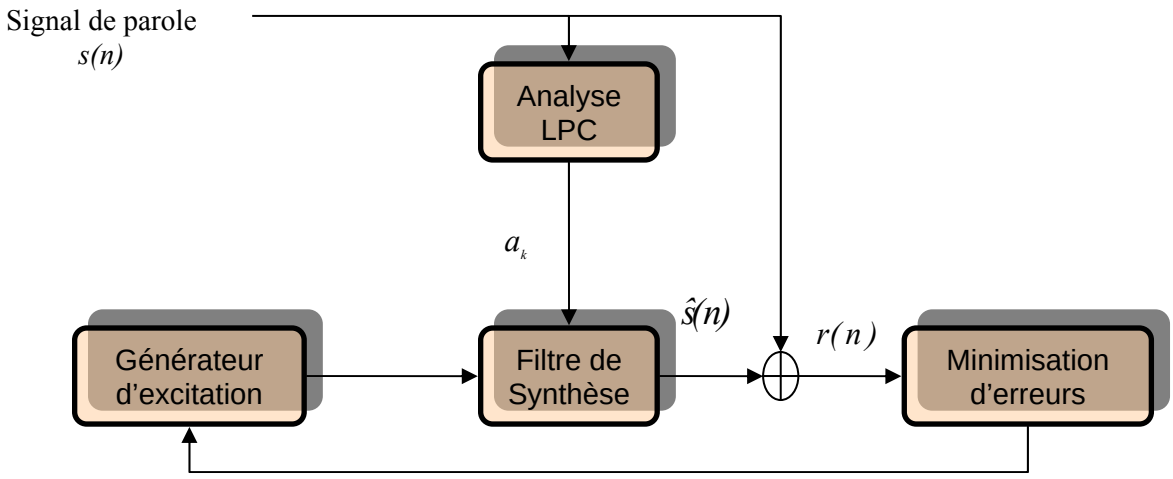


Figure I.3 Principe de l'Analyse par Synthèse

I.3 Le codeur CELP ou standard FS1016

Le codage CELP est une technique de traitement par blocs (trames) qui divise le signal de parole échantillonné en trames d'échantillons. Pour chaque trame une analyse par prédiction linéaire est effectuée pour extraire les coefficients de prédiction. Chaque trame est divisée en sous-trames de 5 à 60 échantillons. Pour chaque sous-trames une excitation est

recherchée à travers une combinaison de deux dictionnaires : un dictionnaire adaptatif et un dictionnaire stochastique [6,7].

I.3.1 Principe du FS1016

FS1016 utilise une fréquence d'échantillonnage de 8 KHZ et des trames de 30 ms divisé en quatre sous-trame de 7.5 ms. L'encodeur CELP basé sur le FS1016 est donné à la figure I.4.

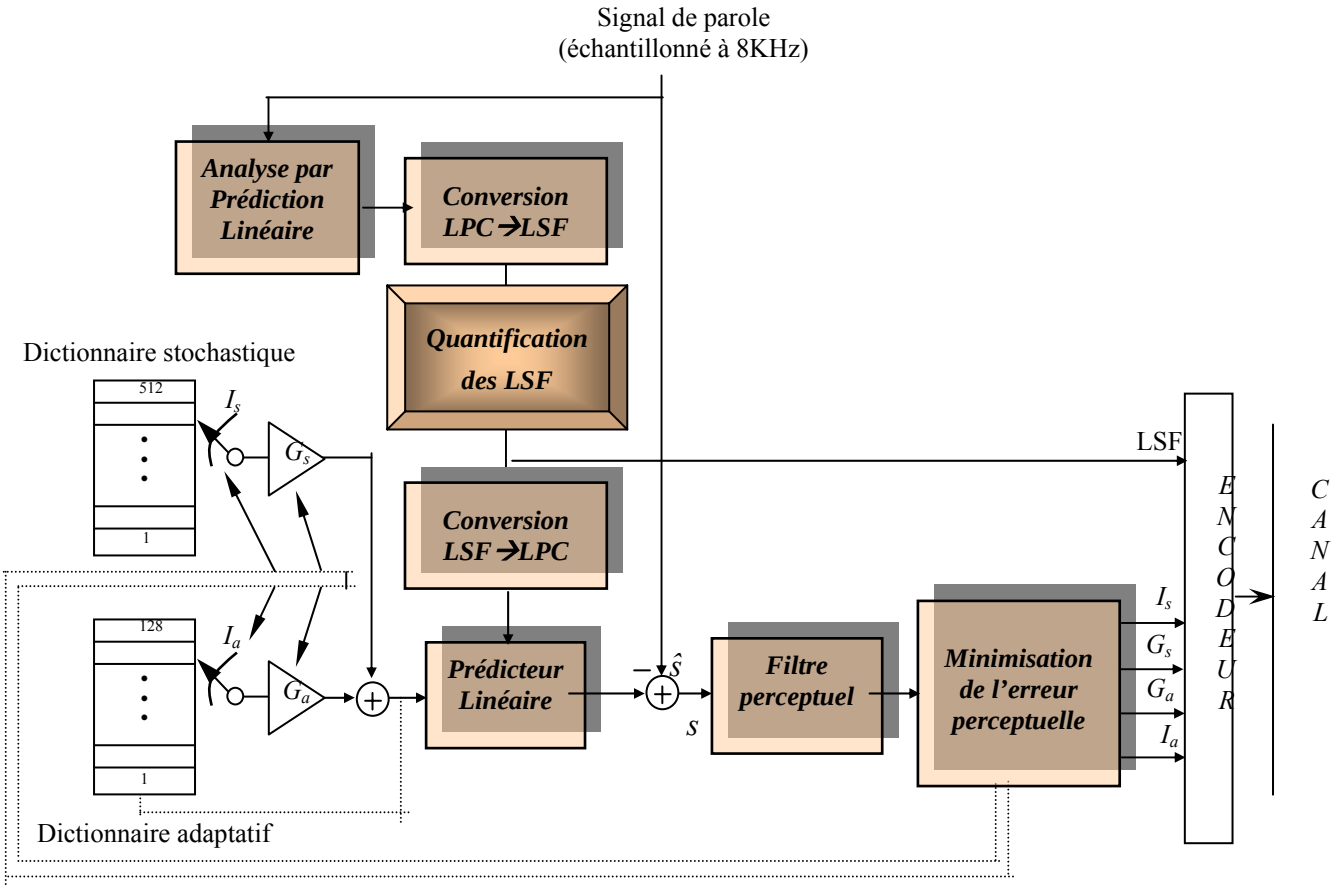


Figure I.4 Schéma de l'encodeur CELP

Le signal synthétique $\hat{s}$ est soustrait du signal original s et la différence est passée à travers un filtre perceptuel. On remarque dans cette figure la présence de deux dictionnaires :

1- Dictionnaire adaptatif: Il est composé de vecteurs-codes numérotés de 0 à La-1. Ces vecteurs représentent les excitations passées. Ce dictionnaire modélise la périodicité long-terme du signal c'est-à-dire le Pitch. Il est actualisé à chaque fois qu'on passe d'une sous-trame à l'autre.

2- Dictionnaire stochastique (fixe): Il est composé de vecteurs-codes numérotés de 0 à Ls-1. Ces vecteurs-codes sont générés d'une façon aléatoire, utilisant une distribution gaussienne. Ils modélisent le signal « résiduel de prédiction » auquel la structure périodique due au Pitch est soustraite.

L'excitation $\hat{r}(n)$ est calculée par sous-trame. Elle est le résultat de l'addition de deux excitations élémentaires: la première est un vecteur-code extrait du dictionnaire adaptatif à l'indice I_a et pondéré par un gain G_a. La deuxième est un vecteur-code extrait du dictionnaire stochastique à l'indice I_s et pondéré par un gain G_s.

Chaque excitation $\hat{r}(n)$ passe à travers le filtre de synthèse $1/A(z)$. Le signal synthétique $\hat{s}$ produit est comparé au signal de parole original $s(n)$. Le signal d'erreur $e(n)$ résultant de cette comparaison est minimisé selon un critère perceptuel afin de ne retenir que l'excitation optimale; c'est-à-dire celle qui produit un signal synthétique le plus proche du signal original (figure I.4). Cette procédure de recherche est appelée « analyse par synthèse »[8,9]. Les coefficients a_i du filtre synthèse sont convertis en coefficients LSF (Line Spectral Frequencies) qui sont plus adaptés à la transmission. Les coefficients LSF seront ensuite codés sur 34 bits (Tableau. I.1).

Une fois l'analyse terminée l'excitation optimale $\hat{r}(n)$ est injectée dans le dictionnaire adaptatif pour actualiser ce dernier et on transmet sur le canal :

1) Les paramètres a_k du filtre de synthèse issu de l'analyse par prédiction linéaire du signal d'entrée
2) Le couple (I_a, G_a) respectivement indice et gain optimal correspondant au dictionnaire adaptatif
3) Le couple (I_s, G_s) respectivement indice et gain optimal correspondant au dictionnaire stochastique.

Tableau I.1: Caractéristique du codeur CELP basé sur le FS1016

	Prédiction linéaire	Dictionnaire adaptatif	Dictionnaire stochastique
Mise à jour	30 ms (240 échantillons)	30/4 = 7,5 ms (60 échantillons)	Fixe
Paramètres	10 coefficients	1 gain, 1 retard (128 vecteurs)	1 gain, 1 indice (512 vecteurs)
Bit /trame	34 bits [3 4 4 4 4 3 3 3 3 3]	Indice : 7 × 4 Gains : 5 × 4	Indice : 9 × 4 Gains : 5 × 4
Débit	1.13 Kbps	1,60 Kbps	1,87 Kbps

I.3.2 Recherche dans le dictionnaire adaptatif

Elle est effectuée en boucle fermée en utilisant le critère de la minimisation de l'erreur quadratique moyenne du signal d'erreur perceptuelle. Le dictionnaire adaptatif contient 128 vecteurs (de 60 échantillons) correspondant à 128 retards [9].

L'excitation globale (résultat de l'addition de l'excitation issue du dictionnaire adaptatif et de l'excitation issue du dictionnaire stochastique) de la sous-trame courante est mise dans le dictionnaire adaptatif aux indices 0 à 59. En effet, le passage d'une sous trame à l'autre entraîne la réactualisation du dictionnaire adaptatif. A cette étape on détermine pour chaque sous-trame l'indice I_a du vecteur issu du dictionnaire adaptatif. Ainsi le gain et cet index du dictionnaire adaptatif sont transmis 4 fois par trame. Le gain est codé entre –1 et +2 utilisant une quantification scalaire non uniforme sur 5 bits comme spécifié dans le FS1016 [5,7].

I.3.3 Recherche dans le dictionnaire stochastique

Elle est effectuée en boucle fermée en utilisant le critère de minimisation de l'erreur quadratique moyenne du signal d'erreur perceptuelle. 9 bits sont réservés pour le codage de 512 vecteurs du dictionnaire stochastique. L'indice et le gain correspondant à l'excitation stochastique sont transmis 4 fois par trame. Le gain est codé sur 5 bits utilisant une quantification scalaire non uniforme.

Le dictionnaire stochastique est un dictionnaire entrelacé : les vecteurs-codes de 60 échantillons sont obtenus par décalage arrière de 2 échantillons, ce qui permet de le stocker dans un vecteur linéaire. Il contient des échantillons de moyenne nulle et de variance unité, et sont générés comme une séquence gaussienne quantifié à des valeurs ternaires {-1, 0, +1}. Il en résulte approximativement 77% de valeurs nulles. Cette forme du dictionnaire permet de réaliser des algorithmes rapides pour la sélection du vecteur optimal en réduisant d'une manière importante le taux de calcul, sans dégrader la qualité de la parole par rapport à d'autres dictionnaires [7].

I.3.4 Calcul des coefficients LSF

Il existe une représentation des coefficients de prédiction a_k dite LSF (Line spectral frequencies). Elle a été introduite par Itakura [9] et devient par la suite largement utilisée dans le codage de la parole.

Les coefficients a_k sont peu appropriés pour la quantification à cause de leur large dynamique de valeurs, causant une grande difficulté dans la détermination des régions de quantification. En effet, cela provoque des erreurs de quantification conduisant à une instabilité du filtre de synthèse $1/A(z)$. Par contre, les coefficients LSF possèdent une gamme bornée de valeurs, avec une allocation de bits variable selon l'importance du coefficient (en fonction de la perception de l'oreille humaine) [9].

I.3.4.1 Définition des coefficients LSF

Les coefficients LSF dérivent du filtre d'analyse *A(z)* d'ordre *P* (généralement *P =10*) donné par l'équation suivante :

$$A(z) = 1 + \sum_{k=1}^{p} a_k z^{-k} \tag{I.8}$$

Deux polynômes *P(z)* et *Q(z)* sont formés par l'addition et la soustraction du terme $z^{-(p+1)}A(z)$ du terme *A(z)* respectivement comme suivant [10] :

$$\begin{aligned} P(z) &= A(z) + z^{-(p+1)}A(z) \\ Q(z) &= A(z) - z^{-(p+1)}A(z) \end{aligned} \tag{I.9}$$

Les deux polynômes *P(z)* et *Q(z)* sont, respectivement, symétrique et antisymétrique et ayant les propriétés suivantes :

- Toute les racines de *P(z)*et *Q(z)* sont sur le cercle unité.
- Les racines de *P(z)*et *Q(z)* sont entrelacées.
- La propriété de la phase minimum de *A(z)* peut être préservée, si les premières propriétés sont préservées après la quantification.

Si l'ordre *P* est paire *P(z)*et *Q(z)* ont respectivement des zéros à z = -1 et z = +1 , et nous définissons deux autres polynômes *P'(z)*et *Q'(z)* tel que

$$\begin{aligned} P(z) &= (1 + z^{-1})P'(z) \\ Q(z) &= (1 - z^{-1})Q'(z) \end{aligned} \tag{I.10}$$

Les polynômes *P'(z)* et *Q'(z)* sont symétriques, et ont les propriétés suivantes :

- Si les racines du filtre *A(z)* sont à l'intérieur du cercle unité, alors les racines de *P'(z)* et *Q'(z)* sont sur ce même cercle et elles sont entrelacées et commençant par une racine de *P'(z)*.
- Inversement si les racines de *P'(z)* et *Q'(z)* sont sur le cercle unité et sont entrelacées et commençant par la racine de *P'(z)* , alors les racines de *A(z)* sont à l'intérieur du cercle unité.

- Du fait que les racines de *P'(z)* et *Q'(z)* sont sur le cercle unité, ces derniers peuvent être complètement définies par la position angulaire de leurs racines. En outre, les polynômes *P'(z)* et *Q'(z)* ont des coefficients réels, leurs racines se présentent sous forme de paires de nombres complexes conjugués, et par conséquent uniquement les positions angulaires situées sur le demi cercle supérieur du plan Z sont suffisantes pour déterminer complètement *P'(z)* et *Q'(z)*. Les coefficients LSF sont définis comme les positions angulaires *(ω)* des racines de *P'(z)* et *Q'(z)* situées dans le demi cercle supérieur du plan Z.

- Les coefficients LSF doivent respecter la condition suivante :

$$0 < \omega_1 < \omega_2 < \ldots\ldots\ldots < \omega_p < \pi \qquad \text{(I.11)}$$

Cette dernière propriété exprime l'ordonnancement des coefficients LSF. C'est une condition nécessaire et suffisante pour la stabilité du filtre de synthèse *1/A(z)*.

Pour une trame de 240 échantillons de parole, les positions des zéros de *A(z)* sont montrées dans la figure I.8 alors que les positions des coefficients LSF sont représentées par la figure I.7.

La figure I.8 montre par des lignes verticales les positions des coefficients LSF par rapport au spectre estimé par les coefficients de prédiction linéaire. De cette figure on peut tirer les propriétés des coefficients LSF suivants [10, 11,12] :

Les coefficients LSF sont toujours concentrés autour des formants. La largeur de bande d'un formant dépend de la proximité de ses coefficients LSF correspondants.

En générale, la sensibilité spectrale de chaque coefficient LSF est localisée. Cela veut dire qu'un changement dans un coefficient causera un changement dans le spectre de puissance près de son voisinage.

Un espacement court entre deux coefficients LSF correspond à un pic (formant), tandis qu'un espacement large entre deux coefficients LSF correspond à une vallée.

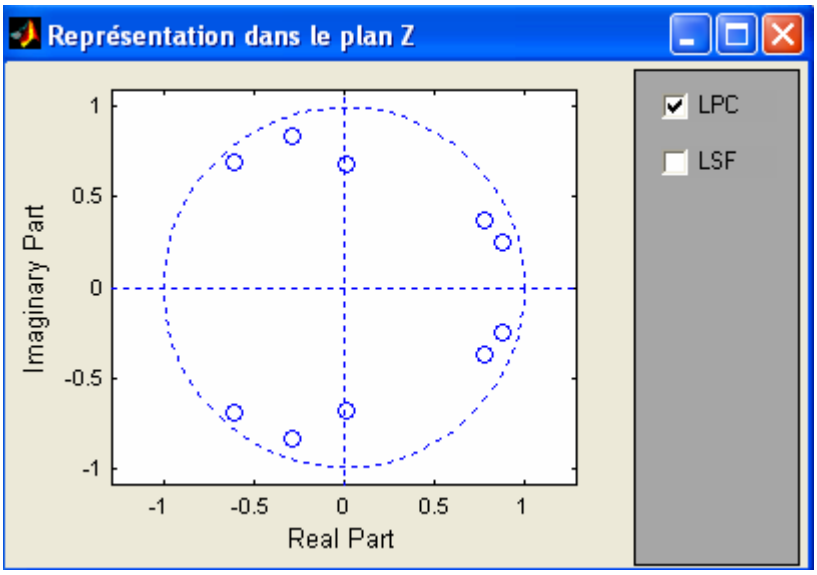


Figure. I.6 Position des racines de *A(z)* d'ordre *P = 10* sur le cercle unité.

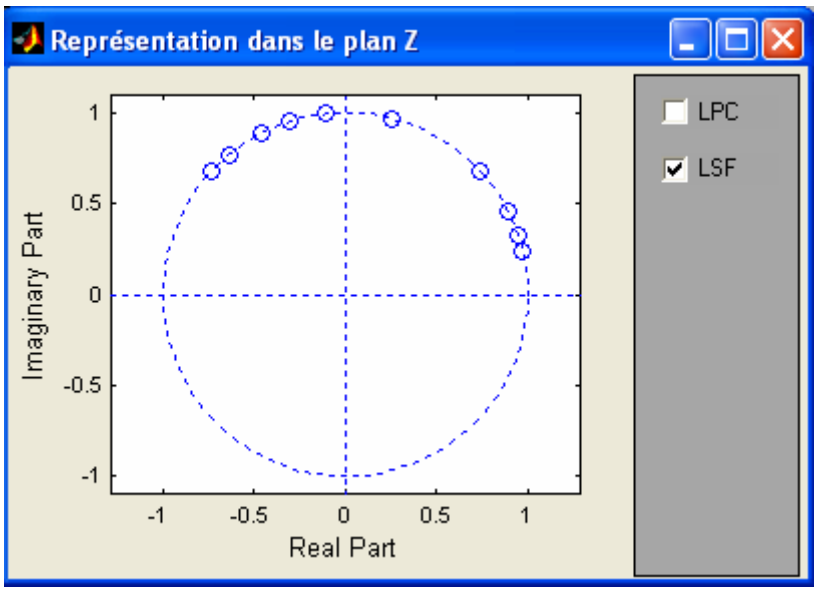


Figure. I.7 Position des LSF d'ordre *P = 10* sur le cercle unité.

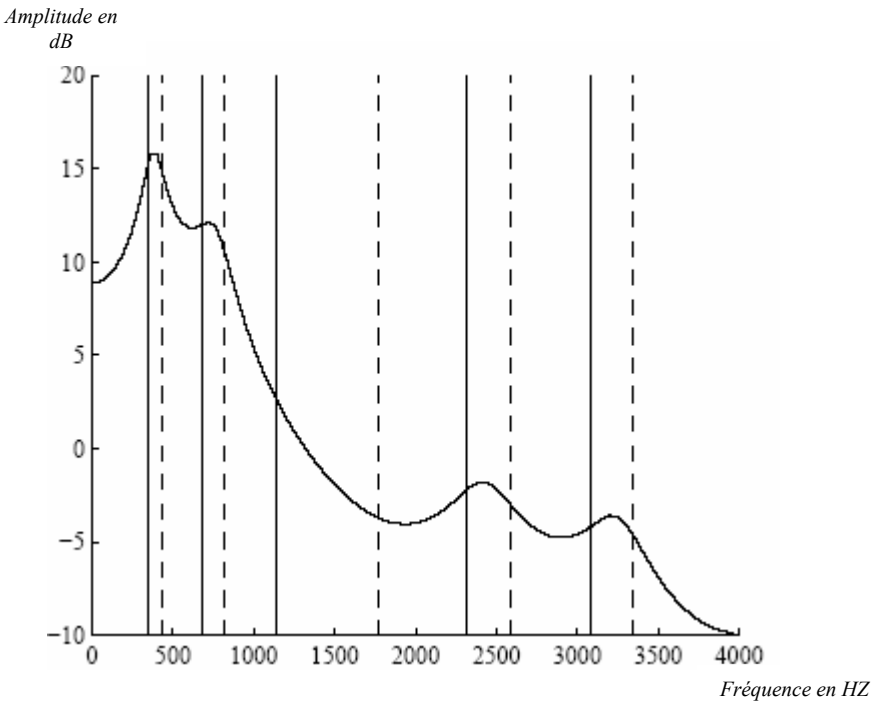


Fig. I.8 Position des LSF d'ordre *P = 10* sur le cercle unité.

I.3.4.2 Utilisation de la représentation LSF

La représentation LSF d'ordre 10 est très utilisée dans presque tous les codeurs de paroles actuels et en particulier les standards dont le débit est inférieur à16 kbps. On cite comme exemple :

- Le standard ITU-T G.729 CS-ACELP à 8Kbpps [13]
- Le standard ITU-T G.723.1, codeur à double débit pour la communication multimédia à 5,3/6,3 Kbps [14]
- Le GSM 6,60, à 13 Kbps [15]
- Le standard japonais pour le cellulaire digital [16]
- Le standard américain de la défense pour la téléphonie sécurisée FS1016 CELP à 4,8 Kbps [17,18].
- Le nouveau standard américain pour la téléphonie sécurisée, MELP (Mixed Exited Linear Prediction), à 2,4 Kbps. [19, 20, 21, 22, 23]

Dans ces standards, la représentation LSF procure une qualité efficace de quantification pour l'information spectrale, avec moins de bits et une meilleure qualité de parole.

I.3.4.3 Caractéristiques des coefficients LSF

En dehors des avantages des coefficients LSF pour la quantification déjà mentionnés précédemment, telles que les valeurs bornées et la stabilité du filtre de synthèse, d'autres caractéristiques et propriétés avantageuses des coefficients LSF viennent se présenter tels que les corrélations intra-trame et inter-trame, la localisation de la sensibilité spectrale et sa relation étroite avec la perception.

a) **Représentation dans le domaine fréquentiel**

La représentation LSF est une représentation fréquentielle dont la quantification peut aisément incorporer des traits très importants pour la perception et la sensibilité de l'oreille humaine en particulier. Un exemple est donné dans [24], où les fréquences basses des LSF ont été quantifiées avec plus de bits que les fréquences plus hautes.

b) **Sensibilité spectrale locale**

La sensibilité spectrale des coefficients est locale [24]. Autrement dit l'existence d'une perturbation dans une valeur d'un coefficient LSF produira un changement dans l'énergie, résidant seulement dans le voisinage de ce dernier. On peut alors conclure que les coefficients LSF peuvent être individuellement quantifiés sans fuite et sans transmission de la distorsion d'une région spectrale à une autre.

I.3.4.4 Calcul des coefficients LSF par la Méthode de Kabal

Pour déterminer les coefficients LSF, les racines des polynômes $P'(z)$ et $Q'(z)$ donnés par l'équation (I.10) doivent être calculées. Plusieurs méthodes plus ou moins complexes on été proposées. La méthode la plus utilisée (vu son efficacité) est la méthode de Kabal [12].

Pour des raisons de compréhension nous allons considérer l'ordre de prédiction fixe correspondant au codage CELP avec P=10. Les polynômes $P'(z)$ et $Q'(z)$ s'écrivent alors

$$P'(z) = z^{-10} + p'_1 z^{-9} + p'_2 z^{-8} + p'_3 z^{-7} + p'_4 z^{-6} + p'_5 z^{-5} + p'_4 z^{-4} + p'_3 z^{-3} + p'_2 z^{-2} + p'_1 z^{-1} + 1 \quad \text{(I.12)}$$

$$Q'(z) = z^{-10} + q'_1 z^{-9} + q'_2 z^{-8} + q'_3 z^{-7} + q'_4 z^{-6} + q'_5 z^{-5} + q'_4 z^{-4} + q'_3 z^{-3} + q'_2 z^{-2} + q'_1 z^{-1} + 1 \quad \text{(I.13)}$$

Dans la méthode de Kabal, la symétrie est utilisée pour grouper les termes comme suit :

$$P'(z) = z^{-5}\left[\left(z^{+5} + z^{-5}\right) + p'_1\left(z^{+4} + z^{-4}\right) + ... + p'_5\right] \quad \text{(I.14)}$$

$$Q'(z) = z^{-5}\left[\left(z^{+5} + z^{-5}\right) + q'_1\left(z^{+4} + z^{-4}\right) + ... + q'_5\right] \quad \text{(I.15)}$$

avec :

$$\begin{aligned} p'_1 &= a_1 + a_{10} + 1 & q'_1 &= a_1 - a_{10} + 1 \\ p'_2 &= a_2 + a_9 + p'_1 & q'_2 &= a_2 - a_9 + p'_1 \\ p'_3 &= a_3 + a_8 + p'_2 & q'_3 &= a_3 - a_8 + p'_2 \\ p'_4 &= a_4 + a_7 + p'_3 & q'_4 &= a_4 - a_7 + p'_3 \\ p'_5 &= a_5 + a_6 + p'_4 & q'_5 &= a_5 - a_6 + p'_4 \end{aligned} \quad \text{(I.16)}$$

Evaluons maintenant les solutions de *P'(z)* et *Q'(z)* sur le cercle unité ($z = e^{-j\omega}$), et remplaçons z dans les équations (I.14) et (I.15) on obtient alors :

$$P'(\omega) = e^{-5j\omega}\left[2\cos(5\omega) + 2p'_1\cos(4\omega) + ... + 2p'_4\cos(\omega) + p'_5\right] \quad \text{(I.17)}$$

$$Q'(\omega) = e^{-5j\omega}\left[2\cos(5\omega) + 2q'_1\cos(4\omega) + ... + 2q'_4\cos(\omega) + q'_5\right] \quad \text{(I.18)}$$

Après la division par le terme $2e^{-5j\omega}$ les équations précédentes deviennent :

$$P'(\omega) = \cos(5\omega) + p'_1\cos(4\omega) + ... + p'_4\cos(\omega) + \frac{p'_5}{2} \quad \text{(I.19)}$$

$$Q'(\omega) = \cos(5\omega) + q'_1\cos(4\omega) + ... + q'_4\cos(\omega) + \frac{q'_5}{2} \quad \text{(I.20)}$$

Kabal fait introduire les polynômes de Chebyshev de premier ordre qui sont définis comme suivant :

$$T_n(x) = cos\ (n\omega) = 2xT_{n-1}(x) - T_{n-2}(x)$$

$$T_0(x) = cos\ (0) = 1 \qquad \text{(I.21)}$$

$$T_1(x) = cos\ (\omega) = x$$

En calculant ces polynômes de l'ordre n = 2 jusqu'à l'ordre n = 5 nous aurons les équations suivantes :

$$T_2(x) = cos\ (2\omega) = 2x^2 - 1$$

$$T_3(x) = cos\ (3\omega) = 4x^3 - 3x$$

$$T_4(x) = cos\ (4\omega) = 8x^4 - 8x^2 + 1 \qquad \text{(I.22)}$$

$$T_5(x) = cos\ (5\omega) = 16\ x^5 - 20\ x^3 + 5x$$

Cependant, en appliquant $x = \cos(\omega)$ aux équations (I.19) et (I.20) tout en utilisant les polynômes de Chebyshev de premier ordre (équation (I.21)), deux polynômes de 5eme degré sont alors obtenus :

$$P'(x) = 16x^5 + 8p'_1 x^4 + (4p'_2 - 20)x^3 + (2p'_2 - 8p'_1)x^2 + (5 - 3p'_2 + p'_4)x + (p'_1 - p'_3 + 0{,}5p'_5) \quad \text{(I.23)}$$

$$Q'(x) = 16x^5 + 8q'_1 x^4 + (4q'_2 - 20)x^3 + (2q'_2 - 8q'_1)x^2 + (5 - 3q'_2 + q'_4)x + (q'_1 - q'_3 + 0{,}5q'_5) \quad \text{(I.24)}$$

Les racines des polynômes *P'(x)* et *Q'(x)* sont les coefficients LSF dans le domaine des x_i avec :

$$-1 < x_{10} < x_9 < \ldots\ldots < x_2 < x_1 < +1 \qquad \text{(I.25)}$$

Comme *P'(z)* et *Q'(z)* sont des polynômes d'ordre « 5 », leurs racines ne peuvent en aucune manière être calculées d'une manière exacte. Aussi Kabal proposa dans sa méthode le passage par l'axe des "zéros" en commençant sa recherche de « x = +1) avec un décrément de Δ = 0,02, et chaque fois qu'un intervalle contenant un passage par l'axe des "zéros" est

retrouvé, la position de la racine est raffinée ; premièrement par l'utilisation de bissections successives, et ensuite par l'utilisation d'une interpolation linéaire.

Il faut noter l'entrelacement des racines des polynômes *P'(z)* et *Q'(z)*. Cet entrelacement doit être pris en compte lors de ces racines.

Les racines ainsi trouvées représentent les coefficients LSF, et ceci après l'application de la transformée inverse $\omega = arccos(x)$.

Une évaluation efficace et récursive des polynômes *P'(z)* et *Q'(z)* pour une valeur donnée de *x*, qui utilise les coefficients p'_i et q'_i (pour *i=1,...5*) définis dans les équations (I.16), est donné comme suit :

$$
\begin{aligned}
&b_7(x) = 0 \qquad b_6(x) = 0 \qquad b_5(x) = 0 \\
&b_4(x) = 2x + p'_1 \\
&b_3(x) = 2xb_4(x) - b_5(x) + p'_2 \\
&b_2(x) = 2xb_3(x) - b_4(x) + p'_3 \\
&b_1(x) = 2xb_2(x) - b_3(x) + p'_4 \\
&P'(x) = xb_1(x) - b_2(x) + 0{,}5\,p'_5
\end{aligned}
\qquad \text{(I.26)}
$$

L'évaluation de *Q'* est similaire à celle de *P',* il suffit de remplacer p'_i par q'_i pour *i=1,...5*. Ainsi, les expressions en puissance de « *x* » données dans l'équation (I.23) et (I.24) ne sont pas nécessaires.

I.4 Principe du décodeur FS1016

Le décodeur (récepteur) CELP est donné à la figure I.5. Après réception des paramètres correspondant à une trame d'analyse, le récepteur décode ces paramètres, incluant la correction d'erreurs de transmission, comme spécifié dans le standard FS1016. Le décodeur synthétise la parole par passage d'une excitation à travers le filtre de synthèse 1/*A(z)*.dont les coefficients ont été reçus. Disposant d'un dictionnaire stochastique identique à celui de

l'encodeur et d'un dictionnaire adaptatif construit de la même manière que celui de l'encodeur, le décodeur peut reconstruire l'excitation globale par addition de l'excitation adaptative (définie par le couple reçu (I_a, G_a)) et de l'excitation stochastique (définies par le couple reçu (I_s, G_s)).

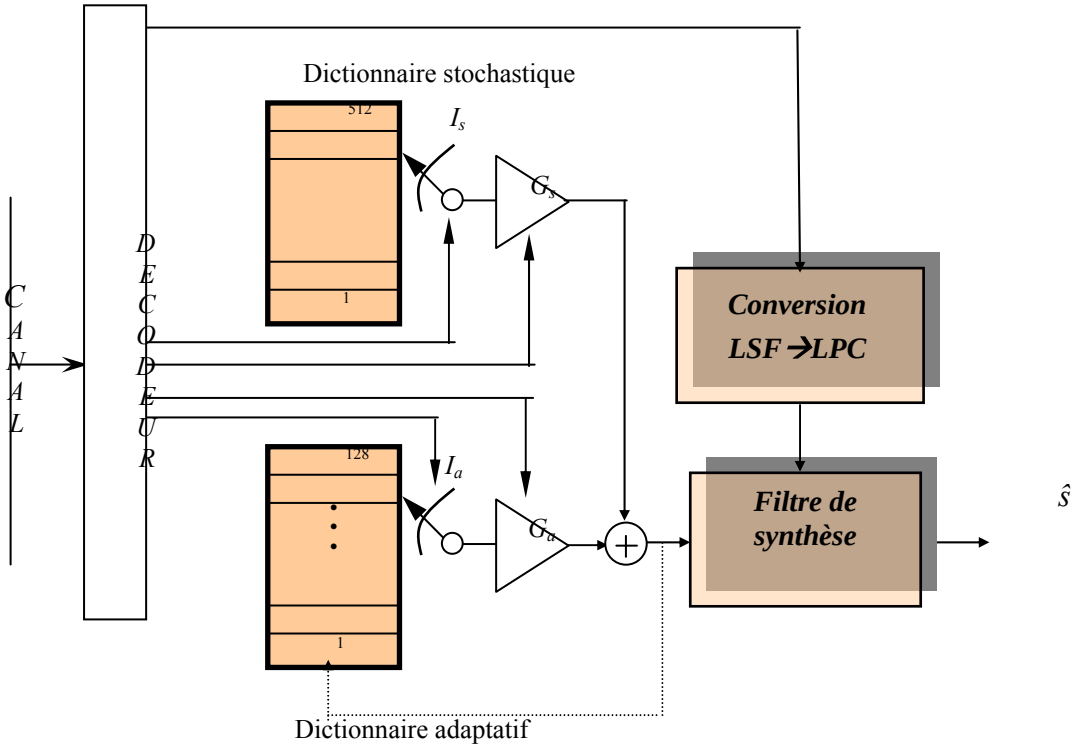


Figure I.5 Schéma de décodeur CELP

I.5 Caractéristiques du codeur CELP implanté

Les caractéristiques du codeur implémenté sont apportées à la table Tableau. I.1. Notons que les coefficients LSF sont quantifié en utilisant une quantification scalaire puis codé sur 34 bits (Tableau. I.2) pour chaque trame de parole de 30 ms ce qui donne un débit de 1,13333 Kbits/s. le but du ce travail est de diminuer ce débit tout en gardant une bonne qualité de la parole synthétique. Pour cela nous avons étudié, conçu et testé d'autres méthodes de quantification des coefficients LSF.

Tableau. I.2 Tables de la quantification des coefficients LSF d'après la norme FS1016. C'est une quantification scalaire codée sur 34 bits

LSF1	
index	**valeur**
0	0.0125
1	0.0213
2	0.0281
3	0.0312
4	0.0350
5	0.0425
6	0.0525
7	0.0625

LSF2	
index	**valeur**
0	0.0262
1	0.0294
2	0.0331
3	0.0369
4	0.0406
5	0.0450
6	0.0500
7	0.0550
8	0.0600
9	0.0650
10	0.0700
11	0.0762
12	0.0838
13	0.0925
14	0.1013
15	0.1100

LSF3	
index	**valeur**
0	0.0525
1	0.0575
2	0.0625
3	0.0675
4	0.0731
5	0.0800
6	0.0881
7	0.0969
8	0.1062
9	0.1188
10	0.1312
11	0.1438
12	0.1562
13	0.1688
14	0.1812
15	0.1938

LSF4	
index	**valeur**
0	0.0775
1	0.0825
2	0.0900
3	0.0994
4	0.1100
5	0.1212
6	0.1350
7	0.1462
8	0.1588
9	0.1713
10	0.1838
11	0.1962
12	0.2088
13	0.2212
14	0.2338
15	0.2462

LSF5	
index	**valeur**
0	0.1250
1	0.1312
2	0.1412
3	0.1512
4	0.1606
5	0.1688
6	0.1788
7	0.1888
8	0.1988
9	0.2088
10	0.2188
11	0.2312
12	0.2438
13	0.2562
14	0.2688
15	0.2812

LSF6	
index	**valeur**
0	0.1838
1	0.1962
2	0.2112
3	0.2288
4	0.2500
5	0.2750
6	0.3000
7	0.3250

LSF7	
index	**valeur**
0	0.2250
1	0.2350
2	0.2450
3	0.2625
4	0.2875
5	0.3100
6	0.3375
7	0.3625

LSF8	
index	**valeur**
0	0.2781
1	0.3000
2	0.3156
3	0.3312
4	0.3500
5	0.3688
6	0.3938
7	0.4188

LSF9	
index	**valeur**
0	0.3450
1	0.3600
2	0.3750
3	0.3875
4	0.4000
5	0.4138
6	0.4288
7	0.4438

LSF10	
index	**valeur**
0	0.3988
1	0.4088
2	0.4188
3	0.4275
4	0.4362
5	0.4488
6	0.4638
7	0.4788

Conclusion

Une introduction sur le codeur CELP basé sur la norme FS1016 a été donnée. Les constituants essentiels de ce standard ont été brièvement expliqués. Une attention particulière a été portée sur les paramètres candidats à la transmission de ce codeur en l'occurrence les coefficients LSF, objet principal de notre travail. Une fine analyse des dictionnaires de la quantification scalaire du FS1016 nous a permis de détecter que cette dernière peut représenter parfois des vecteurs rendant le filtre de synthèse A(z) instable, ce qui nous a conduit à proposer une solution qui fera l'objet du deuxième chapitre.

Chapitre II

Conception d'un système d'énumération en treillis pour le QS du FS1016.

Les quantifications par treillis ont eu cette appellation suite à leurs schémas utilisant des chemins sous forme de treillis. Ces chemins passent, d'un niveau de reproduction à un autre se trouvant dans des dictionnaires différents.

Plusieurs travaux récents [25, 26, 27] se sont intéressés à l'application de ce type de quantification aux coefficients LSF. Généralement la quantification par treillis utilise l'algorithme de Viterbi [27, 28] pour optimiser le parcours à travers les niveaux de reproduction ou mot-codes. Une version de cet algorithme a été proposée dans les travaux de Farshad Lahouti [27].

Dans notre cas, on se propose d'appliquer un système d'énumération en treillis sur les dictionnaires de la quantification scalaire à la norme FS1016 [29].

Une fine analyse faite sur les dictionnaires de la quantification scalaire du FS1016 a révélé que ce dernier exploite une gamme de bits qui introduit quelquefois des vecteurs inacceptables rendant le filtre synthèse *A(z)* instable.

En effet, 34 bits sont exigés pour coder donne $N = \prod_{i=1}^{10} 2^{b_i} = 17179869184$ combinaisons possibles avec b_i nombre de bits accordé pour le i^{eme} dictionnaire(les b_i *sont :* 3, 4, 4, 4, 4, 3, 3, 3, 3, 3).. De ce nombre, on peut extraire seulement 1073741824 (soit 6.25% du nombre total) qui sont des vecteurs LSF acceptables pouvant être codés aux maximum sur 30 bits. Si on compare le premier nombre à celui des vecteurs acceptables on aura 16624910796

(soit 96.76% du nombre total) vecteurs inacceptables. La question qu'on peut se poser est de savoir comment passer rapidement d'un coefficient à l'autre à travers les dictionnaires tout en évitant ces cas inacceptables? Ceci est l'objet de ce chapitre.

II.1 Rappel sur la quantification scalaire

Il est possible, sous certaines conditions, de représenter un signal analogique de parole (ou autre), par un signal de type numérique afin de le transmettre ou le stocker avec un taux d'erreur aussi faible que possible. Cette opération, dite numérisation, comporte deux phases : La première phase consiste à échantillonner le signal et la deuxième procède à quantifier ces échantillons obtenus à l'étape précédente. L'erreur de quantification introduite sur chacun d'eux engendre une distorsion définitive du signal.

Formellement, un quantificateur (figure II.1) associe aux valeurs prises aléatoirement d'une 'source', des valeurs d'un ensemble de niveaux de quantification appelé « alphabet de reproduction ». Les niveaux de quantification sont en nombre fini, et par conséquent le débit d'information est bien déterminé.

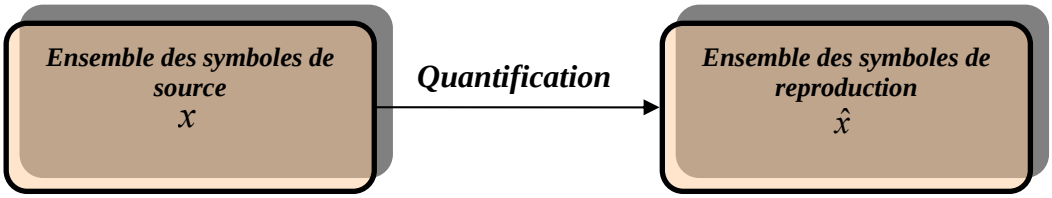


Figure II.1 : Schéma général d'un quantificateur.

En réalité le remplacement de la valeur d'un échantillon x par sa version quantifiée $\hat{x}$ a un coût. Pour mesurer ce coût on définit une métrique appelée distorsion $d(x, \hat{x})$. La mesure de la distorsion la plus utilisée est la distance euclidienne carrée [30] : $d(x, \hat{x}) = (x - \hat{x})^2$. Le problème de la quantification est de chercher à minimiser cette distance pour un débit qui

serait le plus faible possible. Parmi les méthodes de quantification connues on cite la quantification scalaire.

II.1.1 Le principe de la quantification scalaire

Le principe d'un quantificateur scalaire [31] consiste à donner un code à un échantillon sans tenir compte des échantillons voisins, ce qui semble à première vu un choix naturel. En conséquence, le quantificateur scalaire est défini par un ensemble de niveaux de quantification appelé dictionnaire et par une distance.

La quantification scalaire assigne à une valeur d'entrée x sa valeur approximée d'un ensemble fini prédéterminé ou dictionnaire de L valeurs :

$$C=\{y_k /k = 1, \ldots,L\} \quad \text{(II.1)}$$

Les valeurs de sorties y_i sont appelées niveaux de sortie, ou encore niveaux de reproduction. Dans tous les cas, L est fini et on admet que :

$$y_1 < y_2 < \ldots.. < y_L \quad \text{(II.2)}$$

Le quantificateur partitionne la base d'apprentissage R en L intervalles ou classes R_i, la $i^{ème}$ classe est donnée par :

$$R_i=\{x \in R: Q(x)=y_i)\} \quad \text{(II.3)}$$

Selon cette définition, l'intersection de toute les classes ou intervalles R_i est l'ensemble vide:

$$R_i \bigcap R_j = \phi \quad \text{pour i}\neq\text{j} \quad \text{(II.4)}$$

Les classes ou les cellules limites sont appelées cellules granulaires. L'ensemble de ces cellules est appelé région granulaire.

Le débit de ce quantificateur est donné par l'expression suivante :

$$b = \log_2 L \quad \text{[Bits/Echantillon]} \quad \text{(II.5)}$$

Un quantificateur scalaire peut être régulier lorsque deux valeurs d'entrées, *a* et *b* avec $a < b$, sont quantifiées avec la même valeur de sortie *y*, alors n'importe quelle autre entrée comprise entre *a* et *b* sera quantifiée elle aussi avec la même sortie *y*.

La procédure de quantification consiste à décider à quel intervalle appartient $x(n)$ puis à lui associer l'indice $I(n) \in \{0,\ldots,L-1\}$ avec $L=2^b$ et *b* étant le débit correspondant qui sera transmis. La procédure inverse de quantification consiste à associer au numéro ou indice $I(n)$ le représentant correspondant choisi parmi l'ensemble des représentants $\{y_1 \ldots\ldots y_L\}$.

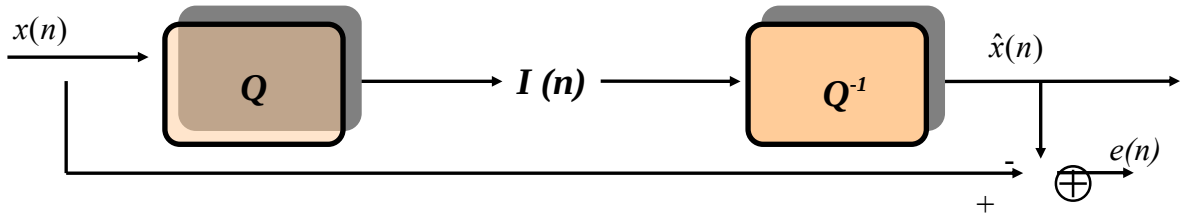


Figure II.2 : Schéma illustrant l'encodage et le décodage d'un quantificateur scalaire.

L'opération de quantification entraîne une perte d'information irréversible, qui se traduit par l'introduction d'une erreur *e(n)*, appelée aussi bruit de quantification :

$$e(n) = x(n) - \hat{x}(n)$$

Notons ici qu'il existe deux types d'erreurs :

- Erreur granulaire qui se produit lorsque la valeur d'entrée *x* se situe dans l'une des cellules $[x_i, x_{i+1}]$. L'erreur résultante qui est la différence entre *x* et *Q(x)* peut être majorée par un demi pas de quantification dans le cas uniforme.

- Erreur de surcharge ou de dépassement qui se produit lorsque la valeur d'entrée se situe en dehors de l'intervalle $[x_0, x_L]$. la valeur de reproduction est soit y_1 ou y_L, et l'erreur résultante est supérieure à un demi pas de quantification dans le cas uniforme.

Le quantificateur scalaire se divise en deux types : le quantificateur scalaire uniforme pour lequel les pas de quantification sont égaux et le quantificateur scalaire non uniforme.

II.1.2 Le quantificateur scalaire uniforme

Le quantificateur scalaire uniforme (figure II.3) est entièrement déterminé par $L=2^b$ et par :

- Les $L+1$ **niveaux de décision** : $x_0, x_1, \ldots, x_L$ qui partitionnent l'axe des réels en L intervalles.
- Les L **valeurs de reproduction** : $y_1, y_2, \ldots, y_L$, qui sont les centres de masse de chacun des intervalles de décision. L'ensemble de ces valeurs s'appelle dictionnaire.

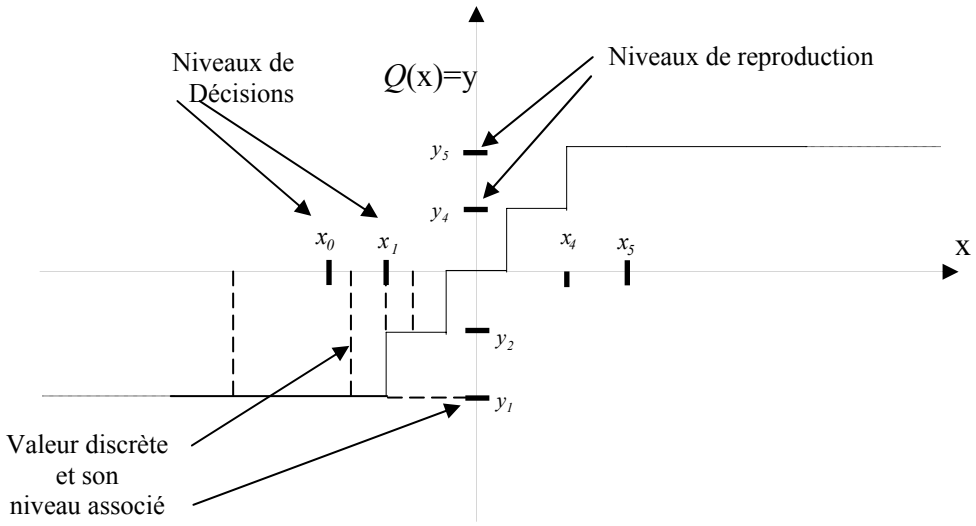


Figure II.3 : Principe de la quantification scalaire uniforme.

II.1.3 Le quantificateur scalaire optimal (non uniforme)

Le QS optimal est celui qui minimise, pour une source donnée et un débit fixé, l'erreur moyenne de reconstruction due aux bruits de quantification et de surcharge. Les niveaux de reconstruction sont donc repartis en tenant compte de la densité de probabilité de la variable à quantifier. Intuitivement nous comprenons que la concentration des niveaux de reconstruction est plus importante dans la zone de l'espace où la densité de probabilité des valeurs à quantifier est plus élevée (Figure II.4).

En fait ce problème d'optimisation à une solution unique et le QS obtenu est connu sous le nom de quantificateur de Lloyd-Max [32, 33]. En définissant le gain en distorsion comme étant le rapport, à débit fixe, des distorsions des deux quantificateurs, le quantificateur de Lloyd-Max offre, du fait de la répartition non uniforme de ses pas de quantification, un gain en distorsion par rapport aux autres quantificateurs scalaires. Cependant, ce gain est loin de rejoindre le maximum annoncé par la théorie de l'information [1].

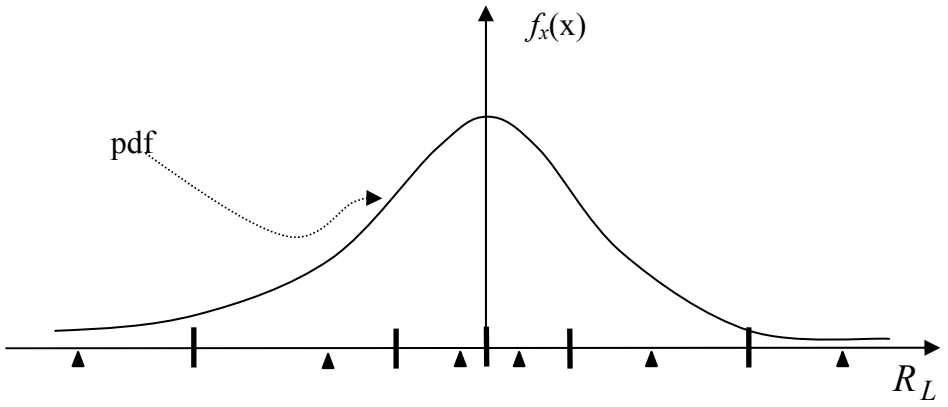


Figure II.4 : Quantificateur scalaire non uniforme.

II.2 Le système d'énumération en treillis

La quantification par énumération en treillis est une méthode qui utilise une quantification scalaire à l'origine. La dénomination "Treillis" vient du fait que la quantification donne un indice relatif au chemin optimal à travers les dictionnaires de la quantification scalaire en question. Le système en treillis dans la quantification scalaire des coefficients LSF profite de l'ordre ascendant des ces coefficients. En effet, ce système est conçu pour éviter les combinaisons donnant des vecteurs non acceptables en sortie du quantificateur (soit par exemple $LSF_2 < LSF_1$). Pour comprendre les algorithmes de quantification on donne les notations suivantes :

- Considérons un QS utilisé pour la quantification scalaire des coefficients LSF. Pour un ordre de prédiction " p ", le nombre maximal des combinaisons que les dictionnaires de cette quantification scalaire pourrait donner est : $N_{\max} = \prod_{i=1}^{p} 2^{b_i}$ où b_i est le nombre de bits accordé pour le i^{eme} dictionnaire.

- $N_{i+1}\left(\hat{LSF}_{i,j}\right)$ est le nombre de combinaisons possibles de coefficients vérifiant $\hat{LSF}_{i,j} < \hat{LSF}_{i+1} < \hat{LSF}_{i+2} < ..... < \hat{LSF}_{p}$ où $\hat{LSF}_{i,j}$ est le représentant du coefficient d'ordre (*i)* dont l'indice est (*j)* dans le dictionnaire C_i correspondant.

Pour une position *j* dans le dictionnaire C_{p-1}, le nombre de combinaisons possibles pour atteindre le dernier coefficient dans le dictionnaire C_p est donné par :

$$N_p\left(\hat{LSF}_{p-1,j}\right) = \sum_{\hat{LSF}_{p,i} > \hat{LSF}_{p-1,j}}^{2^p} 1 \quad \text{avec } 1 \le j \le 2^{b_{p-1}} \text{ reconstructions possibles vérifiant}$$

$$\hat{LSF}_{p-1,j} < \hat{LSF}_{p,i}$$

Autrement dit, de la position *j* dans le dictionnaire C_{p-1} combien on peut former de transitions possibles vers le dictionnaire C_p. La figure II.5 donne un exemple avec les deux dictionnaires 9 et 10 du FS1016.

En général, pour chaque $\hat{LSF}_{k,j} \in C_k$ avec $1 \le k \le p\text{-}2$ on a :

$$N_{k+1}\left(\hat{LSF}_{k,j}\right) = \sum_{\hat{LSF}_{k+1,i} > \hat{LSF}_{k,j}} N_{k+2}\left(\hat{LSF}_{k+1,i}\right) \text{ avec } 1 \le j \le 2^{b_k} \quad \text{(II.6)}$$

Autrement dit, dans un niveau *j* dans le dictionnaire d'ordre *k* on aura N_{k+1} chemins possibles pour une bonne combinaison des coefficients *LSF*.

- Le total des combinaisons possibles est

$$N(p, \overline{C}) = \sum_{i=1}^{l} N_2\left(\hat{LSF}_{1,i}\right)$$

(II.7)

Où $\overline{C} = \{C_1, C_2, ..., C_p\}$ *et* $l = 2^{b_1}$

Avec $N_2\left(\hat{LSF}_{1,i}\right)$ est le nombre de combinaisons possibles de coefficients vérifiant $\hat{LSF}_{1,j} < \hat{LSF}_2$.

Formellement, le nombre de bits nécessaires est : $K = \lfloor \log_2 (N(P, \overline{C})) \rfloor$. On peut remarquer que $K \leq \sum_{i=1}^{p} b_i$.

- Il faut noter que le mot-code du coefficient *LSF* d'ordre *i* est donné, en utilisant la quantification scalaire, dans le dictionnaire C_i par : $L\hat{S}F_{i,j}$. Cette opération sera notée $QS(LSF_i) = L\hat{S}F_{i,j}$.

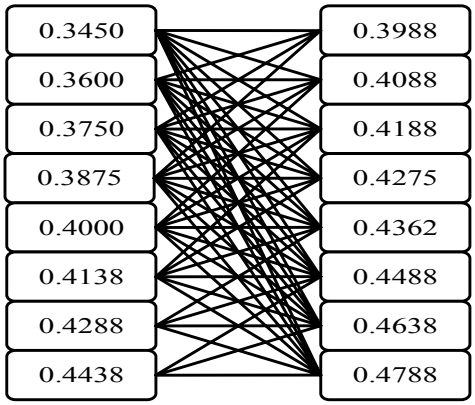


Figure. II.5 Exemple de transition du dictionnaire 9 vers le dictionnaire 10 du QS de FS1016

L'algorithme de quantification est donc :

et mettre $j^* = j$-*1* $1 \leq j \leq 2^{b_1}$ *avec* $QS(LSF_1) = L\hat{S}F_{1,j}$ ***Etape 1*** *Trouver j tel que*

$t = \sum_{k=1}^{j^*} N_2(L\hat{S}F_{1,k-1})$ $j^* = 1$ *sinon* ***Etape 2*** *Initialiser t = 0 si*

Etape 3 *Pour i = 2 à p*

$m = M_i(L\hat{S}F_{i-1,j^*})$ *- Mettre*

et mettre $j^* = j$-*1* $1 \leq j \leq 2^{b_i}$ *avec* $QS(LSF_i) = L\hat{S}F_{i,j}$ *- Trouver j tel que*

- *Mettre* $t = t + 0$ *si* $j^* = m$ *sinon* $t = t + \sum_{k=m}^{j^*} N_{i+1}(L\hat{S}F_{i,k-1})$

Etape 4 *Le mot-code d'énumération est donné par t.*

Au niveau de la réception le nombre *t* sera réceptionné pour être utilisé dans l'opération inverse décrite par l'algorithme suivant :

Etape 1 *Trouver j tel que* $\sum_{k=1}^{j-1} N_2(\hat{LSF}_{1,k-1}) \le t < \sum_{k=1}^{j} N_2(\hat{LSF}_{1,k-1})$ *où*

$1 \le j \le 2^{b_2}$ *et mettre* j^**=j-1.*

Etape 2 *Mettre* $\hat{LSF}_1 = \hat{LSF}_{1,j^*}$

Etape 3 *Mettre* $t = t - \sum_{k=1}^{j^*} N_2(\hat{LSF}_{i,j^*})$

Etape 4 *pour i=2 à p*

- *Mettre* $m = M_i(\hat{LSF}_{i-1,j^*})$
- *Mettre j = m si t = 0 sinon trouver j tel que*

  $\sum_{k=m}^{j-1} N_{i+1}(\hat{LSF}_{1,k-1}) \le t < \sum_{k=m}^{j} N_{i+1}(\hat{LSF}_{1,k-1})$ *où* $1 \le j \le 2^{b_i}$ *et mettre*

  $j^* = j\text{-}1$
- *Mettre* $\hat{LSF}_i = \hat{LSF}_{i,j^*}$
- *Mettre t = t -0 si* $j^* = m$ *sinon* $t = t - \sum_{k=1}^{j^*} N_{i+1}(\hat{LSF}_{i,j^*})$

Etape 5 *Le vecteur LSF reconstitué est donné par* $\{\hat{LSF}_1, \hat{LSF}_2, \hat{LSF}_3, \cdots, \hat{LSF}_p\}$

II.3 La quantification des coefficients LSF pour le codeur CELP

Avant de transmettre les coefficients LSF, il convient d'abord de les quantifier. En effet, ces coefficients contribuent directement à la représentation du spectre vocal. De plus, ils ne présentent pas tous la même loi de répartition (distribution non uniforme).

Dans la norme FS1016, les 10 coefficients LSF *(LSF_1 à LSF_{10})* sont quantifiés en utilisant une quantification scalaire codé sur 34 bits (3, 4, 4, 4, 4, 3, 3, 3, 3, 3). Les tables de quantification pour chaque coefficient LSF_i *(i=1 à 10)* ont été données au tableau I.2.

Notons que les valeurs de ces tables représentent les fréquences normalisées.

Dans ce qui suit, nous exposons l'algorithme de quantification scalaire utilisé par la norme FS1016. On utilisera les notations suivantes pour une bonne compréhension de l'algorithme :

- N_L est le vecteur contenant le nombre de niveaux de quantification des coefficients LSF.
- *TQ* est la table de quantification représentant une matrice d'ordre $P \times N_L(max)$. $N_L(max)$ est la valeur maximale du vecteur N_L et *P* est le nombre de coefficients LSF.
- *Dist* est la distance minimale obtenue après comparaison de chaque coefficient avec les éléments de *TQ*.
- *LSFQ* est le vecteur contenant les coefficients quantifiés.

La quantification s'effectue selon l'algorithme suivant :

```
Pour i = 1,...,P
Début
        Pour j = 1,....N_L(i)
                d(j) = | LSF(i) – TQ(i,j)|

        Dist = d(1), J_min =1
        Pour j = 2,...,N_L(i)
                Si d(j) < Dist alors Début
                                        Dist = d(j)
                                        J_min = j
                                Fin

        LSFQ(i) = TQ(i,j_min)
Fin
```

A noter que cette quantification scalaire (codée sur 34 bits), proposée par la norme FS1016, a donnée une bonne qualité perceptuelle de la parole malgré une distorsion relativement grande *(1.73 dB).* Cette qualité est due à la bonne allocation des bits entre les dictionnaires de la quantification scalaire [17].

II.4 Système d'énumération en treillis pour la quantification scalaire des LSF

Dans le standard américain FS1016 les coefficients LSF sont calculés pour chaque trame de parole de 30 ms et sont codés sur 34 bits après avoir subi une quantification scalaire. Nous avons effectué une analyse fine sur les combinaisons des dictionnaires de la quantification scalaire utilisée dans la norme FS1016. Cette analyse a montré qu'il y'avait un nombre très important des vecteurs LSF qui rendaient le filtre de synthèse instable. Aussi, dans ce qui va suivre nous proposons une méthode d'énumération utilisant un système en treillis réduisant considérablement le nombre de bits nécessaires (30 bits au lieu de 34 bits) sans changer les dictionnaires de la quantification scalaire suscitée. En effet, les résultats de quantification utilisant soit le quantificateur scalaire original de FS1016 soit le système d'énumération en treillis seront les mêmes.

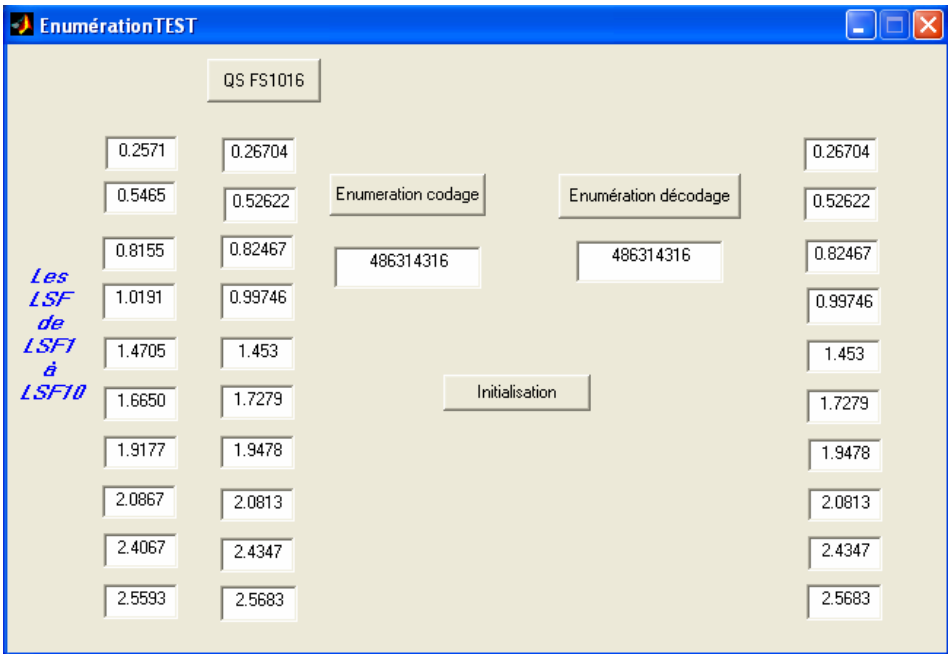


Figure II.6 Exemple du programme de test pour pouvoir examiner le résultat du quantificateur scalaire du FS1016 et celui du système d'énumération. On remarque que ces deux résultats sont égaux.

Les notations pour les coefficients LSF et pour un ordre de prédiction p =10 seront données comme suit :

- $L\hat{S}F_{i,j}$ est le représentant du coefficient LSF_i dans le dictionnaire C_i ayant l'indice j et ceci en utilisant la quantification scalaire du FS 1016 notée : $QS_{34}(LSF_i) = L\hat{S}F_{i,j}$

- $N_{i+1}\left(L\hat{S}F_{i,j}\right)$ est le nombre de chemins possibles allant du mot-code du coefficient LSF_i dans le dictionnaire C_i à la position j pour atteindre enfin le mot-code du dernier coefficient du dictionnaire $C_{10.}$

- $M_i(L\hat{S}F_{i-1,j})$ est le plus petit entier m tel que le mot-code d'ordre m dans le dictionnaire C_i soit directement supérieur au coefficient LSF_i dont l'ordre est j.

Pour diminuer la complexité de l'algorithme on a calculé le nombre de chemins (tableau II.1). Dans cet ordre d'idée, nous donnons un exemple du nombre de chemins possibles si on se place dans la dernière position du dictionnaire C_8 (figure II.6). D'après le tableau II.1 on devrait avoir sept (7) chemins possibles et on peut remarquer que si on se place dans les deux positions 7 et 8 du dictionnaire C_9 on aura respectivement 4 et 3 chemins possibles pour atteindre le mot-code au niveau du dictionnaire C_{10}

Tableau II.1 Le nombre des chemins possibles. Par exemple si on veut quantifier le coefficient LSF_1, N_2 donne le nombre de chemins possibles à n'importe quelle position du dictionnaire C_1 pour continuer la quantification des autres coefficients. Le nombre de bits nécessaires peut être obtenu par la conversion du nombre issu de la sommation des chemins *N2*

	N_i : Nombres de chemin possibles partant du dictionnaire C_{i-1}									
Position dans le dictionnaire C_{i-1}	**N2**	**N3**	**N4**	**N5**	**N6**	**N7**	**N8**	**N9**	**N10**	**N11**
1	97762793	8512383	896795	96022	9455	1904	315	52	8	1
2	97762793	8512383	896795	96022	9455	1904	315	52	8	1
3	89250410	8512383	896795	96022	9455	1904	315	52	8	1
4	80738027	8512383	896795	96022	9455	1589	315	52	8	1
5	72225644	8512383	896795	96022	9455	959	263	44	7	1
6	55200878	8512383	800773	96022	9455	644	211	36	6	1
7	38176112	8512383	704751	77112	9455	381	107	20	4	1
8	23841731	7615588	608729	67657	7551	170	63	7	3	1
9		6718793	512707	58202	5647					
10		5821998	416685	39292	5647					
11		4925203	320663	29837	3743					
12		4028408	243551	22286	2154					
13		3227635	175894	10992	2154					
14		2522884	117692	7249	1195					
15		1914155	78400	5095	1195					
16		1401448	48563	2941	551					

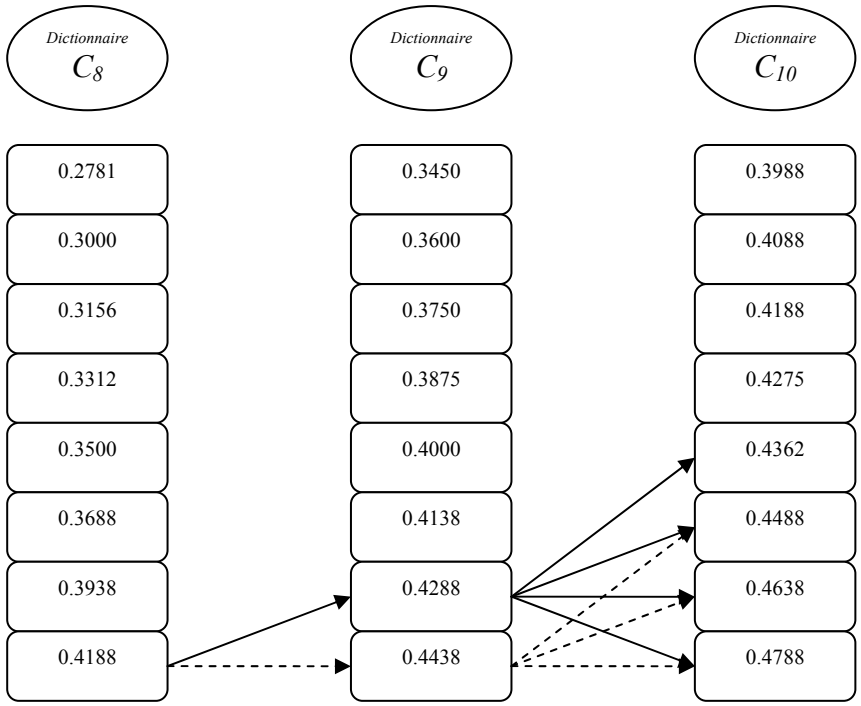


Figure. II.7 Exemple donnant les chemins possibles partant de la $8^{ème}$ position du dictionnaire C_8 pour atteindre le dernier mot-code au niveau du dictionnaire C_{10}. On remarque que ce nombre de chemins est égal à 7 comme indiqué dans le tableau I.1. Aux deux positions 7 et 8 du du dictionnaire C_9, on aura respectivement 4 et 3 chemins possibles pour atteindre le mot-code au niveau du dictionnaire C_{10}.

II.4.1 Algorithme de quantification

Dans ce qui suit on donne l'algorithme de quantification par énumération en treillis pour le quantificateur scalaire de la norme FS1016.

On admet que les valeurs N_i sont calculées comme précédemment (tableau II.1). On notera par t le mot-code de l'énumération en treillis qui sera initialisé à zero.

Pour une meilleure compréhension, nous prenons l'exemple de quantification du vecteur LSF suivant :

LSF = [0.2571 0.5465 0.8155 1.0191 1.4705 1.6650 1.9177 2.0867 2.4067 2.5593] après analyse d'une trame de 30 ms du signal parole.

On donne l'algorithme de quantification par énumération en treillis pour les coefficients LSF d'ordre 10 :

Etape 1 *Trouver j tel que* $QS_{34}(LSF_1) = L\hat{S}F_{1,j}$ *avec* $1 \le j \le 2^{b_1}$ *et mettre* $j^* = j\text{-}1$

Etape 2 *Initialiser t = 0 si* $j^* = 1$ *sinon* $t = \sum_{k=1}^{j^*} N_2(k)$

Etape 3 *Pour i = 2 à p*

-Mettre $m = M_i(L\hat{S}F_{i-1,j^*})$

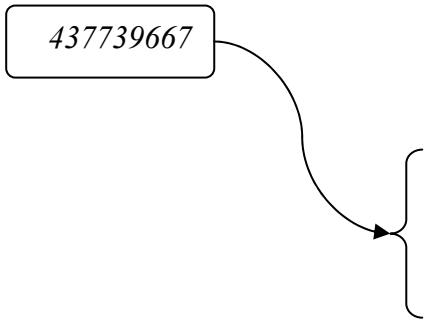
- Trouver j tel que $QS_{34}(LSF_i) = L\hat{S}F_{i,j}$ *avec* $1 \le j \le 2^{b_i}$ *et mettre* $j^* = j\text{-}1$

- Mettre $t = t + 0$ *si* $j^* = m$ *sinon* $t = t + \sum_{k=m}^{j^*} N_{i+1}(k-1)$

Etape 4 *Le mot-code d'énumération en treillis est donné par t.*

Le déroulement de l'opération de quantification suit les étapes suivantes :

Dans les deux premières étapes, on cherche le mot-code du premier coefficient LSF_1 en utilisant la quantification scalaire du FS1016. On s'intéresse ici à son indice. Dans notre cas, il s'agit de l'indice $j=6$ et par conséquent on aura : $j^*=6\text{-}1=5$. Il est clair que $j^* \neq 1$. Donc on calculera t d'après le tableau II.1 comme suit : $t = \sum_{k=1}^{5} N_2(k) = 437739667$ (figure II.8).

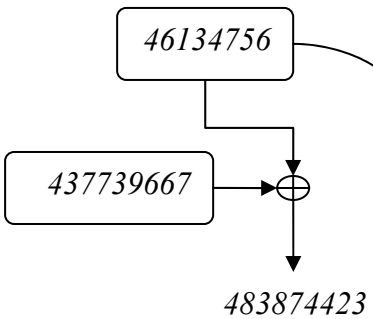


N_i : Nombres de chemin possibles partant du dictionnaire C_{i-1}
N2
97762793
97762793
89250410
80738027
72225644
55200878
38176112
23841731

Figure II.8 Extrait du tableau II.1 montrant le calcul du mot-code *t* pour le premier coefficient LSF

Dans une troisième étape, on déroule une boucle qui commence du coefficient LSF_2 pour atteindre le coefficient LSF_{10}. Pendant le déroulement de cette boucle, on calcule la valeur de m telle que $m = M_i(\hat{LSF}_{i-1,j^*})$. Dans le cas du LSF_2 , $m = 6$. Par la suite on cherche l'indice j du représentant du coefficient LSF_2. Dans cet exemple $j = 13$ et on aura alors : $j^* = j - 1 = 12$.

D'après l'algorithme précédent on remarque que $j^* \neq m$. On continue alors le calcul de t comme suit : $t = t + \sum_{k=m+1}^{j^*} N_{i+1}(k-1)$. Ainsi, pour le cas du LSF_2 on a $t = t + \sum_{k=m}^{12} N_3(k-1)$ $= 483874423$ (figure II.9).



N_i : Nombres de chemin possibles partant du dictionnaire C_{i-1}
N3
8512383
8512383
8512383
8512383
8512383
8512383
8512383
7615588
6718793
5821998
4925203
4028408
3227635
2522884
1914155
1401448

Figure II.9 Extrait du tableau I.1 montrant le calcul du mot-code *t* pour le deuxième coefficient LSF

Les mêmes opérations seront reprises par les coefficients qui restent et on obtient les résultats suivants :

LSF_3	$j^* = 10$	$m = 7$	$t = 486117295$
LSF_4	$j^* = 8$	$m = 7$	$t = 486262064$
LSF_5	$j^* = 11$	$m = 5$	$t = 486313017$
LSF_6	$j^* = 5$	$m = 5$	$t = 486313976$
LSF_7	$j^* = 5$	$m = 5$	$t = 486314239$
LSF_8	$j^* = 3$	$m = 3$	$t = 486314291$
LSF_9	$j^* = 3$	$m = 1$	$t = 486314315$
LSF_{10}	$j^* = 1$	$m = 1$	$t = 486314316$

A ce niveau, et à la quatrième étape, on transmettra l'indice t qui est en réalité codé sur 30 bits. Dans ce cas, l'indice t vaut (en binaire) = 01110011111100100100010100110 0.

II.4.2 Algorithme de reconstruction

A la réception au niveau du décodeur un nombre *t,* codé sur 30 bits, est réceptionné pour l'opération de décodage. Dans le cas de l'exemple précédent, t a été trouvé égal à *486314316*

Edans les deux premières étapes, on cherchera l'indice j tel que t soit borné comme suit :

$$\sum_{k=1}^{j-1} N_2(k-1) \le t < \sum_{k=1}^{j} N_2(k-1) \qquad \text{(II.8)}$$

D'après le tableau II.1 on remarque que la valeur 5 vérifie cette relation car :

$$\sum_{k=1}^{5} N_2(k-1) = 437739667$$
$$\sum_{k=1}^{6} N_2(k-1) = 492940545 \qquad \text{(II.9)}$$

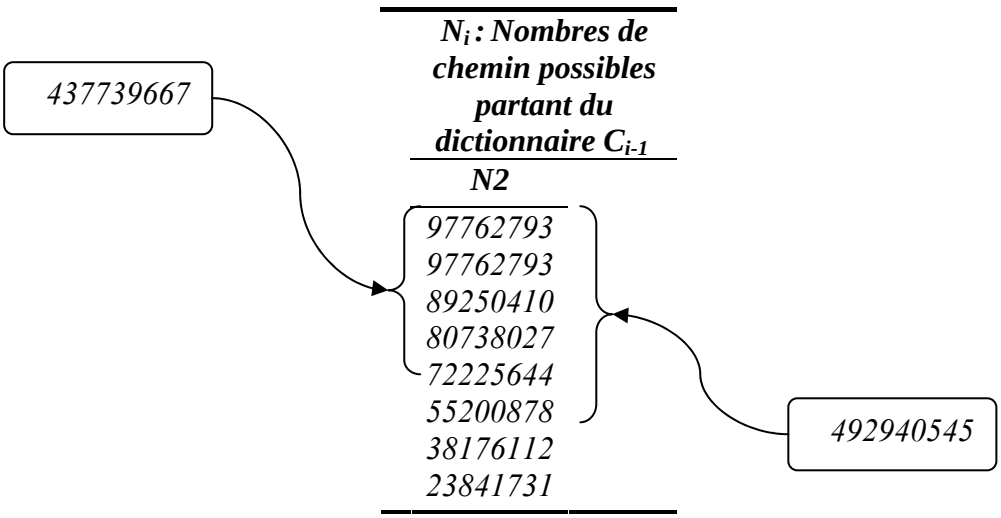


Figure II.10 Extrait du tableau I.1 montrant comment localiser l'indice du premier coefficient LSF

Ainsi le représentant du premier coefficient a un indice égal à *6* d'où $L\hat{S}F_1 = 0.2670$, et on aura d'après l'algorithme l'indice j^* égal à *6 – 1 =5*.

En troisième étape on retranchera de *t* la valeur $\sum_{k=1}^{5} N_2(k-1) = 437739667$ pour avoir un nouveau *t = 48574649*.

A l'étape quatre, une boucle, qui commencera à partir du coefficient LSF_2 jusqu'au dernier coefficient, itérant les opérations suivantes (comme dans le cas du deuxième coefficient) :

- Trouver *m* tel que $m = M_2(L\hat{S}F_{1,6})$. D'après le dictionnaire C_2 de la quantification scalaire du FS1016 *m* vaut *6*.

- On remarque que $t \neq 0$. On cherchera alors *j* vérifiant

$$\sum_{k=m}^{j-1} N_3(k-1) \leq t < \sum_{k=m}^{j} N_3(k-1) \qquad \text{(II.10)}$$

soit *j=13* d'après le tableau II.1 avec les deux bornes inférieure et supérieure égales respectivement à *46134756* et à *49362391*(figure II.11).

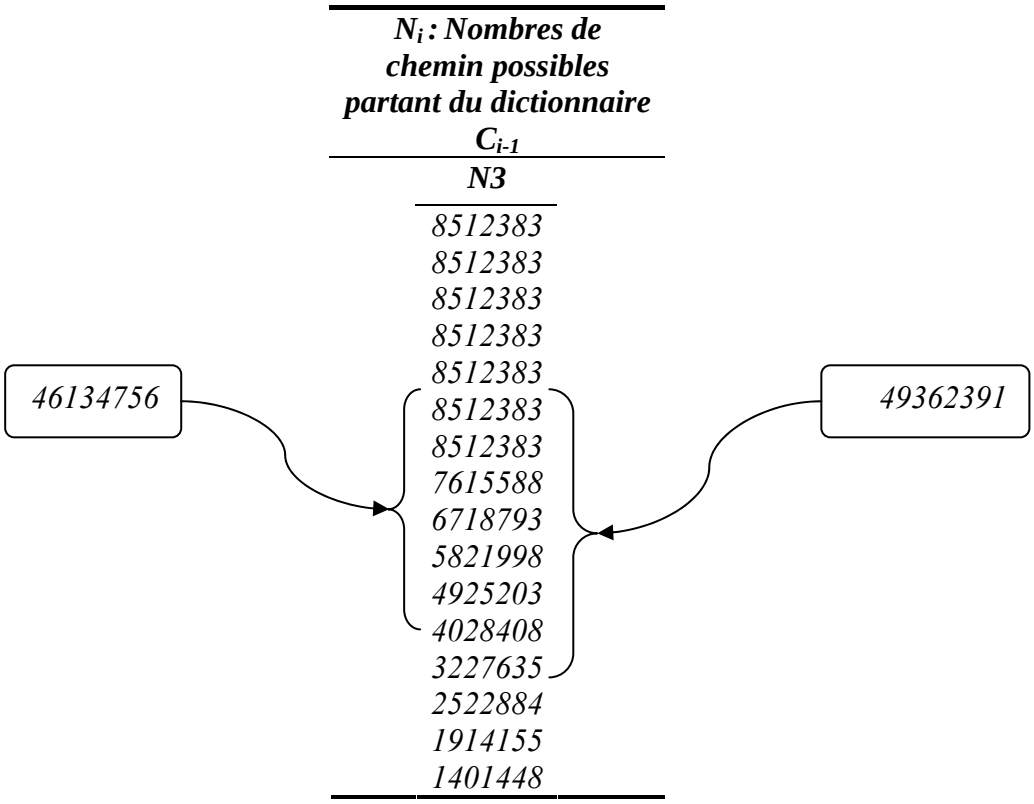


Figure II.11 Extrait du tableau I.1 montrant comment localiser l'indice du deuxième coefficient LSF

- On met $j^* = j - 1 = 12$ et on donne $\hat{LSF}_2 = 0.5262$

- on remarque que $j^* \neq m.\ t$ sera alors comparé à *46134756* et ce pour avoir un nouveau *t* égale à *2439893 (t-46134756 = 2439893).*

Les mêmes étapes que précédemment seront réitérées pour le reste des coefficients *LSF*.

II.5 Résultats et interprétation

Des tests sur le standard FS1016, aussi bien avec la quantification scalaire qu'avec la quantification par système d'énumération en treillis, ont été effectués sur deux bases de données de parole, l'une Américaine (TIMIT) [34] et l'autre Arabe (PAPE) [35].

TIMIT est une base de données de parole échantillonnée à 16Khz comprenant huit dialectes des Etats-Unis (USA). Le tableau II.3 Présente le nombre des locuteurs pour les 8

régions des Etats Unis d'Amérique. La région de dialecte d'un locuteur donné est la région géographique des Etats-Unis où il vivait pendant son enfance.

En pratique, et dans le codage à très bas débit nous utilisons des fichiers de parole échantillonnés à 8 KHz, ce qui nous a conduit à sous échantillonner les fichiers de TIMIT à 8KHz.

Tableau II.2 Distribution des locuteurs dans la base de données TIMIT
On remarque le nombre de phrases prononcées et leurs pourcentages

Indices des dialectes des régions	Nombre de phrases par région et selon le sexe		Total de phrases par région
	Masculin	Féminin	
DR1	31 (63%)	18 (27%)	49 (8%)
DR2	71 (70%)	31 (30%)	102 (16%)
DR3	79 (67%)	23 (23%)	102 (16%)
DR4	69 (69%)	31 (31%)	100 (16%)
DR5	62 (63%)	36 (37%)	98 (16%)
DR6	30 (65%)	16 (35%)	46 (7%)
DR7	74 (74%)	26 (26%)	100 (16%)
DR8	22 (67%)	11 (33%)	33 (5%)
Total	438 (70%)	192 (30%)	630 (100%)

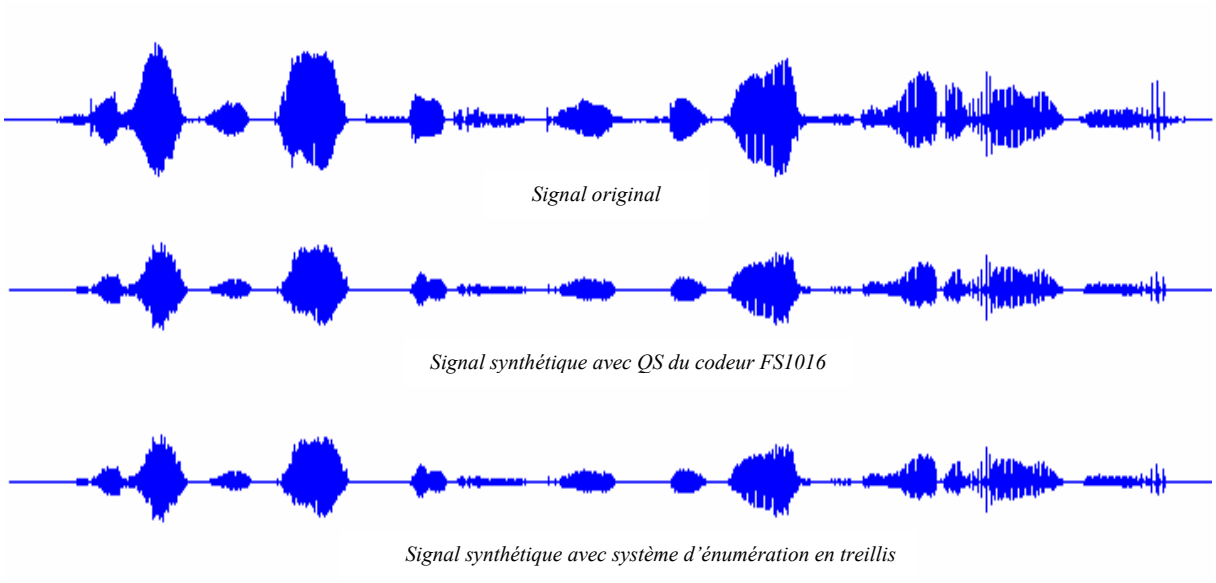
Une autre base de données de parole arabe PAPE (Phrases Arabes Phonétiquement Equilibrées) a été aussi utilisée dans les tests objectifs. Son enregistrement a été effectué en chambre sourde à l'institut ICP de Grenoble par six locuteurs algériens (trois masculins et trois féminins) parlant correctement l'arabe et chacun d'eux a prononcé dix phrases. Les fichiers parole étaient échantillonnés à 10 KHz. Les 10 phrases arabes ont été extraites d'un corpus de 200 phrases phonétiquement équilibrées :

- نمـنم الماء اليوم
- و لن يـشفقا عنها
- أسكت يا خرود !
- كم بعـت ثمرنا ؟
- إذ ظنـنـته جنبنا
- صعد الإمام فوق المنبر
- لا تنسـى أكل برتقالتك
- أذاه زحف رمله
- و سألني هل ضغط
- الويل من لـبنى

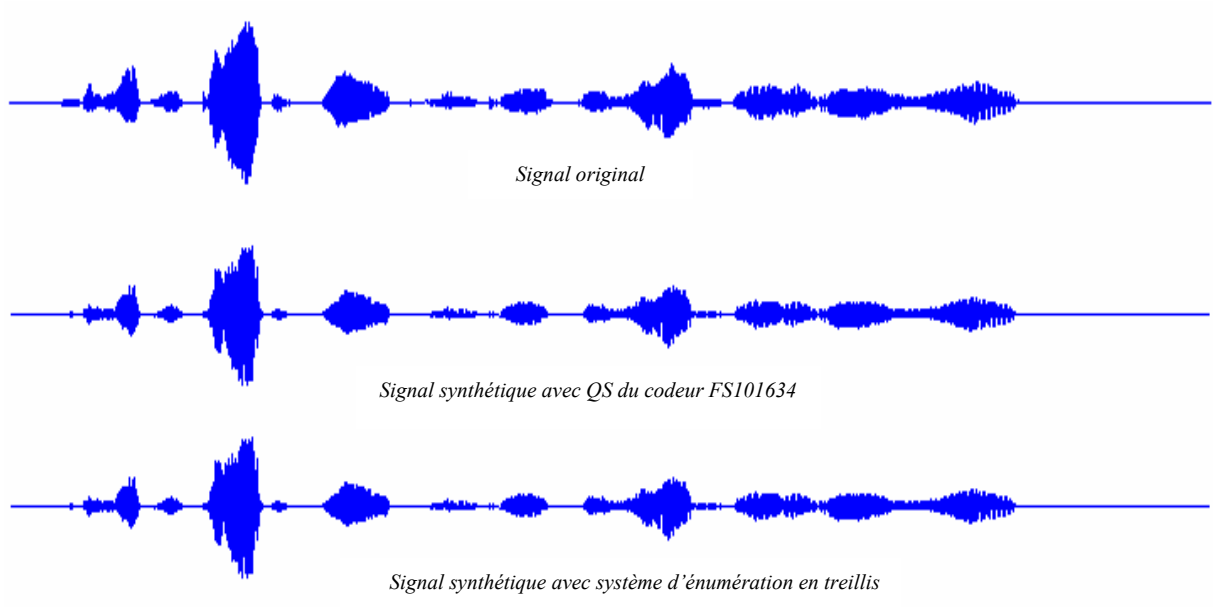
Nous présentons les résultats obtenus pour chacune des deux bases de données, pour un locuteur masculin et pour un locuteur féminin, et ceci pour chacun des deux systèmes à savoir, les standard FS1016 à quantificateur scalaire et le FS1016 avec le système d'énumération en treillis où nous avons observé et écouté une même qualité de parole synthétique (Figures II.12.a, II.12.b, II.13.a et II.13.b).

Les phrases choisies sont:

- She had your dark suit in greasy wash water all year



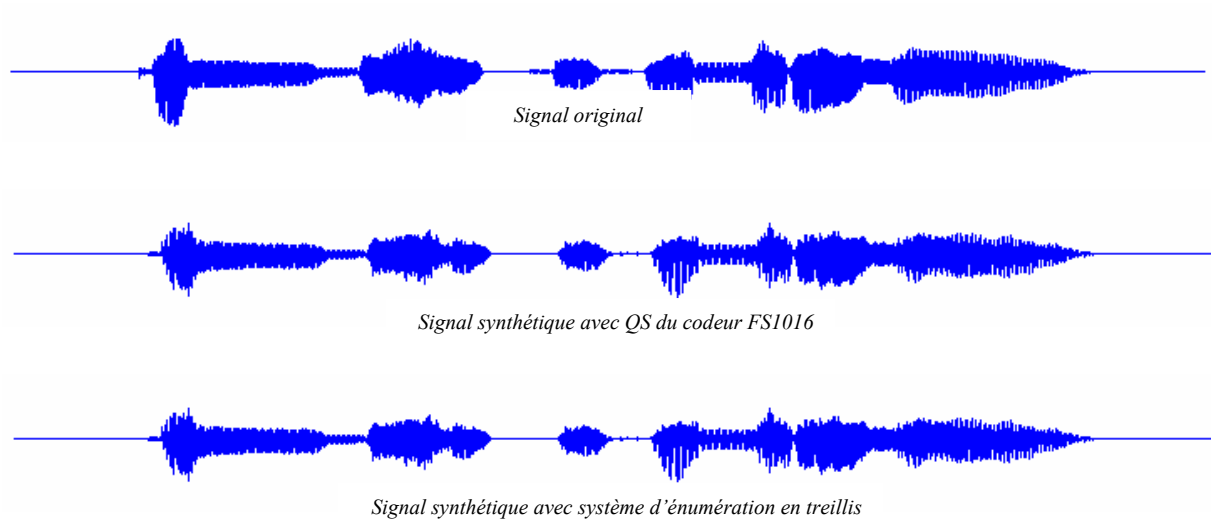
a) Locuteur féminin prononçant la phrase "She had your dark suit in greasy wash water all year"



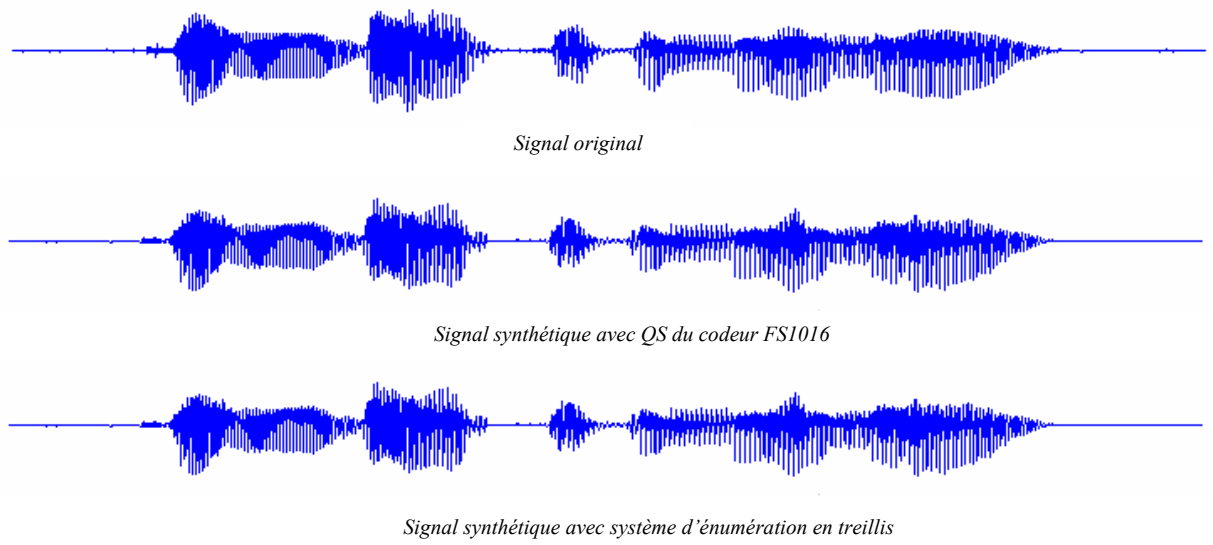
b) Locuteur masculin prononçant la phrase " She had your dark suit in greasy wash water all year "

Figure II.12 Signaux originaux et synthétiques originaux à la sortie du codeur, de la base de données TIMIT.

- كم بعـت ثمرنا ؟



(a) Locuteur féminin prononçant la phrase
" كم بعـت ثمرنا ؟"



(b) Locuteur masculin prononçant la phrase
" كم بعـت ثمرنا ؟"

Figure II.13 Signaux originaux et synthétiques à la sortie du codeur implanté pour la base de donnée arabe PAPE.

Conclusion

Dans ce chapitre, nous avons présenté un système d'énumération en treillis que nous proposons pour la quantification des coefficients LSF et qui vise l'amélioration du

quantificateur scalaire de la norme FS1016. L'algorithme a été détaillé. Nous avons montré, par un exemple, que ce système a diminué le nombre de bits nécessaire à la quantification de 34 bits à 30 bits.

En exploitant certaines propriétés pertinentes à savoir l'ordre ascendant des coefficients LSF et dans le but de diminuer encore plus ce nombre de bits pour la quantification des coefficients LSF nous avons proposé une forme de quantification plus adéquate que celle utilisée dans le standard FS1016, à savoir la quantification vectorielle qui fera l'objet du prochain chapitre.

.

Chapitre III

La quantification vectorielle

Lorsque le débit visé pour coder un signal de parole est très bas, le nombre de bits nécessaires par échantillon doit impérativement suivre la même logique. En effet il est possible d'avoir des débits fractionnaires avec la quantification vectorielle. Cependant, pour coder un signal de parole à bas débit, il devient nécessaire de regrouper les paramètres candidats à la quantification et quantifier l'ensemble. On parlera alors de quantification vectorielle.

III.1 Principe de la quantification vectorielle

La quantification vectorielle est connue comme une technique de codage de source efficace dans plusieurs applications de la communication digitale. Elle joue un rôle important dans le domaine du codage de la parole et de l'image à bas débit et dans les systèmes de communication vidéo. A l'inverse de la quantification scalaire qui quantifie chaque paramètre ou valeur d'un signal d'une manière indépendante, la quantification vectorielle quantifie un ensemble de paramètres conjointement comme un seul vecteur. Shannon [36, 37, 38] a démontré que pour un débit donné, coder des blocs d'information permet d'avoir une bonne performance en terme de distorsion.

Ainsi la quantification vectorielle encode chaque vecteur d'une séquence de vecteur source par un indice de vecteur code choisi dans un ensemble fini.

La quantification vectorielle est une extension da la quantification scalaire dans un espace multidimensionnel. En effet, alors que la quantification scalaire affecte une seule valeur d'un ensemble de sorties à une seule entrée, la quantification vectorielle effectue une

correspondance à un bloc de valeurs, ou un vecteur, un vecteur code d'un ensemble de vecteurs de sorties.

Le quantificateur vectoriel est défini par un ensemble fini C de niveaux de quantification (appelé vecteurs codes) et par une fonction qui associe un niveau de quantification (un vecteur code) à chaque vecteur de N échantillons produit par la source.

Un quantificateur vectoriel de dimension N et de taille L peut être défini mathématiquement comme une application Q de l'espace euclidien R^N vers un ensemble fini (dictionnaire) C [21], qui affecte à un vecteur source x un vecteur code y_i de même dimension.

$$\begin{cases} R^N \rightarrow C \\ x \rightarrow Q(x) = y_i \end{cases} \tag{III.1}$$

L'espace est partitionné en L classes R_i tel que :

$$R_i = \{x / Q(x) = y_i\} \tag{III.2}$$

C est le dictionnaire, et les $\boldsymbol{y_i}$ sont des vecteur codes.

Le partitionnement de l'espace des vecteurs de x en L régions dite de Voronoi se fait selon :

$$R_i = \{x / d(x,y_i) < d(x,y_j) , j \neq i\} \tag{III.3}$$

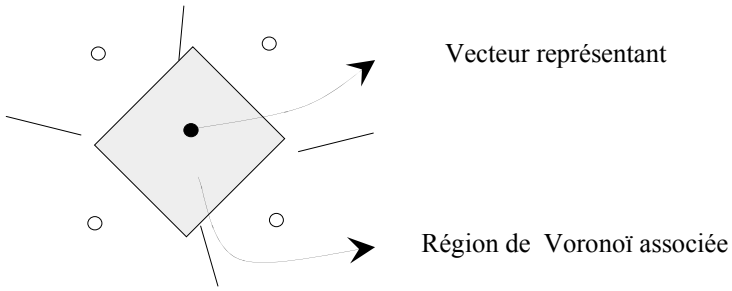


Figure III.1 Principe de la quantification vectorielle.

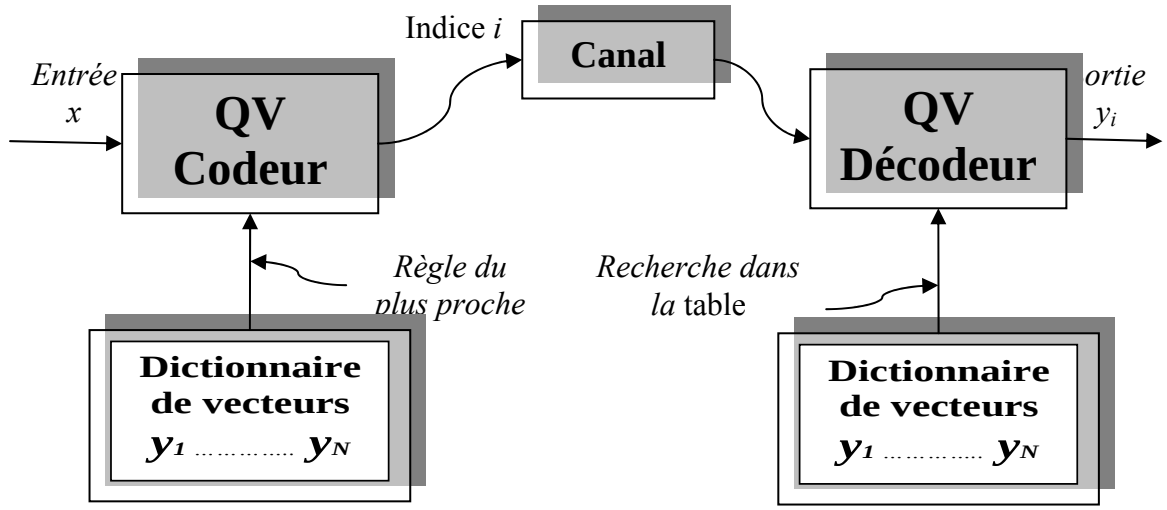


Figure III.2 Modèle de la quantification vectorielle

III.2 Condition d'optimalité pour la quantification vectorielle

La performance de la quantification vectorielle dépend de la partition de l'espace pour le codeur, et la reproduction des vecteurs pour le décodeur.

Un quantificateur vectoriel est optimal quand la distorsion moyenne D est minimale pour un vecteur d'entrée x.

$$D = E(d(x, Q(x))) \tag{III.4}$$

Des méthodes itératives sont disponibles pour la conception d'un quantificateur vectoriel. Ces méthodes nécessitent deux conditions d'optimalité dans leurs conceptions

III.2.1 Condition du plus proche voisin

Soit un quantificateur vectoriel, caractérisé par son dictionnaire des niveaux de sorties C. Lors de l'apprentissage, les partitions optimales peuvent êtres définies comme étant les sous ensembles $\{ R_i \}$ de cette base, qui satisfait [39, 40]:

$$R_i = \{x / d(x, y_i) < d(x, y_j) \forall j \neq i\} \tag{III.5}$$

En d'autre terme la partition est représentée par un vecteur y_i dans C vérifiant :

$$Q(x) = y_i \text{ Seulement si } d(x, y_i) < d(x, y_j) \forall j \neq i \qquad \text{(III.6)}$$

III.2.2 Condition des centroïdes

Soit une base d'apprentissage partitionnée en classes *{Ri | i = 1 ……N}*. Les vecteurs codes les plus optimums de ces classes sont les centres de gravité y_i de chacune d'elles [39, 40] :

$$y_i = \text{Centre de } \{Ri\} \qquad \text{(III.7)}$$

III.3 Conception du dictionnaire de la QV

III.3.1 Algorithme de Lloyd Généralisé (GLA)

Les deux conditions précédentes conduisent à la conception d'un algorithme qui réalise une construction d'un dictionnaire (localement) optimal à partir d'une séquence d'apprentissage représentative de la statistique de la source à coder. Cet algorithme de classification, encore appelé algorithme de K-moyennes [39, 41] est l'extension au cas vectoriel de l'algorithme de Lloyd-Max du cas scalaire. Il s'agit d'un algorithme d'optimisation itératif opérant à partir d'un dictionnaire initial. A chaque itération (dite « itération de Lloyd »), deux opérations distincts sont appliquées :

- Une classification suivant la règle du codage optimal ou « condition du plus proche voisin ».
- Une optimisation suivant la règle du décodage optimal ou « condition des centroïdes ».

Algorithme de Lloyd Généralisé

Etape 1 *Commencer avec un dictionnaire initial* C_0

Etape 2 *Pour un dictionnaire* C_m *donné, utiliser l'itération de Lloyd pour produire un nouveau dictionnaire* C_{m+1}

Etape 3 *Calculer la distorsion moyenne* D_{m+1} *pour* C_{m+1} *et voir son changement par rapport à la dernière itération*

Si $\frac{D_m - D_{m+1}}{D_m} \leq \xi$ *(*$\xi = 0.0001$ *dans notre cas)* ***stop*** *sinon faire m = m +1 et aller à l'****Etape 2***

Etape 2 appelée itération de Lloyd peut être écrite comme suit :

Itération de Lloyd

Etape 2a *Pour un dictionnaire C_m donné, partitionner la base d'apprentissage en cellules Ri en utilisant la règle du plus proche voisin*

$$R_i = \{x / d(x, y_i) \leq d(x, y_j) \forall j \neq i\}$$

Etape 2b *Utiliser la condition des centroïdes pour calculer les centres des cellules trouvées dans* ***Etape 2a*** *et cela pour obtenir un nouveau dictionnaire*

$$C_{m+1} = \{centre\ de\ R_i \mid i = 1 \ldots .L\}$$

III.3.2 Dictionnaire initial

Le choix du dictionnaire initial est capital pour une quantification vectorielle efficace car il conditionne les résultats finaux de l'algorithme GLA ainsi que la vitesse de convergence de ce dernier.

Il y a une variété de techniques pour produire le dictionnaire initial dont on cite :

- **Une initialisation aléatoire**

Le dictionnaire le plus simple est celui qui contient les *L* premiers vecteurs de la suite d'apprentissage ou *L* vecteurs extraits aléatoirement de cette suite. Ces vecteurs peuvent bien sûr ne pas être représentatifs de la suite d'apprentissage, et conduire à des résultats très médiocres, spécialement du point de vue vitesse de convergence. Par ailleurs le temps de traitement peut être très grand.

- **Un algorithme à seuil**

Une distance minimale est fixée entre les éléments du dictionnaire initial. Cette méthode permet d'obtenir une meilleure représentativité comparativement au précédent, mais

elle n'est pas toujours satisfaisante. On effet, le seuil est souvent difficile à déterminer puisque dépendant de la complexité de la séquence d'apprentissage.

- **Méthode par dichotomie vectorielle**

Référencée comme étant l'algorithme de LBG [41], Elle combine à l'itération de Lloyd une technique dite de "Splitting" appliquée initialement au centroïde de la base d'apprentissage.

Le "splitting" consiste à découper chaque vecteur code y_i en 2 nouveaux vecteurs yi+ε et y_i - ε (ε étant un vecteur de perturbation de faible énergie) pour former un dictionnaire de deux vecteurs. On applique à ce nouveau dictionnaire obtenu les itérations de Lloyd. L'algorithme génère ensuite une succession de dictionnaires (à chaque boucle le nombre de vecteurs de reproduction est multiplié par 2, et l'arrêt des itérations se fait à la suite d'un test vérifiant que la distorsion moyenne D ne varie pas plus qu'une valeur très petite fixée préalablement.

Cet algorithme permet :

- De résoudre le problème du choix du dictionnaire initial
- De construire des dictionnaires de taille $L=2^k$ qui se prêteront à une quantification vectorielle arborescente (par choix successifs entre 2 vecteurs-codes)
- D'accélérer non seulement la construction du dictionnaire (qui ne se fait qu'une fois pour une base d'apprentissage donnée) mais aussi la quantification (si on utilise la structure arborescente des dictionnaires successifs).

IV.3.3 Variante de l'algorithme de Katsavounidis

Une dernière méthode à illustrer, est une variante de la méthode proposée par Katsavounidis *et al* [40, 42], nommée la méthode des centroïdes et dont l'algorithme est le suivant :

Algorithme de la variante de Katsavoundis pour l'initialisation du dictionnaire

Etape 1 *Calculer le centroïde de la base d'apprentissage.*

Etape 2 *Calculer la distance euclidienne simple entre chacun des vecteurs de la base d'apprentissage et le centroïde précédent.*

Etape 3 *Choisir le vecteur le plus loin du centroïde précédent comme vecteur référence.*

Etape 4 *Calculer la distance euclidienne simple entre chacun des vecteurs de la base d'apprentissage et le vecteur reférence.*

Etape 5 *Soit M/N le rapport du nombre de vecteur dans la base d'apprentissage et le nombre des vecteurs dans le dictionnaire. Repérer les M/N vecteurs plus proches du vecteur référence, et calculer leur centroïde (le premier vecteur dans le dictionnaire initial).*

Etape 6 *Réduire la base d'apprentissage en retranchant le groupe de vecteurs précédents. et calculer le centroïde du reste de la base d'apprentissage*

Etape 7 *Répéter les étapes de 2 à 7 jusqu'à qu'il ne reste aucun vecteur.*

Le principe de cette technique consiste à accorder une attention particulière aux vecteurs de la base d'apprentissage qui sont les plus éloignés l'un de l'autre, car ils sont susceptibles d'appartenir à des classes différentes.

A noter que cette technique a été utilisée dans tous les travaux de cette thèse.

III.4 Les différentes méthodes d'évaluation pour les phases d'apprentissage et de tests

Les mesures de distorsion jouent un rôle très important dans le codage de la parole et peuvent se diviser en deux groupes de mesures [1] :

- Mesures objectives
- Mesures subjectives

Dans notre étude, nous sommes intéressé à la distorsion spectrale moyenne et à la distance euclidienne.

III.4.1 Mesures objectives

III.4.1.1 Le rapport signal sur bruit

Le *RSB (*ou en anglais *SNR)* [40, 43], peut être défini comme le rapport entre l'énergie du signal de parole d'entrée et l'énergie du signal du bruit mesuré en décibel (dB):

$$SNR = 10 \log_{10} \frac{E_s}{E_\varepsilon} = 10 \log_{10} \frac{\sum_{n=-\infty}^{\infty} s^2(n)}{\sum_{n=-\infty}^{\infty} (s(n) - \hat{s}(n))^2} \quad dB \qquad \text{(III.8)}$$

Où :

$s(n)$ est le signal de parole d'entrée

$\hat{s}(n)$ est le signal de parole synthétisé.

III.4.1.2 Le rapport signal sur bruit segmental

Pour évaluer la qualité de la parole synthétisée, la mesure la plus couramment utilisée dans le domaine temporel est le rapport signal sur bruit segmental *SNRseg*. Le signal est découpé en N_F segments de N échantillons chacun et on calcule une moyenne des *RSB* obtenus sur les N_F segments.

$$SNRseg = \frac{1}{N_F} \sum_{i=0}^{N_F-1} 10 \log_{10} \left(\frac{\sum_{j=0}^{N-1} s(N_i + j)^2}{\sum_{j=0}^{N-1} \left(s(N_i + j) - \hat{s}(N_i + j) \right)^2} \right) \text{(dB)} \qquad \text{(III.9)}$$

N_i étant le numéro de segment

III.4.1.3 La distorsion spectrale

La distorsion spectrale est une autre mesure objective dans le domaine spectrale. La distorsion spectrale pour une trame de parole i est définie par [39]:

$$SD_i = \sqrt{\frac{1}{F_s} \int_0^{F_s} \left[10 \log_{10} \left(\frac{S_i(f)}{\hat{S}_i(f)} \right) \right]^2 df} \text{ (dB)} \qquad \text{(III.10)}$$

Où F_s est la fréquence d'échantillonnage, $S_i(f)$ et $\hat{S}_i(f)$ sont les densités spectrales estimées à partir des coefficients de prédiction de la trame numéro i et sont donnés par :

$$S_i(f) = \frac{1}{\left|A_i(e^{j2\Pi f/F_s})\right|^2} \quad \text{(III.11)}$$

$$\hat{S}_i(f) = \frac{1}{\left|\hat{A}_i(e^{j2\Pi f/F_s})\right|^2} \quad \text{(III.12)}$$

$A_i(z)$ et $\hat{A}_i(z)$ sont les polynômes correspondants respectivement aux coefficients de prédiction originaux et aux coefficients de prédictions quantifiés pour la $i^{ème}$ trame de parole. En passant à la notation numérique, l'équation (III.10) devient [39] :

$$SD_i = \sqrt{\frac{1}{n_1 - n_0}\sum_{n=n_0}^{n_1-1}\left[10\log_{10}\left(\frac{S_i\left(e^{j2\Pi n/N}\right)}{\hat{S}_i\left(e^{j2\Pi n/N}\right)}\right)\right]} \text{ (dB)} \quad \text{(III.13)}$$

Les densités spectrales sont évaluées par DFT (Transformée de Fourier Discrète) en utilisant l'algorithme FFT. n_0 et n_1 correspondent respectivement à 1 et 96, si la distorsion SD_i est calculée dans l'intervalle de fréquence allant de 0Hz à 3kHz (le signal est échantillonné à 8kHz et on utilise N=256 points pour le calcul de la DFT, la résolution entre deux points est donc de 8kHz/256 = 31,25 Hz).

Paliwal et Atal [24] ont montré que cette moyenne n'est pas adéquate et ils ont introduit la notion de la distorsion des paramètres des vecteurs éloignés en calculant le pourcentage des vecteurs pour lesquels la distorsion spectrale (SD) est entre 2 dB et 4dB et le pourcentage des vecteurs ayant la SD supérieur à 4dB. Ces vecteurs ont une appellation Anglo-saxonne qui est " Outliers".

III.4.1.4 La distance euclidienne pondérée pour les LSF

La distance euclidienne pondérée est une mesure qui utilise les coefficients LSF [44]. En effet les coefficients LSF ont une relation avec les formants et les vallées de l'enveloppe spectrale calculée par l'équation (III.14). Si f et $\hat{f}$ sont deux vecteurs composés chacun de P

coefficients LSF originaux et quantifiés respectivement, alors leurs distance euclidienne pondérée et définie par [24]:

$$d(f,\hat{f}) = \sum_{i=1}^{P} c_i \omega_i (f_i - \hat{f}_i)^2 \qquad \text{(III.14)}$$

ω_i est le poids assigné au $i^{ème}$ coefficient LSF, défini par Paliwal et Atal [24] par:

$$\omega_i = |S(f_i)|^r \qquad \text{III.15)}$$

Où $S(f_i)$ est la densité spectrale LPC estimée à la fréquence f_i et associée au vecteur à tester. r est une constante empirique qui contrôle le poids des différents coefficients LSF et est déterminée expérimentalement.

Paliwal et Atal ont trouvé que *r=0.15* est une valeur satisfaisante. Donc la pondération dépend de la valeur de la densité spectrale LPC à la fréquence du coefficient LSF considérée.

Une très simple relation donnée par Rajiv Laroi et *al* [45] ont proposé une relation simple pour exprimer ces poids :

$$\omega_i = \left(\frac{1}{f_{i+1} - f_i} + \frac{1}{f_i - f_{i-1}} \right) \qquad \text{(III.16)}$$

$$Avec\ f_0 = 0\ et\ f_{p+1} = \pi$$

L'oreille humaine est plus sensible aux basses fréquences qu'aux hautes fréquences. Pour exploiter cette caractéristique, Paliwal et Atal [24] ont introduit un coefficients de pondération constant défini par :

$$c_i = \begin{cases} 1.0, & 1 \le i \le 8 \\ 0.8, & i = 9 \\ 0.4, & i = 10 \end{cases} \qquad \text{(III.17)}$$

On peut remarquer que les poids c_i sont fixes alors que les poids ω_i sont des poids adaptatifs à cause de leur changement d'une trame de parole à une autre.

III.4.2 Mesures subjectives

Les essais d'écoute sont nécessaires car le récepteur humain reste le dernier bloc d'un système de codage de parole. Ces mesures sont basées sur l'opinion d'une personne ou d'un groupe de gens qui écoutent et donnent leur avis sur l'intelligibilité de la parole décodée. Cette méthode n'est pas toujours fiable à cause du temps nécessaire aux essais. Cependant, les méthodes les plus utilisées sont [1]:

- Diagnostic Rhyme Test (DRT) qui mesure l'intelligibilité sur un grand nombre de mots.

- Mean Opinion Score (MOS) où l'auditeur évalue un codeur sur une échelle absolue allant de 1 à 5 avec :

  - 1 : Mauvais - 2 : Médiocre
  - 3 : Passable - 4 : Bon
  - 5 : Excellent

Ainsi, on attribue un seul niveau à chaque signal de parole à évaluer durant la procédure d'évaluation subjective.

Puisque la mesure subjective est délicate à réaliser, on se contente en général des mesures objectives dans l'évaluation.

Conclusion

Dans ce chapitre, nous avons introduit la notion de quantification vectorielle qui est utilisée dans les codeurs de parole à bas et très bas débit et en particulier pour coder les paramètres spectraux de la parole. Une introduction à la quantification vectorielle et les différentes méthodologies de conception des dictionnaires a été présentée. En outre, nous avons exposé les mesures de tests objectifs et subjectifs utilisés dans le codage de parole. Dans le chapitre qui va suivre, nous étudierons cinq méthodes à base de la quantification vectorielle qu'on évaluera par les tests objectifs à base de distorsion spectrale et de distances euclidiennes simple et pondérée.

Chapitre IV

Mise en œuvre de cinq méthodes de Quantification vectorielle des coefficients LSF

Une bonne qualité de la parole a été remarquée lors de l'utilisation de la quantification vectorielle simple codée sur 24 bits pour la quantification des coefficients LSF. Cependant deux problèmes sont rencontrés.

Le premier problème est la recherche très complexe au sein du dictionnaire dont la taille est égale à $2^{24} = 16777216$ vecteurs.

Le deuxième problème est l'apprentissage du dictionnaire qui exige une base d'apprentissage de taille supérieure à 50 fois la taille du dictionnaire. En effet, Mahkoul [39, 40, 46] a donné une règle pour un bon apprentissage des dictionnaires de la quantification vectorielle. Cette règle présente deux valeurs pour les tailles des dictionnaires pour leurs bons apprentissages. Pour le cas des dictionnaires de taille $2^{24} = 16777216$, une valeur inférieure égale à 838860800 vecteurs et une valeur supérieure égale à 3355443200 vecteurs.

Des travaux ont fait apparaître certaines méthodes de transformation de dictionnaires en l'occurrence celles proposées par Paliwal et Atal [24]. Ces méthodes visent la diminution de la complexité avec certains changements dans les générations des dictionnaires de la quantification vectorielle.

Parmi les méthodes de quantification vectorielle des coefficients LSF, on peut citer la quantification vectorielle divisée et la quantification vectorielle multi-étages.

Deux approches ont été proposées, conçues et testées et ajoutées à ces méthodes déjà citées pour augmenter leurs performances. On cite en l'occurrence la quantification vectorielle hybride et la quantification vectorielle multi-dictionnaires.

Une des recommandations de l'utilisation de la quantification vectorielle concerne la taille de la base d'apprentissage.

En effet, une règle raisonnable pour une bon apprentissage est de considéré un rapport *M/N* du nombre de vecteur de la base d'apprentissage sur le nombre de vecteurs dans le dictionnaire compris entre 50 et 200 [40, 43, 46].

Ce chapitre expose en détail l'application de ces méthodes dans la quantification des coefficients LSF ainsi qu'une comparaison, entre les méthodes basée sur l'appui de deux bases d'apprentissage et de tests.

IV.1 La quantification vectorielle divisée ou split vector quantization (SVQ).

La quantification vectorielle divisée, appelée aussi Split Vector Quantization SVQ en Anglais [11, 24, 47, 48, 49, 50], consiste à diviser le vecteur candidat à la quantification en plusieurs sous vecteurs de tailles plus petites et plus faciles à quantifier (voir figure IV.1). En d'autre terme on part d'un vecteur de taille N pour avoir n sous vecteurs de tailles plus petites dont la somme est égale à la taille initiale N.

La quantification vectorielle simple était très complexe vu la base de données requise d'après la règle de Makhoul [40, 41, 46]. L'utilisation d'un nombre de bits b très grand dans la quantification simple rend l'apprentissage et même l'opération de quantification délicate. Cependant, la quantification vectorielle divisée utilise le même nombre de bits (b) partagé entre ces sous vecteurs. Le partage de bits n'est forcement pas équitable. De même les tailles des sous-vecteurs peuvent être différentes.

La figure IV.1 expose la façon avec laquelle on effectue la division du vecteur de taille N en sous vecteurs de tailles plus petites. Quand la quantification simple donne naissance à une distorsion D, alors la quantification vectorielle divisée donne naissance à une sommation

de distorsion de chaque sous vecteur. On peut tirer une conclusion sur la valeur de la distorsion qui augmente proportionnellement avec le nombre de sous vecteurs. Par ailleurs si le nombre des sous-vecteurs augmente, on aura moins de complexité car la taille des dictionnaires correspondants sera de plus en plus petites [24].

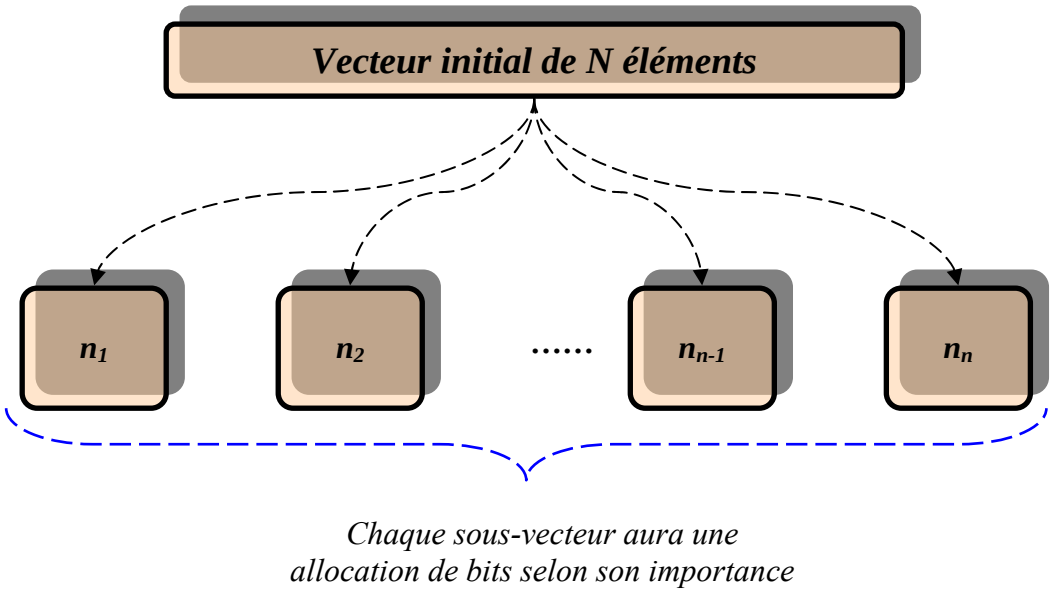


Figure IV.1 Schéma synoptique de la quantification vectorielle divisée

IV.1.1 SVQ des coefficients LSF

Un compromis est à gérer entre le nombre des sous vecteurs et la distorsion qui apparaît suite à l'utilisation de la SVQ. Les travaux de Paliwal et Atal [24] ainsi que d'autres chercheurs [11, 47] ont montré une certaine corrélation entre les coefficients LSF. Aussi une bonne division des vecteurs et une allocation de bits favorisant les basses fréquences, donneront de très bons résultats. La division du vecteur LSF dans la quantification vectorielle est un découpage fréquentiel de l'énergie spectrale LPC. Il faut une allocation de bits plus importante aux fréquences basses qu'aux fréquences hautes à cause de la sensibilité de l'oreille humaine.

Certains travaux ont été fais sur le quantificateur 2SVQ (SVQ à 2 sous vecteurs) avec des allocations plus au moins grandes telles que 12bits et 11bits en faisant intervenir une base d'apprentissage de vecteurs LSF de l'ordre de 204800 vecteurs LSF pour le cas du 12 bits.

Le quantificateur 3SVQ a été étudié avec des allocation de bits pour le dictionnaires allant jusqu'à 8 bits et ce pour des codage des vecteurs LSF entre 21bits et 24bits. Ces valeurs d'allocation exigent des bases d'apprentissage raisonnables allant entre 12800 vecteurs à 51200 vecteurs.

Dans ce travail on a conçu un quantificateur 3SVQ avec quatre allocations de bits entre 21bits et 24bits. Les bits sont partagés entre les sous vecteurs selon la sensibilité de l'oreille humaine. Le nombre de coefficients LSF dans chaque sous vecteur a été déterminé expérimentalement dans [24, 47]. Il a été remarqué que pour un vecteur LSF de 10 coefficients un schéma de division *(3 – 3 – 4)* donne de bons résultats.

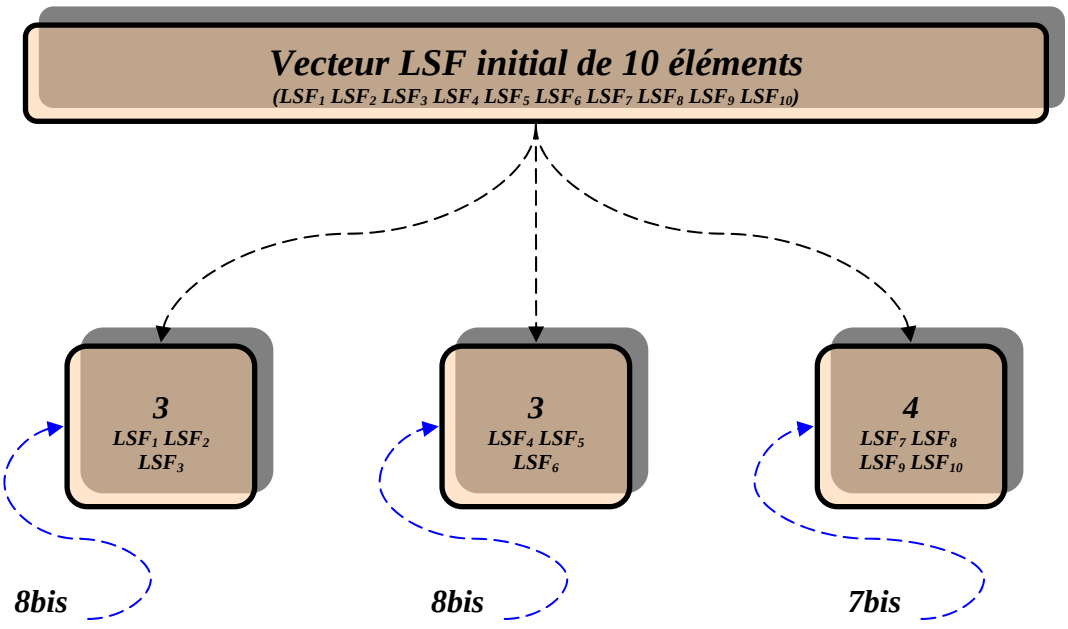


Figure IV.2 Exemple d'allocation de bits pour 3SVQ à 23 bits / trames

IV.1.2 Apprentissage de la 3SVQ

La base de données de parole TIMIT a été utilisée dans l'apprentissage de la quantification vectorielle SVQ. L'extraction des coefficients LSF a été réalisée suite à une opération d'analyse appliquée sur des trames de parole de 240 échantillons.

Les coefficients LSF se présentent sous forme de vecteurs de 10 éléments chacun. La collection des vecteurs LSF a donné naissance à une base d'apprentissage noté *DBASE*.

DBASE contient 25600 vecteurs LSF collectés à partir des fichiers de parole de la base TIMIT pris de l'accent de New York.

L'apprentissage a été fait en deux phases distinctes :

La première phase est la division de la base d'apprentissage *DBASE.* En effet, dans ce travail on a conçu un quantificateur 3SVQ d'où la division du vecteur des 10 coefficients LSF en trois sous vecteurs de taille *(3 – 3 – 4)* De même, les vecteurs LSF de *DBASE* ont subi la même division. Le résultat de cette phase est la création de 3 sous bases d'apprentissages notées $DBASE_1$, $DBASE_2$ *et* $DBASE_3$ relatives aux différents sous vecteurs.

La deuxième phase est la conception d'un quantificateur simple pour chacun des sous vecteurs. A partir de $DBASE_1$ et le nombre de bits alloués au premier sous-vecteur, on conçoit un quantificateur vectoriel simple pour ce sous-vecteur. Le même processus est suivi pour le deuxième et le troisième, sous-vecteurs.

Les deux distances : euclidienne simple et euclidienne pondérée, ont été utilisées avec l'algorithme GLA de Lloyd durant l'apprentissage des dictionnaires du quantificateur 3SVQ.

En effet cet algorithme a deux étapes distinctes : la détermination des régions de voronoï et le calcul des centroïdes de ces régions. Pour le calcul des centroïdes, la distance euclidienne simple a été utilisée. L'utilisation de la distance euclidienne pondérée dans ce dernier cas peut engendrer des vecteurs LSF qui provoquent l'instabilité du filtre synthèse.

Le déroulement de l'algorithme GLA a besoin d'un seuil de la variation de distorsion pour s'arrêter. Les travaux de Soong [11] ont donné de bon résultats avec un seuil de $\xi = 0.0001$. Dans notre travail on a adopté ce même seuil.

L'algorithme d'apprentissage se déroule comme suit

L'algorithme d'apprentissage de SVQ

Etape 1 *Partager la base d'apprentissage en trois sous bases* $DBASE_1$*={vecteurs (* LSF_1,LSF_2,LSF_3*)},* $DBASE_2$*={vecteurs (*LSF_4,LSF_5,LSF_6*)} et* $DBASE_3$*. ={vecteurs (* LSF_7,LSF_8,LSF_9,LSF_{10}*)}*

Etape 2 *Fixer le nombre de bit pour chacun des trois dictionnaires*

Etape 3 *Générer le dictionnaire initial pour chacun des dictionnaires* C_{01}*,* C_{02} *et* C_{03} *en utilisant respectivement les sous-bases* $DBASE_1$*,* $DBASE_2$ *et* $DBASE_3$*. et l'algorithme de Katsavoundis*

Etape 4 *Appliquer l'algorithme GLA pour la génération des dictionnaires* C_1*,* C_2 *et* C_3 *en utilisant les dictionnaires initiaux et les bases d'apprentissages correspondantes.*

IV.1.3 L'opération de quantification dans 3SVQ

L'opération de quantification au niveau de l'encodage est plus délicate que celle au niveau du décodage. Pour déterminer le bon vecteur code on utilise la distance euclidienne avec la règle du plus proche voisin pour chaque sous vecteur *LSF* [50, 51, 52, 53].

En effet, le vecteur LSF sera divisé en trois sous-vecteurs LSF_L, LSF_M et LSF_H. Ces vecteurs vont subir une quantification simple.

$\hat{LSF}_L^i$ est le vecteur-code optimal qui est extrait du 1er dictionnaire C_1 (*i :* étant l'indice du vecteur-code dans C_1) qui représente le vecteur $LSF_L = (LSF_1, LSF_2, LSF_3)$.

De même, $\hat{LSF}_M^j$ est le vecteur-code optimal extrait du $2^{ème}$ dictionnaire C_2 (*j :* étant l'indice du vecteur-code dans C_2) qui représente le vecteur $LSF_M = (LSF_4, LSF_5, LSF_6)$ avec la condition $\hat{LSF}_M^j(1) > \hat{LSF}_L^i(3)$ et qui exprime que la première composante du vecteur $\hat{LSF}_M^j$ doit être plus grande que la troisième composante du vecteur $\hat{LSF}_L^i$.

Enfin $L\hat{S}F_H^k$ est le vecteur-code optimal extrait du $3^{ème}$ dictionnaire C_3 (k : étant l'indice du vecteur-code dans C_3) qui représente le vecteur $LSF_H = (LSF_7, LSF_8, LSF_9, LSF_{10})$ avec la condition $L\hat{S}F_H^k(1) > L\hat{S}F_M^j(3)$ et qui exprime que la première composante du vecteur $L\hat{S}F_H^k$ doit être plus grande que la troisième composante du vecteur $L\hat{S}F_M^j$.

Au niveau du décodeur la reconstitution du vecteur LSF se fait selon les indices déjà transmis *(i, j, k)*

L'opération de quantification dans 3SVQ

Etape 1 *Découpage du vecteur LSF candidat à la quantification en trois sous-vecteurs* $LSF_L = (LSF_1, LSF_2, LSF_3)$, $LSF_M = (LSF_4, LSF_5, LSF_6)$ $LSF_H = (LSF_7, LSF_8, LSF_9, LSF_{10})$

Etape 2 *Recherche dans du dictionnaire* C_1 *du vecteur-code* $L\hat{S}F_L^i$ *du vecteur* LSF_L *tel que*

$$d(LSF_L, L\hat{S}F_L^i) < d(LSF_L, LSF_L^g) \quad \forall g \neq i$$

Etape 3 *Recherche dans le dictionnaire* C_2 *du vecteur-code* $L\hat{S}F_M^j$ *du vecteur* LSF_M *tel que*

$$d(LSF_M, L\hat{S}F_M^j) < d(LSF_M, LSF_M^g) \quad \forall g \neq j$$

et

$$L\hat{S}F_M^j(1) > L\hat{S}F_L^i(3)$$

Etape 4 *Recherche dans le dictionnaire* C_3 *du vecteur-code* $L\hat{S}F_H^k$ *du vecteur* LSF_H *tel que*

$$d(LSF_H, L\hat{S}F_H^k) < d(LSF_H, LSF_H^g) \quad \forall g \neq k$$

et

$$L\hat{S}F_H^k(1) > L\hat{S}F_M^j(3)$$

IV.1.4 Résultats et tests de 3SVQ

Les tests effectués sur ce quantificateur a comme objectif de voir l'efficacité de l'utilisation de la distance euclidienne pondérée. En effet, on a élaboré les dictionnaires du quantificateur 3SVQ, d'abord en utilisant la distance euclidienne simple, ensuite en utilisant la distance euclidienne pondérée.

Dans cette étape, une étude a été faite sur le nombre de vecteurs ayant la distorsion spectrale SD comprise entre 2dB et 4dB (Outliers de 1ère forme) et sur le nombre de vecteurs ayant la SD > 4dB (Outliers de 2eme forme).

Une base de tests formée de 7500 vecteurs pris de trois types de dialectes des Etats-Unis d'Amérique a servi au calcul de la distorsion spectrale moyenne (SD). On a trié les vecteurs selon la distorsion SD en deux parties : en Outliers 1ère forme et en Outliers 2eme forme. Ce test a été fait pour le quantificateur 3SVQ dans les 4 débits différents de 21bits/trame à 24bits/trame. Les résultats obtenus sont donnés aux tableaux IV.1 et IV.2.

La figure IV.3 montre le pourcentage des Outliers de 1ère forme en fonction du nombre de bits par trame. On remarque que, tout le long des allocations de bits, le graphe des Outliers relatif à la distance euclidienne pondérée est inférieur à celui relatif à la distance euclidienne simple.

En effet, pour le codage de 21bits/trame on remarque que le nombre des Outliers 1ère forme a diminué de 3.89% soit égale à 292 vecteurs. Contrairement pour les Outliers 2eme forme qui sont nuls et ce pour le cas de la distance euclidienne pondérée.

Ce résultat est renforcé par la remarque faite sur la distorsion moyenne. A la figure IV.4 on peut remarquer que pour tous les débits, la distorsion est faible quand on utilise la distance euclidienne pondérée.

Une conclusion peut être fait à ce niveau. La division du vecteur LSF peut causer plus de distorsion, seulement l'utilisation d'une bonne mesure de distance peut diminuer ces pertes.

Tab IV.1 Résultats du test dont la 3SVQ a subit en fonction du codage par trame, utilisant la distance euclidienne simple.

Allocation de bits	DS_{moy} (dB)	(%) de Outliers 1ère forme 2<SD<4dB	de Outliers 2eme forme SD>4dB
777	1,52	14,30	0,04
877	1,45	12,02	0,04
887	1,34	08,12	0,02
888	1,26	05,29	0,02

Tab IV.2 Résultats du test dont la 3SVQ a subit en fonction du codage par trame, utilisant la distance euclidienne pondérée .

Allocation de bits	DS_{moy} (dB)	(%) de Outliers 1ère forme 2<SD<4dB	de Outliers 2eme forme SD>4dB
777	1.46	10.41	0
877	1.39	8.53	0
887	1.28	5.62	0
888	1.21	3.86	0

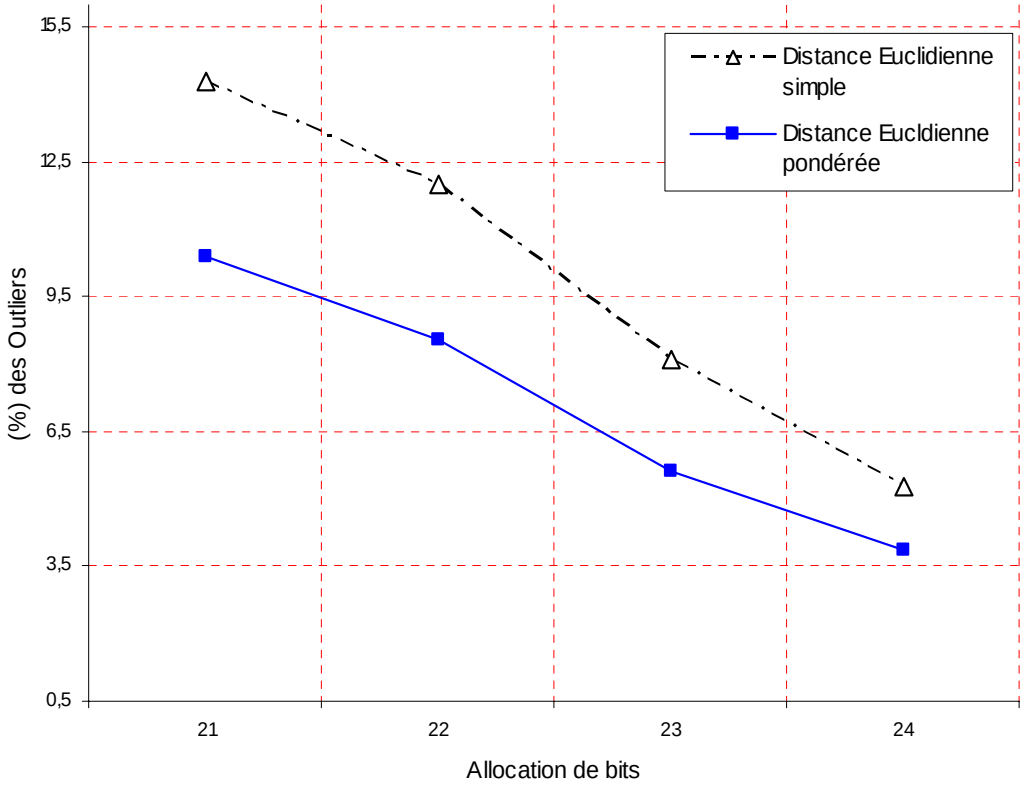


Figure IV.3 Le pourcentage des Outliers 1er et 2ème formes durant le test de la quantification 3SVQ utilisant les deux distances Euclidiennes, simples et pondérées. Le test a été effectué sur une base de vecteurs de test au nombre de 7500 vecteurs.

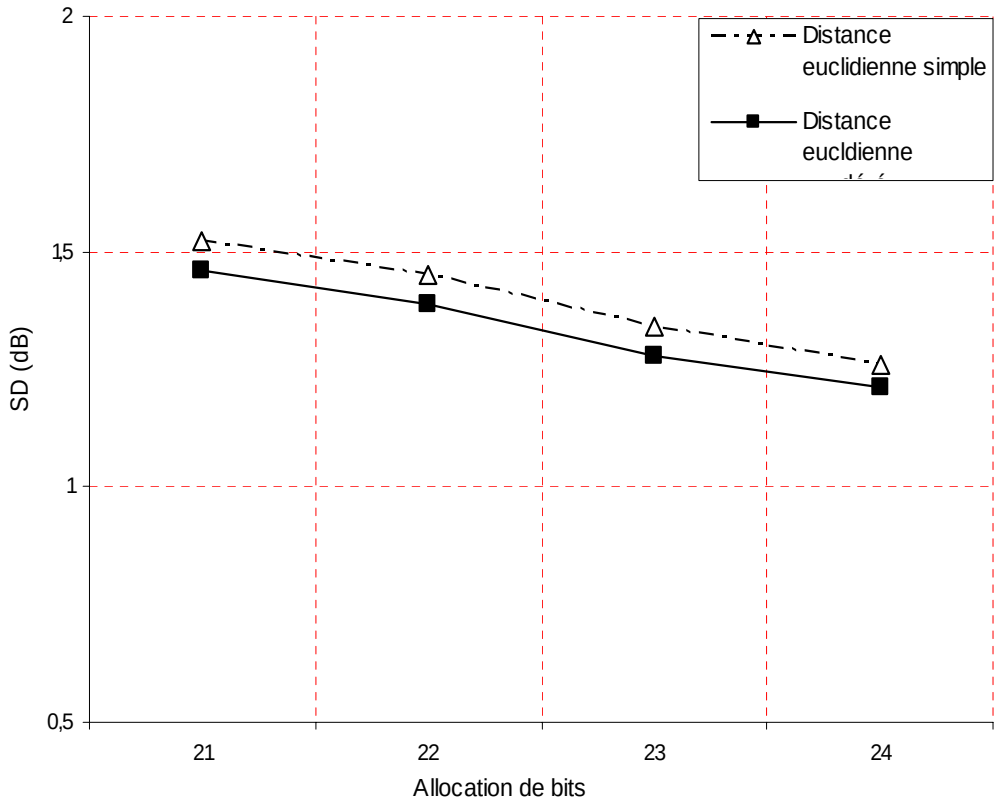


Figure IV.4 Distorsion moyenne générée durant le test pour le quantificateur 3SVQ. Le test a été effectué sur une base de 7500 vecteurs tests. Les deux graphes représentent deux tests selon deux distances, la distance euclidienne simple la distance euclidienne pondérée.

IV.2 La quantification vectorielle multi-étages ou Multi stage vector quantization (MSVQ)

Connue sous le nom de la quantification vectorielle résiduelle, ou en anglais MSVQ (Multi Stage Vector Quantization), elle est composée de plusieurs étages de quantification vectorielle simple[24, 54, 55, 56].

En effet, le vecteur x est quantifié par le premier étage pour avoir le vecteur-code $\hat{f}_1$. L'erreur de quantification après ce premier étage est quantifiée ensuite par le deuxième étage. Ainsi après chaque étage, l'erreur de la quantification sera quantifié par le prochain étage. D'où le nom de la quantification résiduelle.

Pour formaliser ce processus de quantification, le vecteur f_i se présentant à l'entrée de l'étage i est quantifié par ce même étage et le résultat est donné par :

$$f_i = \begin{cases} x & pour \quad i = 1 \\ x - \sum_{j=1}^{i-1} \hat{f}_j & pour \quad i = 2,...,m \end{cases} \qquad \text{(IV.1)}$$

m : étant le nombre d'étage

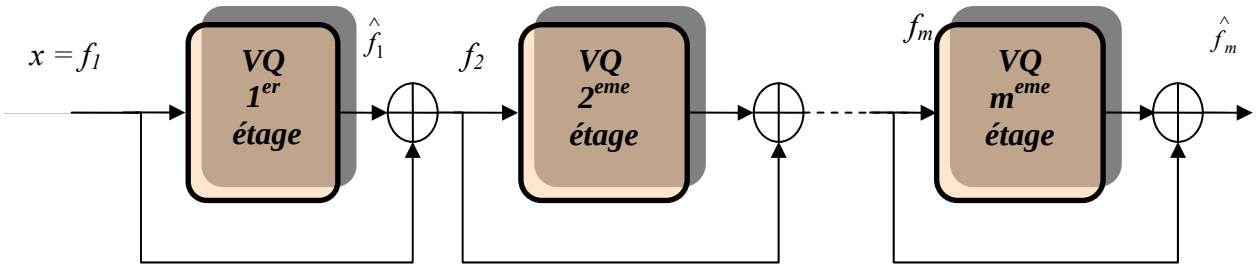


Figure IV.5 Schéma synoptique de la quantification vectorielle à multi étages MSVQ

A la fin du processus, le vecteur code est donné par la sommation des vecteurs codes obtenus après chaque étage. Ainsi, pour un quantificateur MSVQ à m étages, le vecteur code est donné par:

$$\hat{x} = \sum_{i=1}^{m} \hat{f}_i \qquad \text{(IV.02)}$$

A remarquer que les performances d'un quantificateur MSVQ ont tendance à se détériorer quand on augmente le nombre d'étages utilisés. A chaque étage on a une distorsion à calculer. Aussi pour m étages on a m distorsions. En parallèle, l'augmentation du nombre d'étage fait diminuer le nombre de bits alloués à chacun de ces étages et par conséquent la complexité de la recherche au sein de leurs dictionnaires diminue.

IV.2.1 La quantification MSVQ des coefficients LSF

La quantification vectorielle multi-étage a été utilisée dans plusieurs travaux [24, 54, 55]. Les meilleurs résultats ont été obtenus pour un nombre d'étages allant jusqu'à quatre, comme dans le cas du 4MSVQ utilisé par le MELP [57, 58].

Pour concevoir un quantificateur 2MSVQ, le nombre de bits alloué à chaque étage est équitable, sauf dans le cas ou on favorise le premier étage par un bit de plus [24]. En effet, le nombre de bits pour un étage peut atteindre les 12 bits ce qui nécessite un dictionnaire de grande taille. Ceci exige une grande base d'apprentissage.

Dans notre travail, on a réalisé un quantificateur 3MSVQ. Ce quantificateur a été utilisé par la norme d'ITU G729 [51]. Les allocations de bits entre les dictionnaires ont atteint une taille de 8 bits.

IV.2.2 Apprentissage de la 3MSVQ

La même base d'apprentissage utilisée dans la conception du quantificateur 3SVQ, a été aussi utilisée dans la conception du quantificateur 3MSVQ. En effet, pour mieux comparer les deux quantificateurs 3SVQ et 3MSVQ, on a utilisé les mêmes vecteurs de la base d'apprentissage *(DBASE)* et de test. A rappeler que le nombre de vecteurs de la base d'apprentissage est de 25600 vecteurs et le nombre de vecteurs de test est de 7500 vecteurs.

L'apprentissage des dictionnaires des 3 étages a suivi 3 étapes différentes et ce après le partage des bits entre ces dictionnaires : b_1 pour le dictionnaire du premier étage, b_2 pour le dictionnaire du deuxième étage et b_3 pour le dictionnaire du troisième étage.

La première étape consiste à élaborer un dictionnaire C_1 de la quantification vectorielle simple du premier étage. Ce dictionnaire est de taille 2^{b_1}. Une fois le dictionnaire C_1 construit, on commence à quantifier les vecteurs de la base d'apprentissage initiale *DBASE* (appelée dans notre cas $DBASE_1$) vecteur après vecteur. Pour chaque vecteur quantifié, on calcule l'erreur de quantification correspondante créant ainsi la base $DBASE_2$. La base $DBASE_2$ sera la base d'apprentissage du dictionnaire du deuxième étage.

La seconde étape est la construction du dictionnaire C_2, qui sera de taille 2^{b_2}, à partir de la base d'apprentissage $DBASE_2$ en utilisant l'algorithme GLA. Une fois le dictionnaire C_2

élaboré, on commence à quantifier les vecteurs de la base d'apprentissage $DBASE_2$ vecteur après vecteur. Pour chaque vecteur quantifié on calcule l'erreur de quantification correspondante créant ainsi la base $DBASE_3$. La base $DBASE_3$ est la base d'apprentissage du troisième étage. Enfin, à partir de la base d'apprentissage $DBASE_3$ on construit le dictionnaire C_3 (de taille 2^{b_3}) en utilisant l'algorithme GLA. Le dictionnaire C_3 correspond à l'étage *3*.

Le même algorithme GLA a été utilisée avec son seuil d'arrêt $\xi = 0.0001$. Les étapes de l'algorithme se déroule comme suit :

L'algorithme d'apprentissage de 3MSVQ

Etape 1 *Fixer le nombre de bits pour chacun des dictionnaires*

b_1 pour C_1, b_2 pour C_2 et b_3 pour C_3

Etape 2 *Générer le dictionnaire C_1 utilisant la base d'apprentissage $DBASE_1$ et l'algorithme GLA*

Etape 3 *Quantifier la base de données $DBASE_1$, vecteur après vecteur et calculer à chaque fois l'erreur de quantification qui sera enregistrée dans un fichier organisant ainsi la base de données $DBASE_2$ pour le deuxième étage.*

Etape 4 *Générer le dictionnaire C_2 utilisant la base d'apprentissage $DBASE_2$ et l'algorithme GLA*

Etape 5 *Quantifier la base de données $DBASE_2$, vecteur après vecteur et calculer à chaque fois l'erreur de quantification qui sera enregistrée dans un fichier organisant ainsi la base de données $DBASE_3$ pour le deuxième étage.*

Etape 6 *Générer le dictionnaire C_3 utilisant la base d'apprentissage $DBASE_3$ et l'algorithme GLA*

IV.2.3 L'opération de quantification dans 3MSVQ

Au niveau du l'encodeur, le vecteur LSF est initialement quantifié par le premier étage utilisant son dictionnaire C_1 et la règle du plus proche voisin. L'erreur de la quantification précédente sera quantifiée de la même façon avec le deuxième étage utilisant son dictionnaire C_2. On calcul aussi l'erreur de quantification qui sera sujette à une quantification par le troisième étage utilisant son dictionnaire C_3.

Les indices des vecteurs-codes des trois étages seront codés sur $(b_1+b_2+b_3)$ qui seront transmis au décodeur pour l'opération de décodage.

A remarquer ici aussi que les deux distances sont utilisées en l'occurrence, l'Euclidienne simple et l'Euclidienne pondérée.

L'opération de quantification dans 3MSVQ

Etape 1 *Recherche dans le dictionnaire C_1 le vecteur-code $L\hat{S}F_i$ du vecteur LSF tel que*

$$d(LSF\ , L\hat{S}F_i) < d(LSF\ , LSF_g) \quad \forall g \neq i$$

Etape 2 *Calculer l'erreur de la quantification précédente comme suit:*

$$LSF_2 = LSF - L\hat{S}F_i$$

Etape 3 *Recherche dans le dictionnaire C_2 le vecteur-code $L\hat{S}F_j$ tel que*

$$d(LSF_2, L\hat{S}F_j) < d(LSF_2, LSF_g) \quad \forall g \neq j$$

Etape 4 *Calculer l'erreur de quantification précédente comme suit :*

$$LSF_3 = LSF_2 - L\hat{S}F_j$$

Etape 5 *Recherche dans le dictionnaire C_3 le vecteur-code $L\hat{S}F_k$ tel que*

$$d(LSF_3, L\hat{S}F_k) < d(LSF_3, LSF_g) \quad \forall g \neq k$$

Etape 6 *Les représentants sont codés par les indices (i, j, k) qui seront transmis et le vecteur code total sera $L\hat{S}F = L\hat{S}F_i + L\hat{S}F_j + L\hat{S}F_k$*

IV.2.4 Résultats et tests de 3MSVQ

Ce test a visé le même but que celui du quantificateur 3SVQ c'est-à-dire voir l'efficacité de l'utilisation de la distance euclidienne pondérée. En premier lieu on remarque que l'utilisation de la distance euclidienne pondérée a fait diminuer les Outliers de 1ère forme et le Outliers de 2eme forme jusqu'à avoir des valeurs nulles pour les codages de 23bits/trame et 24bits/trame (voir tableaux IV.3 et IV.4).

En effet, pour le même codage par trame on relève une baisse des Outliers 1ère forme. Comme exemple on cite le codage de 21bits/trame. Quand on passe de la distance euclidienne simple à la distance euclidienne pondérée on remarque qu'il y a une diminution des Outliers 1ère forme qui atteint 4.38% cette valeur est l'équivalent de 329 vecteurs. La figure IV.6 expose clairement ces résultats ainsi que les tableaux IV.3 et IV.4.

On relève une valeur de la distorsion moyenne très faible pour le cas du codage de 24bits/trame égale à 1.18dB (tableau IV.4) qui est inférieur à celle obtenu pour le quantificateur scalaire proposé par le FS1016. La figure IV.7 expose l'effet du codage sur la diminution de la distorsion selon le type de codage et selon aussi la distance utilisée.

Tableau IV.3 Résultats du test pour la 3MSVQ en fonction des allocations de bits, utilisant la distance euclidienne simple.

Allocation de bits	moyenne DS (dB)	(%) de Outliers 1ère forme 2<SD<4dB	de Outliers 2eme forme SD>4dB
777	1,48	15,20	0,14
877	1,40	12,04	0,08
887	1,31	08,90	0,02
888	1,23	06,80	0,00

Tab IV.4 Résultats du test pour la 3MSVQ en fonction des allocations de bits, utilisant la distance euclidienne pondérée.

Allocation de bits	moyenne DS (dB)	(%) de Outliers 1ère forme 2<SD<4dB	de Outliers 2eme forme SD>4dB
777	1,46	10,82	0,04
877	1,31	08,12	0,01
887	1,25	06,48	0,00
888	1,18	04,48	0,00

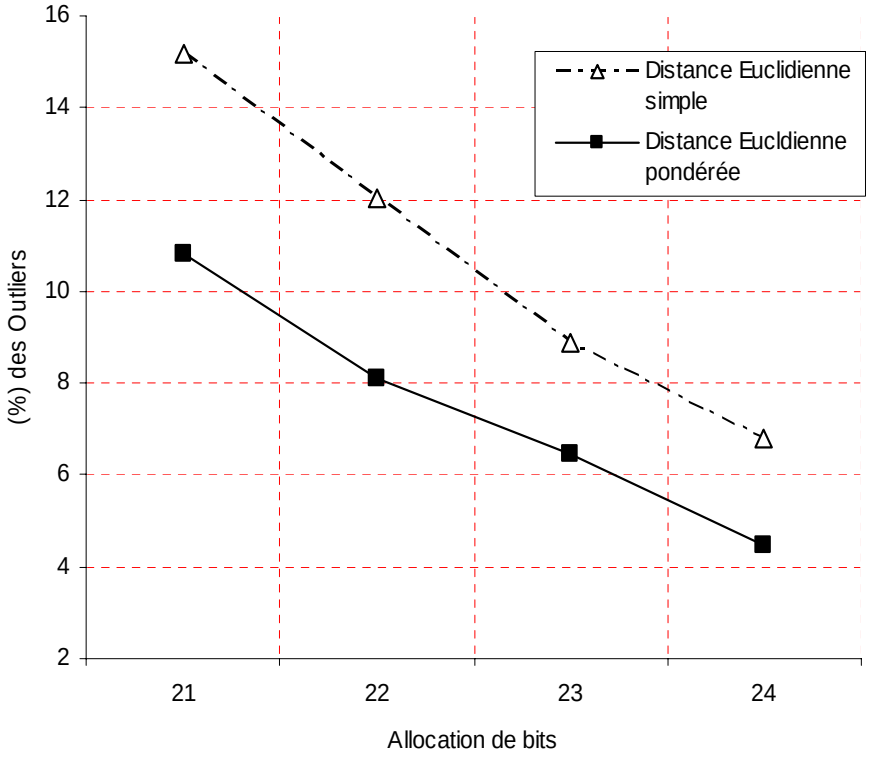


Figure IV.6 Le pourcentage des vecteurs éloignés durant le test de la quantification 3MSVQ utilisant les deux distances euclidiennes simple et pondérée. Le test a été effectué sur une base de vecteurs de test au nombre de 7500 vecteurs

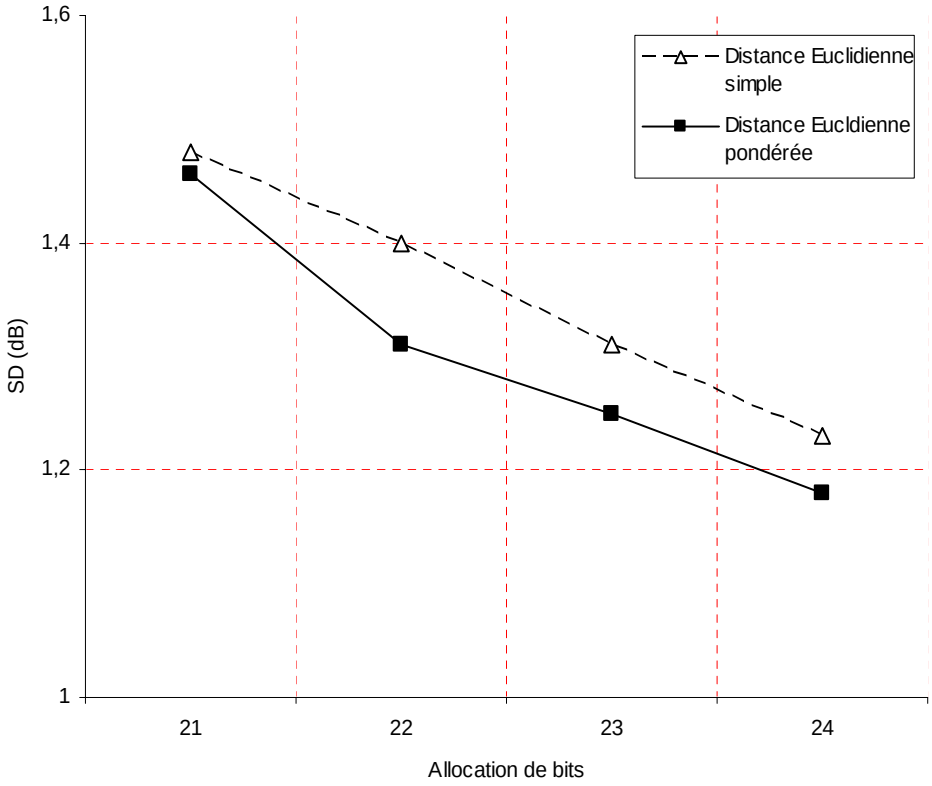


Figure IV.7 Distorsion moyenne générée durant le test pour le quantificateur 3MSVQ.

.

IV.2.5 Comparaison entre 3SVQ et 3MSVQ

L'évaluation des deux quantificateurs 3SVQ et 3MSVQ a révélé que leur utilisation avec la distance euclidienne pondérée donne de bons résultats par apport à ceux trouvés avec la quantification scalaire proposée par le FS1016 [7].

La figure IV.8 expose la comparaison entre les deux quantificateurs 3SVQ et 3MSVQ du point de vu distorsion spectrale moyenne. On remarque qu'il y a plus de distorsion avec le quantificateur 3SVQ qu'avec le quantificateur 3MSVQ. Cette différence est d'environ 0.03dB.

Du point de vu complexité, le quantificateur 3SVQ est moins complexe que le quantificateur 3MSVQ. En effet, les sous vecteurs du quantificateur 3SVQ doivent avoir une taille de 10 pour arriver à la complexité du quantificateur 3MSVQ.

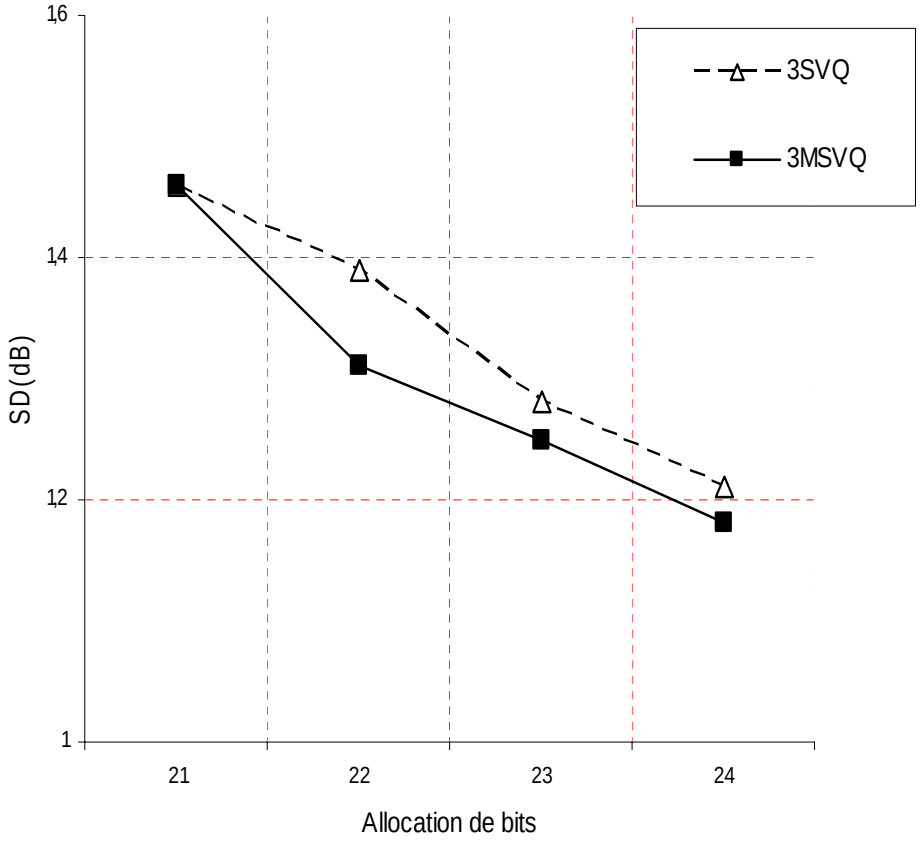


Figure IV.8 Comparaison de la distorsion moyenne obtenue entre 3SVQ et 3MSVQ

IV.3 Quantification vectorielle hybride ou hybrid vector quantization (H-VQ)

La conception des quantificateurs précédents a révélé que les deux schémas 2SVQ et 2MSVQ requièrent une base d'apprentissage très grande. On remarque aussi une chute de la distorsion lors du test du quantificateur 3SVQ. Quand on passe du codage de 21bits/trame à 24bits/trame, il y a une différence de 0.24dB.

Par ailleurs, la 3SVQ conçue sur un partage en trois parties, favorise des pertes, surtout dans le cas des très basses allocations de bits tel que 21bits/trame et 22bits/trame. Cependant, on a imaginé et conçu un quantificateur 2SVQ avec un étage de correction en dernier avec une bonne allocation de bits. On appellera ce schéma la quantification hybride qu'on notera H-SVQ/MSVQ.

La quantification H-SVQ/MSVQ est une quantification vectorielle de deux étages 2MSVQ, dans le premier étage se trouve une quantification vectorielle divisée de deux parties (2SVQ). Voir figure IV.11.

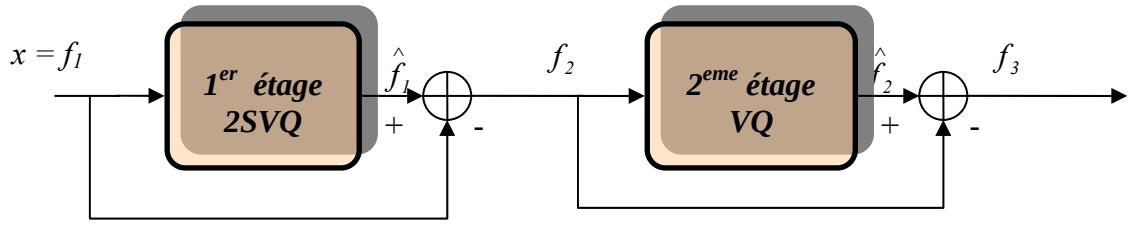


Figure IV.10 Schéma synoptique de la quantification vectorielle H-SVQ/MSVQ

IV.3.1 La quantification H-SVQ/MSVQ des coefficients LSF

Le schéma synoptique de la figure IV.10 expose clairement l'application de la quantification hybride sur les coefficients LSF.

En effet, on a conçu un quantificateur MSVQ à deux étages. Dans le premier étage il y a un quantificateur 2SVQ.

Le nombre de bits alloué à chaque dictionnaire est le suivant : On garde 8 bits pour le deuxième étage et on partage le reste des bits entre les dictionnaires du 2SVQ qui se trouvent dans le premier étage. Le cas particulier du codage de 21bits/trame n'est pas traité car les allocations de bits pour les dictionnaires seraient faibles et par conséquent la distorsion augmentera.

Pour la conception du quantificateur 2SVQ on a suivi les mêmes étapes que celles expliquées dans la section (§IV.1). Les travaux de Paliwal et Atal [24] ont montré une division optimale du vecteur LSF. La subdivision du vecteur LSF en 4 et 6 éléments offre de bons résultats du point de vue distorsion spectrale.

L'allocation de bits pour les dictionnaires du 2SVQ doit respecter la perception de l'oreille humaine. De ce fait, on doit favoriser le premier sous vecteur. On expose dans ce qui suit un exemple de codage à 23bits/trame.

1^{er} étage ⟶ 2SVQ ⟶ 8bits + 7bits

2^{eme} étage ⟶ MSVQ ⟶ 8bits

IV.3.2 L'apprentissage du H-SVQ/MSVQ

La conception du quantificateur hybride H-SVQ/MSVQ a suivi deux étapes distinctes:

La première étape consiste à l'élaboration du premier étage qui englobe un quantificateur vectoriel divisé 2SVQ. En effet on construit deux dictionnaires relatifs aux deux sous vecteurs issu de la division du vecteur LSF. L'allocation de bits était équitable entre les dictionnaires sauf le cas où on favorise le premier sous vecteur par un bit de plus (si le nombre de bits est impair).

La division du vecteur LSF a suivi le schéma ressemblant à celui proposé par [24] à savoir:

$$\underbrace{(LSF_1\ LSF_2\ LSF_3\ LSF_4)}_{1^{er}\text{ sous vecteur de 4 coefficients}} - \underbrace{(LSF_5\ LSF_6\ LSF_7\ LSF_8\ LSF_9\ LSF_{10})}_{2^{eme}\text{ sous vecteur de 6 coefficients}}$$

L'apprentissage des dictionnaires de ce 2SVQ est analogue à celui relatif aux dictionnaires du 3SVQ. Dans l'algorithme d'apprentissage, le 2SVQ sera pris comme bloc et il ne sera pas détaillé.

Une fois cette étape terminée, on procède à la quantification des vecteurs de la base d'apprentissage notée $DBASE_1$ vecteur après vecteur. Après chaque opération de quantification on calcule l'erreur de cette dernière. Ces erreurs seront stockées dans un fichier appelé $DBASE_2$. Le fichier $DBASE_2$ est la base d'apprentissage du dictionnaire du deuxième étage du MSVQ.

La seconde étape consiste en l'élaboration du dictionnaire du deuxième étage. Cet étage englobe une quantification vectorielle simple. On utilise dans cette étape la base d'apprentissage $DBASE_2$ pour cela on réserve 8 bits pour cet étage.

On utilise la distance euclidienne pondérée dans toutes les opérations d'apprentissage des dictionnaires du H-SVQ/MSVQ vus les résultats concluants obtenus.

L'algorithme d'apprentissage du H-SVQ/MSVQ

Etape 1 *Fixer le nombre de bits b_1, b_2 et b_3 pour chacun des trois dictionnaires.*

Etape 2 *Faire l'apprentissage des dictionnaires du 1^{er} étage pour concevoir 2SVQ en utilisant la base des vecteurs LSF $DBASE_1$ et l'algorithme GLA avec les allocations de bits b_1 et b_2.*

Etape 3 *Quantifier la base des vecteurs LSF $DBASE_1$, vecteur après vecteur et à chaque fois calculer l'erreur de quantification qui sera enregistrée dans un fichier créant ainsi la base de données DBASE2 pour le deuxième étage.*

Etape 4 *Elaborer le dictionnaire du 2^{eme} étage du MSVQ qui contient une quantification vectorielle simple.*

IV.3.3 L'opération de quantification

La quantification vectorielle hybride fait passer le vecteur LSF par deux étages MSVQ.

Le premier étage englobe une quantification vectorielle divisée 2SVQ. Son mode de quantification est analogue à celui du 3SVQ déjà expliqué. En effet, on aura à calculer dans cette étape deux indices *(i)* et *(j)* relatifs aux deux dictionnaires du 2SVQ. A la fin de cette opération on calcule l'erreur de la quantification.

En second étage, l'erreur de quantification calculée précédemment sera elle-même quantifiée par une quantification vectorielle simple pour générer un indice d'un vecteur code *(k)*.

Les indices *(i, j, k)* seront transmis au récepteur pour l'opération inverse de la quantification.

L'opération de quantification dans H-SVQ/MSVQ

Etape 1 *quantification du vecteur LSF par le 2SVQ.*

Etape 2 *Calcul de l'erreur de la précédente quantification.*

Etape 3 *Quantification de l'erreur précédente par le deuxième étage du MSVQ.*

IV.3.4 Résultats de test de la H-SVQ/MSVQ

Avant d'examiner les résultats, on rappelle le but de la proposition de cette technique d'hybridation dans la quantification vectorielle. En effet, le partage du vecteur LSF en trois sous vecteurs pour la quantification vectorielle divisée, engendre une distorsion relative à chaque sous vecteur. Cette observation a été plus remarquable dans les allocations faibles telles que 22bits/trame et 23bits/trame. Comme solution, on a proposé un étage de correction.

Du point de vue distorsion spectrale moyenne, on a fait une comparaison entre le quantificateur 3SVQ et le quantificateur H-SVQ/MSVQ. On remarque une diminution de la distorsion quand on passe du 3SVQ au H-SVQ/MSVQ pour le codage de 22bits/trame et ce de 0.02dB (voir table IV.5).

Pour le pourcentage des Outliers 1ère forme, une diminution a été relevée. Dans le cas du codage 22bits/trame, le nombre des Outliers de 1ère forme d'un codage à multi-dictionnaires a diminué de 0.69%, ce qui correspond à environ 53 vecteurs. La même remarque a été faite pour le codage de 23bits/trame où on a observé une diminution du nombre des Outliers 1ère forme d'environ 0.554% soit 50 vecteurs. Le tableau IV.5 montre les valeurs des distorsions spectrales moyenne et les pourcentages des Outliers et ce en fonction des allocations de bits.

A noter que la quantification hybride H-SVQ/MSVQ a été utilisée par la norme d'ITU G.729 [13] pour le codeur *« CODING OF SPEECH AT 8Kbit/s USING CONJUGATE STRUCTURE ALGEBRAIC-CODE-EXCITED LINEAR PREDICTION(CS-ACELP) »,* sauf que le quantificateur SVQ a été placé dans le deuxième étage .

Tableau IV.5 Résultats du test dont le H-SVQ/MSVQ a subit selon les allocations De bits, utilisant la distance euclidienne pondérée

Allocation de bits	DS_{moy} (dB)	(%) de Outliers 1ère forme2<SD<4dB	(%) de Outliers 2eme forme SD>4dB
22 (7-7/ 8)	1.37	07.84	0.02
23(8-7/ 8)	1.28	05.08	0.01
24(8-8/ 8)	1.21	03.89	0.01

IV.4 Quantification vectorielle à multi dictionnaires

Comme son nom l'indique, c'est une quantification vectorielle utilisant plusieurs dictionnaires dont les tailles peuvent être différentes et dans lesquels le choix du vecteur représentant est effectué selon la minimisation simultanée de la distorsion et du débit [59, 60, 61].

IV.4.1 Quantification SVQ à Multi dictionnaires

Proposée par Han *et al* [59] où ils ont imaginé un schéma d'un quantificateur SVQ dans lequel à chaque sous-vecteur existe un ensemble de dictionnaire de différentes tailles. Han *et al* ont proposé un schéma de quantification minimisant simultanément la distorsion et le débit en même temps, et cela en cherchant le meilleur représentant de chaque sous vecteur dans un ensemble de dictionnaires qui lui sont associés.

La recherche du vecteur code dans les dictionnaire suit la règle du plus proche voisin. Il y a un cas spécial où deux vecteurs de deux dictionnaires différents peuvent avoir la même distance par rapport au vecteur LSF. Si ce cas se présente on applique la deuxième règle de décision qui est la diminution du débit.

A noter que les vecteurs codes trouvés doivent respecter la règle de la stabilité du filtre d'analyse. Donc, ils doivent respecter l'ordre des coefficients LSF.

Avec ces remarques on a conçu un quantificateur 3SVQ à multi dictionnaires (figure IV.11). Pour chaque sous-vecteurs, un ensemble de dictionnaires sera élaboré.

Le débit dans ces conditions est variable. Le vecteur code du premier sous vecteur LSF appartient au dictionnaire avec une allocation égale à (m_1). Le vecteur code du second sous vecteur LSF appartient au dictionnaire avec une allocation égale à (m_2) et de même (m_3) pour le troisième sous vecteur LSF. Aussi, le codage par trame du vecteur LSF sera de $(m_1+m_2+m_3)$ bits/trame [59, 60].

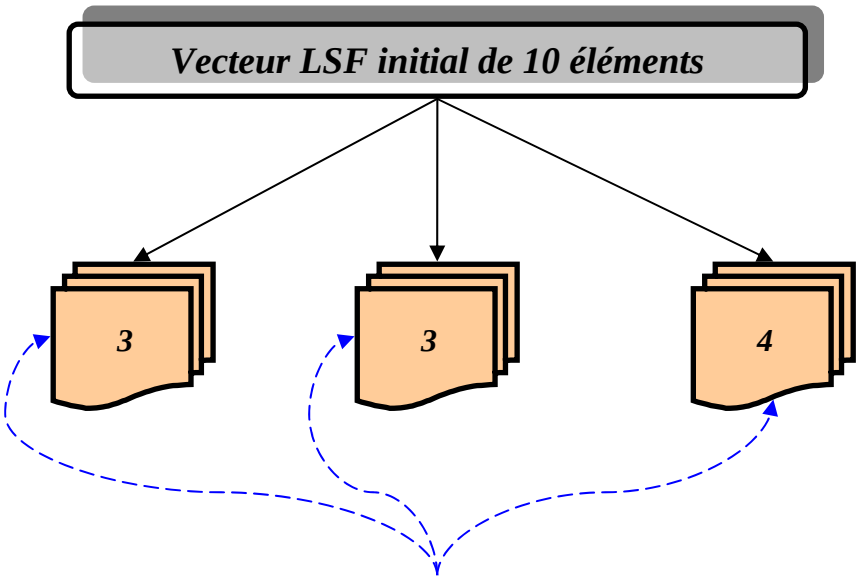


Figure IV.11. Schéma synoptique de la quantification à multi dictionnaires appliquée au SVQ

IV.4.2 Quantification MSVQ à Multi dictionnaires

Comme application de l'idée de Han *et al* [59] on a proposé l'utilisation de la technique multi-dictionnaires pour concevoir un MCMSVQ (Multi codes books MSVQ) [61].

L'idée a été exploitée visant la création d'un ensemble de dictionnaires pour chaque étage du MSVQ (voir figure IV.12).

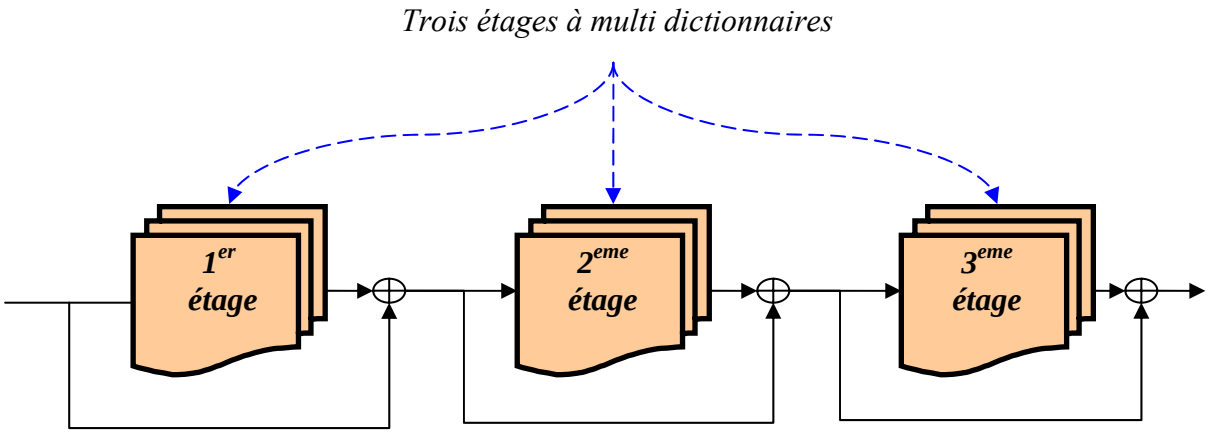


Figure IV.12. Schéma synoptique de la quantification à multi dictionnaires appliquée au MSVQ

IV.4.3 Fixation du débit

En pratique la plupart des codeurs de la parole travaillent avec des débits fixes. Han *et al* [59] ont proposé une méthode pour fixer le débit de la quantification MCSVQ.

L'idée est de concevoir pour un débit donné, quatre quantificateurs SVQ. Durant la quantification, le choix entre ces quatre quantificateurs est basé sur la distorsion. Le quantificateur qui donne un vecteur code avec moins de distorsion sera pris. Le choix entre les quantificateurs est codé sur 2 bits.

Par exemple pour un codage de 24bits/trame les quantificateurs 3SVQ auront 22 bits et le choix sera codé sur 2 bits. Les allocations de bits entre les dictionnaires seront différentes d'un quantificateur à un autre tout en préservant le total de 22bits. Le tableau IV.6 expose le partage des bits entre les différents dictionnaires C_1, C_2 et C_3 du 3SVQ.

Tab IV.6.: Allocation de bits en fonction du codage par trame pour les dictionnaires de MCSVQ et MCMSVQ.

	Codage par trame de 21 bits			**Codage par trame de 22 bits**			**Codage par trame de 23 bits**			**Codage par trame de 24 bits**		
	C_1	**C_2**	**C_3**	**C_1**	**C_2**	**C_3**	**C_1**	**C_2**	**C_3**	**C_1**	**C_2**	**C_3**
1^{er} 3SVQ / 1^{er} MSVQ	8	6	5	8	7	5	8	8	5	8	8	6
2^{eme} 3SVQ / 2^{eme} MSVQ	8	5	6	8	6	6	8	7	6	8	7	7
3^{eme} 3SVQ / 3^{eme} MSVQ	6	7	6	6	7	7	6	8	7	7	8	7
4^{eme} 3SVQ / 4^{eme} MSVQ	6	6	7	7	7	6	7	7	7	8	7	7

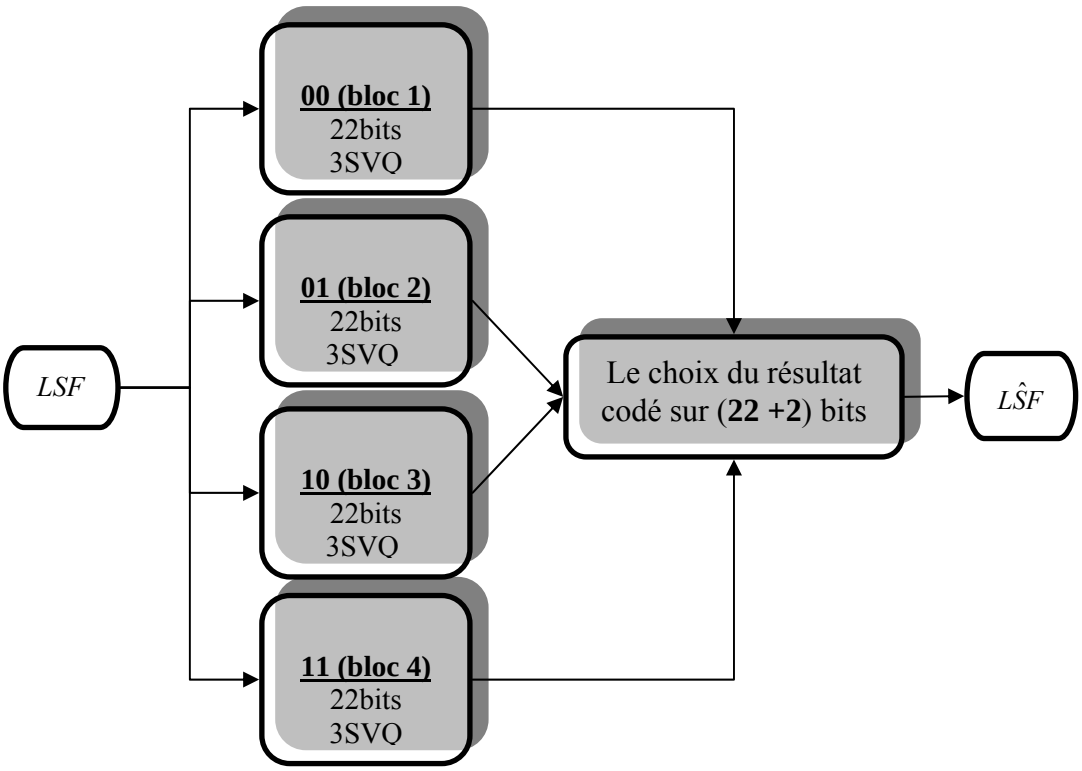


Figure IV.13 : schéma bloc donnant le mode de quantification MCSVQ avec un débit fixe de 24bits/trame.

Nous avons exploité l'idée précédente pour concevoir un quantificateur MCMSVQ. Pour cela, dans le schéma précédent on a changé les quantificateurs 3SVQ par des quantificateurs 3MSVQ. On donne à la figure IV.14 un exemple d'un MCMSVQ à 24 bits/trame.

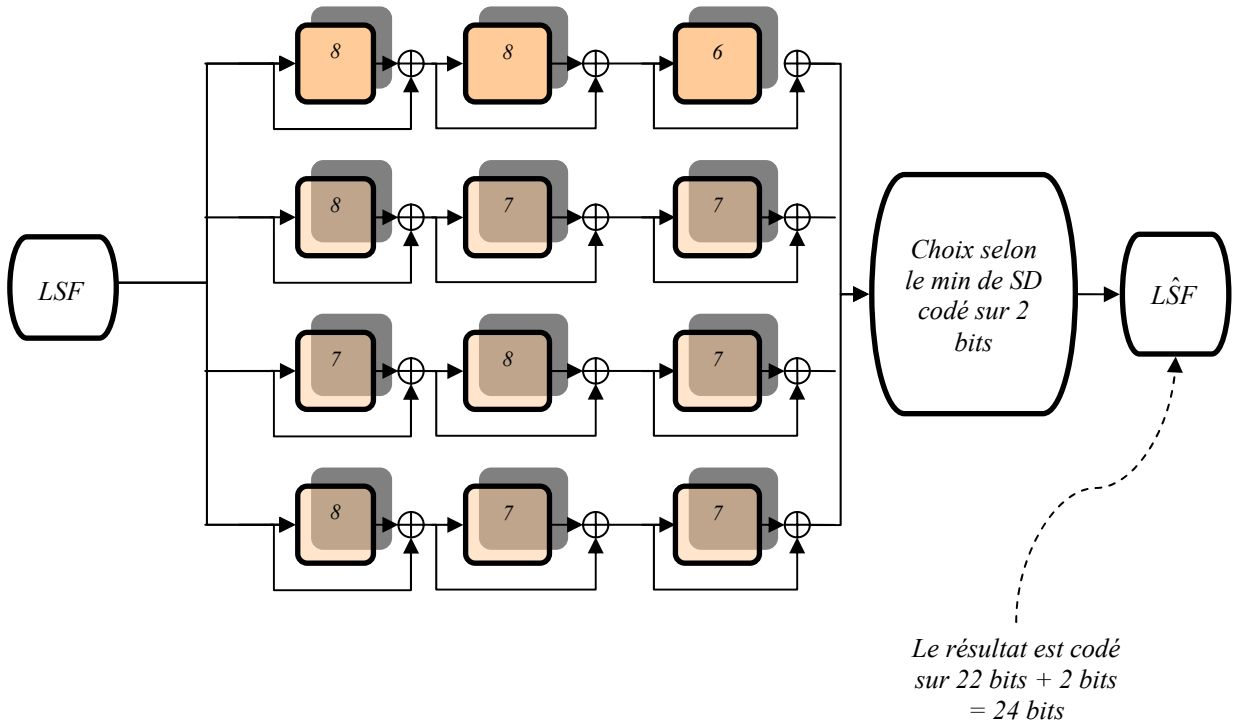


Figure IV.14 : schéma bloc donnant le mode de quantification MCMSVQ avec codage par trame fixé à 24bits/trame.

IV.4.4 Conception du 3MCSVQ et du 3MCMSVQ

Pour un codage par trame donné, la conception se résume dans la réalisation de quatre quantificateurs 3SVQ dans le cas d'un quantificateur MCSVQ (respectivement quatre 3MSVQ dans le cas de MCMSVQ) (voir figures IV.12 et IV.13).

Les dictionnaires de ces quantificateurs sont élaborés en utilisant la distance euclidienne pondérée avec l'algorithme GLA.

Les allocations de bits sont présentées dans le tableau IV.6. On remarque que pour chaque codage par trame on réserve 2 bits pour le choix entre les quantificateurs du point de vu distorsion [59]. En effet, le vecteur LSF va subir quatre quantifications. Pour chaque quantificateur on calcule la distorsion moyenne. Le quantificateur ayant la distorsion minimale sera choisi.

L'opération de quantification dans MCSVQ/ ou MCMSVQ

Etape 1 *Quantifier le vecteur LSF par quatre quantificateurs 3SVQ (respectivement quatre quantificateurs 3MSVQ) de différentes allocations (tableau 4.6).*

Etape 2 *Calculer la distorsion pour les quatre opérations de quantifications précédentes, SD_1, SD_2, SD_3 et SD_4.*

Etape 3*.Choisir le quantificateur 3SVQ (respectivement 3MSVQ) ayant généré la distorsion la plus basse. Le choix est codé sur 2 bits.*

IV.4.5 Résultats et tests du MCSVQ et du MCMSVQ

Les résultats de test des quantificateurs MCSVQ et MCMSVQ sont représentés dans les tableaux (IV.7, IV.8, IV.9 et IV.10). Dans ces tableaux on donne aussi les anciens résultats des quantificateurs 3SVQ et 3MSVQ. Cette réunion de résultats vise l'évaluation de la performance obtenue quand on passe du 3SVQ au MCSVQ et respectivement du quantificateur 3MSVQ au MCMSVQ.

Dans les différents tableaux on remarque que les quantificateurs sont réunis deux à deux 3SVQ et MCSVQ et puis du 3MSVQ au MCMSVQ. Aux premières colonnes on donne les valeurs des distorsions moyennes.

On remarque aussi que la technique de quantification multi dictionnaire a contribué d'une manière efficace pour la diminution de cette grandeur. La distorsion moyenne a subi une diminution d'environ 0.02 dB quand on passe du quantificateur 3SVQ au quantificateur

MCSVQ et ce à travers toutes les allocations de bits de 21bits/trame à 24bits/trame. La même remarque faite quand on passe du quantificateur 3MSVQ au quantificateur MCMSVQ avec une diminution de 0.04 dB.

Tableau IV.7 : Comparaison entre les méthodes de quantification vectorielle conçues pour 21bits/trame.

Les modes de QV	moyenne DS (dB)	(%) de Outliers 1ère forme	(%) de Outliers 2ère forme
SVQ	1.46	10.41	0
MSVQ	1.46	10.82	0.04
MCSVQ	1.41	9.93	0.02
MCMSVQ	1.36	8.6	0.01

Tableau IV.8 : Comparaison entre les méthodes de quantification vectorielle conçues pour 22bits/trame.

Les modes de QV	moyenne DS (dB)	(%) de Outliers 1ère forme	(%) de Outliers 2ère forme
SVQ	1.39	8.53	0
MSVQ	1.31	8.12	0.01
MCSVQ	1.28	5.77	0
MCMSVQ	1.28	5.92	0

Tableau IV.9 : Comparaison entre les méthodes de quantification vectorielle conçues pour 23bits/trame.

Les modes de QV	moyenne DS (dB)	(%) de Outliers 1ère forme	(%) de Outliers 2ère forme
SVQ	1.28	5.62	0
MSVQ	1.25	6.48	0
MCSVQ	1.24	4.36	0
MCMSVQ	1.20	4.45	0

Tableau IV.10 : Comparaison entre les méthodes de quantification vectorielle conçues pour 24bits/trame.

Les modes de QV	moyenne DS (dB)	(%) de Outliers 1ère forme	(%) de Outliers 2ère forme
SVQ	1.21	3.86	0
MSVQ	1.18	4.48	0
MCSVQ	1.19	3.26	0
MCMSVQ	1.15	3.78	0.01

En conclusion, on peut dire que la technique de multi dictionnaires a donné quatre chances pour le vecteur LSF pour trouver un vecteur code et ce à travers 4 quantificateurs différents. L'allocation de bits change. Si on prend par exemple le quantificateur 3SVQ à 24 bits/trame on a la distorsion qui est égale à 1.21dB et ce en utilisant tous les 24 bits pour les dictionnaires. Au même codage, 24bits/trame, le quantificateur MCSVQ a une distorsion moyenne égale à 1.19 dB avec l'utilisation effective de 22bits seulement. Aux mêmes conditions on a eu des distorsions moyennes de 1.18 et 1.15 dB pour respectivement les quantificateurs MCSVQ et MCMSVQ.

L'avantage de la chance donné au vecteur LSF par dans la quantification à multi dictionnaires a contribué d'une manière efficace dans la diminution de la distorsion moyenne.

La figure IV.15 montre les variations de la distorsion moyenne en fonction du codage par trame. Ces graphes sont relatifs aux quantificateurs 3SVQ, 3MSVQ, MCSVQ et MCMSVQ. On remarque que la distorsion générée par le quantificateur MCMSVQ est meilleur et ce tout le long des allocations de bits.

Les tableaux présentent aussi les pourcentages des Otliers 1ère forme en fonction des allocations de bits. On remarque une diminution des Outliers 1ère forme quand on passe d'un quantificateur à un autre. Dans ce cas le quantificateur qui a eu les valeurs les plus basses est le MCMSVQ. Ces valeurs ont été observées pour les codages par trame allant de 22bits/trame à 24bits/trame. Ces résultats sont illustrées à la figure IV.16

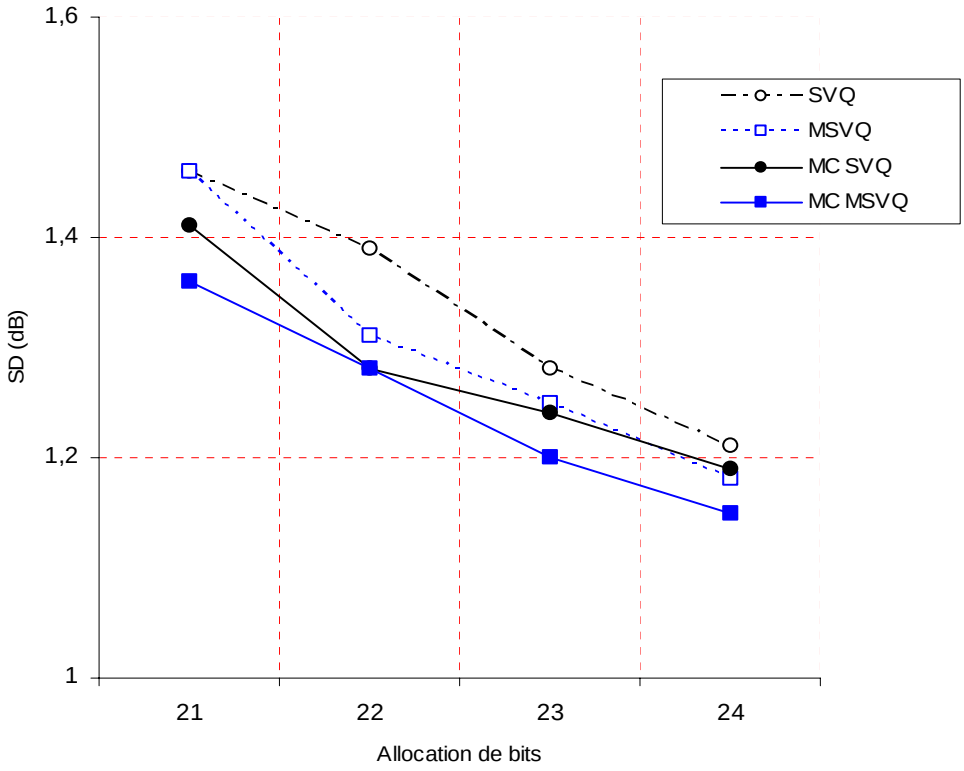


Figure IV.15: La distorsion spectrale moyenne en fonction des codage par trame pour les différents quantificateurs, SVQ, MSVQ, MCSVQ et MCMSVQ.

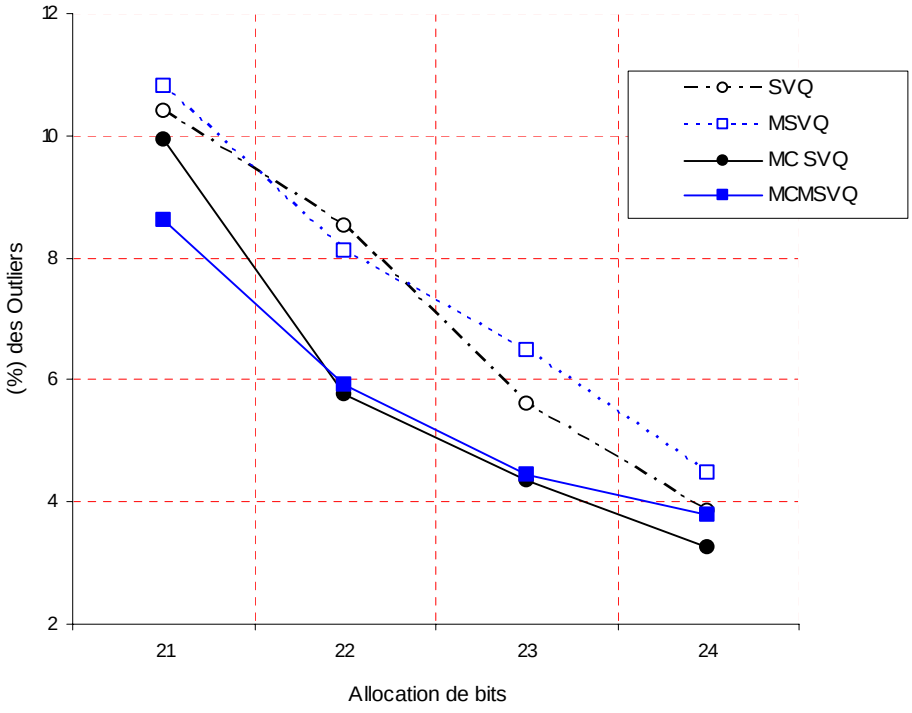


Figure. IV.16 Le pourcentage des vecteurs éloignés ayant le la distorsion spectrale SD entre 2dB et 4dB en fonction des codages par trame pour les différents quantificateurs, SVQ, MSVQ, MCSVQ et MCMSVQ.

Conclusion

Dans cette partie, nous avons proposé cinq méthodes relatives aux transformations des dictionnaires de la quantification vectorielle des coefficients LSF. Deux de ces cinq méthodes se trouvant dans la littérature et les trois autres sont de nouvelles approches que nous proposons et sont implantées dans le codeur CELP (le standard FS1016). Pour évaluer ces méthodes, deux bases de données parole Américaine TIMIT et Arabe PAPE ont été utilisée. Des résultats des tests utilisant la distorsion spectrale et les distances euclidiennes simple et pondérée ont été donnée. Des comparaisons entre les différentes approches ont été faites où nous monté la diminution du nombre de bits

L'implantation de ces méthodes dans le codeur CELP a donné un rapport signal sur bruit qui n'a pas eu un grand changement quand on passe d'un codage par trame à un autre et ce pour le même quantificateur. Cet rapport a été pour ce quantificateur égale à 7,6 dB pour le 3SVQ, à 8.0 dB pour le MSVQ, à 7.8 dB pour le MCSVQ et enfin à 8.0 dB pour le MCMSVQ.

L'idée de la quantification à multi-dictionnaires a bien prouvée son efficacité sur les méthodes SVQ et MSVQ déjà existantes.

Annexe du chapitre IV

Pour la mise au point de nos méthodes, nous avons été amenés à élaborer des logiciels pour le traitement, l'affichage, l'observation des signaux de parole, la génération des différents dictionnaires (apprentissage et test), le codage et le décodage CELP etc. ….

Pour présenter nos résultats d'une manière condensée, nous avons extrait un fichier d'un locuteur masculin, et un autre d'un locuteur féminin pour la base de données arabe (pour la phrase **كم بعـت ثمرنا ؟**) et pour les cinq méthodes vues précédemment.

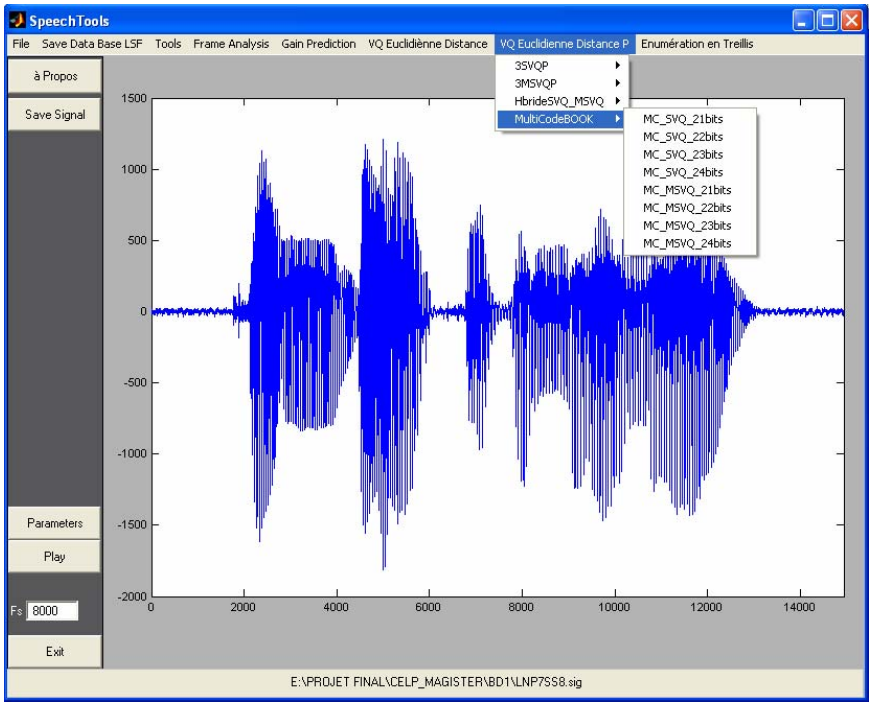


Figure.1 Le software"Speech Tools" conçu pour la génération des dictionnaires de la quantification vectorielle

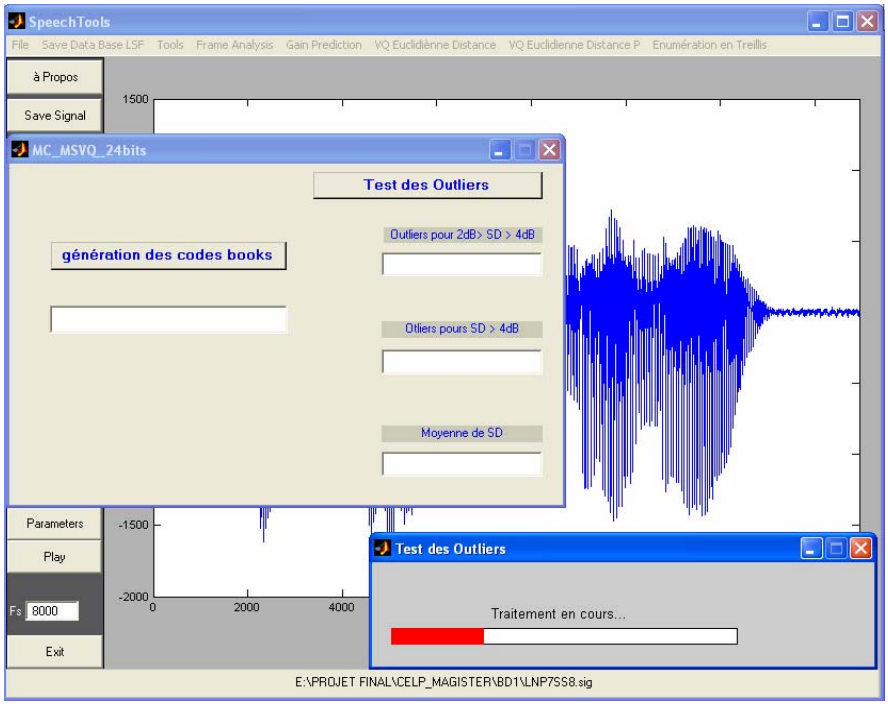


Figure.2 Exemple de déroulement de test de la quantification vectorielle MCMSVQ à 24bits/trame

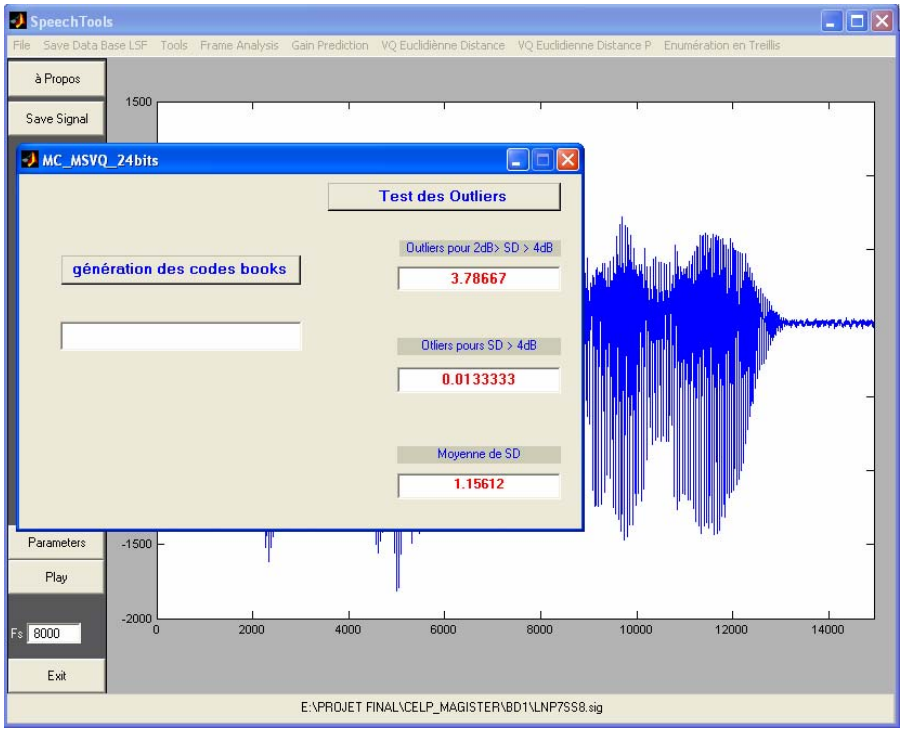


Figure.3 Résultat du test observé à la figure 2

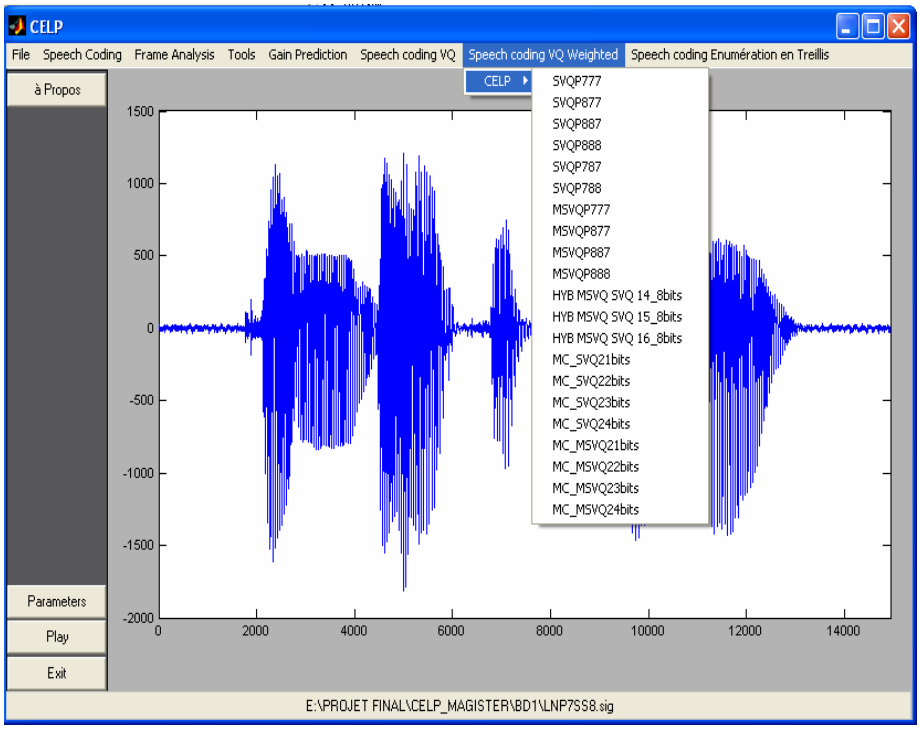


Figure.4 Le codeur CELP implanté basé sur la norme FS1016.

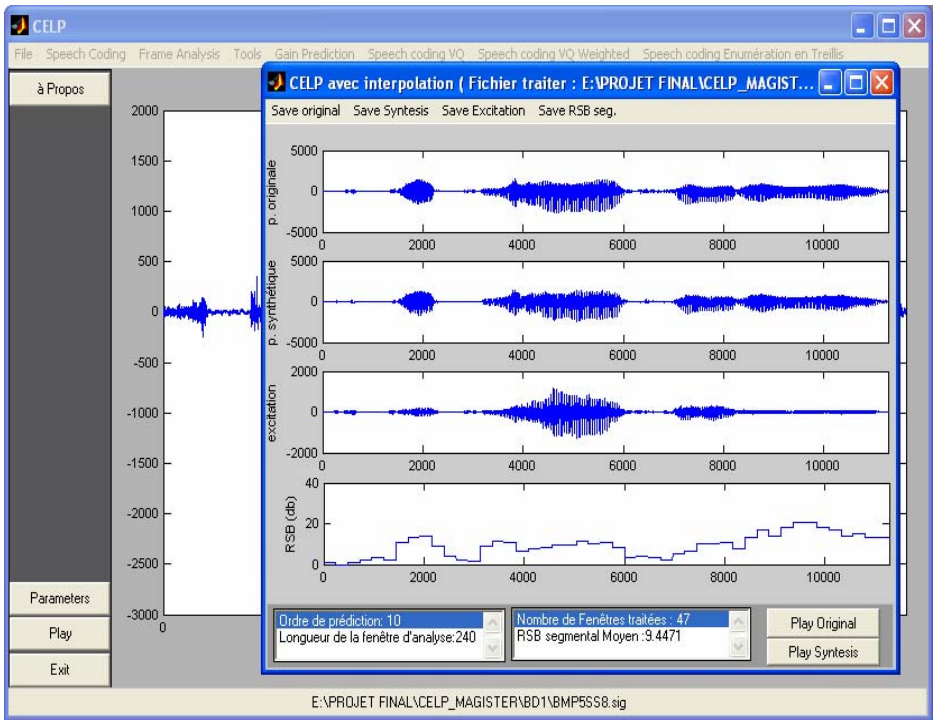
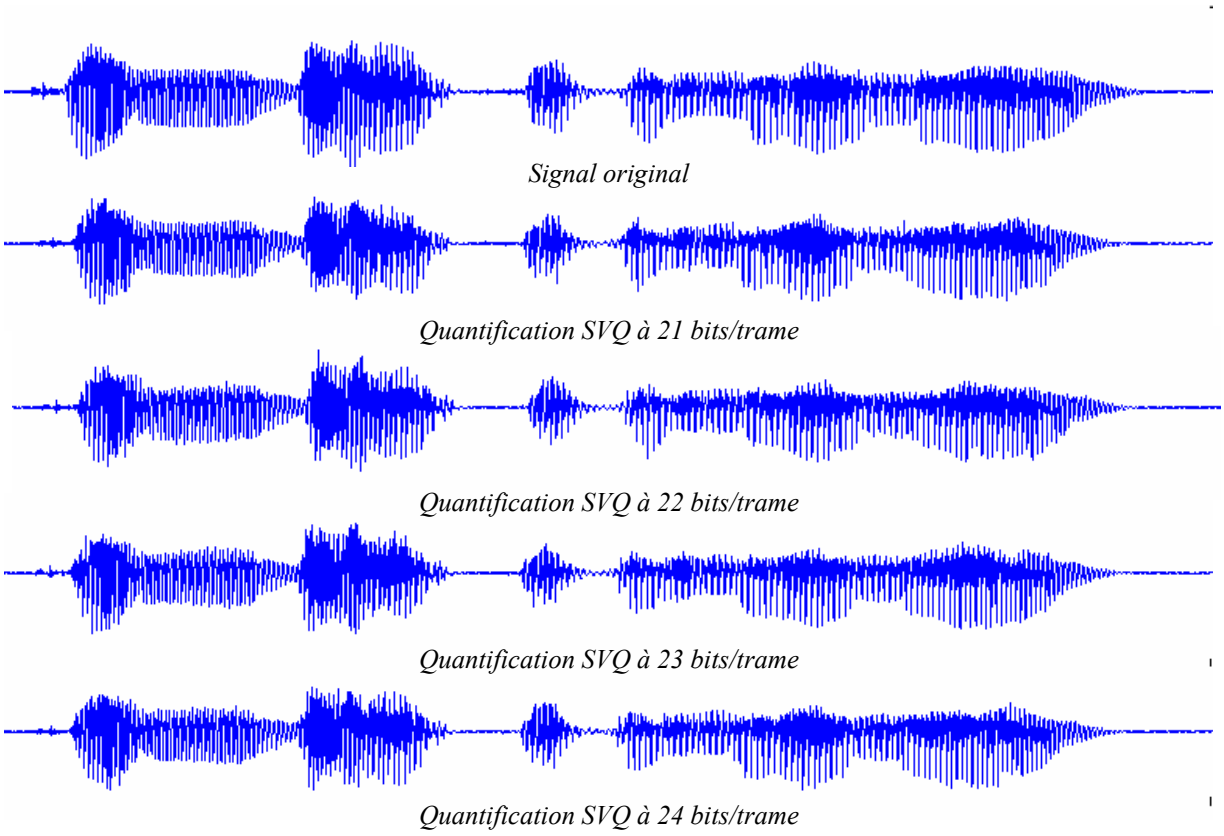
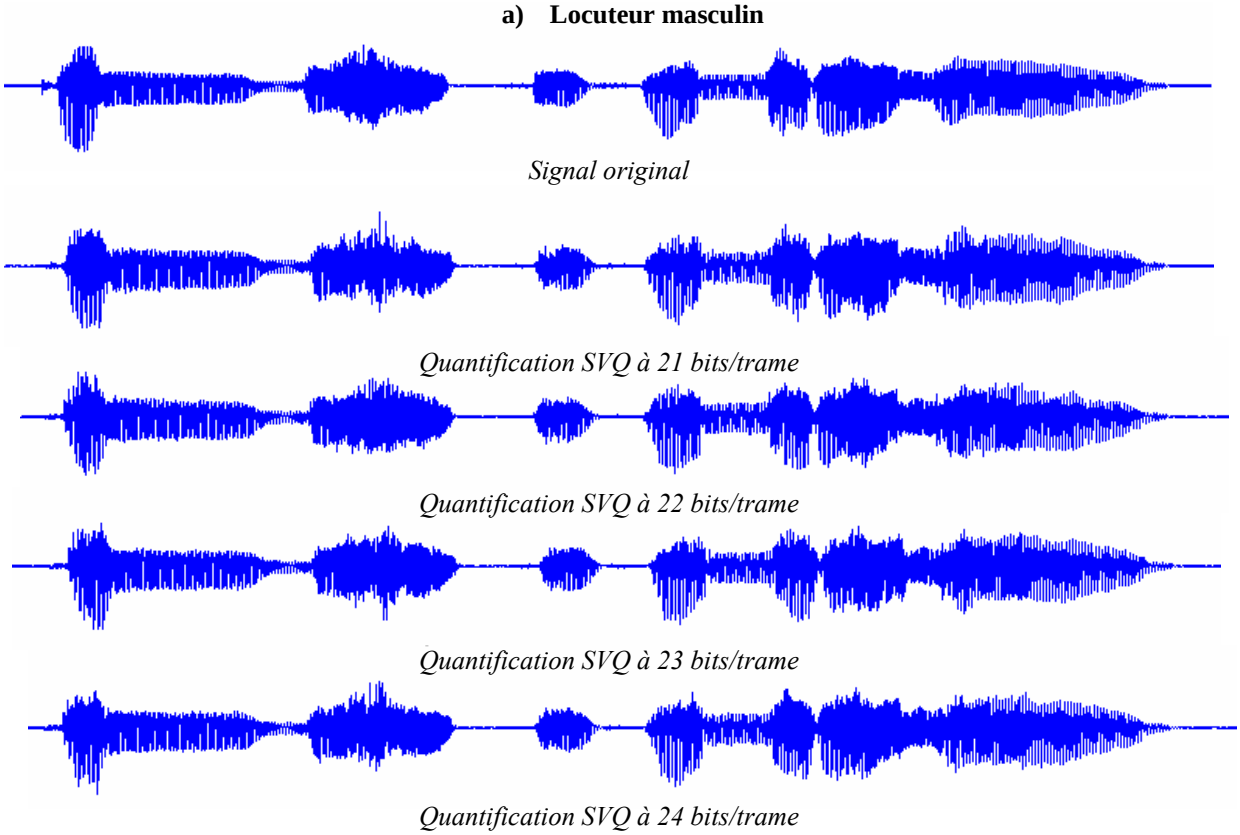


Figure 5. Exemple de résultats obtenus après le déroulement de la procédure du codage et de décodage

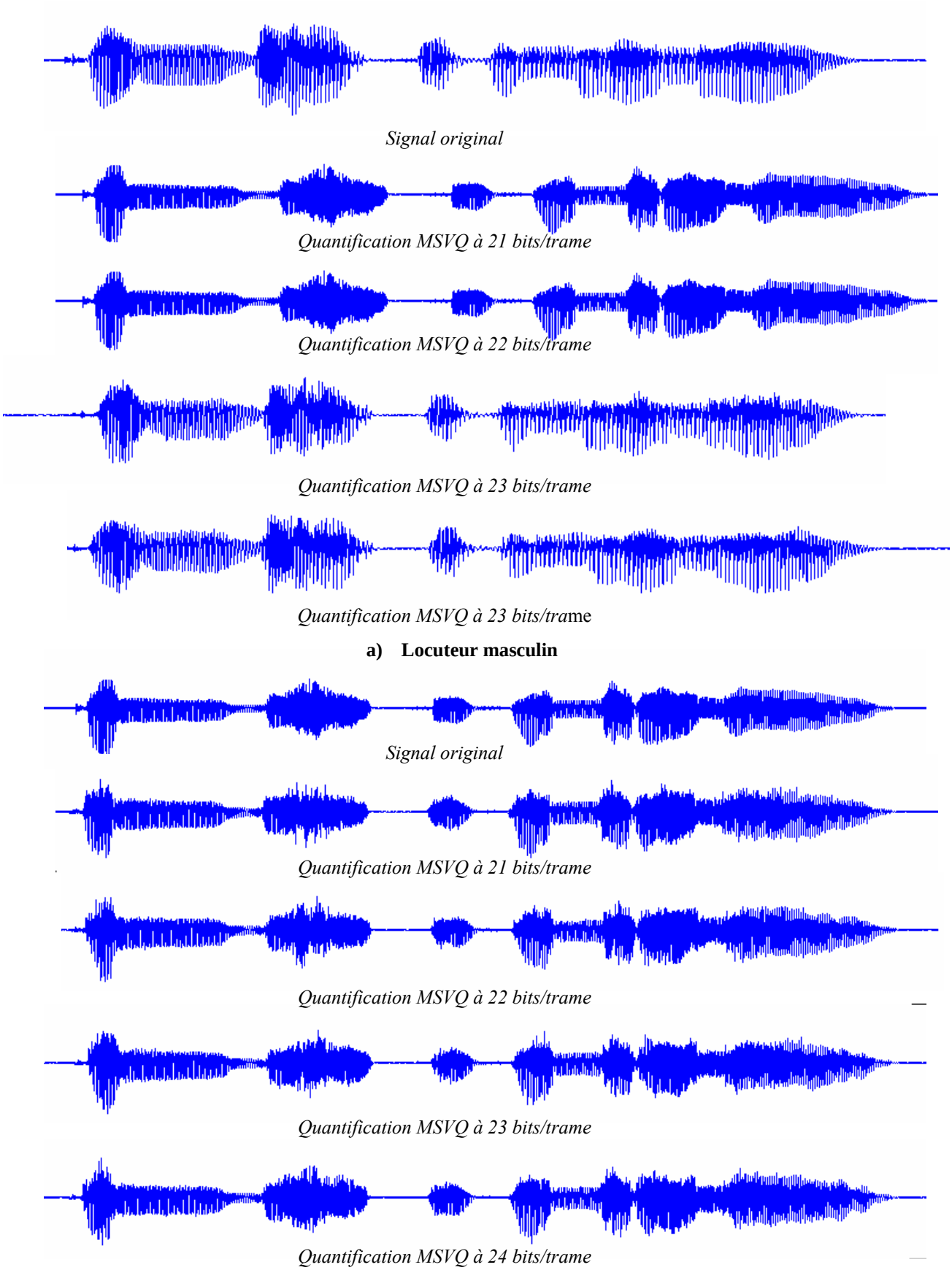


a) Locuteur masculin



b) Locuteur féminin

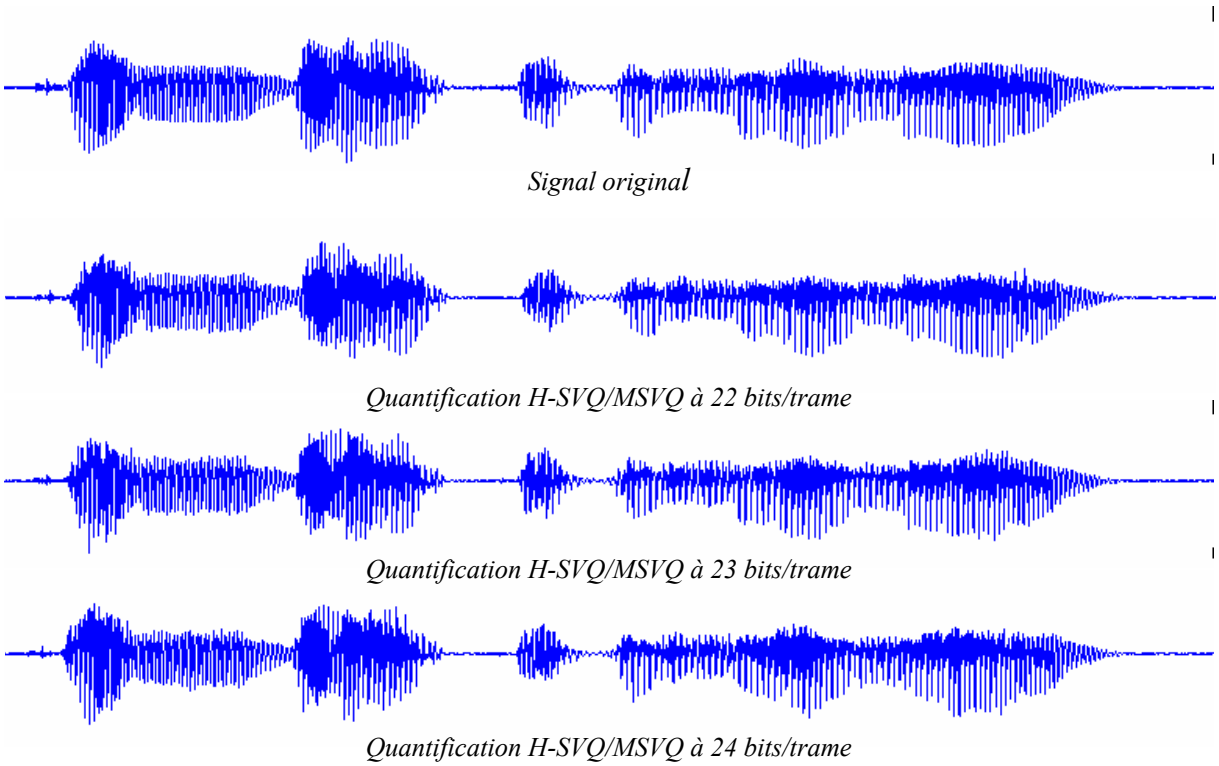
Figure 6 Les signaux de parole originaux et synthétiques pour le CELP utilisant la quantification SVQ



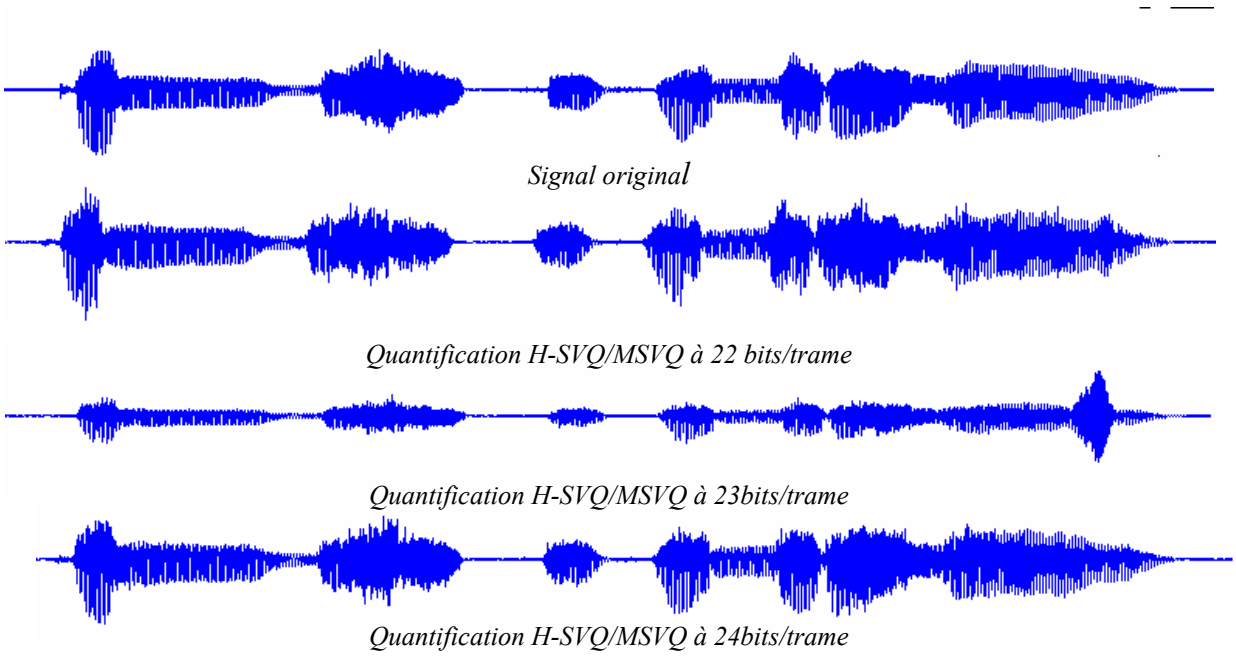
a) Locuteur masculin

b) Locuteur féminin

Figure 7 Les signaux de parole originaux et synthétiques pour le CELP utilisant la quantification MSVQ

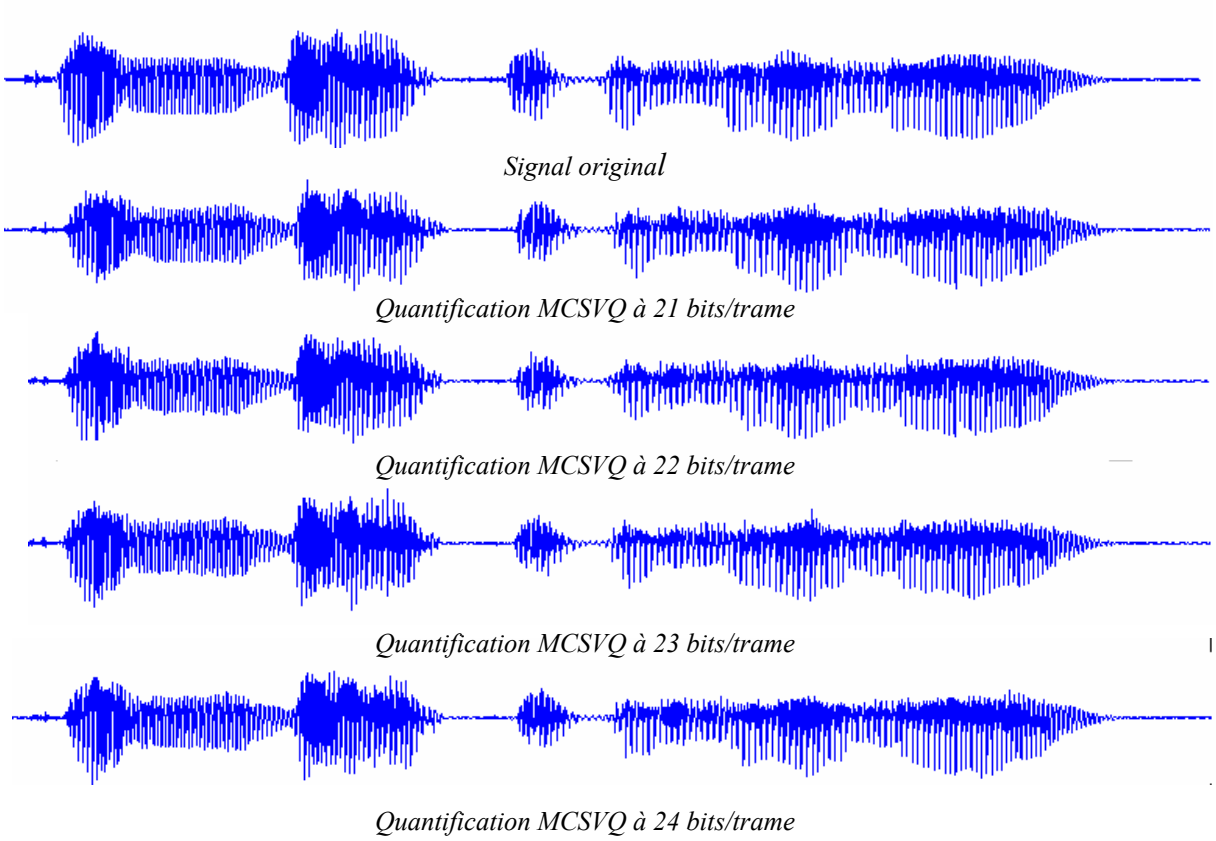


a) Locuteur masculin

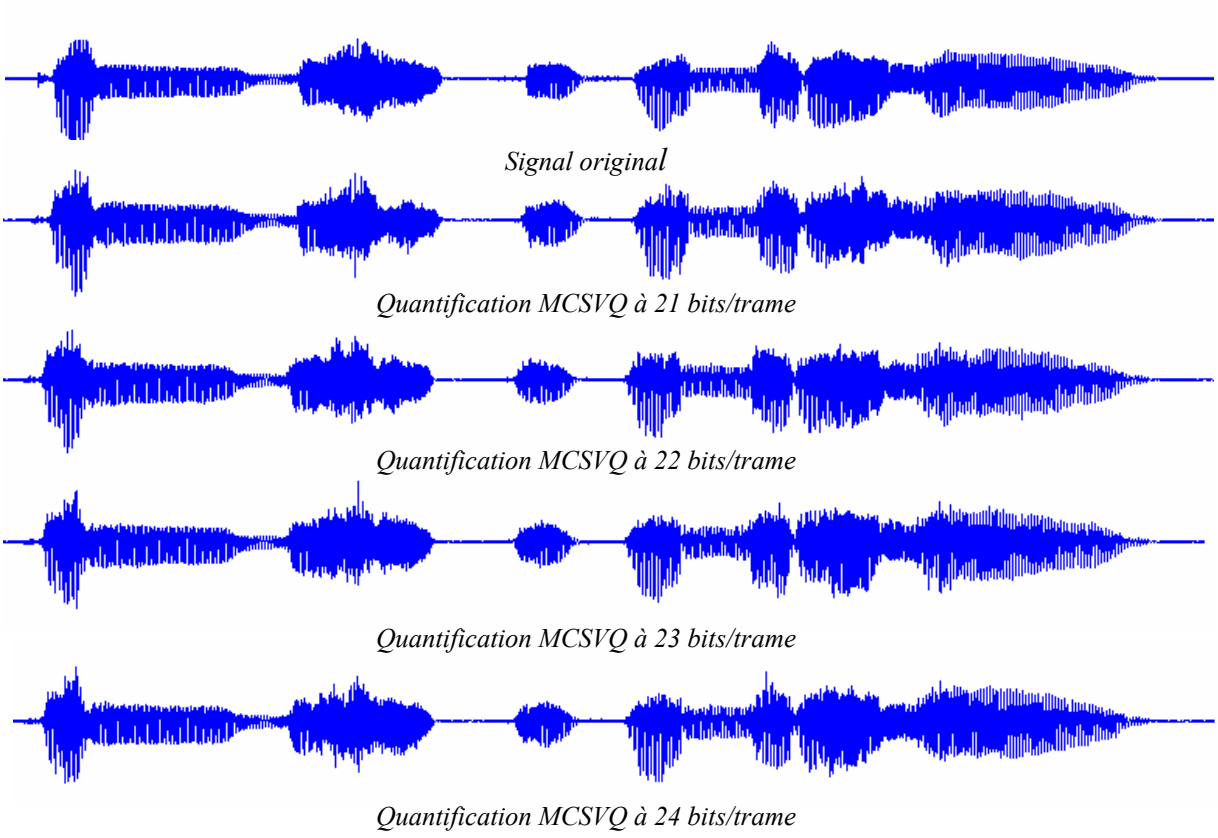


b) Locuteur féminin

Figure 8 Les signaux de parole originaux et synthétiques pour le CELP utilisant la quantification H-SVQ/MSVQ

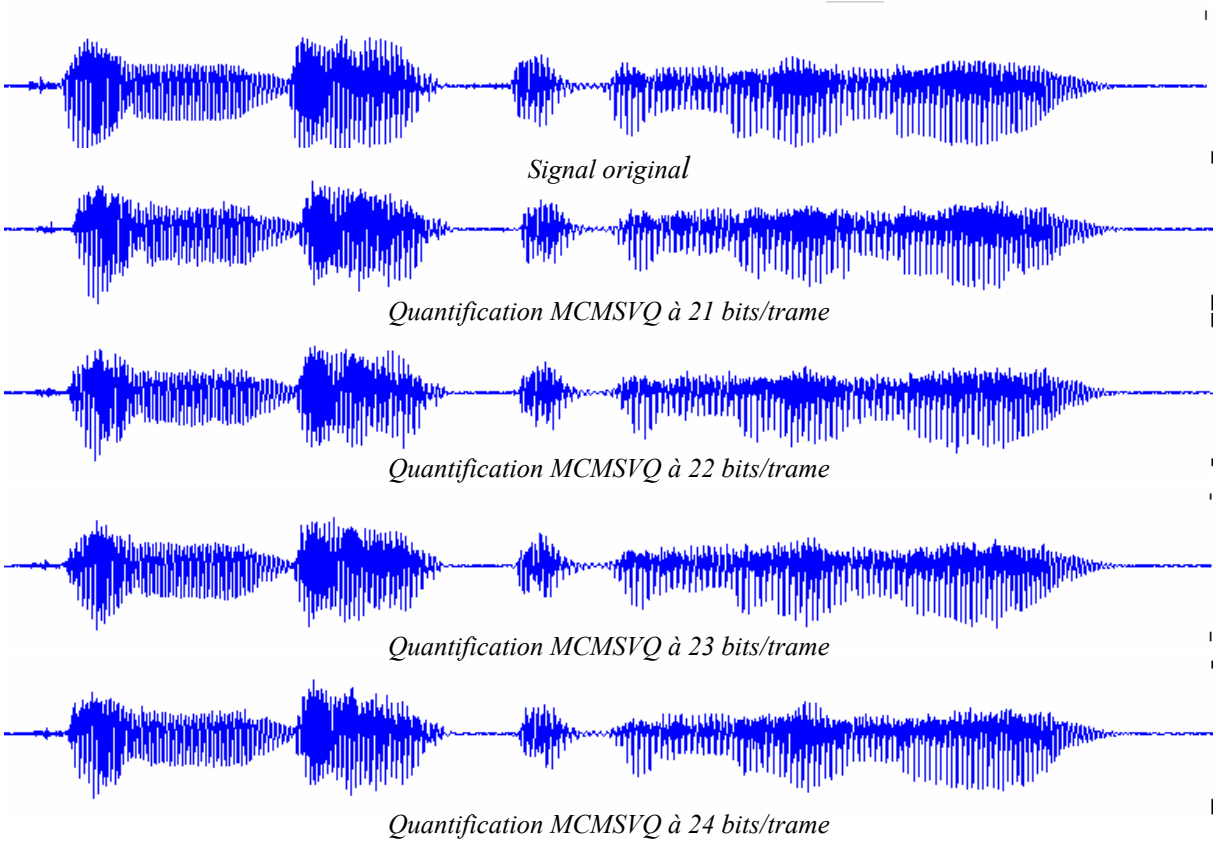


a) Locuteur masculin

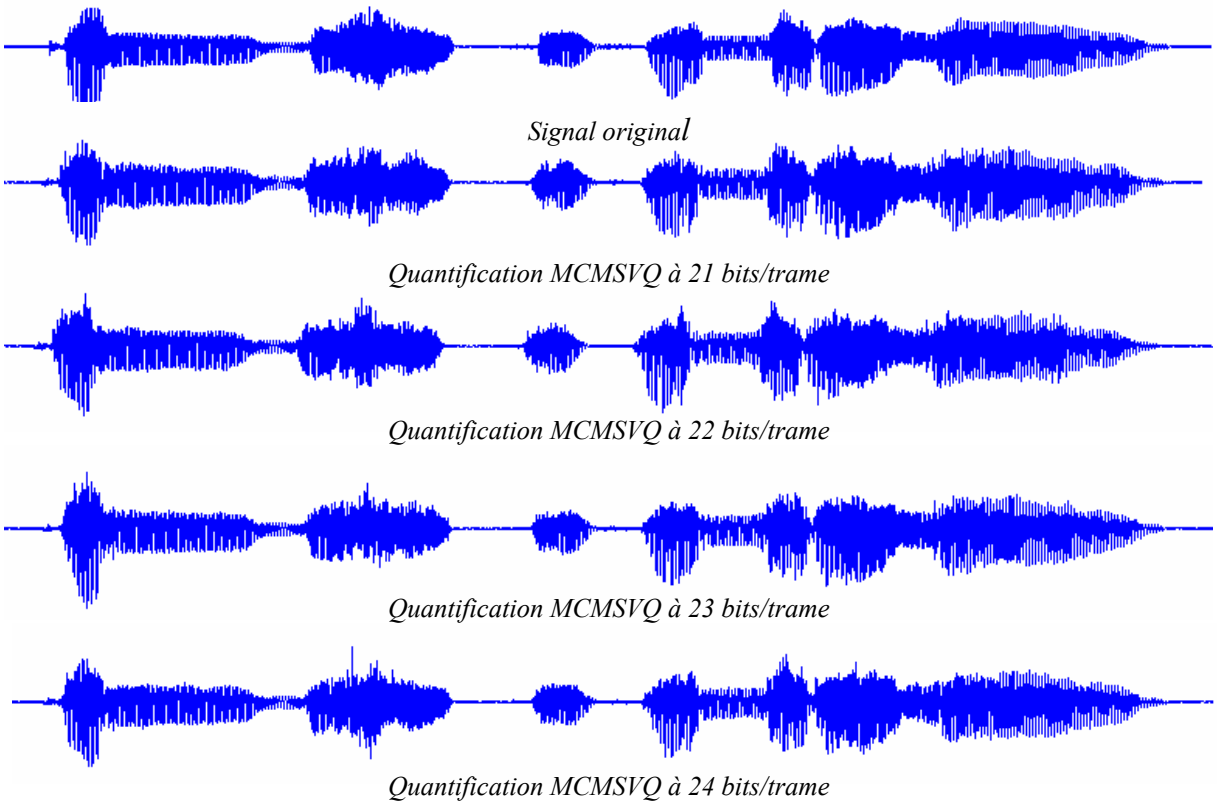


b) Locuteur féminin

Figure 9 Les signaux de parole originaux et synthétiques pour le CELP utilisant la quantification MCSVQ



a) Locuteur masculin



b) Locuteur féminin

Figure 10 Les signaux de parole originaux et synthétiques pour le CELP utilisant la quantification MCMSVQ

Conclusion

Le but recherché par cette étude était la diminution du débit observé par le codeur CELP basé sur la norme FS1016 (standard développé par le département de la défense des Etats Unis d'Amérique "DoD" et le laboratoire AT&T Bell).

Pour ce faire, on s'est intéressé à la quantification des coefficients LSF (Line Spectrum Frequencies). Dans son origine, la norme FS1016 propose une quantification scalaire de ces coefficients codée sur 34 bits. Nous avons jugé ce débit élevé et nous avons proposé des méthodes visant la diminution de celui-ci.

Dans une première étape on a amélioré la quantification scalaire utilisée dans cette norme. Pour cela, on a effectué une analyse fine des différentes combinaisons données par les dictionnaires de cette quantification et on est parvenu à réduire le nombre de bits alloués à ces dictionnaires de 34 bits à 30 bits. Pour un vecteur LSF donné, le même résultat de la quantification peut être obtenu soit par le quantificateur scalaire classique soit par le système d'énumération en treillis que nous avons conçu. Cela explique que le FS1016 n'a pas eu de changement dans ses performances avec cependant un gain de 4 bits.

Dans une seconde étape, on a élaboré cinq méthodes de quantification vectorielle qui ont permis une diminution de la distorsion.

Les deux première méthodes (SVQ et MSVQ) ont été implantées et testées en utilisant deux distances : la distance euclidienne simple et la distance euclidienne pondérée. Deux bases de données parole : TIMIT (américain) et PAPE (arabe standard), ont été utilisées pour les tests. Les résultats ont montré l'intérêt de l'utilisation des poids spectraux dans la distance euclidienne pondérée.

Les trois dernières méthodes de quantification vectorielle (H-SVQ/MSVQ, MCSVQ et MCMSVQ) sont les variantes nous avons proposées pour l'amélioration des deux premières méthodes en diminuant la distorsion induite dans leurs utilisations. Le meilleur résultat de distorsion, égal à 1.15 dB, a été enregistré pour le test de la MCMSVQ pour un codage de 24 bits/trame. Ce quantificateur, MCMSVQ, utilise effectivement 22 bits pour la quantification

au lieu de 24 bits utilisés dans le quantificateur classique MSVQ. Pour le même codage par trame on a observé une nette amélioration du quantificateur SVQ rapport au MCSVQ avec une distorsion de 1.19 dB.

De plus, une comparaison entre nos méthodes et la quantification scalaire proposée par la norme FS1016 a été faite. Cette comparaison a porté sur l'allocation de bits et la distorsion induite. Nous avons obtenu d'une part une diminution du nombre de bits (10 bits) et d'autre part une diminution de la distorsion qui atteint la valeur de 0.42 dB.

Les perspectives futures de ce travail sont nombreuses. Sur le plan de la quantification on souhaiterait voir l'utilisation de l'effet adaptatif sur les méthodes de quantification présentées et en particulier le quantificateur MCMSVQ.

Sur le plan de l'extension vers d'autre standard, il serait utile d'appliquer nos résultats au nouveau standard fédéral MELP (Mixed Exited Linear Prediction) qui utilise une quantification à multi-étages (MSVQ) et qui pourrait être substituée par la quantification multi-étages multi-dictionnaires (MCMSVQ). On peut imaginé des schémas de communication pour les réseaux IP (VoIP: Voice over IP), pour la téléphonie sécurisées par cryptophonie, pour la messagerie vocale, pour les communications par satellite par relais de trame (frame relay), etc.

Bibliographie

[01] Calliope "la parole et son traitement automatique" Edition Masson, *1989*.

[02] Thierry Dutoit "Introduction au traitement automatique de la parole", *faculté Polytechnique de Mons, Edition 2000.*

[03] M'hamed DABOUZ "Transmission de la parole à faible débit par vocodeur à Classification" t*hèse de doctorat, département systèmes et Communications, ENST, janvier 1984.*

[04] Tamana Islam "Interpolation of linear prediction coefficients for speech coding" *Master Engineering, department of electrical engineering Mc GILL University Montreal, Canada, Avril 2000.*

[05] Benjamin W. Wah et Dong Lin "LSP-Based Multiple-Description Coding for Real-Time Low Bit-Rate Voice Over IP", *IEEE Transaction on Multimedia, VOL. 7, NO. 1, pp 167-178 Fév 2005*

[06] MR Schroeder et BS Atal "Code Excited Lineaire Prediction (CELP): High quality speech at very low bit rates" *Proc Int CASSP, pp:940-973, 1985.*

[07] M. Djamah, M.Boudraa, B. Boudraa, M. Bouzid,"Un logiciel de codage de la parole base sur le FS1016" *XXIV^{eme} journée d'étude sur la parole (JEP), NANCY, 24-27 juin 2002.*

[08] Taman Islam et Peter Kabal " Partial energy weighted interpolation of lineair prediction coefficients" *Proc IEEE, workshop on speech coding, pp: 105-107, Sep 2000.*

[09] Djamah Mouloud "Codage par prediction linéaire à excitation par code (CELP), étude à débits variable et réduction de la complexité", *thèse de magister, USTHB juin 1996.*

[10] Tom Bäckström, Paavo Alku, Thomas Paatero, Bastian Kleijin Atime domain interpolation for the LSP decomposition", *IEEE Transaction on Speech and Audio Processing, vol 12, No6, pp: 554-560, November 2004.*

[11] F K Soong et B Juang "Line spectrum pair LSP and speech data compression", *Proc ICASSP, pp:1.10.1 – 1.10.4, 1984.*

[12] P Kabal, P Ramachandran "The computation of line spectral Frequencies using Chebychev polynomial ", IEEE Trans on Acoustics, Speech ans Signal Processing. Vol 34 N°6, pp 1419 – 1426, 1986.

[13] ITU "Coding speech at 8Kbps using conjugate structure Algebraic code excited linear prediction", *ITU Recommendation G.729, Telecommunication standardization sector of ITU (03/96).*

[14] ITU " Dual rate speech coder for multimedia communication transmitting at 5.6 and 6.3Kbps ", *ITU Recommendation G.729.1, Telecommunication standardization sector of ITU (03/96).*

[15] K. Järvinen et *al* "GSM Enhanced full rate speech coder", *Proc IEEE, ICASSP'97 pp 771-774. 1997.*

[16] T. Ohya et *al* "5.6 Kbps PSI-CELP of the half rate PDC speech coding standard", *Proc IEEE Vehicular technology conference, Vol 1 , pp 1680 – 1684. 1994.*

[17] " Federal standard 1016, Telecommunications analogue to digital conversation radio

voice by 4800 bits/s code excited linear prediction (CELP)", National communication systems office of technology and standards. Washington DC. 20305 –20310, 1991.

[18] Vladimir Cuperman, Allen Gersho, Jan Linden, Ajit Rao Tung-Chiang Yang "A novel

approach to excitation coding in low bit rate high quality CELP coders", *IEEE ,pp 14 - 16 March 2000*

[19] Jack Stachurski, Alan Mc Cree, Vishu Viswanhan "High quality MELP coding at bit

rate around 4 kbps" *IEEE, pp 485 – 488, March 1999*

[20] Weiran Lin, Soo Ngee Koh, Xiao Lin, "Mixed excited linear prediction coding of wide band speech at 8kbps", *IEEE, pp 1137 – 1140, April 2000.*

[21] Tian Wang Kazuhito Koishida, Vladimir Cuperman, Alen Gersho, John S Collura "A 1200 bps speech coder based on MELP", IEEE pp: 1375 – 1378, April 2000.

[22] Edward J. Daniel, Keith A. Teague, "Federal standard 2.1 kbps MELP over IP", *43rd IEEE Midwest Symposium on circuits and systems, Lansing MI, pp: 568 – 571, Aug 8.11.2000*

[23] Li wenyuan, MA Xin, zhang Yuzhong, Zeng Yifang "study and development of MELP vocoder", *IEEE, ICSP'02 Proceeding, pp: 1684-1688, Juin 2002*

[24] Kuldip K Paliwal, Bishnu S. Atal "Efficient vector quantization of LPC parameters at 24bits/frame", *IEEE Transaction on Speech and Audio ¨Processing, Vol 1 NO1, pp: 3-14, junuary 1993.*

[25] M.W. Marcellin & T.R. Fischer, "Trellis coded quantization of memoryless and Gauss-Markov sources", *IEEE Transactions on Communications, Vol. 38, pp. 83-93, Jan 1990.*

[26] Sangwon Kang, Youngwon Shin, Thomas R. Fisher, "Low-complexity predictive trellis –coded quantization of speech line spectral frequencies", *IEEE Transaction on signal processing, Vol. 52, N07, pp 2070-2079, July 2004.*

[27] F Lahouti, A K Khandani, "Quantization of LSF Parameters using A Trellis Modeling", *IEEE International Conference on Acoustics, Speech and Signal Processing, Istanbul, Turkey, May 2000*

[28] G. D. Forney, Jr., "The Viterbi algorithm, " *Proc. IEEE (Invited Paper), vol. 61, pp. 268-278, Mar. 1973.*

[29] Kevin T Malone, Thomas R Fisher "enumeration and trellis searched coding schemes for speech LSP parameters", *IEEE Transaction on speech and audio processing, vol1,N03, pp 304-314, July 1993.*

[30] Allen Gersho, "Qauntization", *IEEE Communication Society Magazine, pp: 16-29, Sept 1977.*

[31] Allen Gersho, "Principals of quantization", *IEEE Transaction on circuits and systems, Vol CAS-25, N07, pp: 427-436, july 1978.*

[32] P. Lloyd, "Least squares quantization in PCM", *IEEE, Trans Info theory, Vol IT-28, pp: 129-136, Mar 1982.*

[33] J Max, "quantization for minimum distorsion", *IRE Trans Info: Theory , Vol 6, pp: 07-12, Mar 1960.*

[34] J. Garofolo, L. Lamel, W. Fisher, J. Fiscus, D. Pallett, N. Dahlgren, and V. Zue, "Timit acoustic-phonetic continuous speech corpus", *Linguistic Data Consortium, 1993.*

[35] M. Boudraa, B. Boudraa, Bernard Guerin "Twenty List of Ten Arabic Sentences for Assessment" *ACUSTICA-acta acustica vol.86 (2000) 870-882 Nov 1998*

[36] C. E. Shannon, "A mathematical theory of communication, " *Bell Systems Technical Journal, pp. 27:379-423, 623-656, 1948.*

[37] C. E. Shannon, "Coding theorems for a discrete source with a fidelity criterion," *in Proc. IRE National Convention Rec., Part 4, pp. 142-163, 1959.*

[38] Reni Boite, Murat Kunt, "Traitement de la parole"*, complément au traité d'électricité, Presses polytechniques Romandes, 1er Ed, ISBN 2-88074-140-1, 1987.*

[39] Chang Quian Chen, "An enhanced generalozed Lloyd algorithm", *IEEE signal processing letters, Vol 11. N02, pp 167-170, February 2004.*

[40] Nadim Batri, "robust spectral parameter coding in speech processing", *thesis of master engineering. Department of electrical engineering. Mc gill University. Montreal, Canada. 1998.*

[41] Yosef Linde, Andres Buzo, Robert M. Gray, "An algorithm for vector quantization design", *IEEE transaction on communications, vol COM-28, N01, January 1980.*

[42] Ioamis Katsavounidis, C.C Tay Luo, Zhen Zhang, "A new initialisation technique for generalized Lloyd iteration", *IEEE Signal Processing Letters, Vol 1, N10, pp 144-146,October 1994.*

[43] James H.Y Loo, "Inter frame and intra frame coding of speech spectral parameters", *thesis of master engineering. Department of electrical engineering. Mc gill University. Montreal, Canad. 1996.*

[44] Hai Le Vu, Lazio Lois "Efficient distance measure for quantization of LSF an dits Karhunen –Loeve transformation parameters", *IEEE transaction on speech and audio processing, vol 8, N06, pp744-746, November 200*0.

[45] Rajiv Laroi, Nam Phamdo , Nariman Farvadin "Robust and efficient quantization of speech LSP parameters using structured vector quantization", *IEEE S9.19, pp 641-644, 1991.*

[46] J. Makhoul, S. Roucos, and H. Gish, "Vector quantization in speech coding", *Proceedings of the IEEE, vol. 73, pp. 1551-1558, November 1985.*

[47] A. Boukhari, M. Djamah, M. Boudraa, B. Boudraa, "Evaluation d'une implémentation optimal sous Matlab d'une quantification vectorielle divisée (SVQ) des paramètres LSF pour le codage CELP", *CIGE'2004, pp 100 – 104, Sétif octobre 2004.*

[48] Sung Joo Kim, Yung Hwan Oh, “Split vector quantization of LSF parameters with minimum of dLSF constraint”, *IEEE signal processing letters, vol 6, N09, pp 227-229, September 1999.*

[49] Huilin Xiong, M, N, S Swamy, MO. Ahmed, “Competitive splitting for code book initialization”, *IEEE signal processing letters, vol 11, N05, pp 474-477, May 2004.*

[50] Moo young Kim, Nam Kyu ha, Sang Ryong Kim, “Linked split vector quantization of LPC parameters”, *IEEE pp 741-744, 1995*

[51] S Nadkumar, K Swaminathan, U Bhaskar, “Robust speech mode based LSF vector quantization”, *IEEE 1998.*

[52] Dong Il Chang, Young Kwon Cho, Souguil Ann “Efficient quantization of LSF parameters using classified SVQ combined with conditional splitting”, *IEEE, pp 736-739, May 1995*

[53] P Kabal “ITU-T G.729.1 speech coder: A Matlab implementation”, *department of electrical and computer engineering – Mc Gill University, 21-11-2003.*

[54] D H Lee, D L Neuhoff, K K Paliwal, “Cell conditioned multistage vector quantization”, *IEEE, S9.22, pp 653-656, 1991.*

[55] Wai Yip Chan, alen Gersho “Enhanced multistage vector quantization with constrained storage”, *24 ACSSC, MPLE press, pp 659-663, Dec 1990.*

[56] Venkatesh Krishnan, David V Andersson, Kwan K Truong, “Optimal multistage vector quantization of LPC parameters over noisy chanals”, *IEEE transaction on speech and audio processing, vol 12, N01, pp 1-8, January 2004.*

[57] Pao chi chang, Hsin Min Yu, “Dither like data hiding in multistage vector auqntiwqtion of :ELP qnd G 729 speech coding”, IEEE, pp 1199-1203, September 2002.

[58] F Lahouti, A K Khandani, “Reconstructed of multistage vector quantization sources over noisy channels. Application to MELP codec”, *ICASSP, Montereal QC, Canada, May 2004.*

[59] Woo Jim Han, Eun Kyoung Kim, Yung Hwan Oh, “Multicode book split vector quantization of LSF parameters”, *IEEE signal processing letters vol 9, N12, pp 418-421, December 2002.*

[60] A. Boukhari, M. Djamah, M. Boudraa, B. Boudraa, “Evaluation objective de la quantification vectorielle divisée à multi dictionnaires (MCSVQ) des paramètres LSF du codage CELP”, *conférence nationale de l'ingénierie électrique CNIE'04 Oran Novembre 2004.*

[61] A. Boukhari, M. Djamah, M. Boudraa, B. Boudraa, “Objective evaluation of multi-code-books vector quantization of LSF parameters of the CELP Coding ”, *the 5th International Arab Conference on Information Technology, Acit'2004 Constantine, pp 153-158, Décembre 2004.*

Annexe

Polynômes symétriques et polynômes antisymétriques

On :

$$A(z) = 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_{p-1} z^{-p+1} + a_p z^{-p} \qquad \text{A.1}$$

$$A(z^{-1}) = 1 + a_1 z^{1} + a_2 z^{2} + \cdots + a_{p-1} z^{p-1} + a_p z^{+p} \qquad \text{A.2}$$

Le polynôme *P(z)* est défini par :

$$P(z) = A(z) + z^{-(p+1)} A(z^{-1}) \qquad \text{A.3}$$

Après développement de A.2 on aura :

$$P(z) = 1 + (a_1 + a_p) z^{-1} + (a_2 + a_{p-1}) z^{-2} + \ldots + (a_{p-1} + a_2) z^{-p+1} + (a_p + a_1) z^{-p} + z^{-(p+1)} \qquad \text{A.4}$$

De l'équation A.4 on dit que le polynôme *P(z)* est symétrique parce que le coefficient de z^{-n} est égale au coefficient de $z^{+n} / z^{+(p+1)}$ et ce pour $0 \le n \le p+1$.

De même on donne le polynôme *Q(z)* comme suit :

$$Q(z) = A(z) - z^{-(p+1)} A(z^{-1}) \qquad \text{A.5}$$

Après développement on aura :

$$Q(z) = 1 + (a_1 - a_p) z^{-1} + (a_2 - a_{p-1}) z^{-2} + \ldots + (a_2 - a_{p-1}) z^{-p+1} + (a_1 + a_p) z^{-p} - z^{-(p+1)} \qquad \text{A.6}$$

De l'équation A.6 on remarque que le polynôme *Q(z)* est antisymétrique parce que le coefficient de z^{-n} est égale à l'opposé du coefficient de $z^{+n} / z^{+(p+1)}$ et ce pour $0 \le n \le p+1$.