

8/99

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Ecole Nationale Polytechnique

D.E.R. de Génie Electrique & informatique
Département d'Electronique



المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

Projet de fin d'études

Pour l'obtention du diplôme d'ingénieur d'état
en Electronique

Thème

UTILISATION DU TREILLIS
POUR LE CODAGE DE LA PAROLE

Proposé et dirigé par :

Dr D. BERKANI
M^{elle} F. MERAZKA

Etudié par :

Y. DICHE
A. KADDAI

Promotion : juin 99

E.N.P.10, Avenue Hassen-Badi, EL-HARRACH, ALGER.

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Ecole Nationale Polytechnique

D.E.R. de Génie Electrique & informatique
Département d'Electronique



المكتبة الوطنية المتعددة التخصصات
Ecole Nationale Polytechnique

المعرفة الوطنية المتعددة التخصصات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

Projet de fin d'études

Pour l'obtention du diplôme d'ingénieur d'état
en Electronique

Thème

UTILISATION DU TREILLIS
POUR LE CODAGE DE LA PAROLE

Proposé et dirigé par :

Dr D. BERKANI
M^{elle} F. MERAZKA

Etudié par :

Y. DICHE
A. KADDAI

Promotion : juin 99

E.N.P.10, Avenue Hassen-Badi, EL-HARRACH, ALGER.

استعمل التكميم الشعاعي المشفر بالشبكة لتشفير الوسائط LSP , و قد استعملت طريقتين للتكميم ، في الأولى :
نجزئ الشعاع LSP ذو البعد 10 إلى خمسة أشعة جزئية ثنائية البعد و نكمم الشعاع LSP بالشبكة على خمسة مراحل. أما
في الثانية ، نجزئ شعاعين LSP إلى ستة أشعة جزئية على الشكل 4 3 3 4 3 3 ونكمم الشعاعين LSP بالشبكة على ستة
مراحل . وبدلالة الخطأ الطيفي، نجد أن الشكل الثاني للتشفير هو الأفضل مقارنة بالأخر بحيث يرتكب خطأ طيفي يعادل
1.35dB عند التدفق 22.5bits/trame.
المصطلحات: التكميم الشعاعي المشفر بالشبكة، الوسائط LSP ، الخطأ الطيفي.

Abstract:

Trellis Coded Vector Quantization (TCVQ) is used for LSP parameters encoding. Two encoding schemes are considered. In the first scheme, the LSP vector of ten dimensions is splitting in five two dimensional subvectors and each LSP vector is quantized using five trellis time steps. In the second scheme, two LSP vectors (20 LSP) are splitting in 3 3 4 3 3 4 structure and each two LSP vectors are quantized using six trellis time steps.

The last encoding scheme is best, and can achieve 1.35 dB spectral distortion using about 22.5 bits for each speech frame.

Keywords: Trellis Coded Vector Quantization, LSP parameters, spectral distortion.

Résumé :

La Quantification Vectorielle Codée par Treillis (TCVQ) est utilisée pour coder les paramètres LSP. On considère deux structures de codage : dans la première structure on partitionne le vecteur LSP de dimension 10 en cinq sous vecteurs bidimensionnels quantifiés sur cinq étapes de transition du treillis. La deuxième quantifie deux vecteurs LSP (20) divisés en 3 3 4 3 3 4 sous vecteurs sur six étapes de transition du treillis.

La dernière structure donne des meilleures performances, et peut avoir une distorsion spectrale égale à 1.35 dB à 22.5 bits/trame.

Mots clés : Quantification Vectorielle Codée par Treillis, paramètres LSP, distorsion spectrale.

Remerciements



Nous adressons nos sincères remerciements et notre profonde gratitude à notre promoteur, M^r D. BERKANI, pour l'aide qu'il nous a apporté, ses conseils judicieux qu'il nous a prodigué, son attention et sa grande patience pour nous guider et nous encourager pendant toute la durée du travail.

Nous remercions vivement M^{elle} F. MERAZKA pour son aide, ses conseils avisés et ses efforts qu'elle a consentis pour nous diriger.

Tous nos remerciements :

M^r F. FLITTI pour l'aide qui nous a fourni.

Tous nos enseignants qui ont contribué à notre formation.

A nos camarades et amis.

Nous tenons à remercier tous ceux qui nous ont aidé à élaborer ce modeste travail.

الإهداء

إلى والدي العزيزين أتوجه بهذا العمل
لقد طبعتما في حسي حب الدراسة وحب التطلع
إلى المستقبل الزاهر ، فلم أجد روء الجميل إلا
إهداء كما هذا العمل المتواضع.
كما أتقدم بإهدائي إلى اخوتي الثلاثة «عبري
وابنته، مراد، عبر الرحيم
وإلى أختي العزيزة فوزية وابنتها ، وإلى كل
أقاربي و إلى أصدقائي ، أخص بالذكر صريقي
العزيز وحون علي وكل عائلته.

يوسف

الإهداء

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

إلى كل المخلصين في هذه الأرض
الطيبة ، أهري هذا العمل المتواضع

عبر الله

Table des matières



INTRODUCTION :	1
CHAPITRE I : GENERALITE	3
I.1. La théorie de distorsion :	3
I.1.1. Entropie d'une source :	3
I.1.2. La notion de distorsion :	5
I.2. Système de communication :	9
I.2.1. Chaîne de communication :	9
I.2.2. Critère de performance dans le codage de la parole :	10
I.2.3. Mesure de la qualité :	12
I.3. Les redondances dans le signal parole :	12
I.4. le signal vocal :	14
I.4.1. Caractéristiques psycho-acoustiques :	14
I.4.2. Production du Signal Vocal :	15
I.4.3. La Prédiction Linéaire (LP) :	19
I.4.4. La Méthode d'Autocorrélation :	21
I.4.5. Méthode de covariance :	23
CHAPITRE II : LA QUANTIFICATION VECTORIELLE	26
II.1. préliminaires :	26
II.1.1. Quantification scalaire QS :	27
II.2. Principe de la quantification vectorielle :	28
II.3. Conditions d'optimalité :	32
II.3.1. Condition du plus proche voisin :	33
II.4. Construction de quantificateurs statistiques :	34
II.4.1. Algorithme de Lloyd Généralisé (GLA : Generalized Lloyd Algorithm):	35
II.5. Technique de conception du dictionnaire initial :	40

II.6.Comparaison entre quantificateurs scalaires et vectoriels :	41
CHAPITRE III : LA QUANTIFICATION PAR TREILLIS	43
III.1.Introduction :	43
III.2.La quantification codée par treillis :	44
III.2.1.Description :	44
III.2.2.Représentation :	46
III.2.3.La distorsion :	49
III.2.4.Interprétation :	49
III.2.5.Algorithme de quantification par treillis :	49
III.2.6.Exemple d'application :	50
III.3.La quantification vectorielle codée par treillis (TCVQ) :	52
III.3.1.introduction :	52
III.3.2.La structure du codeur :	53
III.3.3.La complexité du codeur :	54
III.3.4.La conception du codeur :	55
III.3.5.Les propriétés du codeur :	58
CHAPITRE IV : LES PARAMETRES LSF	59
IV.1. Introduction :	59
IV.2. Extraction des paramètres LSF :	59
IV.2.1. Les conditions d'analyse :	59
IV.2.2. Constitution des trames :	60
IV.3. Le filtre de synthèse :	61
IV.3.1. Calcul des LSF :	61
IV.3.2. Algorithme de Calcul des LSF :	63
CHAPITRE V : LA QUANTIFICATION DES PARAMETRES LSF - RESULTATS ET SIMULATIONS	66
V.1. Introduction :	66
V.2. Mesure de la distance entre les LSF :	66
V.3. Mesure des performances de la quantification des LSF :	67

V.4.Simulations et résultats :	68
V.4.1.Introduction :	68
V.4.2.Les conditions d'analyse :	68
V.4.3.Simulation des différents codeurs :	69
Conclusion :	75
CONCLUSION	76
Annexe A :	78
Annexe B :	81
Annexe C :	83
Bibliographie	85

INTRODUCTION



L'évolution de la technologie notamment des DSP (Digital Signal Processor) a permis un développement considérable dans l'application des méthodes de traitement de signal. Les communications sont les grandes bénéficiaires de cette avancée. Ainsi de nombreux travaux de recherche sont consacrés au codage des formes d'ondes appliqué au signal parole.

Dans la transmission digitale, on essaye toujours d'améliorer la qualité du signal transmis à travers le canal tout en réduisant le nombre de bits et en gardant une qualité satisfaisante. Ce problème ainsi posé est appelé " compression de données".

La compression de données s'est avérée indispensable lors de la transmission de l'information vue la largeur de débit exigée et les contraintes en matière de capacité imposée par le canal de transmission.

La représentation numérique des formes d'ondes exigée par les systèmes numériques nécessite la présence de l'opération de quantification. C'est ce qui est rapportée dans ce travail.

La quantification réalise la discrétisation des valeurs échantillonnées du signal parole ; elle convertit ainsi des variables réelles considérées comme un alphabet infini en variables discrètes à alphabet fini.

Depuis longtemps, les systèmes de compression de données utilisent la quantification scalaire (SQ). La SQ ne permet pas de descendre au-dessous d'un bit par échantillon. Ce qui exige un débit alloué relativement élevé.

Ces quantificateurs ne peuvent apporter des solutions concrètes quand le débit imposé est faible ; c'est là que la quantification vectorielle VQ se fait prévaloir. Cette dernière fait quantifier les paramètres du signal par vecteur (Traitement par bloc).

L'idée du TCM (Trellis Coded Modulation) d'Ungerboeck a permis aux chercheurs de mettre au point une nouvelle méthode de quantification appelée TCQ (Trellis Coded Quantization) qui améliore les performances de la SQ.

La quantification vectorielle codée par treillis TCVQ (Trellis Coded Vector Quantization) est une extension de la méthode TCQ au cas vectoriel. La TCVQ permet un débit de transmission fractionnaire. Cependant la complexité se trouve augmentée.

Dans le codage de la parole on cherche toujours à réduire le débit des informations transmises tout en gardant la qualité de ce signal ; pour cela il faut bien connaître ce dernier. Par conséquent, on a obligé de traiter le signal parole et ses principales caractéristiques fréquentielles et temporelles. Donc, on s'intéresse de définir la prédiction linéaire qui est l'approche la plus populaire employée pour la description des coefficients du filtre LPC.

Il est connu, dans le contexte de la compression de la parole que les coefficients de prédiction sont peu appropriés à la quantification à cause de leur large gamme dynamique et aux possibilités d'instabilités du filtre d'analyse LPC. Pour palier ce problème, une nouvelle représentation de ces paramètres a été introduite, ce sont les paramètres LSF (Line Spectral Frequencies).

Le déroulement de notre travail se fait comme suit :

Dans le premier chapitre, on a donné une étude générale sur la théorie d'information les systèmes de communications et le signal vocal. Le modèle classique de la production de la parole y est présenté ainsi que la méthode de la prédiction linéaire permettant d'obtenir l'ensemble des coefficients représentant le filtre de synthèse.

Dans le deuxième chapitre, on décrira la quantification vectorielle (VQ) et on abordera les notions de distorsion, de quantificateurs optimaux et les dictionnaires initiaux... etc.

La méthode de la quantification vectorielle codée par treillis (TCVQ) sera introduite dans le troisième chapitre. Ce chapitre est réservé aussi pour donner les détails sur la quantification codée par treillis (TCQ) qui est un cas particulier de la TCVQ quand la dimension des vecteurs k est égal à 1. On représentera des exemples de quantification et le déroulement du treillis, ainsi que la méthode de codage.

Le quatrième chapitre donne des notions générales sur les paramètres LSF, leurs caractéristiques, leurs intérêts, leurs propriétés ainsi que la méthode employée pour leurs extractions.

Le dernier chapitre est consacré à la quantification des paramètres LSF en utilisant la méthode de quantification vectorielle codée par treillis (TCVQ). Les simulations et les résultats sont aussi décrites.

Chapitre I

GENERALITES

I.1. La théorie de distorsion :

La théorie de distorsion est une discipline de la théorie de l'information qui étudie le codage de canal et de source de séquences stochastique. Parmi les différents types de sources, seules seront considérées les séquences gaussiennes sans mémoire et à moyenne nulle.

La théorie de la distorsion occupe une place importante dans la théorie de l'information [1]. C'est l'étude théorique sur la meilleure façon de représenter la séquence aléatoire de la source qui a pris la dénomination de la théorie de la distorsion (rate distorsion theory). On peut dire que c'est Shannon (1948) qui a lancé les principes de base de cette théorie.

Le but essentiel de cette partie est de montrer que la fonction de la distorsion est l'expression mathématique de la meilleure performance que peut atteindre un quantificateur.

Cette partie introduit également les différents développements des idées et questions afférentes à la quantification.

I.1.1. Entropie d'une source :

Considérons une source stationnaire, ergodique, sans mémoire X [1,10]. Chaque échantillon temporel $x(i)$ est indépendant des échantillons $x(i-1)$, $x(i-2)$, ..., $x(N-1)$ et qu'il ne peut prendre que $N-1$ différentes dans l'ensemble $\{a_0, a_1, \dots, a_{N-1}\}$.

Chaque échantillon est associé à une probabilité $p(x(i)=a(i)) = P_x(a_i)$. On définit ainsi les probabilités : $P(a_0)$, $P(a_1)$, ..., $P(a_{N-1})$.

L'entropie de X est définie comme suit :

$$H(X) = -E[\log_2 P(X)] = -\sum_{i=0}^{N-1} P_x(a_i) \log_2 P_x(a_i) \quad (I.1)$$

La grandeur $I(x) = -\log_2 P(X)$ est appelée l'auto-information ou la « surprise », plus un événement est rare ($P(X)$ est petit), plus la surprise qui lui associée est grande, chaque lettre de la source a donc une surprise qui lui propre. La surprise moyenne de la source n'est autre que l'entropie de la source :

$$H(X) = E[I(X)]. \quad (I.2)$$

$H(X)$ ainsi définie s'exprime en bits, elle mesure l'information moyenne d'une source. dans notre cas nous pouvons écrire :

$$0 < H(X) < \log_2 N$$

de cette double inégalité, on peut souligner les remarques suivantes :

- ◆ Lorsque $H(X)$ est nulle, cela se traduit par une seule probabilité non nulle d'un élément de l'alphabet. Dans ce cas la source est totalement prédictible.
- ◆ On obtient une probabilité uniforme $P_x = 1/N$ dans le cas où $H(X) = \log_2 N$, la source serait alors totalement non prédictible.
- ◆ L'entropie d'une source est toujours positive.

Si nous considérons deux variables aléatoires x et y associées aux ensembles de valeurs $\{a_0, a_1, \dots, a_{N-1}\}$; $\{b_0, b_1, \dots, b_{M-1}\}$, avec une distribution de probabilité conjointe donnée, on peut définir l'entropie conditionnelle déduite de cette notion de densité de probabilité conditionnelle, on obtient la formule suivante :

$$H(x/y) = -E[\log_2 P_{x/y}(x/y)] = -\sum_{n=0}^{N-1} \sum_{m=0}^{M-1} P_{x/y}(a_n, b_m) \log_2 P_{x/y}(a_n / b_m) \quad (I.3)$$

Cette relation exprime l'incertitude moyenne qui reste à propos de X après la connaissance de la valeur de Y , ou de fait, l'information d'une source connaissant une autre source.

L'information mutuelle s'écrit :

$$I(X; Y) = H(X) - H(X/Y) \geq 0 \quad (I.4)$$

Ainsi l'information mutuelle peut être considérée comme la quantité d'incertitudes soustraites de x ou la quantité d'information apprise sur X en observant les valeurs de Y .

La notion d'entropie d'une source discrète telle que définie par les relations précédentes peut être étendue à des variables vectorielles en considérant les probabilités d'apparition de lettres de l'alphabet comme des probabilités conjointes des composantes des vecteurs.

I.1.2. La notion de distorsion :

considérant le schéma d'un quantificateur suivant :

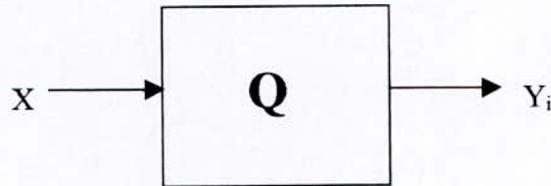


Fig I.1. quantificateur type, X est la grandeur d'entrée à quantifier et Y_i est la valeur de sortie.

En plus de la source déjà considérée plus haut, soient :

- Une des réalisations indépendantes de X composée de k échantillons :

$$X = (x(0), x(1), \dots, x(k-1)).$$

- Une autre séquence ayant ses réalisations dans l'ensemble $\{b_0, b_1, \dots, b_{M-1}\}$,

$$Y = (y(0), y(1), \dots, y(k-1)).$$

- $\rho_k(x, y)$ La mesure de la distorsion, le critère de fidélité ou encore le coût dû à la reproduction de la séquence X par le vecteur Y.

Notre quantificateur Q peut être considéré comme un canal de transmission. il va associer à une lettre de l'alphabet de X une lettre de Y. il consiste donc une matrice de taille $M \times N$ avec en pratique $M \leq N$.

C'est pour cette raison que l'on représente le modèle du quantificateur comme un modèle de probabilité conditionnelle de Y pour X donnée avec la contrainte de critère de fidélité.

On pose l'hypothèse que l'expression de ce critère peut se mettre de la forme :

$$\rho_k(x, y) = \frac{1}{k} \sum_{i=0}^{k-1} \rho_k(x(i), y(i)) \quad (I.5)$$

Où ρ est une mesure de la distorsion non négative entre les composantes correspondantes des séquences X et Y que l'on appelle souvent distorsion à lettre unique ou sans mémoire.

Les fonctions de probabilité et de probabilité conjointe de transition P_x et $P_{x,y}$ supposées connues, on peut obtenir la distorsion moyenne avec l'expression :

$$d(P_{Y/X}) = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} P_X(a_n) P_{Y/X}(b_m/a_n) \rho(a_n, b_m) \quad (I.6)$$

L'expression de l'information mutuelle s'écrit :

$$I(P_{Y/X}) = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} P_X(a_n) P_{Y/X}(b_m/a_n) \log_2 \frac{P_{Y/X}(b_m/a_n)}{P_Y(b_m)} \quad (I.7)$$

$$\text{Où : } P_Y(b_m) = \sum_{n=0}^{N-1} P_X(a_n) P_{Y/X}(b_m/a_n) \quad (I.8)$$

Pour une distorsion donnée D , on définit la fonction de taux de distorsion $R(D)$, par la relation :

$$R(D) = \min_{P_{Y/X} \in P_D} I(P_{Y/X}) \quad (I.9)$$

$$\text{Avec : } P_D = \{P_{Y/X} : d(P_{Y/X}) \leq D\}$$

Cette équation définit une fonction $R(D)$ qui donne pour une distorsion donnée, un débit minimal R ou réciproquement pour un débit donné, la plus faible distorsion possible D .

Ce qui nous ramène à parler du codage d'une source ou d'un vecteur.

On appelle un code C de taille N et de longueur de bloc k un ensemble de N séquences $\{y_0, y_1, \dots, y_{N-1}\}$ de dimension k chacune.

Chaque vecteur de C est appelé mot de code qui est une séquence binaire de longueur $\log_2 N$ au moins. Pour coder un vecteur X dans C , il faudra choisir le mot de code qui minimise la distorsion, notée :

$$\rho_k(x/c) = \min_{y \in c} \rho_k(x, y) \quad (I.10)$$

Pour un vecteur X de k échantillons, on définit le débit par échantillon du code C par la relation :

$$R(\text{bits}) = (\log_2 N)/k \quad (I.11)$$

La théorie sur le codage nous apprend que $R(D)$ est une limite théorique des performances des systèmes de codage, et qu'il est possible d'atteindre une performance assez proche de cette limite.

Ce qui est donné par l'équation :

$$\lim_{k \rightarrow \infty} \frac{1}{k} \log_2 N = R(D) \quad (I.12)$$

Cependant il nous semble utile de noter certaines remarques :

- L'alphabet de reproduction des sources continues est infini.
- L'entropie différentielle des sources continues peut être positive ou négative ; elle est une mesure relative au repère alors que dans le cas discret elle est absolue. Ainsi on définit l'entropie différentielle par la relation :

$$h(X) = -E[\log_2 f_X(X)] = -\int f_X(X) \log_2 f_X(X) dX \tag{I.13}$$

Où $f_X(X)$ est la densité de probabilité.

L'entropie conditionnelle est définie comme suit :

$$h(X/Y) = -\iint f_{XY}(x,y) \log_2 f_{X/Y}(X/Y) dx dy \tag{I.14}$$

Et l'information mutuelle par :

$$I(X, Y) = \iint f_{XY}(x,y) \log_2 \frac{f_{XY}(x,y)}{f_X(x)f_Y(y)} dx dy \tag{I.15}$$

A partir des relations précédentes, si l'on considère que $\rho(x,y)$ est une mesure de distorsion, et $f_{Y/X}(y/x)$ la densité de probabilité conditionnelle, on définit la distorsion moyenne suivant :

$$d(f_{Y/X}) = \iint f_X(x) f_{Y/X}(y/x) \rho(x,y) dx dy \tag{I.16}$$

Et l'information mutuelle :

$$I(f_{Y/X}) = \iint f_X(x) f_{Y/X}(y/x) \log_2 \frac{f_{XY}(x,y)}{f_X(x)f_Y(y)} dx dy \tag{I.17}$$

$$\text{Où : } f_Y(y) = \int f_X(x) f_{Y/X}(y/x) dx \tag{I.18}$$

On obtient finalement comme pour le cas discret, la limite théorique des performances ou la fonction $R(D)$:

$$R(D) = \min_{f_{Y/X} \in f_D} I(f_{Y/X}) \tag{19}$$

$$\text{Où : } f_D = \{f_{Y/X} : d(f_{Y/X}) \leq D\}$$

Les particularités de chaque type de cas créent une légère différence à l'origine de la fonction $R(D)$ entre les représentations graphiques de cette fonction pour les cas des sources continues et discrètes (Fig I.2).

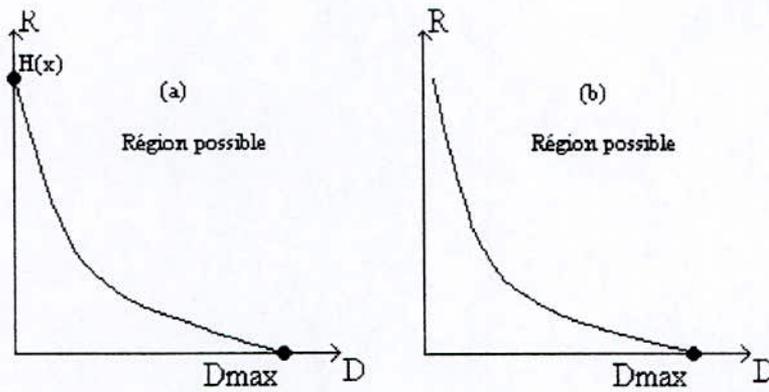


Fig I.2. comparaison de la fonction $R(D)$ des sources discrètes (a) et continue (b).

On remarquera certaines propriétés de la fonction $R(D)$:

- ✓ $R(D)$ est une fonction convexe sur l'intervalle $]0, D_{\max}[$.
- ✓ D_{\max} est la distorsion maximale, elle existe toujours.
- ✓ La valeur extrémal D_{\max} est obtenue comme suit :

Pour le cas discret :

$$D_{\max} = \min_{y \in C} \sum_x P(x) \rho_k(x, y) \tag{I.20}$$

Et pour le cas continu :

$$D_{\max} = \min_y \int f_X(x) \rho(x, y) dx \tag{I.21}$$

- ✓ $R(D)$ est continue, monotone, et décroissante.

La méthode utilisée pour la mesure de la distorsion ou le choix du critère de fidélité reste primordial dans la conception des quantificateurs. Celle de l'erreur quadratique moyenne demeure incontestablement la plus utilisée malgré quelques cas où elle s'avère inadéquate. Appelée encore MSE « mean squared error », elle permet d'exprimer pour une variable aléatoire gaussienne de variance σ_x^2 , la fonction de distorsion $R(D)$:

$$R(D) = \frac{1}{2} \max \left(0, \log_2 \frac{\sigma_x^2}{D} \right) = \begin{cases} \frac{1}{2} \log_2 \frac{\sigma_x^2}{D} & 0 \leq D \leq \sigma_x^2 \\ 0 & \text{ailleurs} \end{cases} \tag{I.22}$$

La limite inférieure de la fonction de distorsion est appelée la borne inférieure de Shannon (Shannon Lower Bound ou LSB). Cette caractéristique peut être obtenue d'une façon explicite dans bien des cas ce qui ne se réalise pas pour la fonction $R(D)$.

La borne inférieure de Shannon s'identifie à cette dernière pour des sources gaussiennes lorsque le MSE est utilisé comme estimateur. [1]

I.2. Système de communication :

L'objectif principal des systèmes de communication est la transmission du signal avec un minimum de distorsion, plus de sûreté, une bonne protection et une occupation minimum de l'espace. Dans la littérature les termes codage de source, codage de canal, compression de données et compression des signaux sont tous utilisés pour désigner les techniques utilisées pour achever une représentation numérique compacte du signal. La figure(I.3) représente un schéma bloc de codage numérique [5]. On notera que le but final est le récepteur humain.

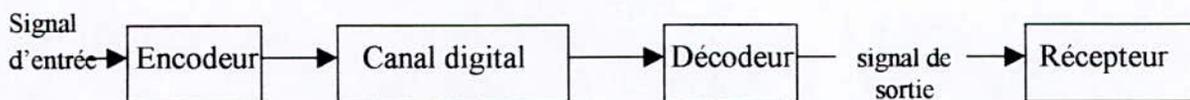


Fig I.3. Système de codage pour compression de signaux.

I.2.1. Chaîne de communication :

La figure (I.4) décrit les différents blocs d'une chaîne de communication numérique.

- ◆ Le codeur source essaie de minimiser le débit binaire nécessaire pour représenter fidèlement le signal d'entrée.
- ◆ Les blocs modulateur et démodulateur servent à maximiser le débit binaire que peut supporter un canal donné sans causer un niveau inacceptable de probabilité d'erreur binaire.
- ◆ Le bloc de codage canal ajoute des redondances au flux binaire pour la protection contre les erreurs.

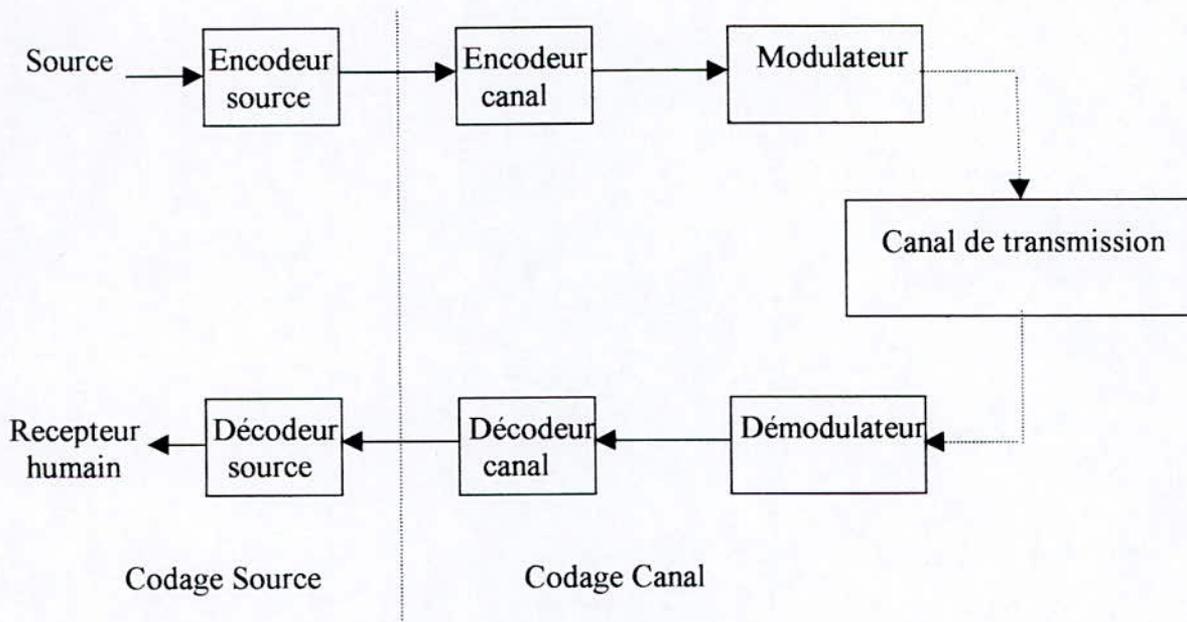


Fig I.4. Schéma bloc d'un système de communication numérique.

La capacité de compression des signaux a été une majeure préoccupation de la technologie de la communication longue distance, stockage de haute qualité et de la cryptage de message.

I.2.2. Critère de performance dans le codage de la parole :

Le problème essentiel dans la compression du signal est de minimiser le débit binaire dans la représentation numérique du signal tout en maintenant des niveaux adéquats de qualité du signal, de complexité d'implantation et de retard de communication.

1.2.2.1. Qualité du signal :

La qualité du signal perçu est souvent évaluée sur une échelle de 5 points qui est connue comme étant l'échelle MOS (Mean Opinion Score) dans les tests de la qualité de la parole : Une moyenne à travers un grand nombre d'entrée parole, locuteurs et testeurs d'écoute évaluant la qualité du signal. Les cinq points de la qualité sont associés à un ensemble d'adjectifs de description : mauvais, médiocre, inacceptable, bon, excellent. On attribue ainsi un seul niveau à chaque signal parole à évaluer durant la procédure d'évaluation subjective.

1.2.2.2. Débit binaire :

On mesure le débit binaire d'une représentation digitale en bits par échantillon ou bits par seconde selon le contexte. Le débit en bits par seconde n'est que le produit de la fréquence d'échantillonnage et le nombre de bits par échantillon. La fréquence d'échantillonnage doit être au moins deux fois plus grande que la largeur de bande du signal

correspondant. Dans le cas de la téléphonie, pour une bande de 3.2 KHz, la fréquence d'échantillonnage de 8 KHz est utilisée.

1.2.2.3.Complexité :

La complexité d'un algorithme de codage est l'effort de calcul exigé pour implanter les processus de l'encodage et du décodage dans les cartes de traitement du signal (hardware), mesuré en terme de la capacité arithmétique et l'espace mémoire utilisé. D'autres mesures de complexité peuvent être signalées telles que la taille physique de l'encodeur ou du décodeur, le prix et la consommation de puissance ce dernier étant un important critère dans un système portable.

1.2.2.4.Retard de communication :

La complexité dans un algorithme de codage est souvent accompagnée d'une augmentation de la durée de traitement dans l'encodeur et le décodeur. Bien que l'évolution de capacités des processeurs de traitement du signal est un facteur en faveur d'utilisation d'algorithme plus sophistiqué, le besoin de limiter le retard de communication ne doit pas être d'une importance moindre. Ce besoin impose des restrictions pratiques importantes dans l'utilisation des algorithmes. Selon l'environnement de communication, le retard total permis à un sens peut être aussi bas qu'une milliseconde.

Le retard de codage à un seul sens est défini comme étant le temps écoulé entre l'instant où l'échantillon du signal de parole arrive à l'entrée de l'encodeur et l'instant où le même échantillon apparaît à la sortie du décodeur, moins tout retard introduit par les autres équipements de communication entre la paire encodeur-décodeur et le retard de propagation du signal qui dépend de la distance. En d'autres termes, c'est comme si l'encodeur et le décodeur sont directement connectés par fils sans aucun équipement entre eux. Cette définition fait que le retard de codage dépend seulement de l'algorithme de codage.

Le retard de codage consiste en trois catégories de retard :

- ◆ Retard algorithme de bufferisation.
- ◆ Retard de traitement.
- ◆ Retard de transmission binaire.

Le premier retard est dû à l'analyse LPC (Linear Prediction Cading) adaptative progressive, et on peut réduire le retard de traitement en utilisant des processeurs plus rapides.

I.2.3. Mesure de la qualité :

Les mesures objectives de la qualité de la parole sont purement des mesures mathématiques évaluées en utilisant des distances euclidiennes et les mesures subjectives de la qualité évaluent la qualité de codage par des tests d'écoute.

La mesure objective de la qualité la plus couramment utilisée, pour les codeurs qui essaient de préserver la forme du signal, reste le rapport signal sur bruit (RSB).

Si s ; s est le signal de parole original. \tilde{s} est le signal de parole synthétisé.
alors le signal d'erreur est donné par :

$$e(n) = s(n) - \tilde{s}(n) \quad (I.23)$$

pour un signal de N échantillons, on définit l'énergie du signal par :

$$E_s = \sum_{n=0}^N s^2(n) \quad (I.24)$$

et l'énergie de l'erreur :

$$E_e = \sum_{n=0}^N e^2(n) \quad (I.25)$$

le RSB est alors donné par :

$$\text{RSB(dB)} = 10 \log \frac{E_s}{E_e} \quad (I.26)$$

Le signal parole est par nature non constant. Certains segments du signal peuvent avoir une énergie plus ou moins grande. En supposant que l'énergie de l'erreur soit à peu près constante, le RSB pourra être très important comme très faible.

I.3. Les redondances dans le signal parole :

On va essayer de déterminer la cadence maximale à laquelle un auditeur peut assimiler un message [5].

Pour cela, définissons d'abord l'information associée à un message constitué par des éléments discrets x_i appartenant à un ensemble donné X .

Soit P_k la probabilité à priori d'occurrence du symbole x_k : $P_k = P_r\{X(n) = x_k\}$.

Exemple :

Soit deux codages binaires d'une source de quatre messages (tableau I.1) :

Message	Probabilité	Code C ₁	Code C ₂
M ₁	0.5	00	0
M ₂	0.25	01	10
M ₃	0.125	10	110
M ₄	0.125	11	111

Tableau I.1 : exemple d'un codage binaire.

L'entropie de la source est :

$$H(x) = 0.5 \log_2 2 + 0.25 \log_2 4 + 0.125 \log_2 8 + 0.125 \log_2 8 = 1.75 \text{ bits}$$

Dans le code C₁ (n = 2) :

$$\text{L'efficacité vaut : } \eta = \frac{H}{n} = 0.875$$

$$\text{La redondance vaut : } \rho = 1 - \eta = 0.125$$

L'efficacité de 87.5% (ou la redondance de 12.5%) provient du fait qu'on avantage de la même manière des messages de fréquences différentes.

Le code C₂, au contraire, affecte les mots les plus courts aux messages les plus fréquents.

$$\text{On a : } n = 1 \times 0.5 + 2 \times 0.25 + 3 \times 0.125 + 3 \times 0.125 = 1.75.$$

$$\text{Donc : } \eta = 1 \quad \text{et} \quad \rho = 0$$

Le code est efficace à 100%. Sa redondance est nulle.

Dans la conversion courante, environ dix phonèmes sont prononcés par seconde ; l'information moyenne est donc inférieure à 50 bits/s.

De l'autre côté, pour garder une haute qualité de la parole avec une représentation numérique du signal de parole, l'utilisation d'un système de conversion A/N réclame plus de 100 000 bits/s. Il y a donc apparemment une redondance énorme dans le signal de parole.

La suppression partielle des redondances permet une représentation plus efficace des données. La compression de données peut se faire sans pertes d'information ou avec

perles en exploitant dans ce cas la tolérance de l'organe récepteur (e.g. l'oreille). La compression du signal consistera à réduire les redondances du signal de parole. Ces dernières sont essentiellement dues :

- Au manque de platitude du spectre court-terme.
- A la quazi-périodicité des signaux voisés.
- A la limitation des formes et des vitesses de mouvement possibles du conduit vocal.
- Aux distributions non uniformes des valeurs des paramètres de transmission.

Les trois premières sont dues à des propriétés physiques du mécanisme de production de la parole. La dernière est due à la fonction du codage utilisée.

Le manque de platitude du spectre court-terme est lié au fait que les échantillons de parole adjacents sont corrélés entre eux. On peut décorréler ces échantillons par un filtrage spectral adapté. La quazi-périodicité des signaux paroles voisés peut être supprimée en utilisant un prédicteur long-terme. La lenteur du conduit vocal permet d'envoyer les paramètres des filtres toutes les 10-30 ms. La dernière des redondances citées peut être exploitée par un codage approprié.

I.4. le signal vocal :

I.4.1. Caractéristiques psycho-acoustiques :

L'oreille est un récepteur complexe, nous nous contenterons d'énumérer quelques grandeurs caractéristiques[5,37].

Le seuil d'audition de l'oreille est non linéaire par rapport aux fréquences. L'oreille atteint sa sensibilité maximale entre 3 et 4 KHz.

L'oreille est extrêmement sensible à la période du signal de parole. Elle est capable de déceler des variations de la fréquence du fondamental de 0.5 à 1% d'où le problème de la mesure précise de cette fréquence. La perception est peu sensible à la phase du signal.

Seuil d'audition masqué : il est bien connu que les sons faibles cessent d'être entendus en présence de son fort. Tel est l'effet de masque. On distingue en fait le masquage temporel du masquage fréquentiel.

Le masquage fréquentiel : peut apparaître lorsqu'on entend en même temps deux sons purs de fréquences différentes. Il arrive que l'un d'entre eux, le son masqué, devienne inaudible. Cet effet de masque, qui peut être partiel ou total, dépend des intensités et des fréquences relatives des deux sons.

Le masquage temporel : apparaît lorsque deux sons ne sont pas présentés en même temps mais sont séparés par un bref silence. Ce masquage est la conséquence du fait que la forme temporelle des excitations auditives produites est supérieure à la durée physique des signaux.

I.4.2. Production du Signal Vocal :

Le signal vocal est produit par l'appareil phonatoire (fig.I.5), qui se compose essentiellement de trois étages :

- *Etage de respiration* : Les poumons insufflent l'air. Ils se comportent comme un générateur d'air qui alimente le larynx.
- *Les cordes vocales*, qui servent de vibrateurs. Ainsi le débit d'air fourni par les poumons est modulé par les cordes vocales ; celles-ci vibrent à des fréquences données appelées PITCH. Elles fournissent ainsi le signal glottal.
- *Le conduit vocal* de longueur 17 Cm environ est composé d'un ensemble de cavités.

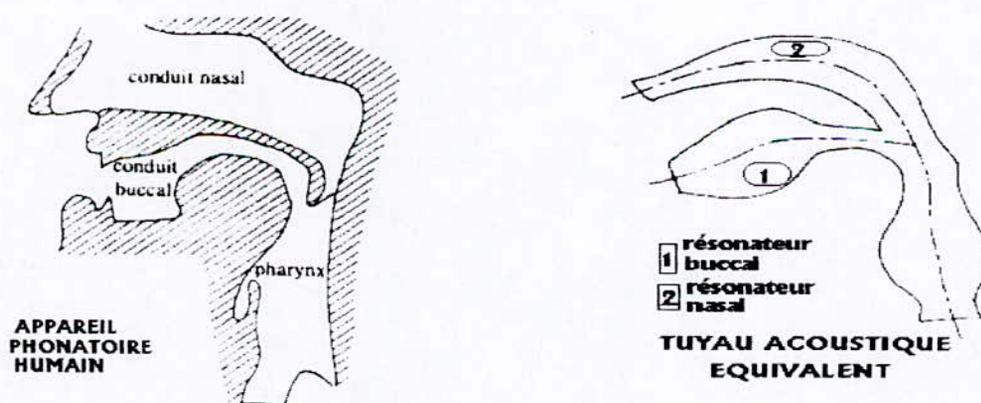


Fig I.5. Le conduit vocal et le tuyau acoustique équivalent.

En ce qui concerne le modèle de simulation, on considère que la parole résulte de l'excitation du conduit vocal par deux types de sources :

Les sons voisés comme les voyelles, qui résultent d'une vibration périodique des cordes vocales ; des impulsions périodiques de pression sont ainsi appliquées au conduit vocal. Ce dernier est un ensemble de cavités (cavité pharyngienne, cavité buccale et cavité nasale) situées entre la glotte et les lèvres.

Le conduit vocal peut être considéré comme une succession de tubes ou cavités acoustiques de sections différentes.

Un son est dit voisé lorsque les cordes vocales sont excitées. Elles s'ouvrent progressivement, sous l'action de la pression de l'air, et laisse apparaître une ouverture de forme triangulaire et d'étendue variable : la glotte. Elles se mettent alors à vibrer donnant naissance à l'onde glottique caractérisée par des variations impulsionnelles périodiques de la pression et du débit de l'air, les sons voisés sont générés par une onde glottique excitant les cavités supra glottiques (pharynx, cavité buccale et nasale). L'ouverture de la glotte changera en fonction des longueurs, épaisseurs et tensions des cordes vocales. Plus la tension des cordes vocales est élevée, plus la fréquence fondamentale sera élevée.

L'excitation du conduit vocal par des impulsions périodiques, se fait par une fréquence de récurrence appelée fréquence fondamentale (PITCH), qui peut varier de : [40]

80 à 200 Hz	pour les hommes.
150 à 450 Hz	pour les femmes.
200 à 600 Hz	pour les enfants.

Cette différence de fréquence est due à la longueur et la masse des cordes vocales. La figure (I.6) représente la forme d'onde et le spectre d'un son voisé. On y observe les raies qui correspondent aux harmoniques du pitch (F_0) ; l'enveloppe de ces raies présente des maximums appelés formants, qui correspondent aux fréquences propres F_i ($i=1,2,3, \dots$) dont les trois premiers sont essentiels pour caractériser le spectre vocal.

- Les sons non voisés, fricatifs ou plosifs, comme certaines consonnes sont engendrées par un signal aléatoire (généralement considéré comme un bruit blanc) en laissant volontairement les cordes vocales ouvertes. Pour les fricatives (f , s, ch), le débit d'air, forcé à travers un resserrement du conduit vocal, donnera naissance à un bruit. Un resserrement lié aux vibrations des cordes vocales générera des fricatives voisées (e, g, v). Un son plosif ou occlusif est produit par une occlusion momentanée du conduit vocal en un point donné suivi par une ouverture brusque. La figure (I.7) montre la forme d'onde et le spectre d'un son non voisé.

Les sons non voisés ne présentent pas de structure périodique. Ils peuvent être considérés comme un bruit blanc filtré par la transmittance de la partie des conduits vocaux et les lèvres ; Son spectre ne présente donc pas de structure pitch.

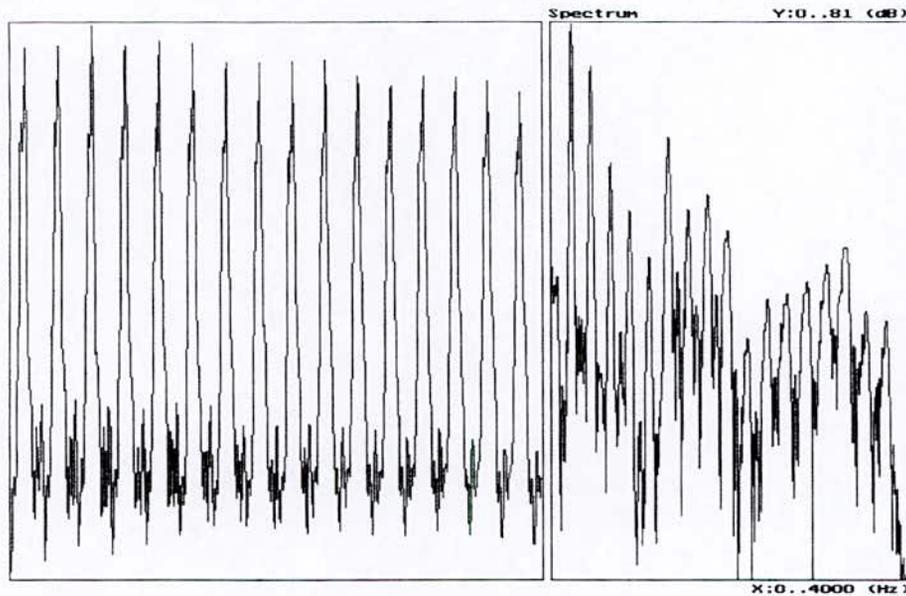


Fig.I.6. la forme d'onde et le spectre d'un son voisé

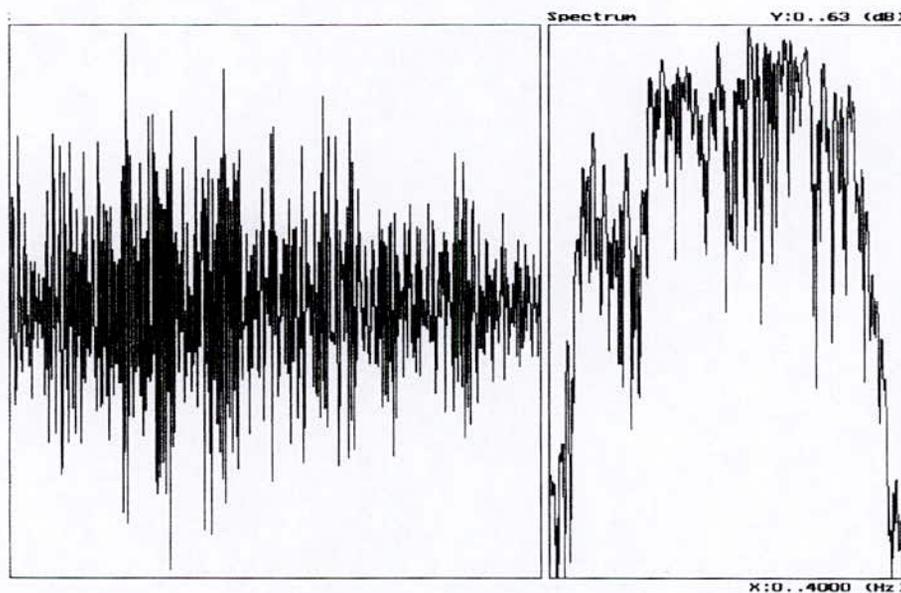


Fig.I.7. la forme d'onde et le spectre d'un son non voisé

Le conduit vocal est modélisé par un filtre tout pôle dont les paramètres varient dans le temps, et sa fonction de transfert est : [32]

$$H(z) = \frac{G}{A(z)} = \frac{G}{1 + \sum_{k=1}^p a_k \cdot z^{-k}} \tag{I.27}$$

Ce modèle de production du signal vocal est appelé autoregressif (AR) ; Il est clair que le modèle AR est une approximation plus au moins approchée de la production du signal vocal ; en particulier, pour les sons nasalisés où la transmittance du conduit vocal est plutôt modélisée par un modèle Auto Régressif à Moyenne Ajustée (ARMA) ; Il est dit d'ordre (p, q). Néanmoins, l'estimation d'un modèle ARMA est beaucoup plus délicate que celle d'un modèle AR ; On préfère souvent utiliser un modèle AR dont l'ordre p est un peu surestimé.

En effet à l'expression (I.27) correspond dans le domaine temporel l'équation suivante :

$$s(n) = - \sum_{k=1}^p a_k \cdot s(n-k) + G \cdot U(n) \tag{I.28}$$

Qui exprime qu'un échantillon s(n) est une combinaison linéaire des p échantillons qui le précèdent, avec un terme d'excitation. Où p est l'ordre du modèle et les coefficients a_k sont appelés paramètres du modèle, G représente le facteur de gain. La fonction H(Z) est appelée filtre de synthèse et A(Z) le filtre inverse ou filtre d'analyse. Les figures (I.8) et (I.9) montrent le modèle de production de la parole dans le domaine fréquentiel et temporel. Le rapport relatif des composantes voisées et non voisées est contrôlé dans le modèle par ajustement des gains correspondants. Pour un signal purement voisé le gain de la source de bruit est nul, et pour un signal purement non voisé le gain de la source de train d'impulsions est nul.

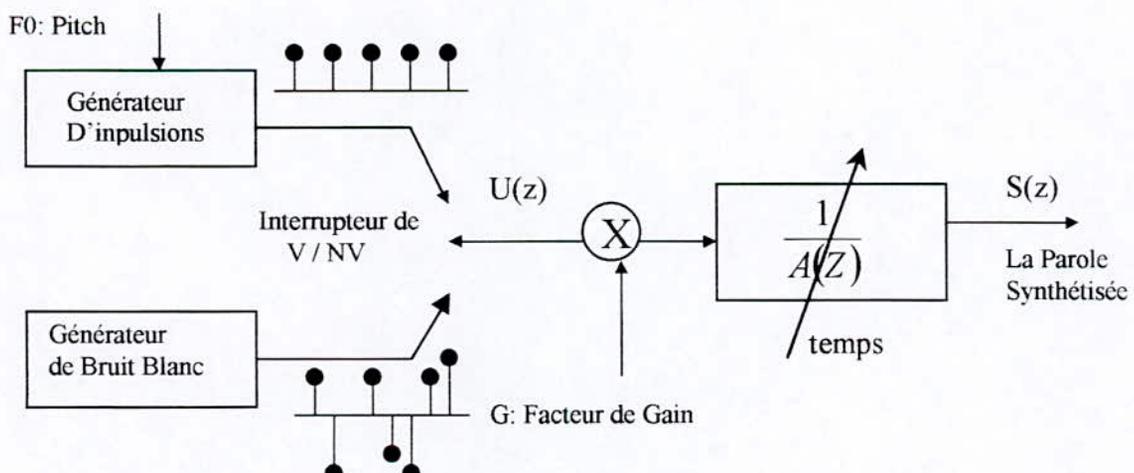


Fig.I.8. Modèle de production de la parole

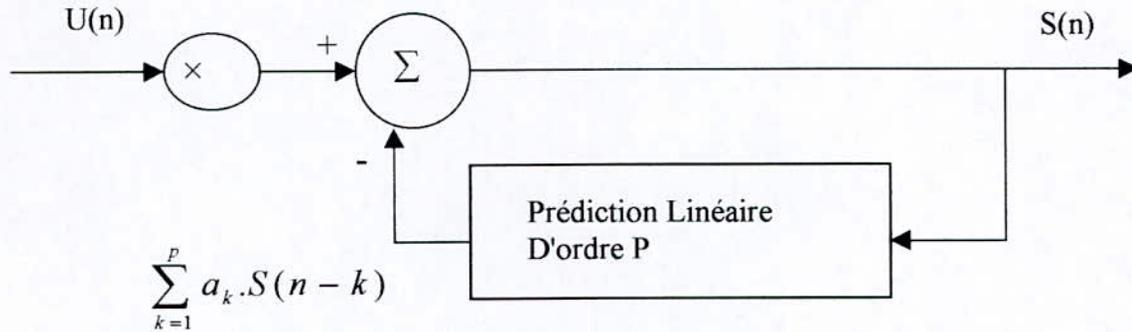


Fig.I.9. Modèle de production de la parole dans le domaine temporel

I.4.3. La Prédiction Linéaire (LP) :

La prédiction linéaire est l'une des méthodes les plus puissantes du traitement de la parole utilisée dans les systèmes de codage et de compression pour l'estimation des paramètres essentiels du signal de la parole. Son succès est dû au fait qu'elle représente une solution linéaire au problème de l'estimation des paramètres du modèle de la production de la parole [30,31,32].

La technique de prédiction linéaire est basée sur le modèle de la production de la parole, décrit dans le domaine fréquentiel par l'équation (I.27) et dans le domaine temporel par l'équation (I.28).

Le principe fondamental de la prédiction linéaire est qu'un échantillon $s(n)$ de parole peut être prédit à partir d'une combinaison linéaire des échantillons finis qui le précède. Ainsi, un échantillon $s(n)$ peut être approximé par une combinaison linéaire de P échantillons précédents ($p=8, \dots, 16$ échantillons) :

$$\tilde{s}(n) = - \sum_{k=1}^P \tilde{a}_k \cdot s(n-k) \quad (I.29)$$

Les coefficients \tilde{a}_k , ($k = 1, 2, \dots, p$) sont appelés paramètres LP ou coefficients de prédiction, et le nombre p des coefficients s'appelle ordre de prédiction. De plus $s(n)$ est l'échantillon de la parole et $\tilde{s}(n)$ est la sortie du prédicteur linéaire à l'instant d'échantillonnage n , qui a la fonction de transfert suivante :

$$P(z) = - \sum_{k=1}^P \tilde{a}_k \cdot Z^{-k} \quad (I.30)$$

Cette prédiction induit une erreur de prédiction $e(n)$ entre l'échantillon original $s(n)$ et l'échantillon prédit $\tilde{s}(n)$ et elle est définie par :

$$e(n) = s(n) - \tilde{s}(n) = s(n) + \sum_{k=1}^p \tilde{a}_k \cdot s(n-k) \quad (I.31)$$

Si on compare (I.28) et (I.31), on constate que lorsqu'on trouve un ensemble de coefficients \tilde{a}_k plus proche de l'ensemble a_k en minimisant l'erreur de prédiction ($\tilde{a}_k = a_k$), ($k = 1, 2, \dots, p$). On obtient alors l'égalité $e(n) = G.U(n)$. Ceci montre que l'erreur de prédiction $e(n)$ contient beaucoup d'informations sur l'excitation du signal de parole. La transformée en Z de l'équation (I.31) donne :

$$E(Z) = S(z) \cdot (1 + \sum_{k=1}^p \tilde{a}_k \cdot Z^{-k}) = S(Z) \cdot A(Z) \quad (I.32)$$

$A(z)$ est l'inverse de $H(z)$ dans (I.27), $A(z)$ est appelé filtre inverse ou filtre d'analyse, car on peut obtenir $e(n)$ à partir du signal parole en utilisant le filtre $A(z)$, suivant l'équation (I.32).

Le problème de l'analyse par prédiction linéaire se réduit donc à trouver un ensemble de coefficients \tilde{a}_k de façon à minimiser l'erreur de prédiction $e(n)$ dans un intervalle donné. L'approche de base est de calculer l'ensemble des coefficients a_k qui minimisent l'erreur quadratique moyenne de prédiction sur un court segment de la parole, cette erreur est définie par :

$$E = \sum_n e^2(n) = \sum_n \left[s(n) + \sum_{k=1}^p a_k \cdot s(n-k) \right]^2 \quad (I.33)$$

Les valeurs de a_k qui minimisent E sont obtenues en annulant les dérivées partielles, par rapport à chaque coefficient prédictif $a(i)$: $\frac{\partial E}{\partial a_i} = 0$ pour $i = 1, 2, \dots, p$ donc :

$$\frac{\partial E}{\partial a_i} = +2 \cdot \sum_n \left\{ \left[s(n) + \sum_{k=1}^p a_k \cdot s(n-k) \right] \cdot s(n-i) \right\} = 0 \quad (I.34)$$

Ce qui conduit au système :

$$-\sum_n s(n) \cdot s(n-i) = \sum_n \sum_{k=1}^p a_k \cdot s(n-k) \cdot s(n-i) \quad (I.35)$$

$$\text{Où : } -\sum_n s(n).s(n-i) = \sum_{k=1}^p a_k \cdot \sum_n s(n-k).s(n-i) \quad \text{pour } i = 1,2,\dots,p \quad (\text{I.36})$$

$$\text{Soit : } \phi(i,k) = \sum_n s(n-k).s(n-i) \quad \text{pour } i, k = 1,2,\dots,p \quad (\text{I.37})$$

Par conséquent l'équation (I.36) devient :

$$\sum_{k=1}^p a_k \cdot \phi(i,k) = -\phi(i,0) \quad \text{pour } i = 1,2,\dots,p \quad (\text{I.38})$$

Le problème de la technique LP se réduit donc à la résolution d'un ensemble de P équations avec P inconnus, afin d'obtenir les coefficients a_k : $k = 1,2,\dots,p$. Premièrement, on commence par le calcul des valeurs $\phi(i,k)$ pour $i = 1,2,\dots,p$ et $k = 0,1,\dots,p$ en utilisant l'équation (I.37) dont la limite de sommation doit être spécifiée. Deux grandes approches sont utilisées dans l'analyse LP, la méthode d'autocorrélation et la méthode de covariance.

I.4.4. La Méthode d'Autocorrélation :

La méthode d'autocorrélation garantit la stabilité du filtre LP et réduit l'espace mémoire.

Les hypothèses de cette méthode sont les suivants [2]:

- ♦ Le signal est défini pour toutes les valeurs du temps ; il est identiquement nul en dehors d'une séquence de N échantillons ; Ceci équivaut à multiplier le signal de parole par une fenêtre de longueur finie correspondant à N échantillons.

$$S_f(n) = \begin{cases} w(n)s(n) & 0 \leq n \leq N-1 \\ 0 & \text{ailleurs} \end{cases} \quad (\text{I.39})$$

La fonction de pondération la plus courante est la fenêtre de Hamming :

$$w(n) = \begin{cases} 0.45 - 0.46 \cos(2\pi n/(N-1)) & 0 \leq n \leq N-1 \\ 0 & \text{ailleurs} \end{cases} \quad (\text{I.40})$$

- ◆ Chaque échantillon peut être prédit approximativement à partir de p échantillons précédents. Ceci est valable pour toutes les valeurs du temps.

Donc l'énergie de l'erreur de prédiction (I.33) est calculée sur un intervalle infini, cependant pour respecter les considérations pratiques on suppose que $s(n) = 0$ en dehors de l'intervalle : $0 \leq n \leq N-1$, où N est la durée de la fenêtre d'analyse LP. Cette hypothèse sur la nature de $s(n)$ simplifiée le calcul du signal $\tilde{s}(n)$, donc :

$$\phi(i, k) = \sum_{n=0}^{N+p-1} s(n-i).s(n-k) \quad \text{pour } i = 1, \dots, p ; k = 0, \dots, p \quad (\text{I.41})$$

On voit que seuls les valeurs définies pour $0 \leq n \leq N + p - 1$, devront être calculés. Par un changement de variable $m = n - i$, l'équation (I.41) devient :

$$\phi(i, k) = \sum_{m=0}^{N-1-(i-k)} s(m).s(m+i-k) \quad (\text{I.42})$$

Donc, $\phi(i, k)$ est l'autocorrelations à court terme de $s(m)$ évaluée à $(i - k)$. Par conséquent :

$$\phi(i, k) = R(i - k) \quad (\text{I.43})$$

Où :

$$R(j) = \sum_{n=0}^{N-1-j} s(n).s(n+j) = \sum_{n=j}^{N-1} s(n).s(n-j) \quad \text{pour } j = 0, 1, \dots, p \quad (\text{I.44})$$

L'ensemble de p équations dans (I.38) devient :

$$\sum_{k=1}^p a_k \cdot R(|i - k|) = -R(i). \quad i = 1, 2, \dots, p \quad (\text{I.45})$$

L'équation (I.45) est utilisée pour évaluer les coefficients a_k du modèle ; elle peut être représentée sous forme matricielle comme suit :

$$\begin{pmatrix} R(0) & R(1) & R(2) & \dots & R(p-1) \\ R(1) & R(0) & R(1) & \dots & R(p-2) \\ R(2) & R(1) & R(0) & \dots & R(p-3) \\ \cdot & \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot \\ R(p-1) & R(p-2) & R(p-3) & \dots & R(0) \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ \cdot \\ \cdot \\ \cdot \\ a_p \end{pmatrix} = - \begin{pmatrix} R(1) \\ R(2) \\ R(3) \\ \cdot \\ \cdot \\ \cdot \\ R(p) \end{pmatrix} \tag{I.46}$$

On constate que la matrice des valeurs de l'autocorrélation est symétrique et que les éléments situés sur la diagonale sont identiques, cette forme matricielle est appelée forme de TOEPLITZ, cette propriété peut être exploitée pour obtenir un Algorithme efficace pour résoudre cette équation.

La procédure récursive de Wiener – Levinson - Durbin (WLD) représente la solution efficace pour déterminer les coefficients de prédiction minimisant la moyenne quadratique de l'erreur de prédiction. [voir l'Annexe B].

I.4.5. Méthode de covariance :

La méthode d'autocorrélation et la méthode de covariance diffèrent dans l'emplacement de la fenêtre d'analyse[2]. dans la méthode de covariance, le signal erreur est fenêtré au lieu du signal parole de façon que l'énergie à minimiser soit :

$$E = \sum_n e_f^2(n) = \sum_n e^2(n)w^2(n) \tag{I.47}$$

En annulant les dérivations partielles par rapport aux coefficients du filtre : $\frac{\partial E}{\partial a_i} = 0$ pour $i = 1, 2, \dots, p$, on a donc p équations linéaires.

$$\sum_{k=1}^p a_k \cdot \phi(i, k) = -\phi(i, 0) \quad \text{pour } i = 1, 2, \dots, p \tag{I.48}$$

où la fonction de covariance $\phi(i, k)$ est définie par :

$$\phi(i, k) = \sum_n w^2(n) s(n-i) \cdot s(n-k) \quad \text{pour } i = 1, \dots, p ; k = 0, \dots, p \tag{I.49}$$

sous forme matricielle, les p équations deviennent :

$$\begin{pmatrix} \phi(1,1) & \phi(1,2) & \dots & \dots & \phi(1,p) \\ \phi(2,1) & \phi(2,2) & \dots & \dots & \phi(2,p) \\ \cdot & \cdot & \dots & \dots & \cdot \\ \cdot & \cdot & \dots & \dots & \cdot \\ \cdot & \cdot & \dots & \dots & \cdot \\ \phi(p,1) & \phi(p,2) & \dots & \dots & \phi(p,p) \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ \cdot \\ \cdot \\ a_p \end{pmatrix} = \begin{pmatrix} \phi(1,0) \\ \phi(2,0) \\ \cdot \\ \cdot \\ \cdot \\ \phi(p,0) \end{pmatrix} \quad (I.50)$$

Cette matrice n'est pas une matrice Toeplitz, elle est symétrique et définie positive. La matrice de covariance peut être décomposée en matrices triangulaires supérieures et inférieures. Cholesky a proposé une décomposition pour la résolution de ce système.

Contrairement à la méthode d'autocorrélation, la méthode de covariance n'assure pas la stabilité du filtre.

Les Considérations de Choix des Conditions d'Analyse LP :

Les variables dans l'analyse LP sont :

- La méthode d'analyse :

Les deux méthodes d'autocorrélation et de covariance donnent presque les mêmes résultats. La méthode de covariance ne stipule aucune hypothèse sur la morphologie du signal $s(n)$, en dehors des N points disponibles. Ainsi, la minimisation de l'erreur quadratique présentée à l'équation (I.33) ne s'effectue que sur les N points de $s(n)$. Il faut noter que cette approche, contrairement à la technique d'autocorrélation, peut induire une solution instable du modèle. Par ailleurs, la méthode la plus utilisée est celle de l'autocorrélation.

- La durée de la trame d'analyse :

Pour assurer une bonne résolution spectrale, la durée N doit être l'intervalle où le mouvement du conduit vocal est négligeable. Généralement, cet intervalle est d'environ **15-20 ms** pour la majorité des voyelles, et plus réduit dans le cas des sons non voisés. On peut dire que cet intervalle est suffisant pour maintenir une bonne qualité de la parole, bien qu'il peut introduire une faible dégradation, surtout pour les sons transitoires, qui ont des changements rapides des caractéristiques spectrales.

La figure I.10 donne une représentation complète de la prédiction linéaire sous forme d'organigramme.

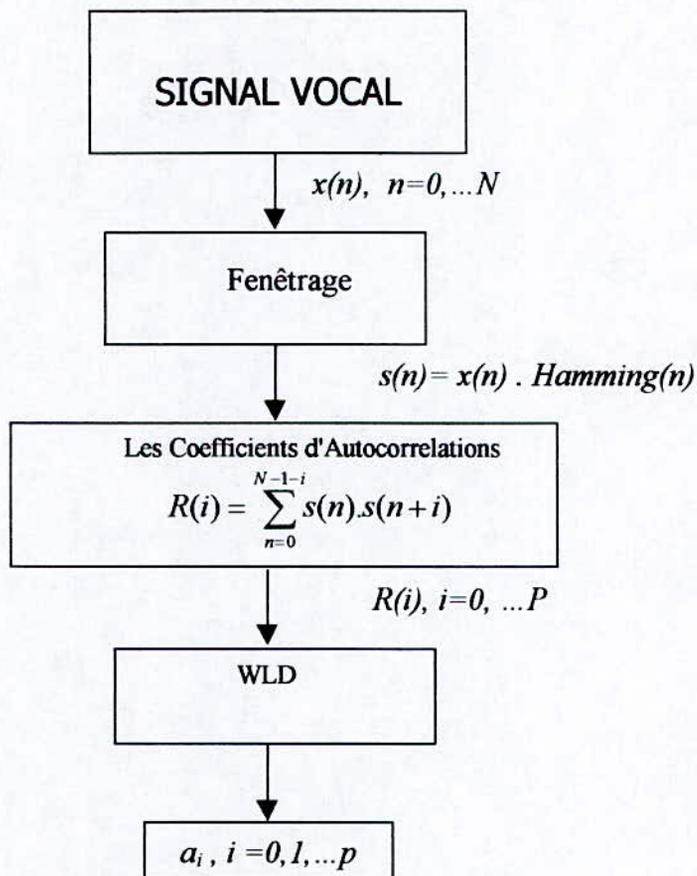


Fig. I.10. Organigramme de la prédiction linéaire (LP)

Chapitre II

LA QUANTIFICATION VECTORIELLE

II.1. préliminaires :

Pour être traités par les systèmes numériques, les échantillons successifs d'un signal doivent être transformés en signaux numériques par quantification. Mais avant de passer à la forme d'échantillons, les signaux doivent être traités par un échantillonneur [1,4,11].

L'échantillonnage d'un signal $x(t)$ consiste à remplacer ce signal par un autre signal $x_e(t)$ qui est égal en valeur instantanée à $x(t)$ pendant de brefs instants de durée, répétés périodiquement avec une fréquence f_e appelée fréquence d'échantillonnage, et nul entre ces brefs instants.

Le théorème d'échantillonnage dû à Shannon, s'énonce comme suit :

« Un signal primaire $x(t)$ qui ne contient que des composantes de fréquence inférieure à f_{max} (spectre borné) peut être entièrement déterminé par des échantillons équidistants prélevés avec une fréquence f_e tel que : $f_e \geq 2f_{max}$ ».

Cependant, les échantillons successifs de valeurs continues, doivent être approximatés par des valeurs discrètes. C'est l'objet de la quantification.

La quantification est l'opération de discrétisation d'une ou plusieurs variables. C'est aussi l'approximation de la valeur instantanée exacte d'un signal par la plus voisine valeur tirée d'un assortiment de N valeurs discrètes.

Si on désigne par x une variable aléatoire, un quantificateur est un appareil qui fait associer à une entrée x comprise dans un intervalle, une sortie y comprise dans le même intervalle. Donc la quantification est le processus de substitution des échantillons d'un signal analogique par des valeurs arrondies prises parmi un nombre fini de valeurs possibles.

La quantification peut être scalaire ou vectorielle selon que les variables x sont à une ou plusieurs dimensions.

II.1.1. Quantification scalaire SQ :

Dans la quantification scalaire, chaque échantillon du signal d'entrée est quantifié séparément des autres échantillons. La figure (II.1) illustre la caractéristique en marche d'escalier d'un quantificateur scalaire.

Le quantificateur fait correspondre à la variable d'entrée x la variable approchée y_i choisie dans l'ensemble de N valeurs prédéterminées.

Soit S l'ensemble ordonné des N valeurs arrondies disponibles : [1]

$$S = \{y_i / i = 0, 1, 2, \dots, N - 1\}$$

On peut alors écrire la fonction de quantification sous la forme :

$$y_i = Q(x) ; \text{ où } i \in [0, N - 1] . \quad (\text{II.1})$$

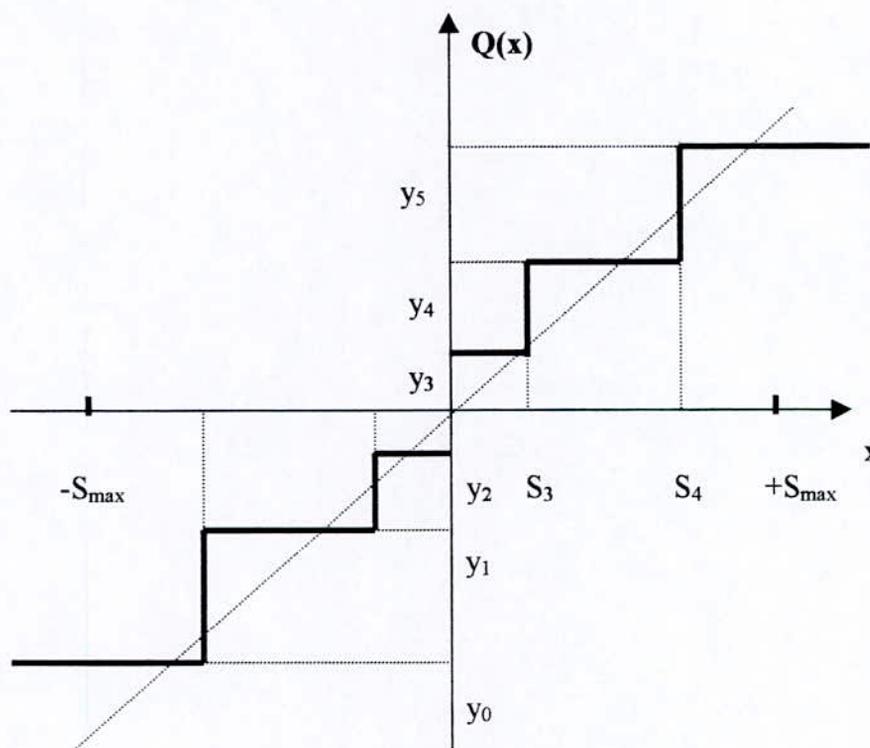


Fig II.1. Caractéristique typique d'un quantificateur scalaire de $N=6$.

Un quantificateur fournit une seule information de sortie mais celle-ci est disponible sous deux formes (y et i). La figure (II.2) donne quatre diagrammes blocs équivalents qui mettent en relief la distinction. [1]

Dans le premier cas (Fig II.2.a.), le quantificateur fournit sous forme intégrée, une même information, dans le second cas (Fig II.2.b) celle-ci est présentée sous deux formes distinctes. Le troisième schéma décrit par la figure (Fig II.2.c) illustre le point de vue adopté en

DPCM (Differential Pulse Code Modulation) où la valeur arrondie Y est réinjectée dans la boucle de rétroaction alors que l'indice i est directement transmis, enfin la quatrième alternative fournit tout d'abord l'indice i , tandis que la valeur arrondie s'obtient par une table de décodage.

Les chercheurs préfèrent inclure dans la quantification ainsi présentée le codage qui associe un indice à chacune des valeurs arrondies ; de là ils contournent le concept de seuil et font correspondre à l'indice i une table de décodage où sont contenues les valeurs arrondies de sortie Y_i ; ils arrivent ainsi à généraliser la notion de quantification à plus d'une variable et réduisent celle-ci à une recherche de l'indice entre 0 et $N-1$ lequel la meilleure approximation est obtenue.

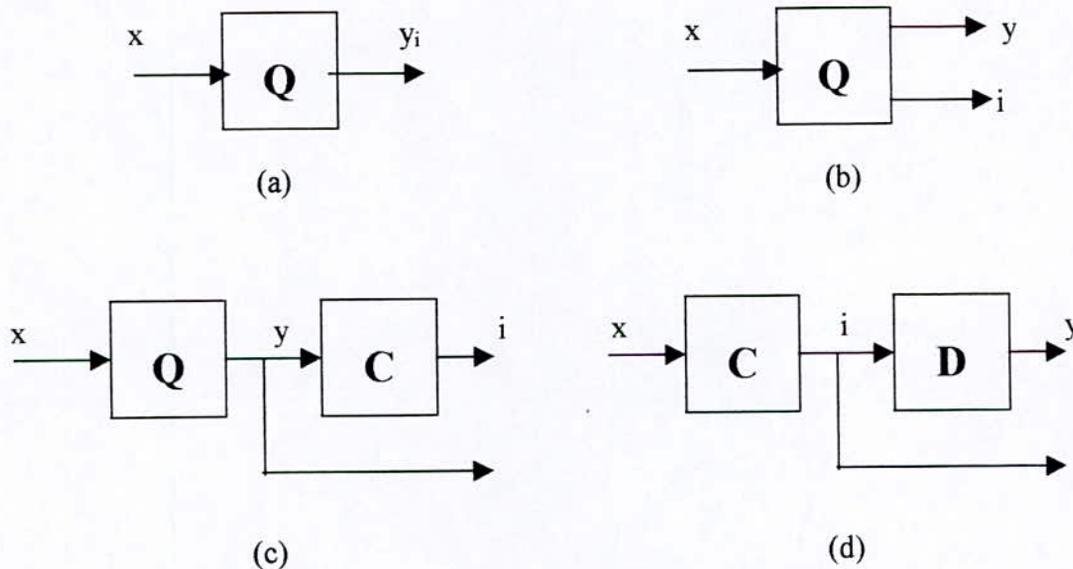


Fig. II.2. le Schéma type de quantificateurs

Cette manière de faire permet de considérer toutes les grandeurs, en particulier l'entrée x et les valeurs arrondies Y , sous forme vectorielle. La figure(II.3) illustre un exemple de quantification d'un vecteur x à deux dimensions. C'est le principe de la quantification que nous allons développer ci dessous.

II.2. Principe de la quantification vectorielle :

Dans le cas où la grandeur à quantifier est composée de plusieurs variables, on parlera de Quantification Vectorielle (VQ).

L'objectif principal de la quantification vectorielle est la compression de données. Cette opération rend l'utilisation des canaux de communication plus efficace grâce à la réduction de la bande de fréquence occupée par le signal. Elle réduit aussi la capacité des systèmes de stockage de données [1,2,33,34].

La quantification vectorielle s'avère d'un intérêt particulier pour la compression du signal de parole. Ce signal qui est de débit assez large sera ramené à un débit faible. Avant de passer à l'aspect distorsion et définition du quantificateur optimal, qu'en est-il de la signification exacte de la quantification vectorielle.

La quantification vectorielle est une opération qui fait correspondre à un vecteur d'entrée x ayant k composantes à valeurs réelles et continues, un vecteur arrondi y_i appartenant à un ensemble fini S tel que :

$$C = \{y_i \in \mathbb{R}^k / i=1,2,\dots,N\}. \tag{II.2}$$

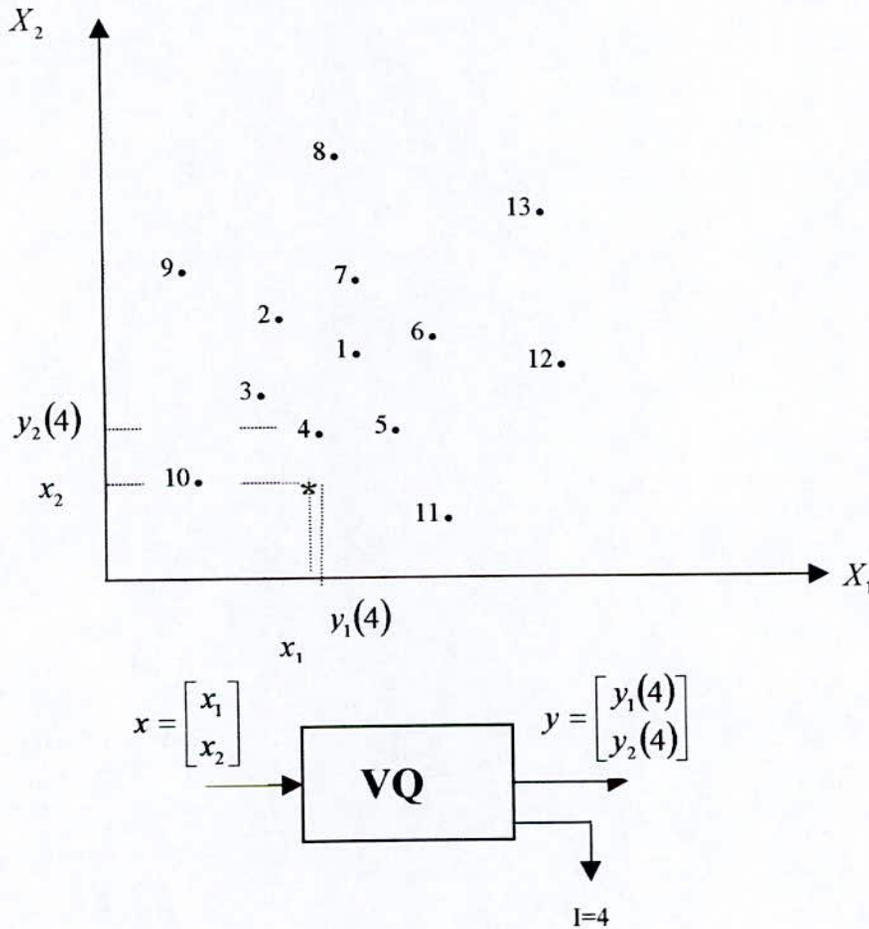


Fig.II.3. Principe de la quantification vectorielle. L'ensemble S des $N=13$ vecteurs arrondis du quantificateur vectoriel sont présentés sous forme de points dans l'espace des coordonnées du vecteur d'entrée x . une entrée x particulière est représentée dans le par une croix. Le vecteur arrondi le plus proche dans la table est le vecteur y pour $i=4$. On suppose ici la distance euclidienne est la distance à minimiser.

Donc le quantificateur vectoriel est l'application de l'espace euclidienne de dimension k (\mathbb{R}^k) à l'ensemble fini C , contenant N vecteurs codes de \mathbb{R}^k :

$$Q : \mathbb{R}^k \rightarrow C$$

Telle que :

$$C = \{y_0, \dots, y_{N-1}\} \text{ et } y_i \in \mathbb{R}^k \text{ pour chaque } i \in J = \{0, \dots, N-1\} \tag{II.3}$$

L'ensemble C est appelé Dictionnaire (Codebook) ayant N éléments chacun est un vecteur de \mathfrak{R}^k . En posant, Nous dirons que les vecteurs d'entrées sont quantifiés sur N niveaux et codés sur Rk bits. R s'appelle le débit par dimension ou débit par échantillon.

Nous appellerons distance entre x et y_i , généralement notée par $d(x, y_i)$, le degré de distorsion dû à l'approximation du vecteur d'entré x par le vecteur «arrondi » y_i . un quantificateur vectoriel est alors complètement défini par le dictionnaire C et la distance d. $d(x, y)$ est défini par l'application :

$$\begin{array}{ccc} & d & \\ \mathfrak{R}^k \times \mathfrak{R}^k & \longrightarrow & \mathfrak{R} \\ (x, y) & \longrightarrow & d(x, y) \end{array} \tag{II.4}$$

Doit avoir les propriétés suivantes :

- ◆ $d(x, y) \geq 0$
- ◆ $d(x, y) = 0$ si $x = y$
- ◆ $d(x, y) = d(y, x)$ (Symétrie)
- ◆ $d(x, z) \leq d(x, y) + d(y, z)$ (Inégalité triangulaire)

Dans le cas de la parole, la distance doit avoir deux propriétés supplémentaires :

- ◆ $d(x, y)$ doit avoir une interprétation physique .
- ◆ $d(x, y)$ doit être simple à calculer.

La mesure de la distorsion doit avoir une certaine signification dans le domaine spectrale selon les propriétés spectrale de la parole. les différences entre l'enveloppe spectrales du signal original et l'enveloppe spectrale du signal codé qui peuvent conduire à des sons phonétiquement différents sont les suivantes :

- ◆ Les formants de l'enveloppe spectrale du signal original et ceux de l'enveloppe spectrale du signal codé se produisent à des fréquences différentes.
- ◆ Les bandes de ses formants diffèrent significativement.

Exemples de distance :

$$x = (\alpha_1, \alpha_2, \dots, \alpha_n)$$

$$y = (\beta_1, \beta_2, \dots, \beta_n)$$

$$d(x, y) = \sum_{i=1}^n |\alpha_i - \beta_i| \quad \text{Distance de Minkowsky.}$$

$$d(x, y) = \left(\sum_{i=1}^n |\alpha_i - \beta_i|^2 \right)^{1/2} \quad \text{Distance Euclidienne.} \tag{II.5}$$

$$d(x, y) = \text{Max}_i |\alpha_i - \beta_i| \quad \text{Distance de Chebychev.}$$

D'autres mesures de distorsion spectrale peuvent être utilisées selon le contexte telles que : la mesure de distorsion spectrale logarithmique, la mesure d'ITAKURA SAITO, etc.

Par exemple, la distorsion d'ITAKURA SAITO équation mesure le rapport d'énergie entre le signal résiduel obtenu en utilisant le filtre LP avec coefficients quantifiés et le signal résiduel obtenu en utilisant le filtre LP avec les coefficients non quantifiés.

$$d_{IS} = \frac{1}{2\pi} \int_{-\pi}^{\pi} [e^{V(\omega)} - V(\omega) - 1] d\omega$$

$$\text{avec : } V(\omega) = \text{Log} \left(S(\omega) - \text{Log} \left(S(\hat{\omega}) \right) \right) \quad (\text{II.6})$$

$$S(\omega) = \frac{G}{|A(e^{i\omega})|} \quad G \text{ facteur gain du filtre LP}$$

en supposant que la grandeur d'entrée est un vecteur aléatoire distribué selon une loi, les performances du quantificateur peuvent être mesurées par la distorsion moyenne D_Q introduite, c'est à dire par l'espérance mathématique de la distance d :

$$D_Q = E[d(x, Q(x))] = \int d(x, Q(x)) * p(x) * dx \quad (\text{II.7})$$

Dans la pratique, la distribution des points d'entrée étant généralement inconnue, on approximera D_Q par une distorsion moyenne calculée sur un large nombre d'échantillons $\{x_1, x_2, \dots, x_N\}$ de vecteurs d'entrée. l'ergodicité et la stationnarité nous permettent d'écrire :

$$D_Q \cong \frac{1}{N} \sum_{j=1}^N d(x_j, Q(x_j)) \quad (\text{II.8})$$

la distance introduit implicitement une partition de l'ensemble des vecteurs d'entrée en N classes, la classe S^i étant l'ensemble des vecteurs associés à y_i par le quantificateur :

$$S^i = Q^{-1}(y_i) = \{x; Q(x) = y_i\} \quad (\text{II.9})$$

nous appellerons centroïde de la classe S^i le vecteur c^i tel que sa distance moyenne à tous les éléments de la classe soit minimale (en géométrie euclidienne, le centroïde est le centre de gravité) :

$$E[d(x, c^i), x \in S^i] = \text{Inf}_x \{E[d(x, x^i), x \in S^i]\} \quad (\text{II.10})$$

Etant donné une distance et une taille de dictionnaire, on cherche un quantificateur optimal qui minimise la distorsion moyenne ou qui se rapproche de l'optimalité.

II.3. Conditions d'optimalité :

Pour une distribution statistique donnée de la source et un débit fixé : [5]

- ◆ Le quantificateur globalement optimal est celui qui minimise la distorsion moyenne.
- ◆ Un quantificateur localement optimal a un dictionnaire qui peut être légèrement perturbé sans que la distorsion moyenne augmente.

Il n'existe pas de méthode qui décrit la façon de concevoir de dictionnaire globalement optimal pour les quantificateurs vectoriels. Seuls des propriétés suffisantes sont connues qui permettent de construire des dictionnaires localement optimaux.

Un quantificateur se décompose en deux applications : un codeur et un décodeur (Fig.II.4). le quantificateur (localement) optimal est alors celui qui réunisse les points suivants :

- ◆ Un codage optimal (pour un dictionnaire fixé), celui-ci respecte «la règle du plus proche voisin » que nous allons décrire ;
- ◆ Le décodage optimal (pour une partition S^i donnée), le vecteur code y_i doit minimiser la distorsion associée au Voronoï, y_i est donc le centroïde de cette cellule : $y_i = \text{cent}(S^i)$;

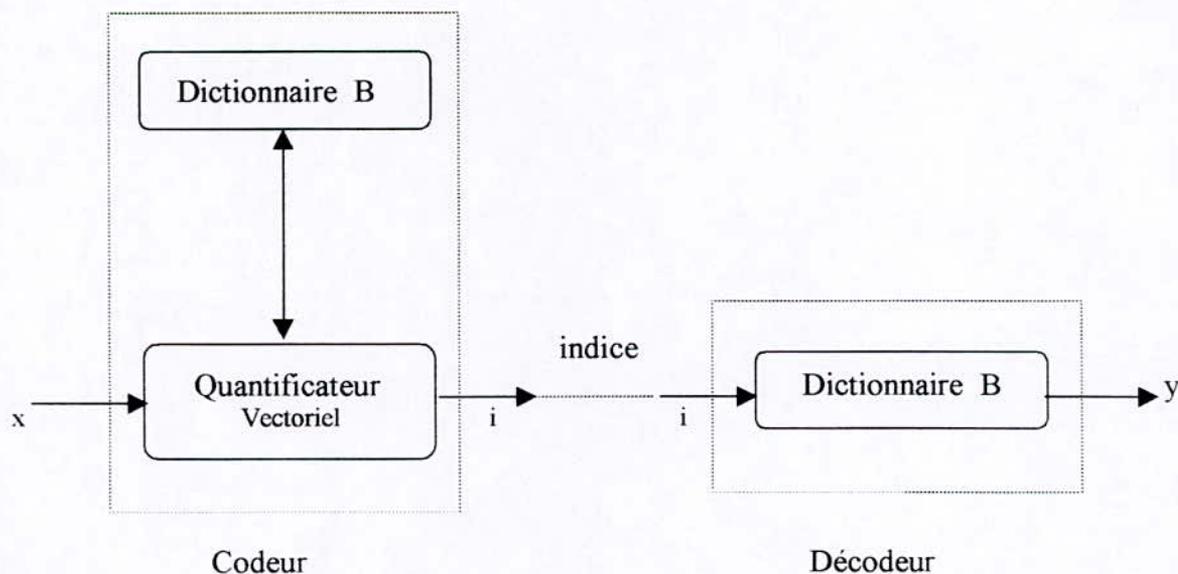


Fig II.4. Schéma d'un quantificateur vectoriel, x est la séquence à coder, i l'indice à transmettre et y est la séquence reproduit .

II.3.1. Condition du plus proche voisin :

Etant donné un décodeur et son ensemble fini de mots codes de sortie C, les classes de partitions S^i de l'encodeur sont optimales si, elles satisfont :

$$S^i \subset \{x / d(x, y_i) \leq d(x, y_j); \forall j\} \tag{II.11}$$

les régions de partitions sont définies par les mots codes y_i dans C :

$$Q(x) = y_i \text{ si seulement si } d(x, y_i) \leq d(x, y_j); \forall j \tag{II.12}$$

Condition de centroïde :

Etant donné une partition d'encodeur $P = \{S^i | i = 1, \dots, N\}$;

Les mots codes optimaux y_i dans C sont les centroïdes dans chaque partition S^i :

$$\begin{aligned} y_i &= \text{Cent}(S^i) \\ y_i &= \min E(d(x, y) | x \in S^i) \end{aligned} \tag{II.13}$$

Application dans la transmission : [1,2]

Soit Q un quantificateur vectoriel défini par un dictionnaire $B = \{Y_0, \dots, Y_{N-1}\}$ et par une partition $S = \{S_0, \dots, S_{N-1}\}$ de l'espace euclidien \mathcal{R}^k . Un exemple de partition dans l'espace \mathcal{R}^2 est représenté par la figure (II.5).

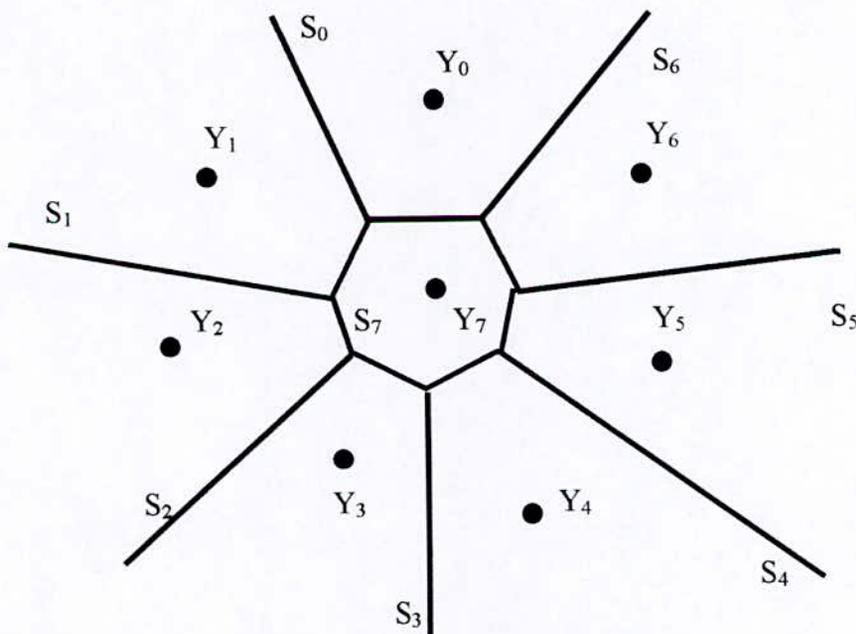


Fig II.5. Exemple de partition (N=8).

Trois opérations successives sont nécessaires pour quantifier un vecteur donné X (Figure II.6) :

1. Retrouver la région de quantification contenant x , c'est à dire la région de voronoï S_i ce qui revient à déterminer le plus proche voisin de x dans le dictionnaire B .
2. Chercher l'indice i de la région de voronoï contenant x qui est aussi l'indice du mot code y le plus proche de x .
3. Régénérer à la réception le mot code Y_i à partir de l'indice i .

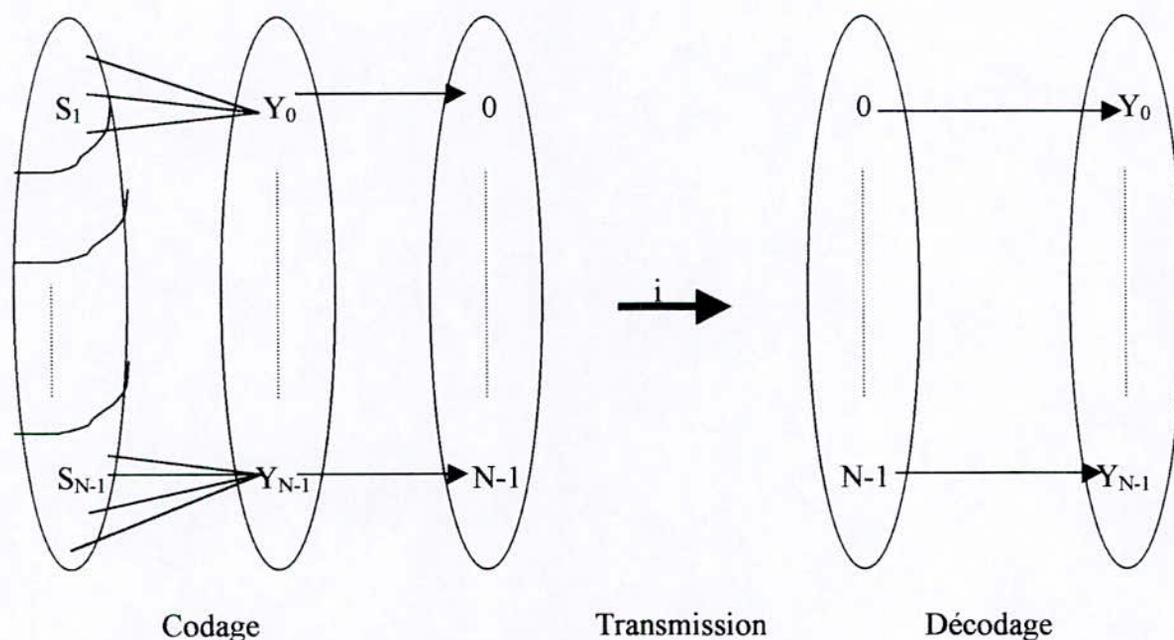


Fig II.6. Schéma de transmission.

II.4. Construction de quantificateurs statistiques :

Supposons que nous disposons d'une certaine distance d . construire un quantificateur revient donc à établir une stratégie de choix du dictionnaire associé. Cette stratégie est entièrement liée à la nature de la distribution des valeurs à quantifier.

Dans le cas où les points d'entrée sont distribués d'une façon non uniforme, on adoptera une approche statistique visant à tirer parti de cette non-uniformité. Le dictionnaire sera construit par l'apprentissage à partir d'une large base de vecteurs d'entrée où sera sélectionnée un nombre réduit de points susceptibles d'en refléter les propriétés statistiques.

En revanche, si la distribution des vecteurs d'entrée est plutôt uniforme, on aura intérêt à conférer à l'espace de représentation une structure mathématique forte, indépendamment de la «réalité» des données à traiter. Cette approche algébrique utilise généralement les propriétés des réseaux réguliers de points. On s'intéressera exclusivement à la première approche.

II.4.1. Algorithme de Lloyd Généralisé (GLA : Generalized Lloyd Algorithm):

Les conditions d’optimalité citées précédemment conduisent à la conception d’un algorithme qui réalise, à partir d’une séquence d’apprentissage représentative de la statistique de la source à coder, la construction d’un dictionnaire (localement) optimal. Cet algorithme de classification appelé aussi algorithme des k-moyens (k-means) est l’extension au cas vectoriel de l’algorithme de Lloyd-Max (cas scalaire) [30].

Il s’agit d’un algorithme d’optimisation itératif opérant à partir d’un dictionnaire initial. Pour chaque itération, dite itération de Lloyd, deux opérations distinctes sont appliquées :

- ◆ Une classification suivant la règle du plus proche voisin.
- ◆ Une optimisation suivant la condition du centroïde.

L’algorithme se déroule comme suit :

Soit un nombre N de vecteurs dans un espace à k dimensions formés à partir de kN échantillons. Nous disposons au départ d’un alphabet vectoriel Y’ contenant m vecteurs types y’ disposés d’une façon quelconque dans l’espace. Notre but est d’obtenir un alphabet vectoriel Y qui permettra de coder les échantillons de X avec une erreur moindre que celle que nous aurions obtenu en utilisant l’alphabet Y’. L’algorithme emploie un procédé itératif pour trouver Y à partir de Y’.

Lors du codage, chacun des vecteurs x sera associé à l’un des vecteurs y qui causera la plus petite erreur d(x,y) possible. Cette erreur est obtenue par :

$$d(x, y) = \sum_{i=0}^{k-1} (x_i - y_i)^2 \tag{II.14}$$

Chacun des vecteurs y servira donc à coder ou à représenter avec plus ou moins de précision un certain nombre de vecteurs de l’ensemble x.

Pour obtenir l’alphabet vectoriel Y à partir de Y’, nous commençons par associer chacun des vecteurs x à l’un des vecteurs y’ qui est le moins éloigné (au sens de la distance euclidienne). La figure (II.7) montre cette association. [3]

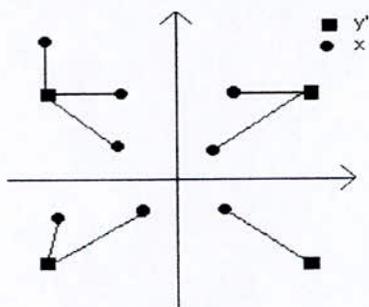


Fig II.7 chacun des vecteurs x est associé au vecteur y' le plus proche (au sens de la distance euclidienne).

La deuxième étape consiste à trouver le centre de chacune des classes ainsi constituées et à y placer un vecteur y comme le montre la figure (II.8).

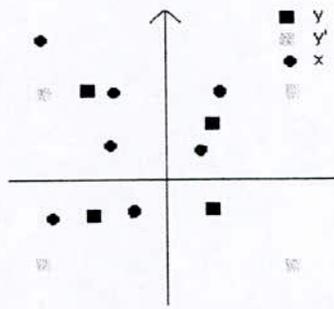


Fig II.8 on positionne les vecteurs de l'alphabet résultant Y au centre des classes des vecteurs x .

Comme les vecteurs y changent de position, il est probable que plusieurs vecteurs x changent d'allégeance. Nous recommencerons alors le processus de l'étape 1 jusqu'à ce qu'au moment où une certaine stabilité sera atteinte dans la formation des classes. On peut schématiser cette opération par la figure (II.9).

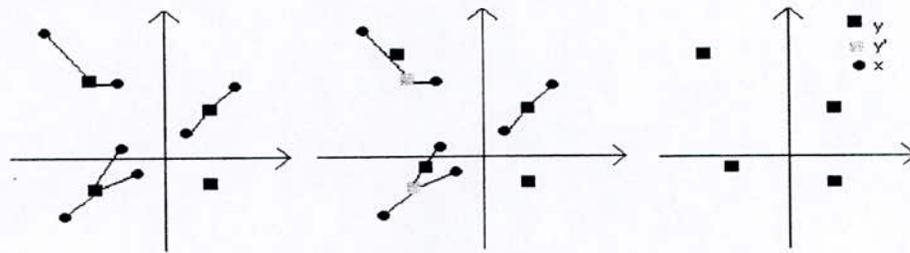


Fig II.9 succession des assignments x, y et des changements d'allégeances des vecteurs x

Chaque itération de Lloyd, en modifiant localement le dictionnaire, réduit ou laisse inchanger la distorsion moyenne. L'algorithme converge en un nombre fini d'itérations vers un minimum local varie en fonction du choix du dictionnaire initial. L'algorithme de la k moyenne peut être résumé comme suit :

Etape 1 : on commence par le dictionnaire initial C_1 , mettre $m=1$;

Etape 2a : ayant un dictionnaire $C_m = \{y_i\}$, partitionner la séquence d'entraînement en un ensemble de classe (cluster) S^i en utilisant la condition du plus proche voisin,

$$\text{Où } S^i = \{x \in T \mid d(x, y_i) \leq d(x, y_j), \text{ pour tout } j \neq i\}$$

Etape 2b : en utilisant la condition de centroïde, calculer les centroïdes pour l'ensemble des classes trouvées en étape 2a pour obtenir le nouveau dictionnaire :

$$C_{m+1} = \{\text{Cent}(S^i) \mid i = 1, \dots, N\}$$

Etape 3 : calculer la distorsion moyenne pour C_{m+1} , si elle a changé d'une petite quantité par rapport à l'itération précédente stop, sinon, mettre $m=m+1$ et répéter étape 2 et 3.

Le diagramme de la figure (II.10) illustre la procédure de fonctionnement de l'algorithme de Lloyd.

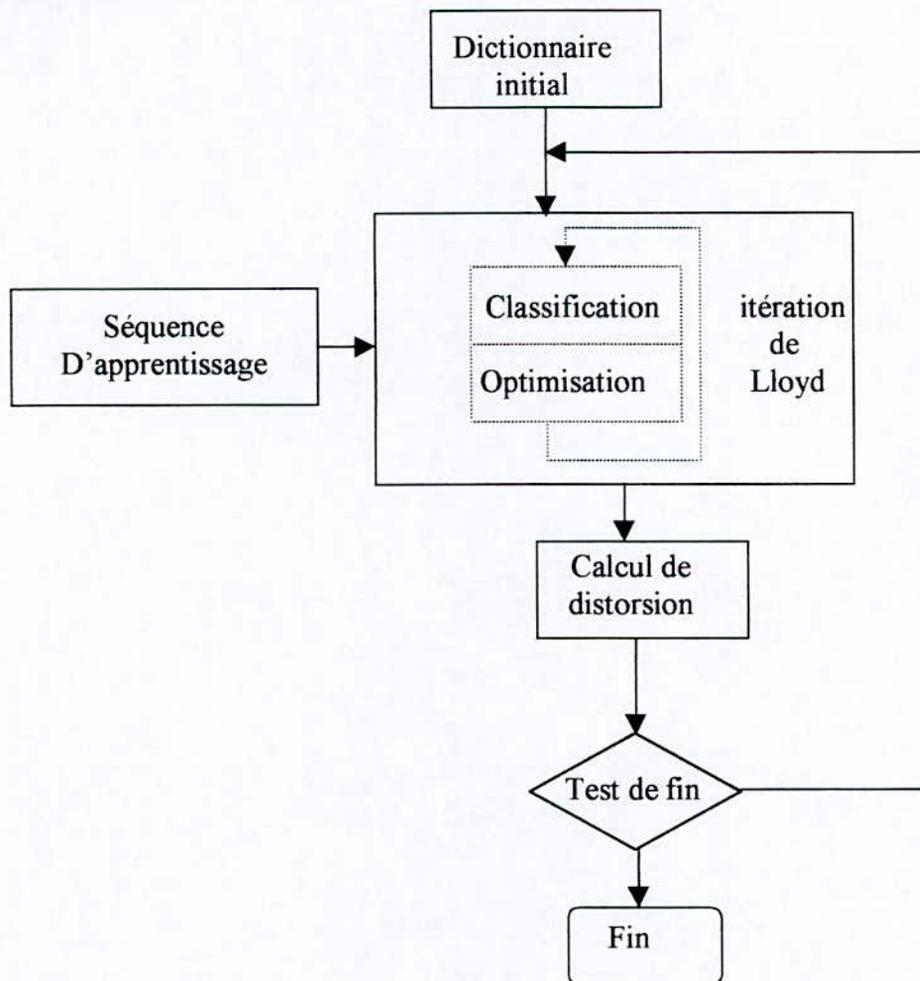


Fig.II.10. Schéma de fonctionnement de l'algorithme de Lloyd.

Pour plus de détail sur l'algorithme des k-moyens (LBG), consultez l'annexe A.[6]

Exemple d'application :

Soit la séquence d'apprentissage de vecteurs suivante :

La séquence S est de taille $n = 7$, avec une dimension K (longueur de vecteur) $K = 2$, et un alphabet initial de niveau $N = 4$.

La séquence d'apprentissage : $S = \{X_i, i = 1, \dots, 7\}$.

X_1	0.000	2.7027
X_2	13.513	16.2162
X_3	0.000	16.2162
X_4	10.8108	16.2162
X_5	8.1081	10.8108
X_6	2.7027	13.5135
X_7	2.7027	10.8108

- Alphabet initial :

$$A_0 = \{Y_{01} = (0,2), Y_{02} = (4,6), Y_{03} = (8,10), Y_{04} = (12,14)\}.$$

- Seuil d'erreur : $\varepsilon = 0.01$

▪ $m = 0$

Signal d'entrée	Signal de sortie	Distance minimale
X_1	Y_{01}	0.7027
X_2	Y_{04}	2.68372
X_3	Y_{03}	10.13121
X_4	Y_{04}	2.51511
X_5	Y_{03}	0.81799
X_6	Y_{03}	6.35658
X_7	Y_{02}	4.98266

$$D_{0\text{ moy}} = 4.02714.$$

- Les partitions obtenues sont :

$$S_1 = \{X_1\}$$

$$S_2 = \{X_7\}$$

$$S_3 = \{X_3, X_5, X_6\}$$

$$S_4 = \{X_2, X_4\}$$

- Le nouvel alphabet A est :

$$A_1 = \{Y_{11} = (0.000 ; 2.7027), Y_{12} = (2.7027 ; 10.8108), Y_{13} = (3.603 ; 13.513), Y_{14} = (12.162 ; 16.216)\}.$$

- $m = 1$

Signal d'entrée	Signal de sortie	Distance minimale
X_1	Y_{11}	0.0000
X_2	Y_{12}	1.35135
X_3	Y_{13}	4.50450
X_4	Y_{14}	1.35135
X_5	Y_{15}	5.25311
X_6	Y_{16}	0.90090
X_7	Y_{17}	0.0000

- pas de changements dans les partitions : $P(A_1) = P(A_2)$.
- $D_{1\text{moy}} = 1.908$.

Donc : $(D_{0\text{moy}} - D_{1\text{moy}}) / D_{1\text{moy}} > \varepsilon$.

- le nouvel alphabet A_2 est le même que A_1 .

- $m = 2$

Les distances minimales sont les même que celles de $m = 1$ donc pas de changements dans les partitions $P(A_1)$, $P(A_2)$ et $P(A_3)$.

$$D_{1\text{moy}} = D_{2\text{moy}} = 1.908.$$

$$(D_{1\text{moy}} - D_{2\text{moy}}) / D_{2\text{moy}} = 0 < \varepsilon.$$

- l'alphabet A_3 est le même que A_2 .

A la fin : on prend donc A_2 comme alphabet final.

$$A_2 = \{ Y_{21} = (0.000 ; 2.7027), Y_{22} = (2.7027 ; 10.818), Y_{23} = (3.603 ; 13.513), Y_{24} = (12.162 ; 16.216) \}.$$

- donc la distorsion moyenne est égale à 1.9087.

Remarques :

Les études effectuées sur l'algorithme LBG ont montrées qu'il permet à la fois d'avoir un alphabet final optimal (réduction de la distorsion moyenne), et réduire le nombre de bits nécessaires pour la transmission du signal.

Le choix de cet algorithme s'est porté pour les avantages suivants :

- L'ordre des vecteurs de la séquence d'apprentissage n'influe pas sur la construction du dictionnaire optimal.
- Le dictionnaire initial n'est pas fixé au préalable, il peut être choisi aléatoirement ou à partir de la séquence d'apprentissage.
- Le critère d'arrêt est conditionné uniquement par l'obtention d'un nouveau alphabet dont l'erreur est inférieure ou égale à celle de l'alphabet précédent.

II.5. Technique de conception du dictionnaire initial :

L'algorithme ainsi décrit est une procédure d'optimisation itérative basée sur la méthode de projections successives et donc conduit vers un minimum local.

La vitesse de convergence des itérations de la GLA et les performances du dictionnaire obtenu après convergence dépendent du dictionnaire initial C_i . Par conséquent, il est important de trouver un bon dictionnaire initial. L'une des techniques d'initialisations est décrite ci-dessous.

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

Soit v_i la séquence d'apprentissage des vecteurs. la procédure peut être exprimée comme suit :

- ♦ Calculer les normes de tous les vecteurs de l'ensemble d'apprentissage. Puis choisir le vecteur ayant la norme maximum comme mot code.
- ♦ Calculer la distance de tous les vecteurs d'apprentissage par rapport au premier mot code, et choisir le vecteur ayant la plus grande distance comme second mot code. On a alors un dictionnaire de taille 2.
- ♦ Généralement, avec un dictionnaire de taille $i=2,3,\dots,L$, nous calculons la distance entre les vecteurs d'apprentissage restant et tous les mots codes existants. La plus petite valeur trouvée est appelée distance entre le vecteur d'apprentissage et le dictionnaire. Alors, le vecteur d'apprentissage ayant la plus grande distance par rapport au dictionnaire est choisi pour être le $i^{\text{ème}}$ mot code. La procédure s'arrête quand on obtient un dictionnaire de taille L .

L'idée de base de cette procédure est d'utiliser le vecteur le plus différent des codes vecteurs existants comme un nouveau mot code.

II.6. Comparaison entre quantificateurs scalaires et vectoriels :

Distribution Gaussienne :

Ce qui peut paraître curieux de prime abord, c'est que, même si les variables sont indépendantes, il y a avantage à utiliser la quantification vectorielle. Dans quelques cas pathologiques, les performances sont égales. Ceci s'explique très simplement par l'observation d'un ensemble de M quantificateurs scalaires sont équivalent à un quantificateur vectoriel particulier opérant sur des vecteurs de dimension M . En effet la figure (II.12) illustre ce fait sur un exemple à deux dimensions [34]. On y compare la quantification d'un vecteur x selon deux approches différentes. On suppose que les composantes du vecteur sont indépendantes et distribuées chacune selon la loi normale. Dans la première situation, on emploie une paire de quantificateurs scalaires que l'on applique à chaque composante séparément. Ces quantificateurs sont optimaux avec un débit de 2 bits, soit $N_s = 4$ points (niveaux). Dans la deuxième situation, on utilise un quantificateur vectoriel ayant le même débit par dimension de 2 bits soit pour deux dimensions : $2 \cdot 2^2 = 8$ points.

Il apparaît clairement qu'une paire de quantificateurs scalaires est équivalent à un quantificateur vectoriel dont les points sont au sommet d'une grille cartésienne. Supposons maintenant que les composantes du vecteur d'entrée aient en fait une certaine corrélation. Rien ne se change du côté des quantificateurs scalaires par contre le quantificateur vectoriel est capable de disposer ces vecteurs arrondis à bon escient.

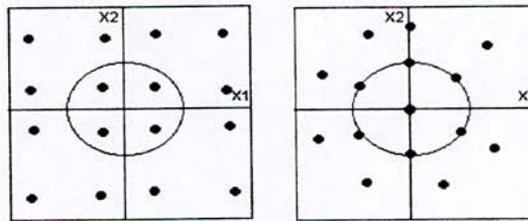


Fig II.11. comparaison de la quantification d'un vecteur à deux dimensions par une paire de quantificateurs scalaires optimaux (a) et par un quantificateur vectoriel (b). les composantes du vecteur sont indépendantes et distribuées selon une loi normale.

Distribution uniforme :

Supposons que nous ayons toujours à faire à un vecteur de deux dimensions mais distribué cette fois d'une façon uniforme sur un certain domaine et que nous comparions de nouveau les performances d'une paire de quantificateurs scalaires et un quantificateur vectoriel. On suppose que le nombre de points est suffisamment grand pour pouvoir négliger les effets de bord. La figure (II.13) nous montre une portion de chacun des deux quantificateurs. A noter que l'on cherche de nouveau des quantificateurs à débit comparable ; par conséquent ils ont le même nombre de points et par suite la même densité moyenne.

Pour conserver des densités de points comparables, il faut poser $d_2/d_1 = 1.075$. Le point le plus mal quantifié est représenté dans les deux cas par un point blanc à distance r_1 et r_2 respectivement. On voit bien que r_2 est inférieur à r_1 ; Par conséquent le quantificateur vectoriel est la plus performante question erreur de quantification.

Remarque : on peut citer une autre insuffisance de la quantification scalaire ; c'est qu'on ne peut pas quantifier au-dessous de 1 bit par échantillon ; tandis que dans la quantification vectorielle, on peut descendre à un débit de $1/k$ bit par échantillon.

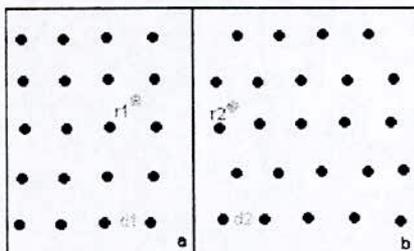


Fig II.12. Comparaison de deux méthodes de quantification pour un vecteur x à 2 dimensions réparti uniformément sur le domaine . Deux quantificateurs scalaires (a) et un quantificateur vectoriel (b).

Chapitre III

LA QUANTIFICATION PAR TREILLIS

III.1.Introduction :

La théorie de la modulation est celle du codage de source, ont bénéficiés ces dernières années des larges applications du codage par treillis. L'exploitation de treillis, fût l'œuvre pour la première fois d'Ungerboeck (c'est le TCM). En utilisant une technique appelée « Mapping by set partitionning » à des constellations de dimension 1 et 2, et des codes convolutifs, Ungerboeck à obtenu de bonnes performances dans le codage de canal à bande limitée. Ces travaux ont retenus une attention particulière des chercheurs à travers le monde. Plusieurs schémas de TCM ont été élaborés. [14]

Forney et Al ont formalisés leur conception en montrant qu'ils peuvent être décrits par les trois éléments suivants (fig III.1) : [1]

1. Un encodeur binaire par blocs ou convolutif qui permet de générer un nombre de bits plus grand à sa sortie que le nombre de bits à son entrée.
2. Ces bits codés servent à sélectionner une classe ou sous-ensemble d'une constellation partitionnée.
3. Des bits additionnels permettent de choisir un point dans la classe précédemment sélectionnée.

L'utilisation d'un ensemble dilaté de signaux fournit une redondance au codage tout en permettant de pouvoir maximiser la distance euclidienne minimale, appelée « free distance » entre les séquences des signaux codés.

La dualité entre la modulation et la quantification a conduit Marcellin et Fischer à appliquer le concept du « Mapping by set partitionning » dans la quantification par treillis, qui a été nommé « treillis coded quantization » (TCQ).

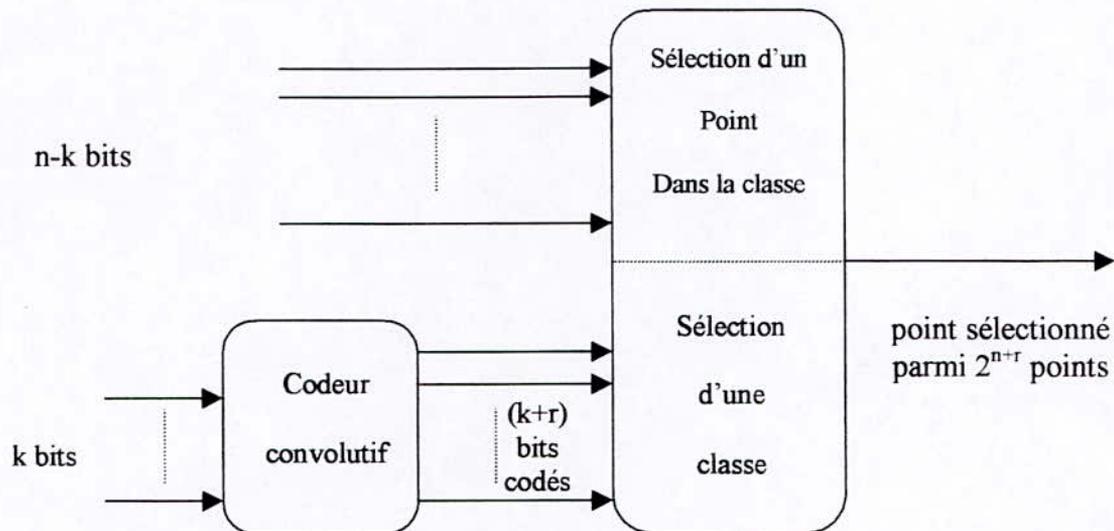


Fig V.1. Structure d'un codeur TCM

III.2. La quantification par treillis :

La VQ classique utilise un alphabet de 2^m vecteurs Y en k dimensions pour coder un ensemble X de vecteurs échantillons ; Ce qui donne un débit de m bits par vecteurs, ou encore de m/k bits par échantillons.

Avec la TCQ, il s'agit tout d'abord de doubler la taille de l'alphabet scalaire tout en conservant le débit intact. Il nous faudra donc imposer quelques règles sur cette utilisation du quantificateur afin de conserver le débit initial donné.

III.2.1. Description :

Supposons que nous ayons à quantifier une source X à un débit R . la formation de l'alphabet se fait de la manière suivante :

Soit Y qui est composé de 2^m échantillons. Quand on double la taille on obtient 2^{m+1} échantillons qui vont former l'alphabet Y' appelé « constellation ». On divise cette dernière en 2^{m+1} classes (Q_0, Q_1, \dots, Q_m), contenant chacune $2^{m-m'}$ échantillons. Les classes seront assignées aux branches de treillis et chacune d'elle doit être un bon quantificateur. Un exemple d'alphabet utilisé dans le cas d'un treillis à quatre états, avec un débit d'un demi-bit par échantillon est illustré par la figure (III.2).[1,3]

Le codeur convolutif publié par Ungerboeck est donné par la figure (III.3).

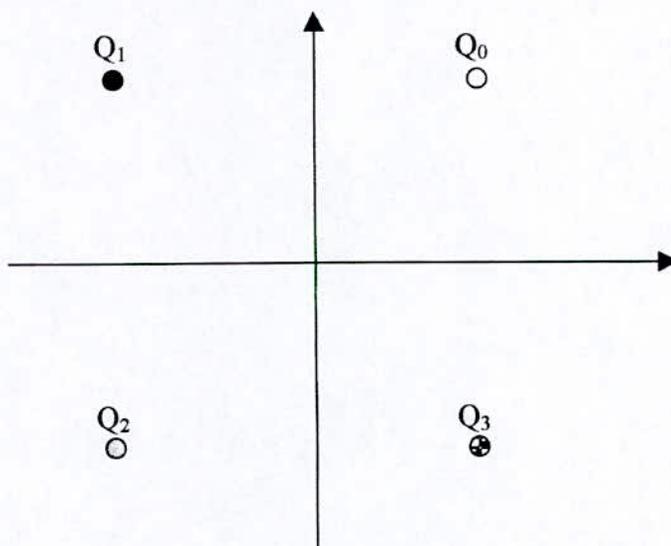


Fig III.2. Alphabet d'un TCQ à un demi-bit par échantillon

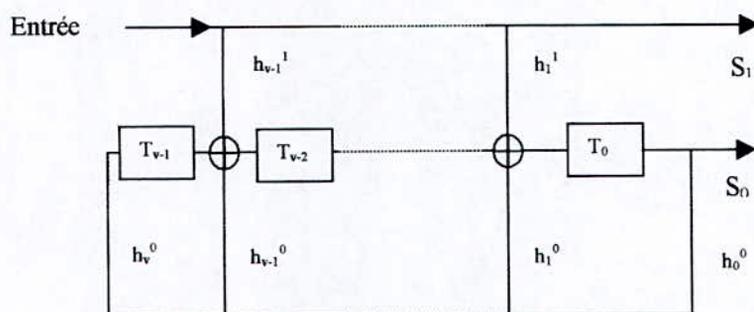


Fig. III.3. codeur convolutif avec un nombre d'états égal à 2^v

Dans le cas d'une utilisation d'un quantificateur vectoriel classique, il faudrait $m+1$ bits et $m-m$ bits pour identifier respectivement la classe et le vecteur type, ce qui donnerait un débit total de $m+1$ bits.

Cependant, en faisant appel à un treillis, il nous offrira la possibilité de ramener le débit à R bits.

La technique consiste à n'utiliser que 2^m des 2^{m+1} échantillons types disponibles dans la constellation mais en changeant les nombres de ce groupe de 2^m échantillons selon des règles connues tant de codeur que du décodeur.

III.2.2.Représentation :

Nous allons détailler un TCQ à 4 états avec un alphabet scalaire de 16 échantillons types séparés en 4 classes Q_0, Q_1, Q_2, Q_3 [1,3,8,11]. Notre codeur est représenté dans la figure ci dessous (fig III.4).

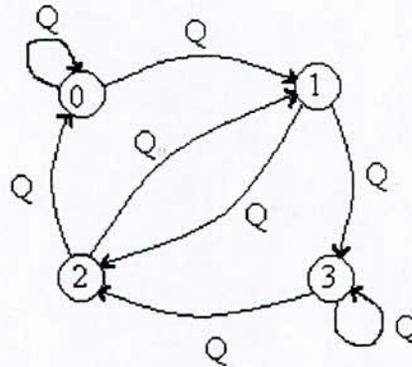


Fig III.4.Diagramme de transition.

Grâce au codeur convolutif qui sélectionne une classe parmi les quatre, on peut donner une autre représentation du diagramme d transition (fig III.5).

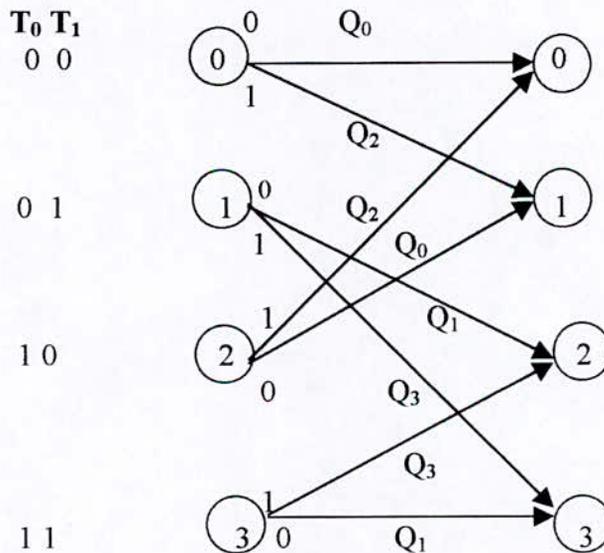


Fig III.5. Exemple de treillis à quatre états.

On remarque que les états du quantificateur autorisent d'utiliser soit les classes Q_0, Q_2 soit les classes Q_1, Q_3 , de plus, le choix d'un sous-alphabet permis occasionne une transition vers un nouvel état.

En effet, supposons que nous nous trouvions à l'état 0 du treillis et que nous ayons à quantifier un vecteur $X=(x_0, x_1)$ quelconque. Dans cet état, nous avons le choix d'utiliser l'un des échantillons types de Q_0 ou de Q_2 pour représenter x_0 . Si nous choisissons notre vecteur type dans Q_0 , nous serions tenus de quantifier x_1 avec le même groupe Q_0 Q_2 .

Si par contre nous utilisons Q_2 pour x_0 , x_1 serait quantifier par l'un des quantificateurs Q_1 Q_3 . Comme le choix des deux ensembles dépend des prochains échantillons, nous attendrons de les connaître pour prendre une décision. Entre temps, nous déterminons la distorsion ou l'erreur causée par chaque meilleur représentant de Q_0 et Q_2 et établissons l'arbre suivant (fig III.6) :

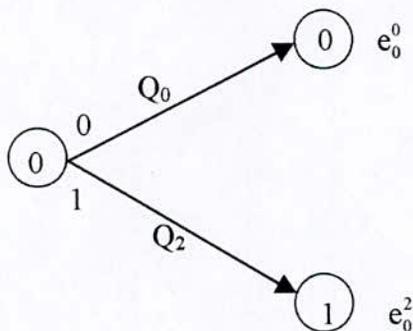


Fig III.6. La première transition du vecteur.

Considérons maintenant l'échantillon x_1 . Si, pour coder x_0 notre choix s'est porté sur Q_0 nous répétons alors la figure précédente, dans l'autre cas il faudra se servir de Q_1 et Q_3 pour coder x_1 . En choisissant encore le meilleur représentant de chacun des sous-alphabets Q_n et en reprenant la procédure qui a permis d'obtenir la figure, un arbre plus développé peut être établi (fig III.7).

On voit déjà apparaître l'essence de la quantification par treillis. En effet, l'erreur cumulée représente dans tous les cas la somme des distorsions produites sur les échantillons x_0 et x_1 . La branche sur laquelle l'erreur est la plus faible indique la meilleure stratégie de quantification. Il est possible d'accepter une erreur plus grande sur x_0 si ce choix s'avère payant pour les échantillons suivants.

On remarque que pour le troisième vecteur que les branches de l'arbre 0 et 2 offrent les mêmes quantificateurs Q_0 et Q_2 et ont des chemins qui aboutissent aux mêmes transitions ; C'est aussi le cas pour les branches 1 et 3 avec les quantificateurs Q_1 et Q_3 . En fait, à partir du troisième vecteur, l'arbre se replie sur lui-même et devient un treillis (Fig III.8).

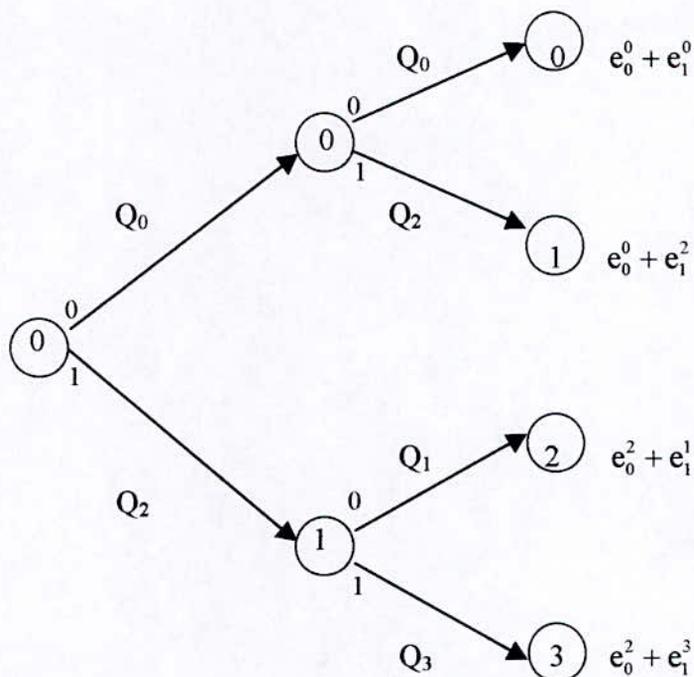


Fig III .7. Etat de l'arbre binaire après la quantification de x_1 .
Chacune des feuilles de l'arbre porte l'erreur accumulée.

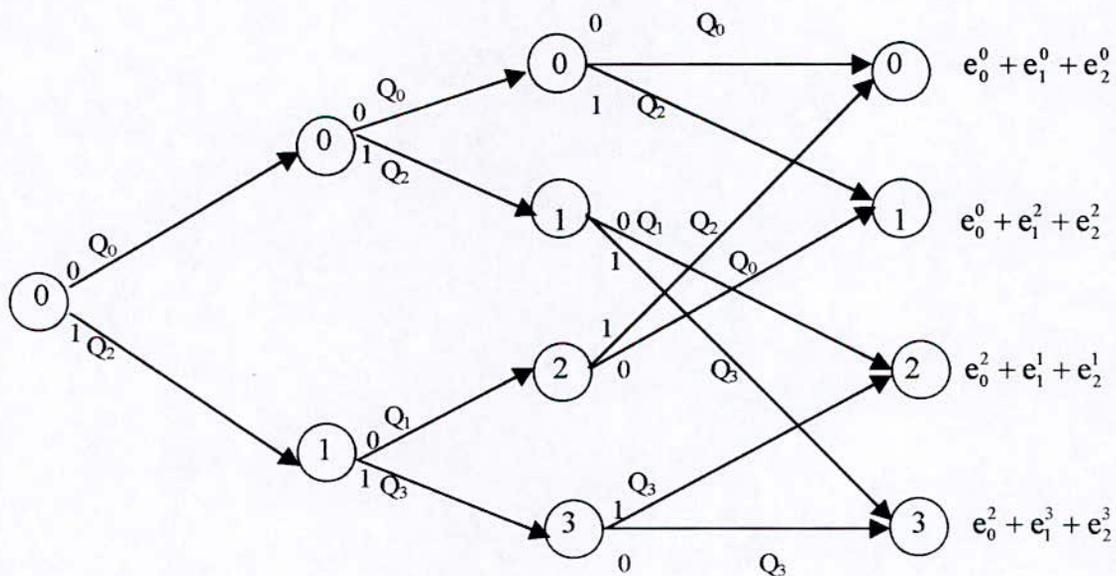


Fig III.8. Etat du treillis pour le troisième vecteur avec les erreurs
accumulées pour les premières branches seulement.

III.2.3. La distorsion :

La distorsion calculée ici c est le carré de la distance euclidienne qui est définie par :

$$e_j^i = |x - Q(x)|^2 \quad (\text{III.1})$$

x : le vecteur à quantifier.

$Q(x)$: son représentant.

j : c est l'indice représentant le classement des branches.

i : c est l'indice du quantificateur utilisé.

Pour minimiser la distance, on utilise l'algorithme de Viterbi.

Au départ, on assigne une erreur cumulée nulle à l'un des états du treillis et une erreur infinie aux autres. Le codeur et le décodeur s'entendent préalablement sur le choix de l'état ayant une erreur nulle.

III.2.4. Interprétation :

Chaque nœud dans le treillis représente un état et les branches correspondent aux transitions. Des deux chemins arrivant à chacun des états, un seul présente un certain intérêt.

En effet, chacun de ces chemins représente un passé différent, mais ils auront le même avenir car ils aboutissent au même état.

Le chemin qui aura accumulé la plus faible erreur sera le chemin restant ou « le survivant ». Il représente pour les chemins arrivants à un état donné la meilleure stratégie de quantification pour les échantillons passés.

Il y aura quatre de ces survivants pour chaque nouveau vecteur X . après un nombre n des échantillons, on choisira parmi ces quatre chemins celui qui a cumulé la plus faible erreur.

III.2.5. Algorithme de quantification par treillis :

On se propose de quantifier n échantillons à la fois par un treillis de quatre états. Les indices i des échantillons sont tel que : $i = 1, 2, \dots, n$. l'erreur e_i^j sera la plus petite erreur ou distance euclidienne entre un vecteur d'indice i à quantifier et un vecteur le plus proche dans le quantificateur Q_j ($j = 0, \dots, 3$) considéré [18].

L'algorithme se subdivise en cinq étapes :

- ❖ **Etape 1** : Au départ, assigné une erreur cumulée nulle à l'un des états du treillis et une erreur infinie aux autres. Le codeur et le décodeur s'entendent préalablement sur le choix de l'état ayant une erreur nulle.

- ❖ **Etape 2 :** prendre un vecteur de l'échantillon et calculer les erreurs avec les quatre quantificateurs ; l'erreur e_i^j sera la plus petite erreur causée par le quantificateur Q_j .
- ❖ **Etape 3 :** pour chaque état du treillis, calculer les deux erreurs cumulées en sommant l'erreur cumulée de l'état parent avec l'erreur e_i^j associée à la transition considérée. Conserver le chemin donnant l'erreur la plus faible.
- ❖ **Etape 4 :** pour chacun des états, mémoriser le chemin survivant (celui ayant l'erreur la plus faible des deux chemins arrivant à un état). Cela requiert un bit. Mémoriser aussi l'indice du vecteur type dans le quantificateur Q_j associé au chemin. Si chaque quantificateur contient $2^{m-m'}$ échantillons cela requiert $m-m'$ bits.
- ❖ **Etape 5 :** si moins de n échantillons ont été considérés, reprendre à l'étape 2. Sinon, choisir parmi les quatre chemins celui ayant la plus faible erreur cumulée, et coder les informations le concernant telles que les transitions et indices d'échantillons des quantificateurs pour chaque étage.[3]

III.2.6.Exemple d'application :

Supposons que nous ayons à quantifier une source X , telle que $X = \{ 0.5, 0.2, -0.1, 0.7 \}$, à un débit de $R = 2$ bits par échantillon.

Pour cela, nous allons détailler un TCQ à quatre états avec un alphabet vectoriel de 8 échantillons types, séparés en quatre classes Q_0, Q_1, Q_2, Q_3 .

La quantification de la source X vers l'alphabet vectoriel se fait par l'intermédiaire du diagramme de transition présenté par la figure (III.9).

Q_0	Q_1	Q_2	Q_3
-0.2	-0.6	-1	-2
2	1	0.6	0.2

Tableau III.1. Les échantillons des quantificateurs.

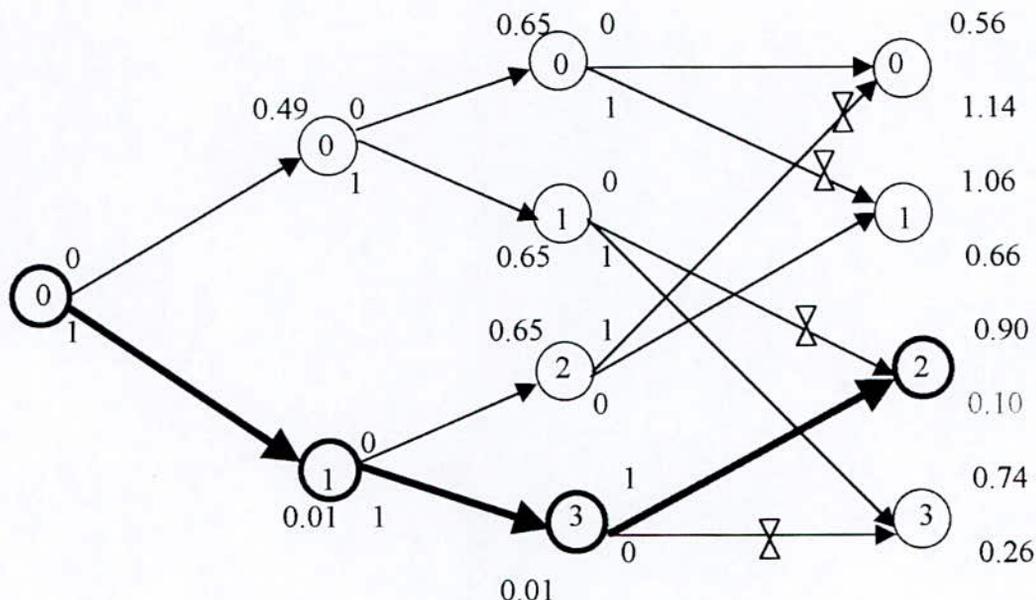


Fig III.9. Quantification du vecteur X par treillis

On remarque que le quantificateur Q_0 qui contient les éléments $q_0 = -0.2$ (codé par 0) et $q_1 = 2$ (codé par 1) est utilisé dans le cas de la première branche horizontale. Pour reconnaître laquelle des valeurs cause l'erreur la plus petite, on affecte respectivement deux indices 0 et 1, on doit calculer la distance euclidienne entre l'échantillon 0.5 du vecteur X et celle de Q_0 . On remarque que $= (0.5 - (-0.2))^2 = 0.49$ et $= (0.5 - 2)^2 = 1.25$; donc on quantifie 0.5 à -0.2 qui représente une distance plus petite de celle de 2. de même pour Q_2 qui donne $(0.5 - 0.6)^2 = 0.01$. On fait le même calcul pour toutes les branches et les quantificateurs jusqu'à dernière composante du vecteur X.

Pour décider lequel des chemins correspond au codage optimal, on laissera le treillis se développer jusqu'à la dernière composante du vecteur X à coder. Le nœud du dernier étage du treillis donnant la distance la plus petite sera la base de cette décision. En effet, on remarque que le chemin Q_2, Q_3, Q_3 (en gras) à accumuler la plus faible erreur ($e = 0.1$) par rapport aux autres chemins du treillis réalisé. Il représente donc la meilleure stratégie de quantification de la source X vers l'alphabet vectoriel.

Pour que ce chemin soit transmis, on doit, pour chaque branche, mémoriser le numéro de la transition (0 ou 1). dans notre exemple, on a les transitions optimales suivantes :

$$I_1 = [1 \ 1 \ 1] \text{ qui correspond aux échantillons : } 0.6, 0.2, 0.2 \text{ respectivement.}$$

Pour ce qui est des valeurs quantifiées $Y(i)$ correspondant au q_j , elles sont données ainsi que leurs indices I_0 dans la table ci dessous :

X(i)	0.5	0.2	-0.1
Y(i)	0.6	0.2	0.2
I ₀	1	1	1
I ₁	1	1	1

Tableau III.2.. résultat de quantification.

I₀ représente le code du quantificateur utilisé.

I₁ représente le chemin suivi.

Finalement seuls les bits de I₁ (chemin optimal) et ceux de I₀ (indice de l'alphabet) sont à transmettre. A la réception le chemin optimal est reconstitué à partir de I₁= [1 1 1] et les valeurs Y(i) correspondants sont données à partir des indices contenus dans I₀= [1 1 1].

III.3.La quantification vectorielle codée par treillis (TCVQ) :

III.3.1.introduction :

On a vu précédemment que la quantification vectorielle se base sur la quantification des vecteurs, c'est le cas général de la quantification scalaire qui sert à quantifier les échantillons d'un vecteur.

La quantification codée par treillis est une méthode qui améliore les performances de la quantification scalaire, elle utilise le treillis pour quantifier les échantillons.

Le mélange entre la quantification vectorielle et celle codée par treillis donne la naissance à une nouvelle méthode efficace nommée «la quantification vectorielle codée par treillis (TCVQ) » [7,9] ; ayant le même principe de la TCQ mais au lieu de quantifier des échantillons d'un vecteur dans plusieurs étapes, on quantifie ce même vecteur dans une seule étape.

Les expériences obtenues par FISCHER montrent que la quantification vectorielle codée par treillis produit une distorsion plus petite que celle de la TCQ et donne des débits fractionnaire ce qui ne permet pas d'avoir le TCQ.

Les paramètres ou grandeurs suivantes sont considérés :

R : est le débit de codage (en bits/ dimension).

L : est la dimension des vecteurs de source.

l : est la dimension des vecteurs de reproduction.

N : le nombre d'états pour un treillis donné.

M : un entier positif, tel que 2^M est le nombre de branches entrantes (sortantes) dans l'état du treillis.

R' : le facteur d'expansion d'un dictionnaire (en bits/ dimension).

III.3.2. La structure du codeur :

Soit R le débit du codage en bits/dimension, L la dimension du vecteur à coder, l un entier positif tel que L/l est un entier, et M un entier positif. On assume que le produit RL est un entier. Considérant N états de treillis contenant chacune 2^M branches entrantes et sortantes. On choisit des sous alphabets de dimension l et soit R' le nombre de multiplication du dictionnaire.

Pour $R > 0$, $R' \geq 0$, $M \geq 1$, $L = nl$ et $RL \geq M$. n est un entier positif.

Soit $K = R'l + M$ et 2^K sous alphabets S_0, S_2, \dots, S_{2^K} chacun contient 2^{RL-M} vecteurs.

L'algorithme ou la procédure peut être décrit comme suit :

- On forme l'alphabet vectoriel ρ contenant $2^{(R+R')l}$ symboles de reproduction. On remarque que la taille de ρ est $2^{R'l}$ fois la taille du dictionnaire de la nominale VQ.
- On partitionne l'alphabet ρ en 2^K sous alphabets notés S_0, S_2, \dots, S_{2^K} . Donc chaque sous alphabets à exactement 2^{Rl-M} vecteurs.
- Pour un treillis de N états et 2^M branches, on associe à chaque branche un sous alphabet S_i , en effet, si chaque sous alphabet est associé au moins à une branche, il faut que $N \geq 2^{R'l}$.

Le choix du treillis et l'association branche-sous alphabet sont des aspects importants dans le design, ceci sera détaillé plus tard.

- On utilise M bits/vecteur pour identifier les sous alphabets et $Rl - M$ bits/vecteur pour identifier le code mot, ce qui nous donnera un débit de Rl bits/vecteur.
- Le codage est accompli en deux étapes :

Dans une première étape d'exécution, pour chaque vecteur de source X , on cherche le plus proche voisin code mot dans chaque sous alphabet qui correspond à une distorsion d_i .

Dans une deuxième étape, l'algorithme de Viterbi est utilisé pour trouver le chemin qui minimise la métrique cumulée, tel que la métrique d'une branche associée à un sous alphabet S_i est la distorsion trouvée en étape 1.

En effet, RL bits sont utilisés pour spécifier la séquence des code mots correspondants au chemin optimal.

III.3.3. La complexité du codeur :

Il y a $2^{R^{l+M}}$ sous alphabets, chacune contenant $2^{R^{l-M}}$ codes mots.

Dans la première étape de codage, pour trouver le plus proche voisin code mot dans chaque sous alphabet pour un vecteur de source X il faut :

$l2^{R^{l-M}}$ additions.
 $l2^{R^{l-M}}$ multiplications.
 $2^{R^{l-M}}-1$ comparaisons.

Puis chaque état de treillis a 2^M branches, donc pour déterminer le SURVIVOR, il faut :

2^M additions.
 2^M-1 comparaisons.

La complexité totale est :

$l2^{(R+R')l}$ multiplications.
 $l2^{(R+R')l} + N2^M$ additions.
 $2^{R^{l+M}}(2^{R^{l-M}}-1) + N(2^M-1)$ comparaisons.

La complexité de la VQ de dimension l est :

$l2^{R^l}$ multiplications.
 $l2^{R^l}$ additions.
 $2^{R^l} - 1$ comparaisons.

On remarque que pour la même dimension des symboles de reproductions, la TCVQ présente plus de complexité que la VQ.

Remarques :

- D'après M^r Fischer, les performances de la TCVQ pour un débit R fixe augmentent généralement en augmentant l ou N.
- En se basant sur une évaluation expérimentale des performances de codage en TCVQ pour une source donnée, il est possible de modéliser la variation de la distorsion comme une fonction de l et N. en effet, pour un niveau de complexité donné, on peut sélectionner des valeurs optimales de l et N pour une application donnée.

III.3.4. La conception du codeur :

III.3.4.1. construction de l'alphabet initial :

Soit v_i la séquence d'apprentissage des vecteurs. la procédure peut être exprimée comme suit :

- ◆ Calculer les normes de tous les vecteurs de l'ensemble d'apprentissage. Puis choisir le vecteur ayant la norme maximum comme mot code.
- ◆ Calculer la distance de tous les vecteurs d'apprentissage par rapport au premier mot code, et choisir le vecteur ayant la plus grande distance comme second mot code. On a alors un dictionnaire de taille 2.
- ◆ Généralement, avec un dictionnaire de taille $i=2,3,\dots,L$, nous calculons la distance entre les vecteurs d'apprentissage restant et tous les mots codes existants. La plus petite valeur trouvée est appelée distance entre le vecteur d'apprentissage et le dictionnaire. Alors, le vecteur d'apprentissage ayant la plus grande distance par rapport au dictionnaire est choisi pour être le $i^{\text{ème}}$ mot code. La procédure s'arrête quand on obtient un dictionnaire de taille L .

L'idée de base de cette procédure est d'utiliser le vecteur le plus différent des codes vecteurs existants comme un nouveau mot code.

III.3.4.2. construction des sous alphabets :

On adoptera les règles AD-HOC dans la construction des sous alphabets et dans l'association branche-sous alphabets :

Règle 1 : la construction des sous alphabets doit vérifier les règles de set partitioning de Ungerboeck ; c'est à dire, pour chaque niveau de partition il y a une augmentation de la distance euclidienne entre les points des sous alphabets.

Règle 2 : l'union des sous alphabets correspondants aux branches sortantes ou entrantes dans l'état de treillis doit être un raisonnable quantificateur pour la source.

Règle 3 : les symboles dans chaque sous alphabets doivent être équiprobables.

En basant sur ces règles, M^r H.S.Wang et N.Moayeri [9] ont proposé un algorithme de partition qui est décrit comme suit :

Algorithme de partition :

Soit C le dictionnaire initial de taille $M = 2^{l(R+R')}$; R est le débit.

On calcule la distance entre chaque paire de codes vecteurs et on les classe dans un tableau par ordre de grandeurs. Cette opération nous permet d'avoir un tableau de taille $M*(M-1)/2$.

La distance calculée entre deux codes vecteurs c_i et c_i' est donnée par : $d_i = \|c_i - c_i'\|$

Puis on met c_0 et c_0' dans les sous alphabet A_0 et A_1 respectivement.

On doit avoir $M/2$ codes vecteurs dans chaque sous alphabet, donc on répète cette partition selon l'algorithme suivant :

- **Etape 1 :** on cherche dans le tableau l'index j tel que $\forall i < j ; n_i c_i \text{ ni } c_i' \notin A_0 \cup A_1$, mais au moins c_j ou $c_j' \in A_0 \cup A_1$.
- **Etape 2 :** si c_j et $c_j' \in A_0 \cup A_1$; c'est à dire les deux codes vecteurs sont déjà assignés, donc, on met :

$$c_i \longleftarrow c_{i+1} \quad ; \quad c_i' \longleftarrow c_{i+1}' \quad ; \quad d_i \longleftarrow d_{i+1} \quad ; \quad \forall i \geq j .$$

puis ; Aller à l'étape 1.

- **Etape 3 :** Si $c_j \in A_0$ (ou A_1), donc on met c_j' dans A_1 (ou A_0) et Si $c_j' \in A_0$ (ou A_1), donc on met c_j dans A_1 (ou A_0).
- **Etape 4 :** Si la taille de A_0 (ou A_1) est égale à $M/2$, donc on ajoute le reste des codes vecteurs à A_1 (ou A_0). **Stop.**
Si non ; aller à 1.

Pour faire partitionner le dictionnaire en 2^{R+1} sous alphabets on fait répéter cet algorithme R' fois.

III.3.4.3.association branche-sous alphabet :

Cette association s'obéit à la deuxième règle de AD-HOC décrite ci-dessus. Le treillis de 8 états représenté dans la figure (III.9) applique cette règle.

Remarque : Dans la figure ci-dessous, la notation $S_i S_j$ signifie que :
 S_i est associé à la branche de haut.
 S_j est associé à la branche de bas.

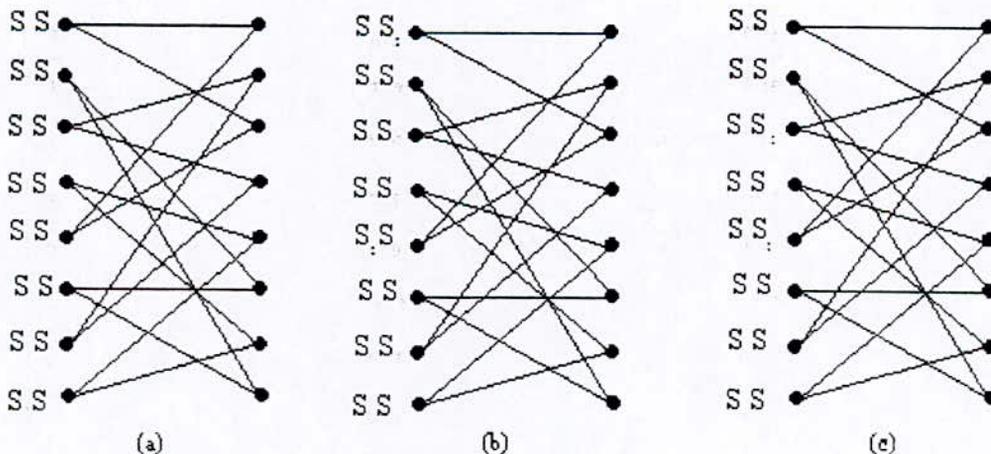


Fig.III.10. structure d'un treillis 8 états

- (a) 4 sous alphabets.
- (b) 8 sous alphabets.
- (c) 16 sous alphabets.

III.3.4.4. optimisation de l' alphabet de reproduction :

Après ces trois étapes, il est désirable d'optimiser l'alphabet de reproduction pour le treillis donné. Dans la référence [8] une méthode de quasi-newton est utilisée pour optimiser un alphabet de reproduction scalaire, la complexité des calculs de cette méthode augmente rapidement avec la taille de l'alphabet de reproduction, donc cette technique est non efficace dans le cas d'un alphabet multidimensionnel. En alternative, L'algorithme de Lloyd généralisé proposé par Stewart, Gray et Linde [35] peut être utilisé pour optimiser l'alphabet de reproduction, on incorpore cette approche dans le contexte de la TCVQ.

On résume l'algorithme de l'optimisation comme suit :

Etant donné :

- Une séquence d'apprentissage χ .
- Les branches du treillis nommées par les sous alphabets S_i .
- Un alphabet initial ρ^1 avec des éléments $y_i^1, i = 1, 2, \dots, \|\rho^1\|$ partitionné en sous alphabets S_i .
- On utilise la distance euclidienne comme un critère de mesure de distorsion.
- On prend $\varepsilon > 0$.
- On pose $\delta(0) = \infty, j = 1$ et à l'itération j :

Etape 1 : on code la séquence d'apprentissage χ en utilisant l'algorithme de Viterbi et la structure de TCVQ avec un alphabet ρ^j . On définit la distorsion résultante comme suit :

$$\delta(j) = \frac{1}{\|\chi\|} \sum_{x \in \chi} d(x, y) \quad (\text{III.1})$$

Si $[\delta(j-1) - \delta(j)] / \delta(j) < \varepsilon$ alors stop et l'alphabet final est ρ^j .
Sinon continue.

Etape 2 : pour la séquence codée $y(n)$ trouvé en étape 1, on partitionne la séquence d'apprentissage en cellules B_i^j , tel que :

$$x(n) \in B_i^j \text{ si et seulement si } x(n) \text{ est codé par } y_i^1$$

Etape 3 : on génère un nouvel alphabet de reproduction ρ^{j+1} , en accordant la relation suivante :

$$y_i^{j+1} = \frac{1}{\|B_i^j\|} \sum_{x \in B_i^j} x \quad i = 1, 2, \dots, \|S\| \quad (\text{III.2})$$

$j \leftarrow j+1$ et aller à l'étape 1.

III.3.5. Les propriétés du codeur :

Si on a une source stationnaire et ergodique, d'après la relation (III.2) on conclut pour une longue séquence d'apprentissage, pour chaque $i = 1, 2, \dots, \|\rho\|$:

$$y_i = \frac{1}{\|\mathcal{B}_i\|} \sum_{x \in \mathcal{B}_i} x \xrightarrow{P} E[x | x \text{ codé par } y_i] \quad (\text{III.3})$$

le bruit de codage est défini comme suit :

$$q(n) = x(n) - y(n) \quad (\text{III.4})$$

l'équation (III.3) implique que :

- La moyenne de bruit de codage est nulle.
- $E[q^T y] = 0$.
- La distorsion égale à la différence de la variance de la source et la variance de séquence de reproduction.

On résume ces propriétés dans le théorème suivant :

Théorème : soit $x(n)$ une source de vecteurs stationnaire et ergodique ; Et le codeur a un alphabet optimisé par l'algorithme d'optimisation décrit au-dessus, on note $y(n)$ comme étant la séquence de reproduction et $q(n) = x(n) - y(n)$ le bruit de codage. Si la séquence d'apprentissage est très longue, on a :

$$E[q] = 0 \quad (\text{III.5})$$

$$\text{Distorsion} = \frac{1}{l} E\{\|q\|^2\} = \text{var}(x) - \text{var}(y) \quad (\text{III.6})$$

$$E[q^T y] = 0 \quad (\text{III.7})$$

telle que :

$$\text{var}(x) = \frac{1}{l} \sum_{i=1}^l \text{var}(x_i)$$

$$\text{var}(y) = \frac{1}{l} \sum_{i=1}^l \text{var}(y_i)$$

$$\|q\|^2 = q^T q$$

la démonstration de ce théorème est donnée en annexe c.

Chapitre IV

LES PARAMETRES LSF

IV.1. Introduction :

Nous avons vu au chapitre précédent que la prédiction linéaire était une des approches la plus populaire employée pour la description des variations temporelles du spectre à terme court du signal de parole. De nombreux travaux ont été effectués sur la quantification scalaire et vectorielle des coefficients du filtre LPC.

Il est connu, dans le contexte de la compression de la parole que les coefficients de prédiction sont peu appropriés à la quantification à cause de leur large gamme dynamique et aux possibilités d'instabilités du filtre d'analyse LPC.

Différentes représentations, telles que les coefficients de réflexion, les paramètres LAR (Log Area Ratio), les coefficients spectraux et les paramètres LSF (Line spectral fréquences), ont été proposés pour contourner ce problème. Parmi ces représentations ; les coefficients LSF (Line Spectral Frequencies).

Le concept d'un LSF à été introduit pour la première fois par Itakura [19]. Leur efficacité est due principalement à : [2]

- La relation étroite qui existe entre les LSF et les fréquences des formants.
- La sensibilité spectrale localisée des LSF, en d'autres termes un changement d'un LSF donné produit une variation du spectre LPC uniquement dans son voisinage.

IV.2. Extraction des paramètres LSF :

IV.2.1. Les conditions d'analyse :

Les considérations à prendre pour le choix des conditions d'analyse par prédiction linéaire sont les suivantes [2,20,22] :

- La méthode d'analyse.
- L'ordre de prédiction m .
- La longueur de la fenêtre de mise à jour.

On a vu dans le chapitre de la prédiction linéaire qu'on peut obtenir les coefficients de prédiction par l'algorithme WLD.

La seconde variable à prendre en compte dans l'analyse LPC est la détermination de l'ordre de prédiction m . le nombre de pôles, ou nombre de coefficients du prédicteur permet de représenter correctement toute séquence de signal de parole. Il est en fonction du nombre de résonances, d'antirésonances du conduit vocal dans la bande de fréquence considérée, de la

nature de la fonction d'excitation ainsi que des caractéristiques du rayonnement au niveau des lèvres. Plusieurs méthodes permettent de calculer approximativement la valeur de m .

Pour donner une représentation correcte des pôles de la fonction de transfert du conduit vocal, la durée de mémorisation du prédicteur linéaire doit correspondre à deux fois le temps mis par l'onde de parole pour se propager depuis la glotte jusqu'aux lèvres. Ainsi, par exemple, si le conduit vocal a une longueur moyenne de 17 cm, le temps équivalent au nombre d'échantillons contenus dans la mémoire doit être égale à 1 ms. Ceci correspond au double du temps mis par l'onde sonore à la vitesse de 340 m/s, pour traverser le conduit vocal.

A la fréquence d'échantillonnage de 8 KHz, la valeur correspondante de m doit être égale au moins à 8 pour un modèle idéal. Et pour une fréquence de 10 KHz, la valeur moyenne du m est 12.

La dégradation de la qualité de la parole devient significative lorsque la valeur de m est réduite à 8 et on a une saturation lorsque m est supérieure à 12. Le choix de $m = 10$ est raisonnable pour une représentation adéquate du conduit vocal.

La troisième variable à prendre en compte est l'intervalle de mise à jour. Comme l'ordre de prédiction m , le choix de l'intervalle L présente un compromis entre la qualité et le débit binaire.

Il est généralement souhaitable d'effectuer l'analyse spectrale dans un intervalle où le mouvement du conduit vocal est négligeable. En codage à bas débit, un intervalle de 20 ms est suffisant pour maintenir une bonne qualité de la parole, cependant ceci introduit une petite dégradation dans quelques sons possédant des caractéristiques spectrales très changeantes.

VI.2.2. Constitution des trames :

La parole est souvent décomposée par analyse LPC en plusieurs paramètres que l'on code à des fins de transmissions, de rehaussement ou autres.

Pour obtenir les trames de la parole à partir d'un signal continu, une méthode couramment employée consiste à le multiplier par une fenêtre appelée "fonction de pondération". La pondération effectuée sur le signal annule celui-ci pour les échantillons situés en dehors de l'intervalle d'analyse.

Le choix de la fonction de pondération est important dans le domaine spectral, en effet, lorsqu'on multiplie le signal par une fenêtre $w(n)$, le spectre résultant se trouve modifié ; dans le domaine fréquentiel le produit simple se transforme en un produit de convolution.

Soit le signal de parole $S(n)$ et $s(f)$ son spectre ; donc : $S_f(f) = S(f) W(f)$.

Pour un spectre d'impulsion de Dirac : $W(f) = \delta(f) \Rightarrow W(n) = 1 \quad \forall n$.

Le produit de convolution donne :

$$S_f(f) = S(f)$$

Le spectre résultant S_f ne subit alors aucune déformation.

Une fonction de pondération sera d'autant meilleure que sa réponse fréquentielle ressemblera davantage à Dirac.

En parole, la fonction de pondération la plus utilisée est celle de Hamming. La longueur de la fenêtre est fixée à 160 échantillons, convenant aux longueurs typiques de quasi-stationnarité pour l'analyse de WLD.

IV.3. Le filtre de synthèse :

L'utilisation directe des paramètres a_i pour la représentation du filtre LPC n'est pas recommandée. Cela, comme on a mentionné déjà, est dû au fait que la quantification de ces paramètres peut introduire des instabilités du filtre de synthèse [19]. Selon Sugamura et Itakura, la synthèse LSP présente une sensibilité plus faible que la synthèse LPC à la quantification des paramètres, l'interpolation introduit aussi moins de distorsion.

IV.3.1. Calcul des LSF :

En analyse par prédiction linéaire, un segment de parole est supposé être généré comme sortie d'un filtre tous pôles $H(z) = 1/A(z)$; où $A(z)$ est un polynôme en z appelé le filtre inverse dont l'expression est donnée par [2,20,24] :

$$A(z) = 1 + a_1 z^{-1} + \dots + a_m z^{-m} \quad (\text{IV.1.a})$$

Les coefficients a_i sont les coefficients de prédiction du filtre, et m l'ordre de prédiction.

Le filtre inverse $A(z)$ associé au filtre de synthèse $H(z)$ satisfait à la relation de récurrence suivante[25] :

$$A_n(z) = A_{n-1}(z) - K_n z^{-n} A_{n-1}(z^{-1}) \quad , n = 1, \dots, p \quad (\text{IV.1.b})$$

avec $A_0(z)=1$, et K_n est le $n^{\text{ième}}$ coefficient de réflexion. Ces coefficients appelés aussi coefficients PARCOR (Partial Corrélation), sont interprétés comme les limites du modèle du tube acoustique. En étendant l'ordre du filtre à $n = p+1$, l'équation (IV.1.b) devient :

$$A_{p+1}(z) = A_p(z) - K_{p+1} z^{-(p+1)} A_p(z^{-1}) \quad (\text{IV.2})$$

On considère les deux conditions limites artificielles $K_{p+1} = 1$ et $K_{p+1} = -1$, qui correspondent, respectivement, à l'ouverture complète et la fermeture complète au niveau de la glotte dans le modèle du tube acoustique. Sous ces conditions, nous obtiendrons les deux polynômes suivants :

$$F_1(z) = A_p(z) + z^{-(p+1)} A_p(z^{-1}) = 1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_2 z^{-(p-1)} + a_1 z^{-p} + z^{-p} \quad (\text{IV.3})$$

pour $K_{p+1} = -1$, et Pour $K_{p+1} = 1$:

$$F_2(z) = A_p(z) - z^{-(p+1)} A_p(z^{-1}) = 1 + a_1 z^{-1} + a_2 z^{-2} + \dots - a_2 z^{-(p-1)} - a_1 z^{-p} + z^{-p} \quad (\text{IV.4})$$

Les polynômes $F_1(z)$ et $F_2(z)$ possèdent les propriétés importantes suivantes : [20]

- 1- Toutes les racines de $F_1(z)$ et $F_2(z)$ sont sur le cercle unité.
- 2- Les racines de $F_1(z)$ et $F_2(z)$ alternent sur le cercle unité.
- 3- $A(z)$ est de phase minimum (la stabilité de $H(z)$ est facilement préservée après la quantification des racines de $F_1(z)$ et $F_2(z)$).

Comme les racines de $F_1(z)$ et $F_2(z)$ sont sur le cercle unité, elles sont données par e^{jw_i} , et il est facile à montrer que pour le cas d'un ordre de prédiction p pair, $F_1(z)$ et $F_2(z)$ sont données par :

$$F_1(z) = (1 + z^{-1}) \prod_{i=1,3,\dots,p-1} (1 - 2 \cos(2\pi f_i) z^{-1} + z^{-2}) \quad (\text{IV.5})$$

$$F_2(z) = (1 - z^{-1}) \prod_{i=2,4,\dots,p} (1 - 2 \cos(2\pi f_i) z^{-1} + z^{-2}) \quad (\text{IV.6})$$

Les fréquences f_i , qui correspondent aux racines des polynômes $F_1(z)$ et $F_2(z)$, dans les équations (IV.5) et (IV.6) sont normalisées par la fréquence d'échantillonnage f_s . Les paramètres f_i , $i = 1, \dots, m$, sont appelés les paires de raies spectrales LSP (line spectrum pairs) ou LSF (line spectrum fréquences). Il est important de noter que $f_0 = 0$ et $f_{p+1} = 0.5$ sont des racines fixes correspondant à $z = 1$ et $z = -1$ respectivement. Ils sont donc exclus de l'ensemble des paramètres LSF nécessaires pour caractériser le filtre de synthèse. Les LSF peuvent être interprétés comme les fréquences de résonances du conduit vocal sous les deux conditions limites au niveau de la glotte (ouverture complète ou fermeture complète). Une deuxième propriété des paramètres LSF peut être donnée sous la forme suivante :

$$f_0 < f_1 < f_2 < \dots < f_{p-1} < f_p < f_{p+1} \quad (\text{IV.7})$$

Avec $f_0 = 0$ et $f_{p+1} = 0.5$. Cette relation est connue sous le nom de propriété d'ordonnement des LSF. Aussi bien que cette propriété est conservée, la stabilité de $H(z)$ est assurée.

Les paramètres LSF se prêtent mieux à la quantification que les autres représentations du filtre LPC à cause des propriétés suivantes :

1- les LSF ont de bonnes propriétés statistiques, et la stabilité du filtre de synthèse est assurée par la préservation de la propriété d'ordonnement. En plus, cette propriété permet la détection des erreurs de transmissions des LSF sans introduire de redondance.

2- il y a une relation évidente entre les LSF et le spectre du filtre LPC. Une concentration des LSF dans une certaine bande de fréquences correspond approximativement à une résonance dans cette bande.

3- Les LSF entre deux fenêtres d'analyse adjacentes sont fortement corrélés.

IV.3.2. Algorithme de Calcul des LSF :

Les deux polynômes $F_1(z)$ et $F_2(z)$, qui sont symétrique et antisymétrique respectivement, ont pour racines fixes $z=1$ et $z=-1$ [20]. Pour un ordre de prédiction pair, ces deux racines peuvent être enlever par division polynomiale. Ceci donne :

$$G_1(z) = F_1(z)/(1+z^{-1}) \quad \text{et} \quad G_2(z) = F_2(z)/(1-z^{-1}) \quad (\text{IV.8})$$

Les résultants $G_1(z)$ et $G_2(z)$ sont des polynômes symétriques pour un ordre p pair. Comme les racines n'apparaissent que par paires complexes conjuguées, il est possible de ne déterminer que ceux situées sur la moitié supérieure du cercle unité. Les racines en question sont e^{jw_i} pour $i=1,2,\dots,p$. Les LSF sont les positions angulaires des racines, $0 < w_i < 0.5$. En faisant le changement de variable suivant : $M=p/2$, on peut exprimer sous forme explicite la symétrie des polynômes $G_i(z)$ comme suit :

$$G_i(z) = 1 + g_i(1)z^{-1} + \dots + g_i(1)z^{-(2M-1)} + z^{-2M} \quad ; i=1,2 \quad (\text{IV.9})$$

En enlevant le terme de la phase linéaire, nous aurons deux développements en séries de cosinus à phase nulle :

$$G_i(e^{jw}) = e^{-jwM} G_i'(w) \quad (\text{IV.10})$$

Avec:

$$G_i'(w) = 2 \cos Mw + 2g_i(1) \cos(M-1)w + \dots + 2g_i(M-1)\cos w + g_i(M) \quad (\text{IV.11})$$

En faisant le changement de variable $x = \cos w$, on obtient :

$$\cos mw = T_m(x) \quad (\text{IV.12})$$

Où $T_m(x)$ est un polynôme de Chebychev d'ordre m . Les polynômes de Chebychev vérifient la récurrence :

$$T_k(x) = 2x T_{k-1}(x) - T_{k-2}(x) \quad (\text{IV.13})$$

Avec les conditions initiales : $T_0(x) = 1$ et $T_1(x) = x$. L'équation (IV.11), en utilisant le développement en polynômes de chebychev, devient :

$$G_i'(w) = 2 T_M(x) + 2g_i(1) T_{M-1}(x) + \dots + 2 g_i(M-1) T_1(x) + g_i(M) \quad (IV.14)$$

Lorsque les racines x_i de $G_1'(x)$ et $G_2'(x)$ sont déterminées, les LSF correspondants sont donnés par $w_i = \arccos(x_i)$. Le changement de variable $x = \cos w$ transforme la moitié supérieure du cercle unité, dans le plan z , dans l'intervalle $[-1,+1]$, sur la droite des x . Ainsi tout les x_i sont situées entre -1 et 1 , avec la racine correspondante à la plus faible fréquence située plus près de 1 .

Soit la série en polynôme de Chebychev :

$$Y(x) = \sum_{k=0}^{N-1} c_k T_k(x) \quad (IV.15)$$

Considérons la récurrence suivante :

$$b_k(x) = 2.x b_{k+1}(x) - b_{k+2}(x) + c_k \quad (IV.16)$$

avec les conditions initiales $b_N(x) = b_{N+1}(x) = 0$. Cette récurrence est utilisée pour calculer $b_0(x)$ et $b_2(x)$. Après, $Y(x)$ peut être exprimée en termes de $b_0(x)$ et $b_2(x)$ par la relation:

$$Y(x) = \sum_{k=0}^{N-1} [b_k(x) - 2.x b_{k+1}(x) + b_{k+2}(x)] T_k(x) = (b_0(x) + b_2(x) + c_0)/2 \quad (IV.17)$$

L'avantage de cette formulation est que les erreurs d'évaluation de $b_0(x)$ et $b_2(x)$ tendent à s'annuler. En se rappelant la propriété d'entrelacement et sachant que la plus proche racine de $x = 1$ correspond au polynôme $G_1'(x)$, la procédure d'extraction des x_i peut être abordée :

$$x_0 = x_1 = 1, i=1$$

Pour $j = 1$ à P

1- Evaluer $G_i'(x_0)$ utilisant IV.16 et IV.17

2- $x_1 = x_0 + \Delta x$

3- Evaluer $G_i'(x_1)$ utilisant IV.16 et IV.17

4- Si $G_i'(x_0) \cdot G_i'(x_1) > 0$ aller à 2

- 5- Recherche de la solution précise x_j par divisions successives de $[x_0, x_1]$
- 6- Si $i = 1$ alors $i=2$ sinon $i = 1$
- 7- $x_0 = x_1$
- 8- aller à 2

Fin .

Enfin, il faut noter que le pas Δx doit être assez faible pour pouvoir détecter toutes les racines, mais non pas trop afin d'accélérer l'algorithme de résolution.

Dans un système de compression de parole basé sur le codage des LSF, ces derniers doivent être quantifiés. Donc, pour garantir la stabilité du filtre de synthèse, on doit s'assurer que les versions quantifiées des LSF vérifient toujours la propriété d'ordonnement. Ceci représente un résultat très important dans le concept des quantificateurs basés sur les paramètres LSF.

Remarque :

La conversion des LSF aux coefficients de prédiction linéaire est réversible de nature. Le nombre d'opérations à calculer pour la conversion des LSF en coefficients de prédiction linéaire est moins important que celui de l'extraction des LSF à partir des coefficients de prédiction.

Chapitre V

LA QUANTIFICATION DES PARAMETRES LSF

RESULTATS ET SIMULATION

V.1. Introduction :

L'objectif primordial de la quantification des paramètres LSF est la réduction, lors de la transmission des trames d'information, du nombre de bits attribués aux LSF. Les LSF nécessitent l'emploi de bons quantificateurs capables de minimiser la distorsion de quantification, de réaliser une compression maximale possible et de conserver la propriété d'ordonnement de ces paramètres quantifiés. Ce qui améliore les performances du codeur.

Cette étude va tenter de servir à réduire le nombre de bits lors de la quantification des LSF tout en gardant une bonne qualité du signal transmis en utilisant la méthode de la quantification vectorielle codée par treillis (TCVQ).

V.2. Mesure de la distance entre les LSF :

La sélection d'une mesure de distorsion est un choix important dans le design et l'opération de la quantification vectorielle codée par treillis.

Comme on a mentionné dans les chapitres précédents, la distance euclidienne est largement utilisée à cause de sa facilité de mettre en œuvre et vu son interprétation directe comme une puissance d'erreur de quantification et sa propriété intéressante. Lors du design du dictionnaire, les valeurs de la base d'apprentissage sont associées à leurs centroïdes et chaque centroïde est représenté par un point dont la somme des distances avec les vecteurs de la classe considérée est minimale.

La distance de Atal [30] consiste à pondérer un LSP en fonction de la densité spectrale du filtre LPC selon la fréquence du LSF. Cette distance est donnée par la formule suivante :

$$d(\text{LSF}, \text{LSFQ}) = \sum_{i=1}^{10} [c_i \omega_i (\text{LSF}_i - \text{LSFQ}_i)]^2 \quad (\text{V.2})$$

tel que :

ω_i : est le poids associé à la $i^{\text{ème}}$ LSF, et il est donné par :

$$\omega_i = [P(\text{LSF}_i)]^r \tag{V.3}$$

c_i : est un poids additionnel associé à la $i^{\text{ème}}$ LSF, les valeurs de $\{c_i\}$ sont données par :

$$\begin{cases} 1.0 & \text{pour } 1 \leq i \leq 8 \\ 0.8 & \text{pour } i = 9 \\ 0.4 & \text{pour } i = 10. \end{cases} \tag{V.4}$$

Où :

$P(\text{LSF}_i)$: est puissance spectrale du LPC.

r : est une constante empirique pour contrôler le poids relatif donné au différent LSF et ils ont trouvé que $r = 0.15$ est une valeur satisfaisante.

V.3. Mesure des performances de la quantification des LSF :

La mesure de la distorsion spectrale est toujours utilisée pour évaluer les performances de codage des paramètres LSP.

La distorsion spectrale moyenne entre un vecteur LSF et un vecteur LSF quantifié est donné par la formule suivante :

$$SD = \frac{1}{N_f} \sum_{n=1}^{N_f} \left(\frac{100}{\pi} \int_0^\pi \left[\text{Log} |A_n(e^{j\omega})|^2 - \text{Log} |\tilde{A}_n(e^{j\omega})|^2 \right]^2 d\omega \right)^{1/2} \tag{V.5}$$

Où : SD est exprimée en dB.

N_f : est le nombre total des fenêtres.

$A_n(e^{j\omega})$ et $\tilde{A}_n(e^{j\omega})$ sont les spectres de la $n^{\text{ème}}$ trame de parole sans et avec quantification respectivement.

La mesure de la SD est connue d'avoir une bonne correspondance avec les mesures subjectives.

Les conditions citées ci-dessous sont nécessaires pour avoir une bonne qualité de la parole synthétisée ; c'est à dire, la quantification LPC n'introduit aucune distorsion audible dans le signal parole. En d'autre terme, les deux versions de la parole codée obtenues par les paramètres de prédiction linéaire quantifiés et les paramètres non quantifiés ne se distinguent pratiquement pas à l'écoute. Telle est l'exigence pour un bon codeur.

Pour assurer la bonne qualité de l'information, il est recommandé de respecter les trois conditions suivantes [30] :

- La distorsion spectrale doit être au autour de 1 dB.
- Le pourcentage des trames ayant une distorsion spectrale entre 2 et 4 dB doit être inférieur à 2%.
- Aucun pourcentage n'est permis pour une distorsion spectrale supérieure à 4 dB.

V.4.Simulations et résultats :

V.4.1.Introduction :

Dans cette partie nous reprenons et expliquons les résultats des différents essais effectués sur notre quantificateur et nous interprétons les performances obtenues.

Nous présentons une étude comparative entre les résultats obtenus par la TCVQ et d'autres quantificateurs.

Dans un premier lieu, on définit les deux structures étudiées.

En deuxième lieu, on fait une comparaison entre ces deux structures.

V.4.2.Les conditions d'analyse :

Méthode d'analyse	Méthode d'autocorrelation
Ordre de prédiction	$m = 10$
Fréquence d'échantillonnage	$f_s = 8000$ Hz
Fenêtre d'analyse	10 ms
Fenêtre de pondération	Fenêtre de Hamming
débordement la fenêtre de pondération de chaque côté de la fenêtre d'analyse	40 échantillons
Longueur de la séquence d'apprentissage	13800 trames
La base des LSF utilisée	138 000

Tableau V.1. conditions d'analyse

V.4.3.Simulation des différents codeurs :

Ayant un ordre de prédiction égal dix, la dimension du vecteur LSF pour chaque trame est $L=10$. Ce vecteur est noté $\{\omega_1, \omega_2, \dots, \omega_L\}$. Pour une utilisation de ce type de codeur dans des applications en temps réel (faible temps de retard), le codage par treillis est utilisé en mode bloc où chaque bloc correspond à un vecteur LSF. La longueur du chemin du treillis est inférieure ou égale à la dimension des vecteurs LSF.

Le vecteur de 10 LSF, représentant chaque trame d'analyse, n'est pas quantifié directement mais divisé en plusieurs sous vecteurs. Selon le nombre et la taille des sous vecteurs résultants, on peut avoir plusieurs structures. Pour notre simulation, on a utilisé deux structures principales à savoir :

- Structure 2 2 2 2 2 : les 10 LSF sont divisés en cinq sous vecteurs de dimension 2.
- Structure 3 3 4 : les 10 LSF sont divisés en trois sous vecteurs de dimension 3, 3 et 4 respectivement.

Pour chaque structure, on a varié le nombre de bit alloué à chaque sous vecteur, obtenant ainsi plusieurs codeurs à des débits variables.

Sachant que les paramètres LSF possèdent des propriétés statistiques différentes (moyenne et variance), nous avons utilisé des dictionnaires différents pour chaque sous vecteur, ainsi à chaque transition du treillis, un sous vecteur est codé en utilisant son propre dictionnaire. A la fin du codage, on transmet le chemin optimal correspondant au vecteur LSF codé et l'état initial du treillis.

Chaque dictionnaire correspondant à un sous vecteur est obtenu en appliquant l'algorithme décrit dans le paragraphe (§ IV.3.4.1) et on le partitionne selon l'algorithme de partition donné dans le paragraphe (§ IV.3.4.2) puis optimisé par l'algorithme de Lloyd généralisé présenté au paragraphe (§ IV.3.4.4).

Pour chaque débit de codage, on attribue trois bits à l'état initial pour un treillis de huit états, où le reste des bits est réservé à la représentation des dictionnaires. Sachant que les LSF d'ordre supérieur ont peu d'influence sur la distorsion spectrale, les dictionnaires des derniers LSF ont des tailles moins importantes que les premiers.

V.4.3.1.Codeur 2 2 2 2 2 :*Description :*

Comme on a mentionné précédemment, le codeur quantifie le vecteur LSF ($L=10$) en cinq étapes de transitions du treillis deux à deux tel que chaque transition a son propre dictionnaire.

Les résultats de codage obtenus de deux différents fichiers parole sont cités ci-dessous :

Débit (Bits/trame)	Allocation de bits de chaque sous vecteur	Distorsion spectrale moyenne (SD) (dB)	Pourcentage des trames ayant une SD entre 2-4 dB	Pourcentage des trames ayant une SD supérieur à 4 dB
25	4 4 3 3 3	2.397	49.113 %	6.139 %
26	4 4 4 3 3	1.352	10.914%	0.000%
28	5 4 4 4 3	1.200	4.774%	0.000%
29	5 5 4 4 3	1.165	3.819%	0.000%
30	5 5 5 4 3	1.069	1.227%	0.000%
31	5 5 5 4 4	1.005	0.818%	0.000%
32	5 5 5 5 4	0.940	0.272%	0.000%

Tableau V.2. performance en terme de distorsion spectrale moyenne SD pour le premier fichier.

Débit (Bits/trame)	Allocation de bits de chaque sous vecteur	Distorsion spectrale moyenne (SD) (dB)	Pourcentage des trames ayant une SD entre 2-4 dB	Pourcentage des trames ayant une SD supérieur à 4 dB
25	4 4 3 3 3	2.214	59.160 %	1.047 %
26	4 4 4 3 3	1.275	2.239 %	0.000%
28	5 4 4 4 3	1.135	0.740 %	0.000%
29	5 5 4 4 3	1.151	0.577 %	0.000%
30	5 5 5 4 3	1.088	0.740 %	0.000%
31	5 5 5 4 4	0.987	0.126 %	0.000%
32	5 5 5 5 4	0.944	0.090 %	0.000%

Tableau V.3. performance en terme de distorsion spectrale moyenne SD pour le deuxième fichier.

On remarque d’après les deux tableaux que pour un débit de 31 bits/trame, on a une distorsion spectrale moyenne d’environ 1 dB. Il faut noter que la distorsion diminue en augmentant le débit.

On peut voir que le codeur 2 2 2 2 2 présente des meilleures performances à 26 bits/trame comparée au quantificateur scalaire utilisé par U.S. Federal (codeur CELP à 4.8 Kbits/seconde) qui a les performances suivantes [30]:

Débit (Bits/trame)	Distorsion spectrale moyenne (SD) (dB)	Pourcentage des trames ayant une SD entre 2-4 dB	Pourcentage des trames ayant une SD > 4 dB
34	1.45	11.16 %	0.010 %

Tableau V.4. performance en terme de distorsion spectrale moyenne SD pour le quantificateur scalaire (U.S.Federal)[].

Nous avons testé également le déroulement de l’algorithme de construction du dictionnaire initial de 16 points pour une séquence d’apprentissage de 256 points. Le résultat est représenté par la figure (V.1).

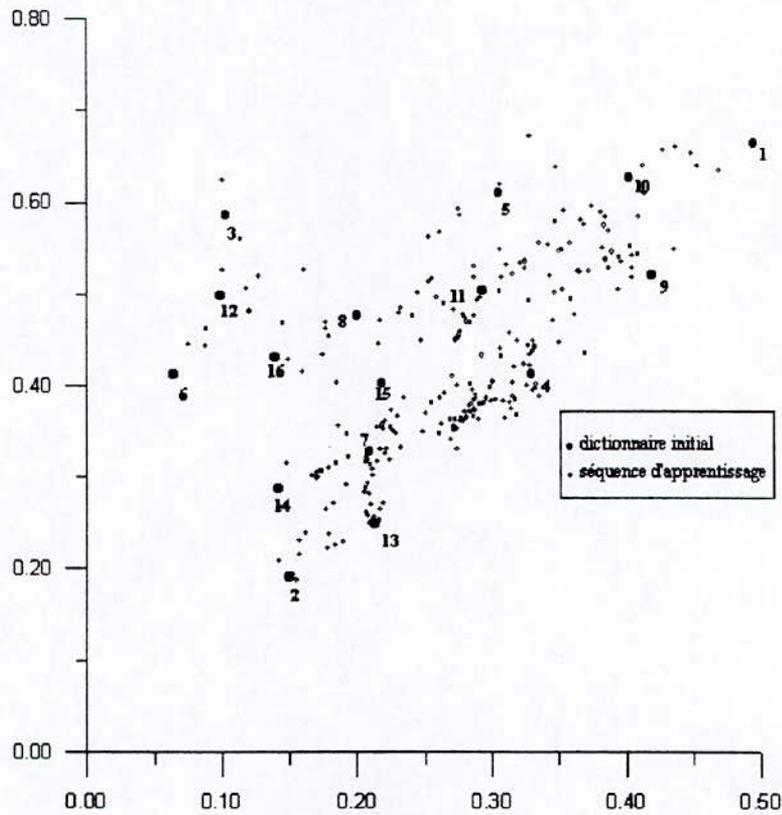


FIG V.1. génération dictionnaire initial de 16 points

La figure ci-dessous donne l'erreur d'optimisation en fonction du nombre d'itération pour différents débits :

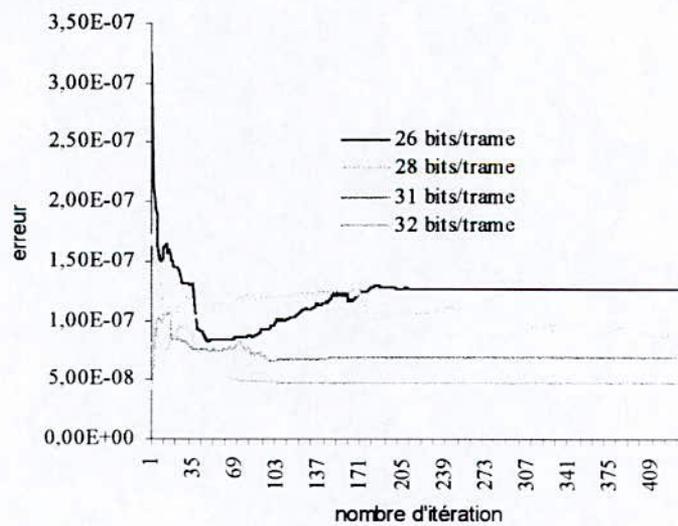


FIG V.2. la variation de l'erreur d'optimisation en fonction du nombres d'itérations

On remarque que l'erreur finale d'optimisation diminue quand on augmente le débit du codage. Il faut noter que L'erreur se stabilise après un certain nombre d'itérations.

Les figures (V.3), (V.4) donnent une comparaison entre les LSF originaux et les LSF quantifiés d'une trame pour différents débits.

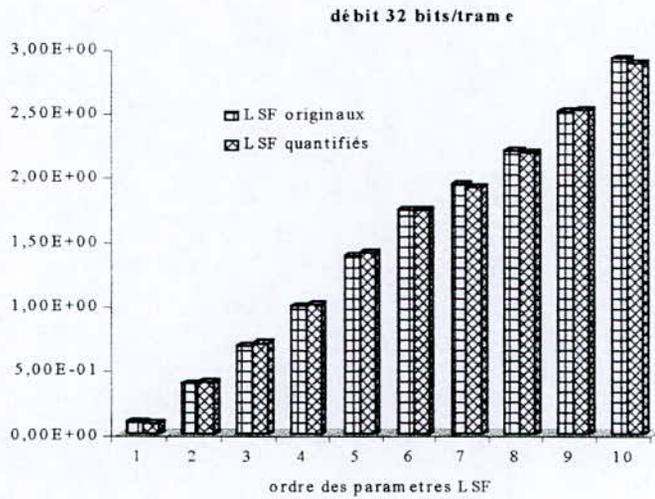


FIG V.3. comparaison entre les LSF et les LSF quantifiés pour un débit de 32 bits/trame

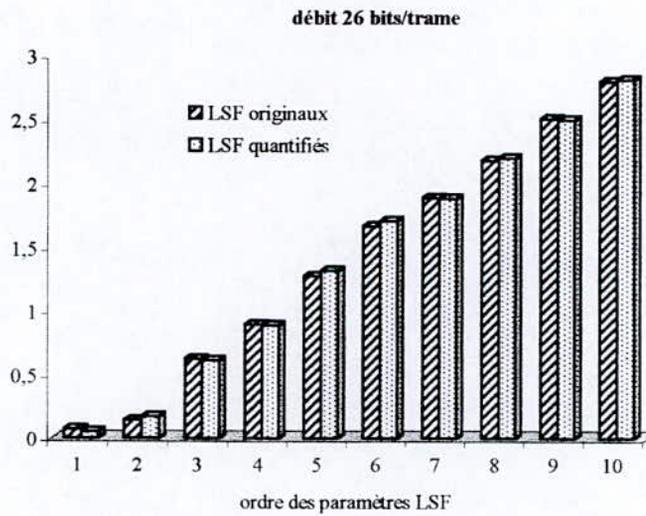


FIG V. 4. comparaison entre les LSF et les LSF quantifiés pour un débit de 26 bits/trame

Pour mieux voir la différence entre les LSF et les LSF quantifiés, la figure (V.5) représente les points (LSF, LSFQ) comparés à la bissectrice.

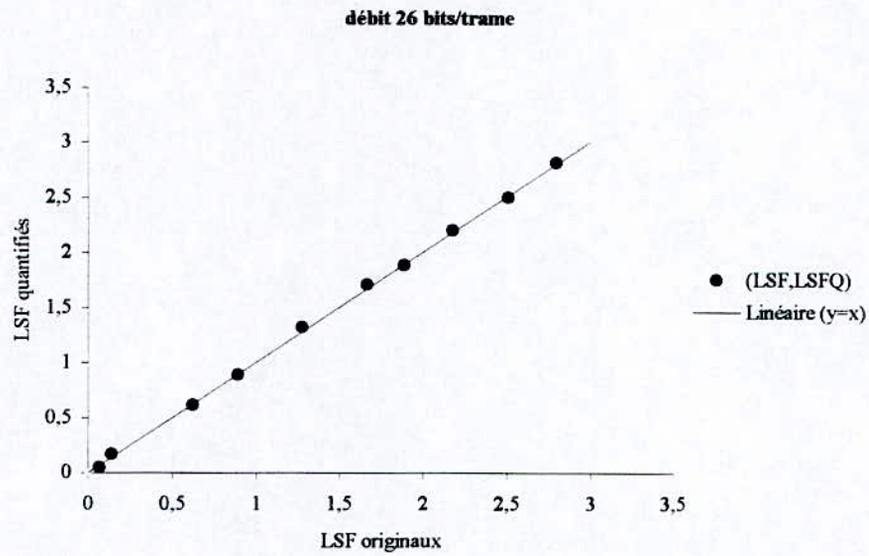


FIG V. 5. comparaison entre les LSF et les LSF quantifiés pour un débit de 26 bits/trame par rapport au bissectrice.

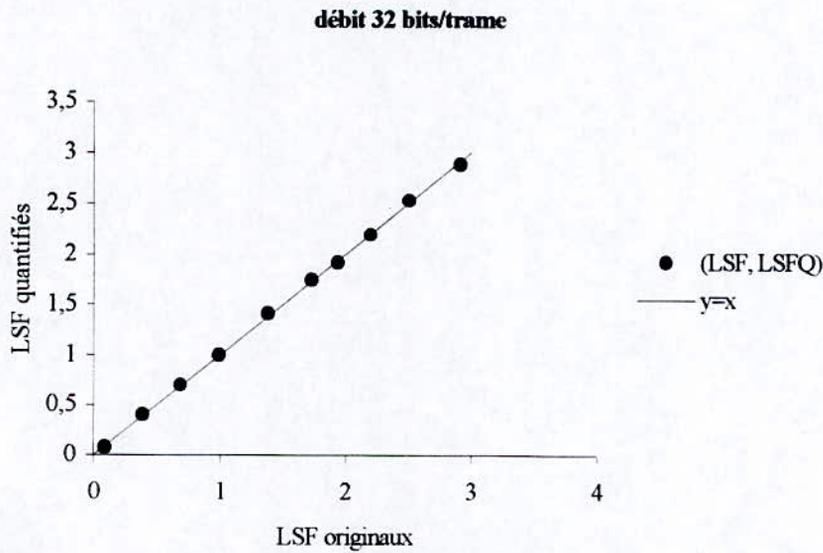


FIG V. 6. comparaison entre les LSF et les LSF quantifiés pour un débit de 32 bits/trame par rapport au bissectrice.

On remarque que la différence entre les LSF et les LSFQ est satisfaisante.

V.4.3.2.Codeur 3 3 4 :

Dans ce codeur, on quantifie deux trames à la fois sur six étapes de transitions du treillis, c'est à dire que les 20 LSF sont décomposés comme suit :

- Trois LSF sont quantifiés dans la 1^{ère}, 2^{ème}, 4^{ème} et la 5^{ème} étape.
- Quatre LSF sont quantifiés dans la 3^{ème} et la 6^{ème} étape.

Le résultat obtenu est représenté par le tableau ci-dessous pour deux signaux de parole avec un débit de 22.5 bits/trame :

Fichier parole	Allocation de bits de chaque sous vecteur	Distorsion spectrale moyenne (SD) (dB)	Pourcentage des trames ayant une SD entre 2-4 dB	Pourcentage des trames ayant une SD supérieur à 4 dB
1 ^{er} fichier	6 6 6 6 6 6	1.352	1.909 %	0.000 %
2 ^{ème} fichier	6 6 6 6 6 6	1.321	0.632 %	0.000 %

Tableau V.5. performance en terme de distorsion spectrale moyenne SD pour deux fichiers parole.

D'après le tableau et les résultats trouvés précédemment, on remarque que ce codeur à un débit égal à 22.5 bits/trame est comparable en terme de performance à ce du premier codeur à un débit de 26 bits/trame et surtout à 25 bits/trame.

Conclusion :

D'après les résultats obtenus par les deux structure on constate qu'on peut obtenir une distorsion spectrale minimale et une meilleure quantification dans la structure 3 3 4 . On a trouvé que la deuxième structure, à un débit de 22.5 bits/trame, est comparable en performance à la première structure à un débit de 26 bits /trame.

La comparaison des résultats du tableau 5 avec ceux des tableaux 2 et 3 pour le premier quantificateur a permet de sauver de 3.5 à 4.5 bits/trame. Aussi, il faut noter que le temps de retard de la première structure est inférieur à celui de la deuxième.

On peut conclure que la quantification par la structure 3 3 4, présente des meilleures performances.

Conclusion :

Dans ce travail, nous avons mis l'accent sur la définition d'une nouvelle technique de quantification des signaux à savoir : " la quantification vectorielle codée par treillis (TCVQ)". Cette technique permet de palier les insuffisances de la quantification scalaire codée par treillis (TCQ) qui a fait l'objet d'une étude détaillée.

Pour la quantification des paramètres LSF on a utilisé la méthode de TCVQ, et on a introduit plusieurs structures de codage pour mieux comparer les résultats obtenus.

Nous avons présentés l'algorithme qui permet l'obtention des bons quantificateurs (optimaux), ainsi que la méthode de partition des dictionnaires.

L'objectif principal de notre travail concerne la quantification des paramètres LSF. Nous avons été amenés à faire l'extraction de ces paramètres à partir des coefficients de la prédiction linéaire.

La distorsion utilisée dans les calculs de quantification c'est la distorsion spectrale exprimée en dB pour permettre une comparaison objective des performances des LSF. Une distorsion spectrale moyenne de 1 dB représente un niveau communément accepté pour une bonne reproduction de la parole reconstruite.

Nous avons élaborés le programme de construction du dictionnaire initial, le programme d'optimisation, le programme de partition ainsi que le programme de déroulement du treillis. Ce dernier programme permet d'obtenir le chemin optimal, représentant le mieux les vecteurs à quantifier, parmi tous les chemins possibles dans le treillis en comparant les erreurs cumulées de tous les chemins à la fin du treillis.

La connaissance du chemin optimal permet d'obtenir les LSF quantifiés les plus proches aux LSF originaux au sens de la distance euclidienne.

On a utilisé deux structures principales pour la simulation des codeurs, en variant le débit de chaque structure, qui sont :

- La structure 2 2 2 2 2, c'est à dire on a quantifié les 10 LSF sur cinq étapes de transitions du treillis.
- La structure 3 3 4, on a quantifié deux trames à la fois par un treillis de six étapes, c'est à dire 3 LSF pour la première, la deuxième, la quatrième et la cinquième étape du treillis, et 4 LSF pour la troisième et la dernière étape.

L'utilisation de la première structure a permis d'obtenir une distorsion spectrale de 1.06 dB à 30 bits/trame. La variation du nombre de bit de 26 à 32 bits/trame, en variant

l'allocation des bits, a permis d'avoir un intervalle de 1.38 à 0.94 dB pour différents signaux de parole.

La quantification par la deuxième structure optimisée par une base d'apprentissage composée de 13800 trames, en codant deux trames à la fois par un treillis de six étapes, a permis d'obtenir une distorsion spectrale égale à 1.35 dB pour un nombre de bit égal à 22.5 bits/trame.

La comparaison des résultats obtenus par les deux structures a permis de constater qu'on peut obtenir une distorsion spectrale minimale et une meilleure quantification dans la structure 3 3 4 . On a trouvé que la deuxième structure ,à un débit de 22.5 bits/trame, est comparable en performance à la première structure à un débit de 26 bits /trame.

Donc la structure 3 3 4 permet de sauver 3.5 à 4.5bits/trame par rapport à la structure 2 2 2 2 . Aussi, il faut noter que le temps de retard de la première structure est inférieur à celui de la deuxième.

Annexe A

A.I. Algorithme d'apprentissage LBG :

Les algorithmes qui existent jusqu'à présent sont restreints à un nombre réduit à quelques variantes près, c'est par exemple l'algorithme de la k moyenne connu généralement sous le nom du LBG(nom des auteurs : Lynde, Buzo, et Gray) .

A.I.1. Présentation de l'algorithme LBG :

Cet algorithme itératif propose une fonction qui transforme les représentants au centre d'une partition pour obtenir une nouvelle partition dont l'erreur est inférieure ou égale à celle de la partition originale.

Pour son utilisation, on se donne un alphabet initial construit soit aléatoirement, en tenant compte de la distribution des vecteurs de la séquence, ou en prenant quelques vecteurs de la séquence, ou bien quelques vecteurs ayant la plus grande distance entre eux (en terme de distance euclidienne).

Par le calcul de la distance euclidienne, le nuage initial est partitionné en cellules ou partitions auxquelles leur est associés des centroïdes respectives, l'ensemble des centroïdes constitueront le nouveau alphabet.

Sans s'étendre sur les caractéristiques mathématiques, le LBG peut être décrit comme suit :

a- utilisation d'une séquence d'apprentissage typique de données ayant une probabilité et une fonction de densité donnée.

b- Utilisation d'un processus itératif à partir d'un dictionnaire initial et chaque itération, un dictionnaire plus performant est construit.

Les paramètres d'itérations du dictionnaire sont :

- La composition de la séquence d'apprentissage.
- La taille n de la séquence.
- Le rapport d'apprentissage $p = n/N$.
- Le choix de l'alphabet initial.

La composition de la séquence à une importance considérable car on désire toujours construire un dictionnaire universel capable de reproduire (coder) toutes données présentées à lui. La taille n de la séquence est aussi importante, car elle détermine le nombre de vecteurs appartenant à un groupe fini.

Pour un dictionnaire représentatif, un rapport p d'apprentissage minimal de 16 est nécessaire, s'il est inférieur a cette limite la source sera mal présentée. Le dernier paramètre à prendre en considération est le choix du dictionnaire initial, ce choix est généralement aléatoire ou à partir d'un échantillon de la séquence.

A.I.2. Algorithme du LBG :

On développe dans cette annexe l'algorithme LBG. Les termes suivants désignent :

- N : dimension du dictionnaire.
- K : largeur du bloc.
- n : taille de la séquence d'apprentissage.

➤ **Etape 0** : initialisation

- ε : seuil d'erreur.
- S : séquence d'apprentissage $S = \{x_i / i=0,1,\dots,N-1\}$.
 X_i est un vecteur de dimension k normalisé.
- A_0 : alphabet initial.
- D_m : distorsion moyenne ($m = 0, D_{-1} = \infty$).

➤ **Etape 1** :

- Soit $A_m = \{Y_j / j = 0, \dots, N\}$.

Cherchons les partitions $P(A_m) = \{S_j, j=0, \dots, N\}$ donnant le minimum de distorsion pour chaque Y_j avec :

$$d(X_i, Y_j) \leq d(X_i, Y_m) \quad \text{pour tout } m. \quad (\text{A.1})$$

- Calculons la distorsion moyenne D_m :

$$D_m = D \{ A_m, P(A_m) \} = \frac{1}{n} \sum_{i=0}^{n-1} \min d(X_i, Y) \quad (\text{A.2})$$

➤ **Etape 2** :

$$\text{Si : } \frac{D_{m-1} - D_m}{D_m} \leq \varepsilon \quad (\text{A.3})$$

Donc A_m trouvée est un bon quantificateur, sinon continuer.

➤ **Etape 3** :

- cherchons l'alphabet optimal :

$X(P(A_m)) = \{X(S_j) / j = 1, \dots, N\}$ pour $P(A_m)$, avec $X(S_j)$ sont les centroïdes des partitions S_j données par :

$$X(S_j) = \frac{1}{\|S_j\|} \sum_{x_i \in S_j} x_i \quad (\text{A.4})$$

$$\text{Si : } \|S_j\| = 0 :$$

faire $X(S_j) = Y_j$, ancien vecteur.

- Nouveau alphabet est $A_{m+1} = X(P(A_m))$.
- Faire $m+1$ et aller à l'étape 1.

A.3. Organigramme :

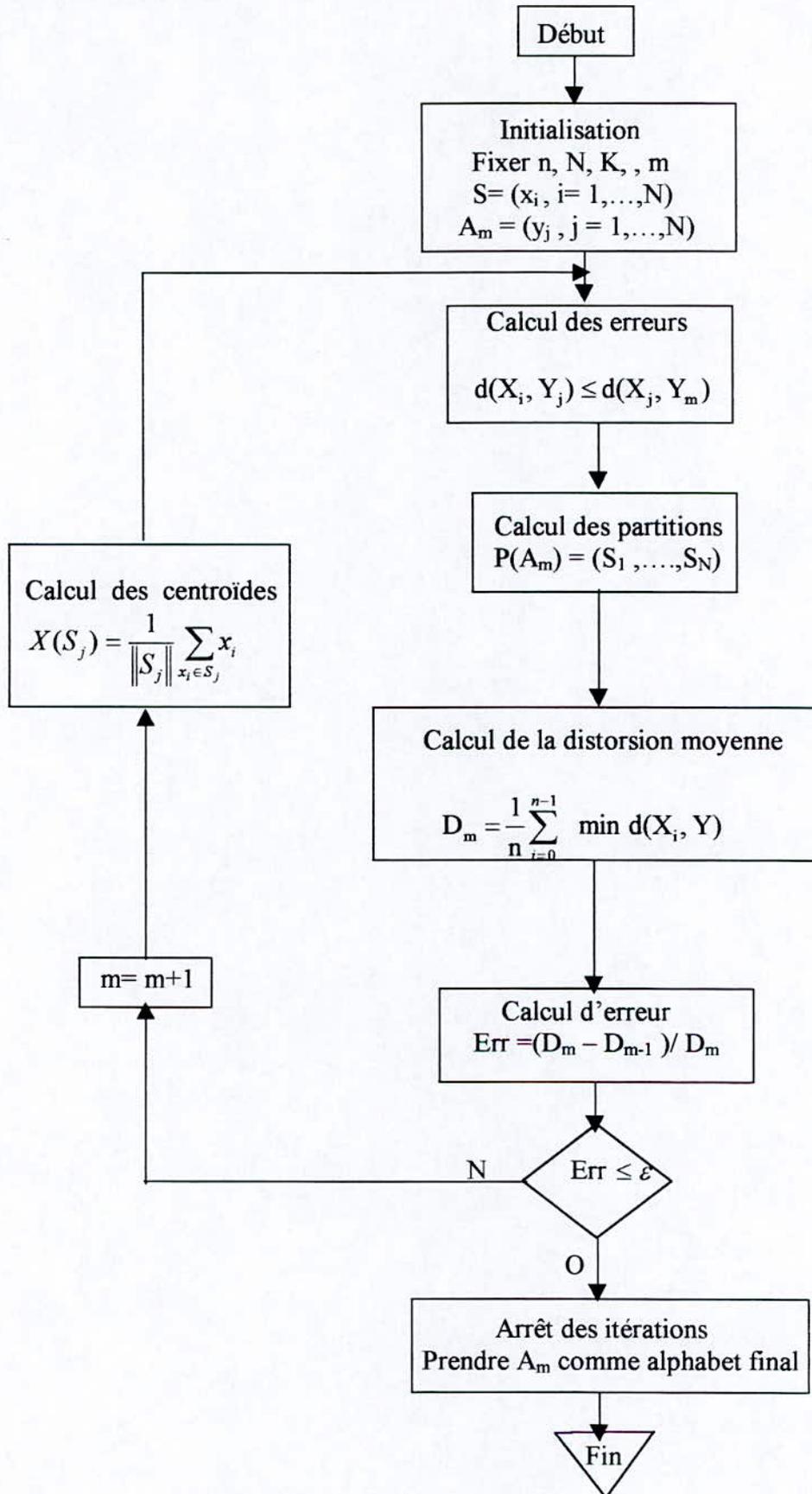


Fig A.1. Structure générale de LBG

Annexe B

Algorithme de résolution WLD :

Nous admettons que la fonction d'autocorrélation est connue pour $k=0,1,\dots,p$. la moyenne du signal est supposée nulle ; Dans le cas contraire ; elle est estimée et soustraite.

Il s'agit donc de résoudre le système matriciel, c'est à dire en fait d'inverser une matrice d'ordre p . les méthodes algébriques classiques exigent pour cela un nombre d'opérations (multiplication + addition) de l'ordre de p^3 . L'algorithme qui va être décrit profite de la structure particulière (Toeplitz symétrique) de la matrice d'autocorrélation pour résoudre le système par une récursion sur l'ordre de prédiction ; Autrement dit, ils fournissent toutes les solutions d'ordre $m = 1,2,\dots,p$, le nombre d'opérations est seulement de l'ordre de p^2 .

La variance de l'erreur de prédiction α_p sera obtenue également par une récurrence sur l'ordre m .

La procédure WLD

Rappelons que la fonction d'autocorrélation est supposée connue ; et que pour un signal stationnaire, on a :

$$R(i, j) = R(|i, j|) = R(k). \quad (\text{B.1})$$

$$\text{initialisation : } \alpha_m(0) = 1 \quad \text{et} \quad E_0 = R(0) = \sigma_x^2$$

$$\text{récursion : } \text{pour } m = 1, 2, \dots, p.$$

$$K_m = - \frac{\left[R(m) - \sum_{k=1}^{m-1} \alpha_{m-1}(k) R(m-k) \right]}{E_{m-1}} \quad (\text{B.2})$$

pour $i = 1, 2, \dots, m-1$

$$\begin{aligned} \alpha_k(m) &= \alpha_k(m-1) - k_m \alpha_{m-k}(m-1) \\ E_m &= E_{m-1} (1 - k_m^2) \end{aligned} \quad (\text{B.3})$$

les coefficients $\alpha_k(m)$ résultants, quand $m = p$, représentent les coefficients de prédiction d'un prédicteur linéaire d'ordre p :

$$a_k = \alpha_k(p) \quad k = 1, 2, \dots, p$$

La valeur de k_m jouit de la propriété : $-1 \leq k_m \leq 1$ (B.4)

La relation (B.4) est une condition nécessaire et suffisante pour que le filtre soit stable. La méthode d'autocorrélation garantit la stabilité du filtre, de plus le calcul de $R(i)$ nécessite un fenêtrage de $s(n)$ par la fenêtre de Hamming.

Annexe C

La démonstration du théorème (§ III.3.5):

D'après l'équation (III.3) et en supposant que $x(n)$ et $y(n)$ sont stationnaires et ergodiques, on a :

$$\begin{aligned}
 E[q] &= \lim_{n \rightarrow \infty} \frac{1}{2n+1} \sum_{j=-n}^n q(j) \\
 &= \lim_{n \rightarrow \infty} \frac{1}{2n+1} \sum_{j=-n}^n x(j) - y(j) \\
 &= \lim_{n \rightarrow \infty} \frac{1}{2n+1} \sum_i \left\{ \left(\sum_{x(j) \in B_i} x \right) - \|B_i\| y_i \right\} \\
 &= \sum_i \lim_{n \rightarrow \infty} \frac{\|B_i\|}{2n+1} \left\{ \left(\frac{1}{\|B_i\|} \sum_{x(j) \in B_i} x \right) - y_i \right\} = 0
 \end{aligned} \tag{C.1}$$

de l'équation (III.5), on a :

$$\begin{aligned}
 E[q^T y] &= \lim_{n \rightarrow \infty} \frac{1}{2n+1} \sum_{j=-n}^n q^T(j) y(j) \\
 &= \sum_i \lim_{n \rightarrow \infty} \frac{\|B_i\|}{2n+1} \left\{ \left(\frac{1}{\|B_i\|} \sum_{x \in B_i} x^T \right) y_i^T \right\} y_i = 0
 \end{aligned} \tag{C.2}$$

de (III.6) :

$$\frac{1}{L} E[\|q\|^2] = \lim_{n \rightarrow \infty} \frac{1}{2n+1} \sum_{j=-n}^n \frac{1}{L} \left\{ \|x(i)\|^2 + \|y(i)\|^2 + 2x^T(j)y(j) \right\} \tag{C.3}$$

en utilisant l'équation (III.3), l'équation (III.7) donne :

$$\frac{1}{L} E[\|q\|^2] = \text{var}(x) + \text{var}(y) - 2 \text{var}(y). \tag{C.4}$$

On peut démontrer le théorème autrement. La convergence de l'équation (III.3) implique qu'on peut considérer que la TCVQ introduit une probabilité conditionnelle liée à la source, tel que $P(y_i/x)$, et en même temps, une densité de probabilité $P(x/y_i)$, donc :

$$E[x] = \sum E[x/y_i]p(y_i) = \sum y_i p(y_i) = E[y] \quad (C.5)$$

Ce qui donne $E[q]=0$. les autres propriétés(III.6), (III.7) sont trouvées directement. La même démonstration est utilisée pour démontrer les propriétés analogues de l'algorithme de Lloyd généralisé utilisé dans la quantification vectorielle [28].

Bibliographie :

- [1] D. Berkani , " Design d'un quantificateur vectoriel à deux dimensions : Le SPIRAL QUANTIZER ," thèse de doctorat, ENP, 1991.
- [2] F. Merrazka, " Quantification des Paramètres LSF," thèse de magistère, ENP, 1997.
- [3] D. Berkani ,G.Turgeon, A.Chekima,B.Derras " Utilisation de l'algorithme LBG et du treillis en quantification vectorielle," AI.Journal of Technology, N° 8,pp.35-52,ENP,1992.
- [4] A. Hallali, " Quantification Vectorielle des Signaux et Codage par Treillis," Projet de Fin d'étude, ENP, 1990.
- [5] M. Djeddou, " Conception et Réalisation d'un Codeur / Décodeur de la Parole à Bande Etroite (300 – 3400 Hz), à 16 Kbit/s et à Faible Retard (< 5 ms)," thèse de magistère, ENP, 1997.
- [6] Xiaolin Wu and Lian Guan , " Acceleration of the LBG Algorithm," IEEE Transactions on Communications, VOL. 42, NO. 2/3/4, pp.1518-1523,February/ March/ April 1994.
- [7] Thomas R. Fisher, Michael W. Marcellin and Min Wang , " Trellis Coded Vector Quantization," IEEE Transactions on Information Theory, VOL. 37, NO. 6,pp 1551-1566, November, 1991.
- [8] Michael W. Marcellin and Thomas R. Fischer , " Trellis Coded Quantization of Memoryless and Gauss Markov Sources," IEEE Transactions on Communications, VOL. 38, NO. 1,pp.82-93, January 1990.
- [9] Hong Shen Wang and Nader Moayeri , " Trellis Coded Vector Quantization," IEEE Transactions on Communications, vol. 40, no. 8,pp.1273-1276, August 1992.
- [10] Michael W. Marcellin , " On Entropy Constrained Trellis Coded Quantization, " IEEE Transactions on Communications, vol. 42, no. 1,pp.14-16, January 1994.
- [11] Roch Lefebvre , " la quantification par code en trellis," Mars 1992.
- [12] Carl Rudolph Nassar and Mohammad Reza Soleymani , " Codebook Design for Trellis Quantization Using Simulated Annealing," IEEE Transactions on Speech and Audio Processing, vol. 1, no. 4,pp.400-404, October .1994.
- [13] Parthasarathy Sriram and Michael W. Marcellin , " Performance of Adaptive Prediction Algorithms for Trellis Coded Quantization of Speech," IEEE Transactions on Communications, vol. 42,no. 2/3/4,pp.1512-1517, February/ March/ April .1994.
- [14] H. Arda Aksu and Masoud Salehi , " Joint Optimization of TCQ-TCM Systems, " IEEE Transactions on Communications, vol. 44, no. 5,pp.529-533, May .1996.
- [15] Michael W. Marcellin, Thomas R. Fischer and Jerry D. Gibson , " Predictive Trellis Coded Quantization of Speech , " in Proc. IEEE ICASSP , pp. 247-250,april.1988.
- [16] Markov F. Larsen and Richard L. Frost , " Complexity constrained Trellis Quantizers.", IEEE Transactions on Information Theory, vol. 34, no. 4,pp.1134-1144, July. 1997.
- [17] G. Zou , " Structured Trellis Ebcoding," IEE Proceedings, vol. 137, Pt 1, no. 6,pp.335-344, December, 1990.
- [18] G. David Forney, JR , " The Veterbi Algorithm ," IEEE Proceedings, pp.268-278, March .1973.

- [19] F. Itakura and N. Sugamura. "LSP Speech Synthesizer, Its Principles and Implementation." Trans. Of the Committee on Speech Research. ASJ, S79, November .1979.
- [20] Peter Kabal and Ravi Prakash Ramachandran . "The Computation of Line Spectral Frequencies Using Chebychev Polynomials," IEEE Transactions on Acoustics, Speech and Signal Processing, vol. ASSP. 34, no. 6, pp.1419-1426, December .1986.
- [21] Frank K. Soong and Biing Hwang Juang . "Optimal Quantization of LSP Parameters ." IEEE Transactions on Speech and Audio Processing, vol. 1, no. 1, pp.15-23, January 1.993.
- [22] Noborn Sugamura and Nariman Farvardin . "Quantizer Design in LSP Speech Analysis and Synthesis ." in Proc. IEEE ICASSP, pp.398-401, april.1988.
- [23] Frank K. Soong and Biing Hwang Juang . "Optimal Quantization of LSP Parameters ." in Proc. IEEE ICASSP ,pp.394-397, April.1988.
- [24] Kevin T. Malone and Thomas R.Fischer , " Enumeration and Trellis Searched Coding Schemes for Speech LSP Parameters ," IEEE Transactions on Speech and Audio Processing, VOL. 1, NO. 3, pp.304-314, July .1993.
- [25] Jianping Pan and Thomas R. Fischer . " Vector Quantization of Speech Line Spectrum Pair Parameters and Reflection Coefficients ." IEEE Transactions on Speech and Audio Processing, vol. 6, no. 2, pp.106-115, March 1998.
- [26] Chung-Hsien Wu and Jau-Hung Chen . " A Novel Two-Level Method for the Computation of the LSP Frequencies Using a Decimation-in-Degree Algorithm ," IEEE Transactions on Speech and Audio Processing, vol. 5, no. 2, pp.106-115, March 1997.
- [27] Jianping Pan and Thomas R. Fischer . " Vector Quantization of Speech LSP Parameters Using Trellis Codes and l_1 -norm Constrained ," IEEE International Conference of ASSP, vol.2, pp.17-20, April 27. 28 1993.
- [28] Yoseph Linde, Andrés Buzo and Robert M. Gray , " An Algorithm for Vector Quantizer Design ," IEEE Transactions on Communications, vol. COM-28, pp.84-95, NO. 1, January. 1980.
- [29] Kevin T. Malone and Thomas R. Fischer . " Trellis-Searched Adaptive Predictive Coding of Speech ." IEEE Transactions on Speech and Audio Processing, vol. 1, no. 2, pp.196-206, April 1993.
- [30] Kuldeep K. Paliwal and Bishnu S. Atal . " Efficient Vector Quantization of LPC Parameters at 24 Bits/Framme ." IEEE Transactions on Speech and Audio Processing, vol. 1, no. 1, pp.3-14, January 1993.
- [31] William R. Gardner and Ehaskar D. Rao , "Theoretical Analysis of the High Rate Vector Quantization of LPC parameters ," IEEE Transactions on Speech and Audio Processing, vol. 3, no. 5, pp.367-381, September. 1993.
- [32] John Makhoul . "Linear Prediction ." Proceeding of IEEE, vol. 63, no. 4, pp.561-578, April 1975.
- [33] Robert M.Gray . " Vector Quantization ." IEEE ASSP Magazine, pp.4-29, April 1984.
- [34] Jean Pierre Adoul . " La Quantification Vectorielle des Signaux : Approche Algébrique ." ANN. Telecommun. 41, N° 3-4, pp.158-177, 1986.
- [35] Lawrence C. Stewart, Robert M. Gray and Yoseph Linde . " The Design of Trellis Waveform Coders ." IEEE Transactions on Communications, vol. COM. 30, no. 4, pp.702-710, April 1982.
- [36] Ioannis Katsavounidis, C. -C. Jay Kuo and Zhen Zhang . " A New Initialization Technique for Generalized Lloyd Iteration ." IEEE Signal Processing Letters, vol. 1, no. 10, pp.144-146, October 1994.
- [37] William H. Equitz . " A New Vector Quantization Clustering Algorithm ." IEEE Transactions on Acoustics Speech and Signal Processing, vol. 37, no. 10, pp.1568-1575, October 1987.

- [38] G. Dalessandro . "Traitement de la Parole ." DEA ATS 1994-1995.
- [39] Ronald W. Schafer and Lawrence R. Rabiner . " Digital Representations of Speech Signals ." Proceeding of IEEE, vol. 63, no. 4, pp.662-675, April 1975.
- [40] D. Berkani and M. Xie . " Amélioration des performances des codeurs de parole ." Submit.to AI.Journal of Technology.N°12,ENP.1997 .
- [41] Yunus Hussain and Nariman Farvardin . " Variable Rate Finite State Vector Quantization and Applications to Speech and Image Coding ," IEEE Transactions on Speech and Audio Processing, vol. 1. no. 1, pp.25-38 ,January 1993.
- [42] John Foster, Robert M. Gray and Mari Ostendore Dunhan . " Finite State Vector Quantization for Waveform Coding ," IEEE Transactions on Information Theory, vol. IT-31. no. 3, pp.348-359,May 1985.