

UNIVERSITE DE SHERBROOKE

Faculté des sciences appliquées
Département de génie électrique et de génie informatique

QUANTIFICATION VECTORIELLE ALGEBRIQUE
ET CODAGE DE PAROLE EN BANDE ELARGIE

Thèse de doctorat (Ph. D.)

Spécialité: génie électrique

XIE Minjie

Sherbrooke, Québec
Canada

Février 1996



National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-21865-1

SOMMAIRE

L'objet de cette thèse est l'étude de la quantification vectorielle algébrique et de son application au codage du signal de parole.

La quantification vectorielle algébrique, appelée aussi la quantification vectorielle par réseaux réguliers, est considérée comme une technique de quantification pleine de promesses. A cause de la nature fortement structurée des réseaux réguliers, cette technique permet de réduire la complexité (en stockage et en calcul) de la quantification vectorielle.

Plusieurs approches efficaces et pratiques de la quantification vectorielle algébrique sont proposées dans cette étude. Le quantificateur vectoriel ellipsoïdal permet de quantifier efficacement les coefficients de transformation distribués en forme d'ellipsoïde dans le codage par transformée. Les quantificateurs vectoriels triangulaires et pyramidaux sont utilisés pour quantifier les paramètres LSF du signal de parole. Et les quantificateurs vectoriels encastrés sont appliqués au codage de la parole en bande élargie (50 - 7000 Hz).

REMERCIEMENTS

J'aimerais exprimer ma reconnaissance la plus sincère à monsieur Jean-Pierre Adoul, mon directeur de thèse. C'est grâce à son aide inestimable que j'ai pu poursuivre mes études graduées à l'Université de Sherbrooke. Ses excellents conseils, ses encouragements constants, sa bienveillance et sa patience m'ont permis de mener à bien mes travaux.

Je tiens à remercier toutes les personnes qui m'ont aidé et encouragé pendant la durée de mes études.

TABLE DES MATIERES

SOMMAIRE	i
REMERCIEMENTS	ii
TABLE DES MATIERES	iii
LISTE DES FIGURES ET TABLEAUX	v
INTRODUCTION	1
1. "SHAPING" ET QUANTIFICATION VECTORIELLE ALGEBRIQUE ELLIPSOIDALE	4
1.1 Quantification vectorielle et réseaux réguliers	4
1.2 Technique du "shaping" pour la quantification	11
1.2.1 Gains de quantification	11
1.2.2 "Shaping"	16
1.3 Quantification vectorielle algébrique ellipsoïdale	18
1.3.1 Principe de la quantification vectorielle algébrique ellipsoïdale	19
1.3.2 Algorithmes rapides d'indexation	21
1.3.3 Résultats expérimentaux	30
2. QUANTIFICATION VECTORIELLE ALGEBRIQUE DES PARAMETRES LSF	33
2.1 Quantification de l'information LPC et paramètres LSF	33
2.2 Quantification vectorielle par transformée des paramètres LSF	37
2.2.1 Codage par transformée	37
2.2.2 Quantification vectorielle ellipsoïdale des paramètres LSF	39
2.3 Quantification vectorielle par différence des paramètres LSF	48
2.3.1 Quantification du vecteur différentiel des paramètres LSF	48
2.3.2 Conception du quantificateur vectoriel triangulaire	51
2.3.3 Conception du quantificateur vectoriel pyramidal	55
2.3.4 Résultats expérimentaux	60
3. APPLICATION DE LA QUANTIFICATION VECTORIELLE ALGEBRIQUE AU CODAGE DE LA PAROLE EN BANDE ELARGIE	65
3.1 Codage de la parole en bande élargie	65
3.1.1 Signal de parole en bande élargie et phénomène de masquage	66
3.1.2 Techniques de codage de la parole en bande élargie	67
3.2 Quantification vectorielle algébrique à partir du réseau RE_8	73
3.2.1 Codes sphériques basés sur le réseau RE_8	74
3.2.2 Approche complémentaire	77
3.2.3 Approche encadrée	81
3.2.4 Codage entropique	84
3.2.5 Comparaison des performances de quantification	91
3.3 Application au codage de la parole en bande élargie	94
3.3.1 Codeur TCX et propriétés de la cible	95
3.3.2 Quantification de la cible	97
3.3.3 Performances et comparaison	103

Table des matières

CONCLUSION	106
ANNEXE	108
BIBLIOGRAPHIE	116

LISTE DES FIGURES ET TABLEAUX

Figure 1. 1	Dictionnaires différents pour la quantification d'un signal gaussien ..	15
Figure 1. 2	Un signal de distribution non uniforme	17
Figure 1. 3	Comparaison des performances entre différents quantificateurs vectoriels	17
Figure 1. 4	Principe du codage du quantificateur vectoriel algébrique ellipsoïdal	20
Figure 1. 5	Principe de l'algorithme de Conway et Sloane	21
Figure 1. 6	Dictionnaire ellipsoïdal basé sur le réseau A_2	22
Figure 2. 1	Modèle de production de la parole	34
Figure 2. 2	Distributions conjointes des coefficients de transformée de KL pour les paramètres LSF	40
Figure 2. 3	Contour d'un dictionnaire ellipsoïdal avec trois facteurs d'échelle	41
Figure 2. 4	Ecarts types normalisés des coefficients de transformée	45
Figure 2. 5	Histogramme des distorsions spectrales (SD) pour le quantificateur vectoriel algébrique ellipsoïdal à 29 bits/trame	46
Figure 2. 6	Distributions conjointes des composantes des vecteurs différentiels normalisés	50
Figure 2. 7	Dictionnaire d'un quantificateur vectoriel triangulaire basé sur D_2 ($S = 14$)	52
Figure 2. 8	Schéma de principe du quantificateur vectoriel triangulaire	55
Figure 2. 9	Dictionnaire d'un quantificateur vectoriel pyramidal basé sur D'_3 ($S = 9$)	56
Figure 2.10	Recherche du mot de code dans un dictionnaire pyramidal ($S = 9$)	59
Figure 2.11	Histogramme des distorsions spectrales (SD) correspondant au quantificateur vectoriel algébrique à 28 bits/trame	62

Liste des figures et tableaux

Figure 3. 1	Schéma de principe du codeur à 64 kbit/s de la parole en bande élargie standardisé par la Recommandation UIT-T G.722	69
Figure 3. 2	Schéma de principe du codeur TCX	71
Figure 3. 3	Structure des dictionnaires des sous-quantificateurs sans recouvrement	78
Figure 3. 4	Structure des dictionnaires des sous-quantificateurs encastrés	81
Figure 3. 5	Algorithme de codage de Huffman	85
Figure 3. 6	Processus d'expansion des intervalles dans le codage arithmétique	88
Figure 3. 7	Schéma de principe de la simulation sur les approches algébriques proposées	91
Figure 3. 8	Dictionnaires des sous-quantificateurs statistiques	93
Figure 3. 9	Comparaison des performances de quantification entre les approches algébriques et l'approche statistique	94
Figure 3.10	Principe de la quantification de la cible dans le domaine fréquentiel pour le codage de la parole en bande élargie (50 - 7000 Hz) à 16 kbit/s ...	97

Tableau 1. 1	Performances des meilleurs réseaux connus	9
Tableau 1. 2	Comparaison de la complexité de calcul entre divers quantificateurs vectoriels	10
Tableau 1. 3	Grains granulaires des meilleurs réseaux connus	13
Tableau 1. 4	Performances des quantificateurs différents pour un signal gaussien	15
Tableau 1. 5	Performances de divers quantificateurs pour un signal de distribution ellipsoïdale	31
Tableau 2. 1	Matrice de transformée de KL des paramètres LSF	44
Tableau 2. 2	Ecart types des coefficients de transformée	44
Tableau 2. 3	Performances (SD) du quantificateur vectoriel algébrique ellipsoïdal à 29 bits/trame	46

Liste des figures et tableaux

Tableau 2. 4	Performances (SD) du quantificateur vectoriel fendu en (3, 3, 4) à 29 bits/trame	47
Tableau 2. 5	Diverses tailles du dictionnaire triangulaire	54
Tableau 2. 6	Mots de code du dictionnaire pyramidal basé sur D'_3 ($S = 9$) et leur indice (en italique)	56
Tableau 2. 7	Diverses tailles du dictionnaire pyramidal	58
Tableau 2. 8	Plan d'allocation de bits pour différents débits dans la région de 26 à 29 bits/trame	61
Tableau 2. 9	Performances (SD) du quantificateur vectoriel algébrique pour 26 - 29 bits/trame	61
Tableau 2.10	Facteurs d'échelle et "offsets" pour le quantificateur vectoriel algébrique à 28 bits/trame	63
Tableau 2.11	Performances (SD) du quantificateur vectoriel fendu en (3, 3, 4) pour 26 - 29 bits/trame	63
Tableau 3. 1	Inventaire des sphères du réseau RE_g	76
Tableau 3. 2	Liste des leaders absolus utilisés dans l'approche complémentaire	79
Tableau 3. 3	Liste des leaders absolus utilisés dans l'approche encadrée	83
Tableau 3. 4	Liste des mots de code pour l'algorithme de codage de Huffman	85
Tableau 3. 5	Détail de la procédure de codage arithmétique	89
Tableau 3. 6	Détail du codage arithmétique avec des nombres entiers	90
Tableau 3. 7	Performances de quantification des approches algébriques proposées	92
Tableau 3. 8	Paramètres du codeur TCX à 16 kbit/s	95
Tableau 3. 9	Distribution des bits pour une trame de 96 échantillons du codeur TCX à 16 kbit/s	96
Tableau 3.10	Leaders pour le codage des numéros des sous-quantificateurs dans le premier mode de quantification de l'orientation \mathbf{x}_o	101
Tableau 3.11	Leaders pour le codage des numéros des sous-quantificateurs dans le deuxième mode de quantification de l'orientation \mathbf{x}_o	102
Tableau 3.12	Comparaison des performances entre l'approche algébrique proposée et l'approche complexe statistique	104

Introduction

Les systèmes numériques de communications permettent d'offrir plusieurs avantages importants en comparaison des systèmes analogiques de communications. A cause des progrès de l'électronique et de l'informatique, les systèmes numériques sont largement utilisés actuellement.

La numérisation et la compression des signaux, tels que la parole, la musique et l'image, ont fait l'objet de très nombreuses études depuis que Claude Shannon a posé les bases théoriques de ce domaine dans les années 1940 et 1950. Ces études couvrent aujourd'hui un vaste champ et se concentrent principalement sur la diminution de la distorsion et l'économie des ressources physiques de transmission ou de stockage.

La quantification vectorielle a été considérablement développée depuis les années 1980 et a émergé récemment comme une puissante technique de compression des signaux. La quantification vectorielle n'est pas simplement une généralisation de la quantification scalaire (unidimensionnelle), elle exploite directement la corrélation existant dans le signal. La théorie de l'information démontre que les performances de quantification peuvent toujours être améliorées en augmentant la dimension du vecteur à quantifier. Mais la quantification vectorielle classique, c'est-à-dire la quantification vectorielle statistique, est limitée dans la pratique à cause d'une complexité (en stockage et en calcul) qui augmente rapidement avec la dimension du vecteur. Par conséquent, il est nécessaire de développer de nouvelles techniques de quantification vectorielle.

La quantification vectorielle algébrique, appelée aussi la quantification vectorielle par réseaux réguliers, est considérée comme une technique de quantification pleine de promesses. Divers quantificateurs vectoriels algébriques ont été étudiés durant ces dernières années et certains d'entre eux ont été appliqués avec succès au codage de la parole ainsi qu'au codage de l'image. Grâce à la nature fortement structurée des réseaux réguliers, des algorithmes rapides sont disponibles pour ces quantificateurs et leurs dictionnaires n'ont pas à être stockés en mémoire.

Cette thèse est consacrée à l'application de la quantification vectorielle algébrique au codage du signal de parole. La compression du signal de parole est un domaine important des télécommunications. L'objectif principal du codage de la parole consiste à économiser les ressources des canaux de communications tout en maintenant une bonne qualité de la parole codée. L'intérêt actuel des études dans ce domaine est de développer des codeurs à bas débit, de bonne qualité et de complexité raisonnable. Et l'utilisation de la technique efficace de quantification a une grande importance pour la conception et la réalisation de ces codeurs de la parole.

La contribution principale de cette étude est de développer des algorithmes efficaces et pratiques permettant de réduire la complexité de la quantification vectorielle pour le codage du signal de parole. Plusieurs approches intéressantes de quantification vectorielle algébrique sont proposées dans cette étude. Le quantificateur vectoriel ellipsoïdal permet de quantifier efficacement les coefficients de transformation distribués en forme d'ellipsoïde dans le codage par transformée. Les quantificateurs vectoriels triangulaires et pyramidaux sont utilisés pour quantifier les paramètres LSF du signal de parole. Et les quantificateurs vectoriels encastrés sont appliqués au codage de la parole en bande élargie (50 - 7000 Hz).

Introduction

Cette thèse se compose de trois chapitres distincts. Le chapitre 1 commence par une présentation de la quantification vectorielle algébrique et la technique du "shaping". Un quantificateur vectoriel ellipsoïdal basé sur des réseaux connus est ensuite proposé. Le chapitre 2 concerne la quantification vectorielle algébrique des paramètres LSF du signal de parole. Le chapitre 3 présente l'application de la quantification vectorielle algébrique au codage de la parole en bande élargie (50 - 7000 Hz).

Chapitre 1

"Shaping" et quantification vectorielle algébrique ellipsoïdale

Pour un quantificateur vectoriel par réseaux réguliers, les performances de quantification dépendent de trois quantités essentiellement séparables: le gain granulaire, le gain de forme et le gain de densité non uniforme. La technique du "shaping" consiste à changer la forme (ou le contour) de la région de support du dictionnaire du quantificateur afin d'obtenir un gain de quantification. Un quantificateur vectoriel algébrique ellipsoïdal pour le codage par transformée est proposé dans ce chapitre. Il combine des algorithmes rapides avec la technique du "shaping".

1.1 Quantification vectorielle et réseaux réguliers

Soit $\mathbf{x} \in \mathbf{R}^N$ un vecteur aléatoire. Un quantificateur vectoriel Q de dimension N et de taille L est défini comme une fonction qui associe au vecteur d'entrée \mathbf{x} un vecteur de sortie \mathbf{y}_i choisi dans un ensemble fini C contenant L vecteurs de dimension N , soit

$$Q: \mathbf{R}^N \rightarrow C$$

$$\mathbf{x} \rightarrow Q(\mathbf{x}) \tag{1.1}$$

où
$$Q(\mathbf{x}) = \mathbf{y}_i \quad \text{si } \mathbf{x} \in V_i. \tag{1.2}$$

On appelle $C = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_L\}$ le dictionnaire et \mathbf{y}_i un vecteur de reproduction ou mot de code.

Le quantificateur vectoriel \mathcal{Q} est complètement déterminé par trois éléments: le dictionnaire C , la partition V_i et la correspondance entre les mots de code \mathbf{y}_i et leurs indices i . La partition V_i selon le critère du plus proche voisin s'exprime comme:

$$V_i \supset \{ \mathbf{x} \in \mathbf{R}^N : d(\mathbf{x}, \mathbf{y}_i) < d(\mathbf{x}, \mathbf{y}_j), j \neq i \} \quad (1.3)$$

et elle s'appelle la région de Voronoï.

La différence entre le vecteur d'entrée \mathbf{x} et le mot de code \mathbf{y}_i est appelée l'erreur (ou le bruit) de quantification. L'erreur de quantification peut être séparée en deux parties: l'erreur (ou le bruit) de granulation et l'erreur (ou le bruit) de dépassement. Lorsque le vecteur d'entrée \mathbf{x} tombe dans la région de support du dictionnaire, l'erreur de quantification s'appelle l'erreur de granulation. Lorsque \mathbf{x} est en dehors de la région de support, l'erreur de quantification s'appelle l'erreur de dépassement. En général, les performances d'un quantificateur sont appréciées selon le débit r (bits par dimension) et le rapport signal à bruit (RSB). Ces deux paramètres sont définis comme

$$r = \frac{1}{N} \log_2 L \quad (1.4)$$

et

$$\text{RSB} = \frac{\sigma^2}{\text{EQM}} \quad (1.5)$$

où $\sigma^2 = \frac{1}{N} E\{\|\mathbf{x}\|^2\}$ est la variance par dimension du signal d'entrée et $\text{EQM} = \frac{1}{N} E\{\|\mathbf{x} - \mathbf{y}_i\|^2\}$ est l'erreur quadratique moyenne par dimension. En théorie de l'information, le RSB pour la source gaussienne sans mémoire augmente de 6.02 dB pour chaque bit/dimension additionnel [1].

La procédure de quantification vectorielle peut être statistique ou algébrique. La quantification vectorielle statistique, pour laquelle le dictionnaire est conçu par l'algorithme de K -moyennes [2], génère un quantificateur vectoriel optimal localement. La conception du quantificateur vectoriel statistique est basée sur une base d'apprentissage composée d'un grand nombre de vecteurs représentatifs du signal à quantifier. Afin de rendre compte des caractéristiques statistiques du signal, on fait l'apprentissage à plusieurs reprises. Donc, on exploite l'histogramme du signal au lieu de sa distribution, que l'on ne connaît pas, pour concevoir le quantificateur vectoriel.

Les principales étapes de l'algorithme de K -moyennes sont décrites ci-dessous:

Etape 1: On initialise le dictionnaire composé de L mots de code.

Etape 2: Selon le critère de distorsion minimisant l'erreur quadratique moyenne, on associe à chaque vecteur d'apprentissage un mot de code du dictionnaire. La partition optimale, $\{P_i; i = 1, 2, \dots, L\}$, est donc déterminée.

Etape 3: On calcule la distorsion moyenne globale correspondant à la base d'apprentissage. Si cette distorsion ne varie plus, on arrête cet algorithme. Sinon, on continue.

Etape 4: On calcule le centroïde de chaque partition P_i et on le choisit comme nouveau mot de code du dictionnaire. Puis on reprend les deux premières étapes.

Dans l'initialisation du dictionnaire, on trouve tout d'abord un seul mot de code optimal en calculant le centroïde de l'ensemble de la base d'apprentissage. A partir du mot de code optimal, on produit un dictionnaire de taille 2, en perturbant faiblement ce centroïde dans deux directions opposées. L'application de l'algorithme de K -moyennes à ce dictionnaire, permet d'obtenir un meilleur dictionnaire à deux mots de code optimaux qui sont à leur tour divisés en deux pour constituer un dictionnaire de taille 4. Le procédé est poursuivi jusqu'à l'obtention d'un dictionnaire de la taille voulue.

La quantification vectorielle statistique permet d'offrir de bonnes performances pour diverses sources, mais son application pratique est sévèrement limitée par la complexité en matière de stockage et de calcul. Plus précisément, la complexité de la réalisation (en stockage et en calcul) augmente exponentiellement avec le produit du débit r et de la dimension N .

La quantification vectorielle algébrique, appelée aussi la quantification vectorielle par réseaux réguliers, est considérée comme une technique de quantification pleine de promesses. Elle a été étudiée par de nombreux chercheurs [2][3][4][5] et appliquée avec succès au codage de la parole [6].

Un réseau régulier dans \mathbf{R}^N est l'ensemble des vecteurs \mathbf{x} qui s'obtiennent par combinaison linéaire de N vecteurs de base indépendants, $\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_N$, avec des coefficients de proportionnalité entiers, k_1, k_2, \dots, k_N :

$$\Lambda_N = \{ \mathbf{x} \mid \mathbf{x} = \sum_{i=1}^N k_i \mathbf{z}_i \}. \quad (1.6)$$

Géométriquement, un réseau régulier est un arrangement régulier de points dans l'espace euclidien de dimension N .

Le réseau le plus simple, noté Z_N , est composé de tous les points à coordonnées entières de \mathbf{R}^N . Il est en forme de grille cartésienne.

Un autre réseau intéressant est le réseau D_N . Il contient les points de Z_N dont la somme des coordonnées est paire. En deux dimensions, D_2 a la forme d'un damier et en trois dimensions, D_3 correspond au réseau cubique à faces centrées.

Un réseau très important en huit dimensions est le réseau E_8 , appelé aussi le réseau de Gosset. Il est défini comme l'union de deux réseaux D_8 translétés par les deux mots de code à répétition: $[0,0,0,0,0,0,0,0]$ et $[\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}]$, soit

$$E_8 = (D_8) \cup (D_8 + [\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}]). \quad (1.7)$$

Une définition alternative représentant le réseau de Gosset tourné RE_8 s'exprime comme

$$RE_8 = (2D_8) \cup (2D_8 + [1, 1, 1, 1, 1, 1, 1, 1]). \quad (1.8)$$

D'autres réseaux souvent utilisés en quantification vectorielle sont:

- le réseau hexagonal A_2 en 2 dimensions,
- le réseau de Barnes-Wall A_{16} en 16 dimensions et
- le réseau de Leech A_{24} en 24 dimensions.

Le détail de ces réseaux est décrit dans la référence [3].

Lorsqu'on translate un réseau régulier pour amener son origine \mathbf{y}_0 à un point quelconque \mathbf{y}_i , il reste congruent à lui-même. Donc, toutes les régions de Voronoï du réseau sont identiques à la région de Voronoï V_0 autour de l'origine.

Pour un quantificateur vectoriel utilisant un réseau régulier Λ appliqué à une source de distribution uniforme, l'erreur quadratique moyenne normalisée EQM_i est la même pour toute région de Voronoï V_i . La performance globale en terme d'erreur quadratique moyenne est uniquement liée aux propriétés géométriques de la région de Voronoï V_0 . Le moment d'ordre 2 normalisé de la région de Voronoï de Λ , $M(\Lambda)$, est défini comme [7]

$$M(\Lambda) = \frac{\int_{\mathbf{x} \in V_0} \mathbf{x}^2 d\mathbf{x}}{\left[\int_{\mathbf{x} \in V_0} d\mathbf{x} \right]^{(N+2)/N}}. \quad (1.9)$$

$M(\Lambda)$ permet la comparaison entre différents réseaux. Le tableau 1.1 résume les performances des meilleurs réseaux connus.

TABLEAU 1.1
Performances des meilleurs réseaux connus

N	Réseau (Λ)	$M(\Lambda)$
1	Z	0.0833
2	A_2	0.0802
4	D_4	0.0766
8	E_8	0.0717
16	A_{16}	0.0683
24	A_{24}	0.0658

Grâce à la nature fortement structurée des réseaux réguliers, des algorithmes rapides sont disponibles pour le quantificateur vectoriel algébrique et son dictionnaire n'a pas à être stocké en mémoire. Conway et Sloane ont développé des algorithmes rapides pour la quantification vectorielle basée sur certains réseaux réguliers connus [8]. A la différence de la quantification vectorielle statistique, le calcul de la distorsion entre le vecteur d'entrée et chaque vecteur du dictionnaire, qui est utilisé pour trouver le meilleur mot de code parmi eux, n'est pas requis.

La quantification d'un vecteur quelconque $\mathbf{x} \in \mathbf{R}^N$ basée sur le réseau Z_N est réalisée en arrondissant individuellement chaque composante de \mathbf{x} au nombre entier le plus proche.

Pour le réseau D_N , le plus proche voisin de \mathbf{x} doit avoir une somme paire de composantes. L'algorithme de quantification se décompose en deux étapes suivantes:

Etape 1: Trouver le plus proche voisin \mathbf{y} de \mathbf{x} dans le réseau Z_N .

Etape 2: Si la somme de composantes de \mathbf{y} est impaire, modifier \mathbf{y} en arrondissant de la "mauvaise façon" la composante de \mathbf{x} qui a une plus grande distorsion de quantification que les autres.

Dans le cas du réseau D_4 , par exemple, un vecteur d'entrée $\mathbf{x} = [3.2, -1.7, -0.4, 2.1]$ est représenté par $\mathbf{y} = [3, -2, -1, 2]$ au lieu de $\mathbf{y}' = [3, -2, 0, 2]$.

En examinant la définition (1.7), on peut voir que le réseau de Gosset E_8 est basé sur le réseau D_8 . Donc, l'algorithme de recherche du plus proche voisin de \mathbf{x} dans le réseau E_8 s'effectue par les trois étapes suivantes:

Etape 1: Trouver le plus proche voisin \mathbf{y} de \mathbf{x} dans le réseau D_8 .

Etape 2: Trouver le plus proche voisin \mathbf{y}_1 de $\mathbf{x} - [\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}]$ dans le réseau D_8 et calculer $\mathbf{y}' = \mathbf{y}_1 + [\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}]$.

Etape 3: Choisir entre \mathbf{y} et \mathbf{y}' le vecteur qui minimise la distorsion de quantification en terme d'erreur quadratique moyenne.

Le tableau 1.2 montre la complexité de calcul du quantificateur vectoriel statistique et celle des algorithmes de quantification pour les réseaux réguliers Z_N , D_N et E_8 . Pour $N = 8$ et $r = 1.25$, par exemple, les rapports de complexité de calcul entre un quantificateur basé sur le réseau E_8 et un quantificateur statistique sont respectivement 1 : 512 en multiplications, 1 : 256 en additions et 1 : 68 en comparaisons.

TABLEAU 1.2
Comparaison de la complexité de calcul
entre divers quantificateurs vectoriels

Opération	Statistique	Z_N	D_N	$E_8(N=8)$
Multiplications	$2^{rN} N$			$2N$
Additions	$2^{rN} (2N-1)$		$2N-1$	$4(2N-1)$
Comparaisons	$2^{rN} - 1$		$N-1$	$2N-1$
Arrondis		N	$N+1$	$2(N+1)$

Cependant, le quantificateur par réseaux réguliers n'est optimal que pour les sources de distribution uniforme et donc il est possible que ses performances soient mauvaises pour d'autres sources. Selon le principe AEP (Asymptotic Equipartition Principle) [1], à mesure que $N \rightarrow \infty$, la densité de probabilité d'une source gaussienne tend à se concentrer à la surface d'une sphère [9] et celle d'une source laplacienne se concentre à la surface d'une pyramide [5]. Pour améliorer les performances de quantification dans les cas des distribution non uniformes, la technique du "shaping" est utilisée dans la conception du quantificateur vectoriel algébrique.

En général, la conception d'un quantificateur vectoriel algébrique procède par les trois étapes suivantes:

Etape 1: Tronquer, selon le débit donné, un réseau régulier infini avec un "shaping" adéquat.

Etape 2: Développer un algorithme de recherche du plus proche voisin dans le réseau utilisé.

Etape 3: Développer un algorithme d'indexation des mots de code.

1.2 Technique du "shaping" pour la quantification

Nous donnons d'abord les définitions des gains qui déterminent les performances d'un quantificateur vectoriel. Ensuite, nous expliquons quel est l'objectif du "shaping" et comment le "shaping" améliore les performances du quantificateur.

1.2.1 Gains de quantification

Dans cette étude, un quantificateur scalaire uniforme est utilisé comme système de référence. Ce quantificateur scalaire uniforme consiste à partitionner uniformément

l'intervalle $[-x_{max}, x_{max}]$ en sous-intervalles de même longueur et choisir les niveaux de quantification au milieu des sous-intervalles. Un ensemble de N quantificateurs scalaires uniformes est équivalent à un quantificateur vectoriel de dimension N dont les mots de code sont les points situés au sommet d'une grille cartésienne. La région de Voronoï de ce quantificateur équivalent est un hypercube et donc il est appelé le quantificateur cubique.

Pour les sources sans mémoire, un quantificateur vectoriel permet d'obtenir trois gains séparés par rapport à un quantificateur cubique. Ces trois gains sont le gain granulaire, le gain de forme et le gain de densité non uniforme. Les deux premiers gains sont définis mathématiquement dans la référence [10] et le dernier est défini qualitativement dans la référence [11]. Nous les résumons ci-dessous.

Supposons que nous avons un quantificateur vectoriel de dimension N basé sur un réseau régulier Λ , dénoté \mathcal{Q}_Λ , et que la distorsion soit mesurée par l'erreur quadratique moyenne.

Soit τ un N -cube ayant le même volume que Λ . Le gain granulaire γ_g du quantificateur \mathcal{Q}_Λ est défini comme le rapport du moment d'ordre 2 normalisé de τ à celui de la région de Voronoï de Λ [10]

$$\gamma_g = \frac{M(\tau)}{M(\Lambda)} = \frac{1}{12M(\Lambda)}. \quad (1.10)$$

Le gain granulaire représente l'amélioration des performances au sens du moment d'ordre 2 normalisé. Autrement dit, si τ est remplacé par Λ , l'erreur de granulation du quantificateur sera réduite d'un facteur égal à γ_g . Lorsque N tend vers l'infini, le gain granulaire γ_g approche de $\frac{\pi e}{6}$ (1.53 dB ou 0.255 bit par dimension) [10]. Cette valeur est le gain granulaire maximum. Le tableau 1.3 montre les gains granulaires des meilleurs réseaux connus [10].

TABLEAU 1.3
Grains granulaires des meilleurs réseaux connus

N	Réseau (Λ)	γ_g (dB)
1	Z	0.00
2	A_2	0.17
4	D_4	0.37
8	E_8	0.65
16	Λ_{16}	0.86
24	Λ_{24}	1.03

Selon la définition (1.10), nous pouvons voir que le gain granulaire γ_g dépend de la mesure de distorsion et est indépendant de la densité de probabilité de la source.

Pour définir le gain de forme, on suppose qu'un N -cube τ a la même probabilité de dépassement que la région de support \mathbf{S} du dictionnaire de \mathcal{Q}_Λ . Soit L la longueur du côté de τ et $V(\mathbf{S})$ le volume de \mathbf{S} . Le gain de forme γ_f du quantificateur \mathcal{Q}_Λ est défini comme le rapport du volume normalisé (ramené à deux dimensions) de τ à celui de la région \mathbf{S} [10]

$$\gamma_f = \frac{L^2}{V(\mathbf{S})^{2/N}}. \quad (1.11)$$

Le gain de forme correspond à l'amélioration des performances qui est obtenue par l'adaptation de la région de support du dictionnaire au contour de la densité de probabilité de la source. Epouser la forme de cette région consiste à placer la plupart des mots de code dans la région de haute densité de probabilité afin de minimiser l'erreur de dépassement du quantificateur. Pour le gain de forme, il n'existe pas en théorie de valeur maximale. Dans la pratique, le gain de forme peut être beaucoup plus grand que le gain granulaire pour les sources non uniformes [10].

Contrairement au gain granulaire, le gain de forme γ_f dépend de la densité de probabilité de la source.

Le gain de densité non uniforme mesure l'amélioration des performances qui résulte de l'adaptation de la répartition des mots de code du dictionnaire à la densité de probabilité de la source. Pour les sources non uniformes, si les mots de code sont faiblement espacés dans la région de haute densité de probabilité et largement espacés dans la région de densité faible, il est susceptible de conduire à un meilleur RSB. En effet, l'importance de l'erreur de granulation sera réduite pour les amplitudes du signal les plus probables. Dans le cas du quantificateur vectoriel par réseaux réguliers, le gain de densité non uniforme peut être obtenu en utilisant plusieurs réseaux pour lesquels les distances entre les points sont différentes.

Il est évident que le gain de densité non uniforme dépend de la densité de probabilité de la source.

L'exemple qui suit permet d'examiner les contributions aux performances de quantification apportées respectivement par le gain granulaire, le gain de forme et le gain de densité non uniforme.

Considérons un signal de 10000 vecteurs à deux dimensions distribué selon la loi gaussienne à moyenne nulle de variance unité. Ainsi qu'on l'a déjà montré dans la section 1.1, la densité de probabilité de ce signal possède une allure circulaire. On cherche à le quantifier en utilisant trois quantificateurs vectoriels basés sur le réseau Z_2 . Les dictionnaires des quantificateurs ont la même taille (256 mots de code), mais les formes de leur région de support et les distances entre les points sont différentes. La figure 1.1 illustre la structure des dictionnaires.

Les trois quantificateurs ne réalisent pas de gain granulaire parce qu'ils sont basés sur le réseau cubique Z_2 . Cependant, les deux derniers quantificateurs (Figure 1.1(b) et (c)) ont un dictionnaire de contour circulaire et donc permettent d'obtenir le gain de forme. Et un gain de densité non uniforme est fourni seulement par le dernier quantificateur (Figure 1.1(c)) à cause de la répartition non uniforme de ses mots de code. Le tableau 1.4 montre les performances en terme de RSB des quantificateurs pour ce signal gaussien.

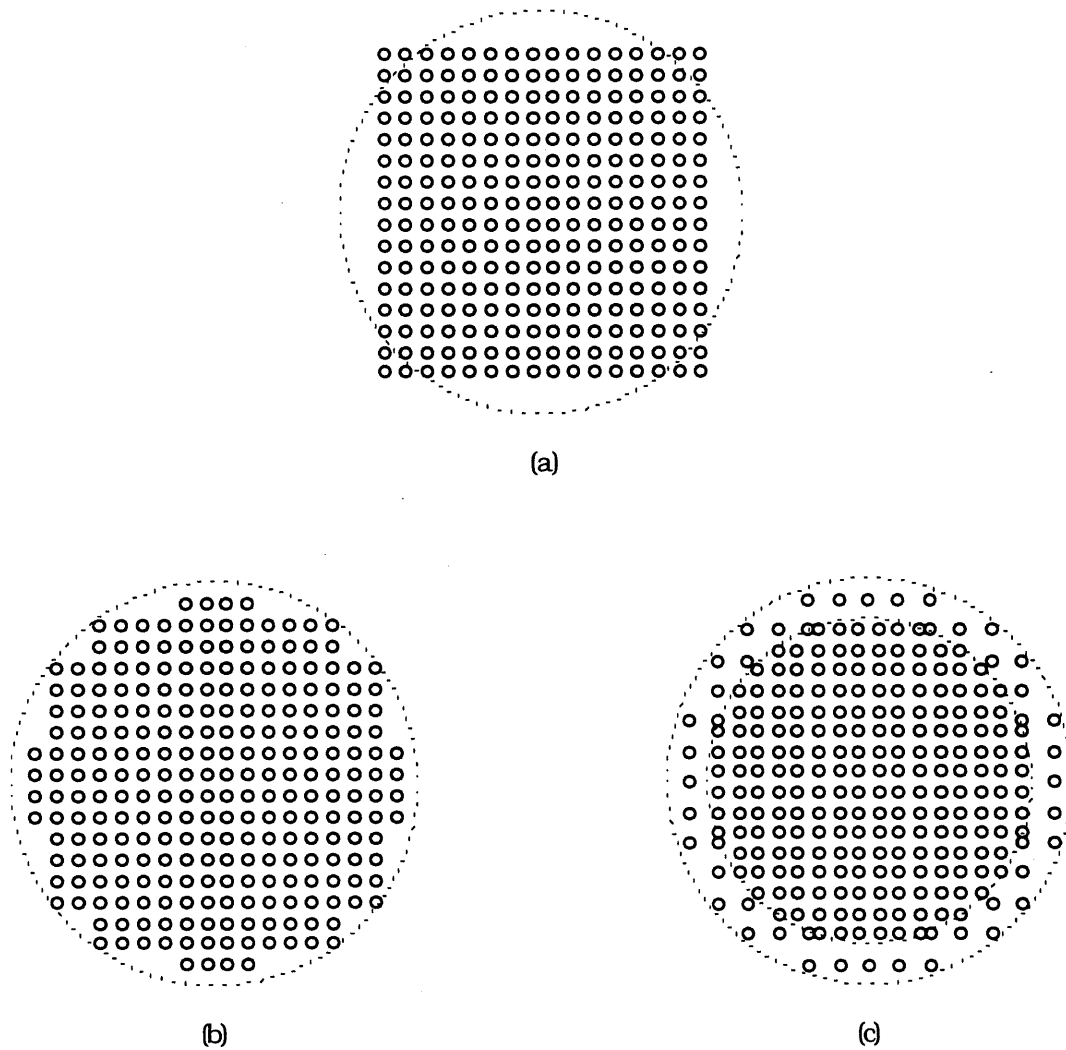


Figure 1.1 Dictionnaires différents pour la quantification d'un signal gaussien.

TABLEAU 1.4
Performances des quantificateurs
différents pour un signal gaussien

Quantificateur	RSB (dB)
(a)	19.41
(b)	19.76
(c)	20.09

On remarque que pour un dictionnaire de forme non cubique et un dictionnaire à répartition non uniforme, l'indexation des mots de code devient très compliquée lorsque la dimension du vecteur, N , tend vers l'infini.

1.2.2 "Shaping"

Le terme technique "shaping" a été utilisé pour la première fois en transmission de données [12][13]. Dans ce contexte, le "shaping" a pour objectif de réduire la puissance moyenne du signal transmis en changeant la forme (ou le contour) de la constellation du signal. Selon la dualité entre la transmission de données et la quantification [10], la constellation du signal dans la transmission de données correspond au dictionnaire du quantificateur. En conséquence, le "shaping" pour la quantification consiste à changer la forme de la région de support du dictionnaire du quantificateur pour obtenir le gain de forme.

En réalité, la technique du "shaping" est utilisée en quantification vectorielle par réseaux réguliers depuis les années 1980. Diverses approches avec "shaping" ont été étudiées par les chercheurs. Conway et Sloane ont présenté un algorithme rapide de quantification pour la source gaussienne en 1983 [14], leur dictionnaire basé sur des réseaux réguliers connus possède une région de support sphérique. Adoul *et al.* ont développé le quantificateur vectoriel sphérique basé sur les réseaux E_8 et A_{24} en 1984 [6] et appliqué avec succès ce quantificateur au codage du signal de parole. Fischer a proposé le quantificateur pyramidal pour la source laplacienne en 1986 [5] et le quantificateur ellipsoïdal pour le codage par transformée en 1989 [15].

Maintenant nous expliquons le "shaping" pour la quantification avec l'exemple suivant. Considérons une source avec la distribution illustrée à la figure 1.2. Trois méthodes sont proposées pour la quantifier avec le même débit. La figure 1.3 montre les formes des régions de support des dictionnaires utilisés par ces trois quantificateurs vectoriels. Dans le premier quantificateur (Figure 1.3(a)), aucun "shaping" n'est utilisé. Le

deuxième quantificateur (Figure 1.3 (b)) et le troisième quantificateur (Figure 1.3 (c)) emploient respectivement un "shaping" simple et un "shaping" adéquat. Nous voyons clairement que le quantificateur avec "shaping" adéquat s'adapte beaucoup mieux aux caractéristiques du signal à quantifier et peut avoir de meilleures performances que les autres.

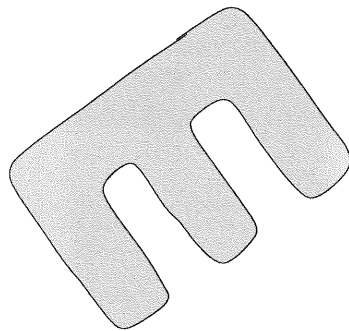
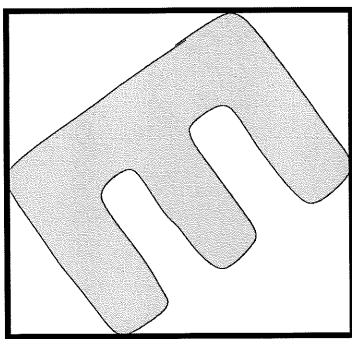
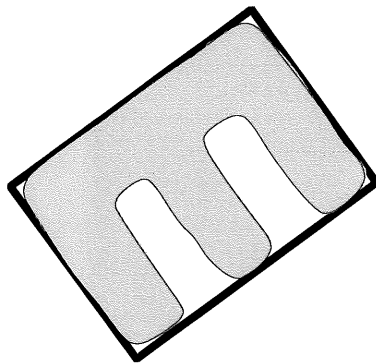


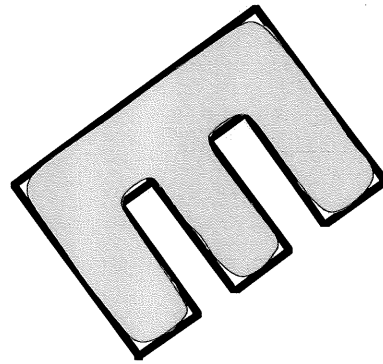
Figure 1.2 Un signal de distribution non uniforme.



(a) Sans "shaping"



(b) Avec "shaping" simple



(c) Avec "shaping" adéquat

Figure 1.3 Comparaison des performances entre différents quantificateurs vectoriels.

Le "shaping" permet d'améliorer les performances de quantification pour les sources non uniformes. Cependant la conception d'un dictionnaire avec un bon "shaping" est une tâche compliquée, même si la forme de la région de support du dictionnaire est très symétrique. D'une part, les frontières du réseau régulier utilisé doivent s'adapter à la densité de probabilité de la source autant que possible. D'autre part, l'indexation des mots de code doit être facile et efficace. Pour ces raisons, l'utilisation du "shaping" est souvent restreinte. La solution à ce problème consiste à développer des algorithmes efficaces et rapides pour des formes simples fréquemment rencontrées.

1.3 Quantification vectorielle algébrique ellipsoïdale

Le codage par transformée est une technique très populaire dans le traitement de la parole et de l'image [16]. Pour des sources gaussiennes avec mémoire, certaines transformées, par exemple la transformée de Karhunen-Loève (KL) et la transformée en ondelettes, conduisent à la distribution ellipsoïdale des coefficients de transformée [15][17]. La quantification vectorielle par réseaux réguliers appliquée à ces signaux sera donc beaucoup plus efficace lorsqu'on utilise un dictionnaire ellipsoïdal.

Fischer a présenté une approche pour le codage par transformée de KL [15], mais son quantificateur est basé sur le réseau cubique Z_N et ne fournit pas de gain granulaire. Moureaux *et al.* ont construit des dictionnaires ellipsoïdaux basés sur les réseaux Z_N , D_N et E_8 pour les coefficients d'ondelette en codage d'image [18]. Cependant leur approche ne résout pas le problème de l'indexation des mots de code. Nous proposons dans cette section un quantificateur vectoriel algébrique ellipsoïdal combinant des algorithmes rapides avec la technique du "shaping". Les réseaux réguliers A_2 , D_N et E_8 sont utilisés pour construire le dictionnaire du quantificateur. Le dictionnaire n'a pas besoin d'être stocké et l'indexation des mots de code est réalisée algébriquement.

1.3.1 Principe de la quantification vectorielle algébrique ellipsoïdale

Pour un quantificateur vectoriel ellipsoïdal par réseaux réguliers, la conception de son dictionnaire se décompose en deux étapes suivantes:

Etape 1: Tronquer, selon le débit donné, un réseau infini de façon ellipsoïdale.

Etape 2: Ajuster l'échelle du réseau tronqué afin de minimiser la distorsion globale en terme d'erreur quadratique moyenne.

La troncation du réseau utilisé est une procédure de "shaping" en vue d'obtenir le gain de forme.

L'algorithme proposé par Conway and Sloane [14] réalise une troncation basée sur la région de Voronoï des réseaux connus. Elle apporte une bonne approximation pour une région sphérique dans l'espace \mathbf{R}^N . De plus, leur algorithme fournit aussi une méthode d'indexation des mots de code. Nous généralisons l'algorithme de Conway and Sloane et l'appliquons aux sources de distribution ellipsoïdale. Cela sera décrit en détail au paragraphe 1.3.2.

Afin d'obtenir un dictionnaire optimal pour le débit donné, on a besoin d'ajuster l'échelle du réseau tronqué. Cette opération garantit que la plupart des vecteurs d'entrée tombent dans la région conservée et permet de minimiser la distorsion globale de quantification. Dans la pratique, au lieu de changer l'échelle du réseau, on réduit (ou agrandit) proportionnellement le signal d'entrée avec un facteur d'échelle de façon à utiliser les algorithmes rapides tels que décrits dans la section 1.1. Au récepteur, les représentants des vecteurs d'entrée sont obtenus en changeant l'échelle du réseau avec l'inverse du facteur d'échelle. Ce facteur d'échelle est choisi expérimentalement.

Après avoir réduit (ou agrandi) le signal, il y a des vecteurs qui tombent à l'extérieur de l'ellipsoïde. Pour quantifier ces vecteurs, on les réduit homothétiquement jusqu'à ce que leur plus proche voisin appartienne au dictionnaire.

L'algorithme de codage se résume comme suit:

Etape 1: Réduire (ou agrandir) proportionnellement un vecteur d'entrée \mathbf{x} par un facteur d'échelle α . Dénoter \mathbf{x}' le vecteur obtenu.

Etape 2: Trouver le plus proche voisin \mathbf{y} de \mathbf{x}' dans le réseau en utilisant les algorithmes rapides de quantification décrits dans la section 1.1.

Etape 3: Retrouver le mot de code \mathbf{y}' à partir de l'indice i et comparer \mathbf{y} avec \mathbf{y}' . Si $\mathbf{y} = \mathbf{y}'$, \mathbf{y} est situé dans la région ellipsoïdale et donc il est le représentant de \mathbf{x}' . Sinon, \mathbf{x}' est réduit homothétiquement jusqu'à ce que son plus proche voisin appartienne au dictionnaire.

Etape 4: Fin.

La figure 1.4 illustre le principe du codage du quantificateur vectoriel algébrique ellipsoïdal.

La procédure de décodage se fait essentiellement par les opérations inverses.

L'algorithme est montré ci-dessous:

Etape 1: Retrouver le mot de code à partir de l'indice i .

Etape 2: Changer l'échelle du mot de code par l'inverse du facteur α .

Etape 3: Fin.

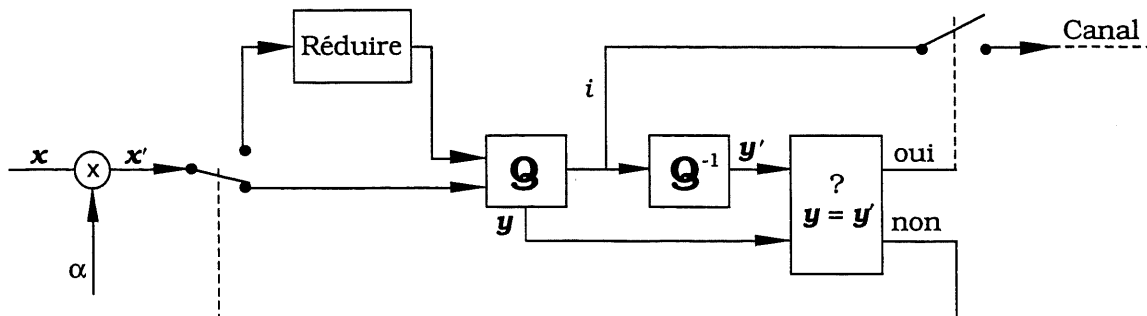


Figure 1.4 Principe du codage du quantificateur vectoriel algébrique ellipsoïdal.

1.3.2 Algorithmes rapides d'indexation

L'algorithme d'indexation de Conway et Sloane [14] consiste à associer à chaque point du réseau régulier Λ situé dans la région de Voronoï $m\Lambda$ ($m = 1, 2, 3, \dots$) un vecteur $\mathbf{j} = [j_1, j_2, \dots, j_N]$ formé des N coefficients entiers qui spécifient les points du réseau Λ à partir des vecteurs de base dans la définition (1.6). Les composantes j_i prennent leurs valeurs dans l'intervalle $[-m, m]$. A partir de ce vecteur \mathbf{j} , on obtient le vecteur indice \mathbf{k} en effectuant l'opération modulo m sur chacune des composantes j_i . Grâce à une propriété de la région de Voronoï $m\Lambda$, les vecteurs indices associés aux m^N points de cette région sont tous distincts. La matrice génératrice G [3] du réseau Λ , qui est composée de N vecteurs de base, et son inverse G^{-1} sont utilisés respectivement dans le décodage et le codage. Soit \mathbf{y} le point de Λ dans la région $m\Lambda$, \mathbf{k} le vecteur indice et \mathbf{z} le vecteur de reproduction. La figure 1.5 montre le principe de l'algorithme de Conway et Sloane.

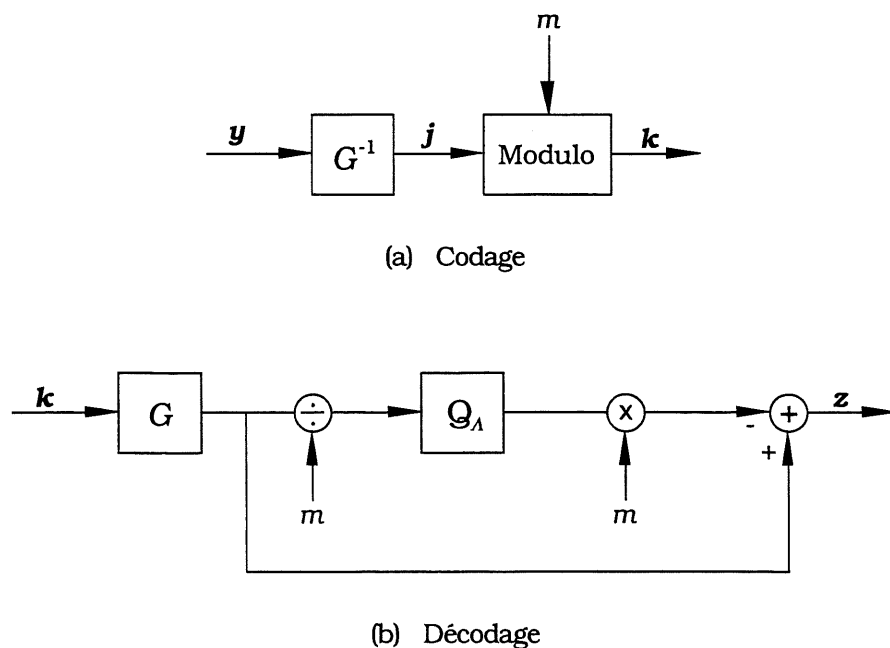


Figure 1.5 Principe de l'algorithme de Conway et Sloane.

Dans notre quantificateur vectoriel algébrique ellipsoïdal, on utilise un vecteur $\mathbf{m} = [m_1, m_2, \dots, m_N]$ au lieu du nombre entier m pour associer au point du réseau situé dans la région ellipsoïdale le vecteur indice. Les composantes de \mathbf{m} sont des nombres entiers positifs et peuvent être différentes. Les valeurs de ces composantes dépendent de la caractéristique statistique du signal de source. Plus précisément, chaque composante de \mathbf{m} est déterminée par la variance de la composante correspondante du vecteur d'entrée. De plus, on apporte des restrictions aux valeurs des composantes de \mathbf{m} pour exploiter des algorithmes rapides.

Le nombre de vecteurs indices, c'est-à-dire la taille du dictionnaire ellipsoïdal, peut être calculé par

$$L = \prod_{i=1}^N m_i \tag{1.12}$$

où les m_i sont des nombres entiers positifs.

La figure 1.6 illustre un dictionnaire ellipsoïdal basé sur le réseau A_2 pour lequel on choisit le vecteur $\mathbf{m} = [16, 8]$.

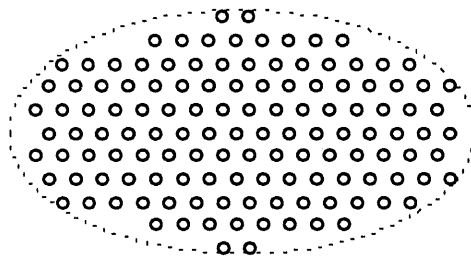


Figure 1.6 Dictionnaire ellipsoïdal basé sur le réseau A_2 .

Maintenant on décrit les algorithmes rapides d'indexation pour les réseaux réguliers A_2 , D_N et E_8 . Le détail des étapes ayant permis de déterminer ces algorithmes est décrit en annexe.

Algorithme pour A_2

La matrice génératrice du réseau A_2 est définie comme [14]

$$G = \begin{bmatrix} 1 & 0 \\ \frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix}. \quad (1.13)$$

Et son inverse s'exprime par

$$G^{-1} = \begin{bmatrix} 1 & 0 \\ -\frac{\sqrt{3}}{3} & \frac{2\sqrt{3}}{3} \end{bmatrix}. \quad (1.14)$$

Dans cet algorithme, on impose une contrainte au vecteur \mathbf{m} : ses composantes doivent toutes avoir la même parité (toutes paires ou toutes impaires).

- Codage (mot de code \mathbf{y} \rightarrow vecteur indice \mathbf{k})

Etape 1: Calculer $\mathbf{j} = \mathbf{y}G^{-1}$.

Etape 2: Si $j_2 < 0$, modifier j_1 selon l'expression suivante:

$$j_1 = j_1 + \frac{1}{2}(m_1 - m_2). \quad (1.15)$$

Etape 3: Calculer $k_i = j_i \pmod{m_i}$ pour trouver $0 \leq k_i < m_i$ ($i = 1, 2$) et obtenir le vecteur indice $\mathbf{k} = [k_1, k_2]$.

Etape 4: Fin.

- Décodage (vecteur indice \mathbf{k} \rightarrow mot de code \mathbf{y})

Etape 1: Calculer $\mathbf{y}' = \mathbf{kG}$.

Etape 2: Calculer $\mathbf{z} = [\frac{y'_1}{m_1}, \frac{y'_2}{m_2}]$.

Etape 3: Trouver le plus proche voisin \mathbf{v} de \mathbf{z} dans le réseau A_2 .

Etape 4: Calculer $y_i = y'_i - m_i v_i$ ($i = 1, 2$) et obtenir le mot de code $\mathbf{y} = [y_1, y_2]$.

Etape 5: Fin.

- Exemple $\mathbf{m} = [16, 8], \quad \mathbf{y} = [\frac{1}{2}, -\frac{5\sqrt{3}}{2}]$.

$\mathbf{y} \rightarrow \mathbf{k}$:

$$\mathbf{j} = \mathbf{yG}^{-1} = [3, -5],$$

$$j_1 = 3 + \frac{1}{2}(16 - 8) = 7,$$

$$\mathbf{k} = [7, 3].$$

$\mathbf{k} \rightarrow \mathbf{y}$:

$$\mathbf{y}' = \mathbf{kG} = [\frac{17}{2}, \frac{3\sqrt{3}}{2}],$$

$$\mathbf{z} = [\frac{17}{32}, \frac{3\sqrt{3}}{16}],$$

$$\mathbf{v} = [\frac{1}{2}, \frac{\sqrt{3}}{2}],$$

$$\mathbf{y} = [\frac{1}{2}, -\frac{5\sqrt{3}}{2}].$$

Algorithme pour D_N

Pour le réseau D_N , on a [14]

$$G = \begin{bmatrix} 2 & 0 & 0 & \dots & 0 \\ 1 & 1 & 0 & \dots & 0 \\ 1 & 0 & 1 & \dots & 0 \\ \cdot & \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot \\ 1 & 0 & 0 & \dots & 1 \end{bmatrix} \quad (1.16)$$

et

$$G^{-1} = \begin{bmatrix} \frac{1}{2} & 0 & 0 & \dots & 0 \\ -\frac{1}{2} & 1 & 0 & \dots & 0 \\ -\frac{1}{2} & 0 & 1 & \dots & 0 \\ \cdot & \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot \\ -\frac{1}{2} & 0 & 0 & \dots & 1 \end{bmatrix} \quad (1.17)$$

Pareillement à l'algorithme pour le réseau A_2 , les composantes du vecteur \mathbf{m} doivent toutes avoir la même parité.

- Codage (mot de code \mathbf{y} → vecteur indice \mathbf{k})

Etape 1: Calculer $\mathbf{j} = \mathbf{y}G^{-1}$.

Etape 2: Modifier j_1 à partir des signes des autres composantes de \mathbf{j}

$$j_1 = j_1 + \frac{1}{2} \sum_{i=2}^N (s(j_i) (m_1 - m_i)) \quad (1.18)$$

avec

$$s(x) = \begin{cases} 0, & x \geq 0 \\ 1, & x < 0 \end{cases}.$$

Etape 3: Calculer $k_i = j_i \pmod{m_i}$ pour trouver $0 \leq k_i < m_i$ ($i = 1, 2, \dots, N$) et obtenir le vecteur indice $\mathbf{k} = [k_1, k_2, \dots, k_N]$.

Etape 4: Fin.

• Décodage (vecteur indice $\mathbf{k} \rightarrow$ mot de code \mathbf{y})

Etape 1: Calculer $\mathbf{y}' = \mathbf{kG}$.

Etape 2: Calculer $\mathbf{z} = [\frac{y'_1}{m_1}, \frac{y'_2}{m_2}, \dots, \frac{y'_N}{m_N}]$.

Etape 3: Trouver le plus proche voisin \mathbf{v} de \mathbf{z} dans le réseau D_N .

Etape 4: Calculer $y_i = y'_i - m_i v_i$ ($i = 1, 2, \dots, N$) et obtenir le mot de code $\mathbf{y} = [y_1, y_2, \dots, y_N]$.

Etape 5: Fin.

• Exemple $\mathbf{m} = [7, 5, 3, 3],$ $\mathbf{y} = [2, -2, -1, 1].$

$\mathbf{y} \rightarrow \mathbf{k}$:

$$\mathbf{j} = \mathbf{yG}^1 = [2, -2, -1, 1],$$

$$j_1 = 2 + \frac{1}{2}((7 - 5) + (7 - 3)) = 5,$$

$$\mathbf{k} = [5, 3, 2, 1].$$

$\mathbf{k} \rightarrow \mathbf{y}$:

$$\mathbf{y}' = \mathbf{kG} = [16, 3, 2, 1],$$

$$\mathbf{z} = [\frac{16}{7}, \frac{3}{5}, \frac{2}{3}, \frac{1}{3}],$$

$$\mathbf{v} = [2, 1, 1, 0],$$

$$\mathbf{y} = [2, -2, -1, 1].$$

Algorithme pour E_8

La matrice génératrice du réseau E_8 s'écrit [14]

$$G = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \quad (1.19)$$

Et donc on a

$$G^1 = \begin{bmatrix} \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{1}{2} & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -\frac{1}{2} & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ -\frac{1}{2} & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -\frac{1}{2} & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ -\frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \frac{5}{2} & -1 & -1 & -1 & -1 & -1 & -1 & 2 \end{bmatrix} \quad (1.20)$$

Dans ce cas, l'algorithme impose les trois contraintes suivantes aux composantes du vecteur m :

- toutes les composantes ont la même parité (toutes paires ou toutes impaires);
- la somme de composantes est un multiple de 4;
- $m_i \geq m_8$ ($i = 1, 2, \dots, 7$).

- Codage (mot de code \mathbf{y} \rightarrow vecteur indice \mathbf{k})

Etape 1: Calculer $\mathbf{j} = \mathbf{yG}^{-1}$.

Etape 2: Définir un vecteur $\mathbf{u} = [1, u_2, u_3, u_4, u_5, u_6, u_7, u_8]$ comme suit:

$$u_8 = \begin{cases} -0.5, & m_8 \leq j_8 \\ 0, & 0 \leq j_8 < m_8 \\ 0.5, & -m_8 \leq j_8 < 0 \\ 1, & j_8 < -m_8 \end{cases} \quad (1.21)$$

Si $u_8 = 0, 1$ alors

$$u_i = \begin{cases} 0, & 0 \leq (j_i - u_8 m_8) \\ 1, & -m_i \leq (j_i - u_8 m_8) < 0, \\ 2, & (j_i - u_8 m_8) < -m_i \end{cases} \quad i = 2, 3, \dots, 7; \quad (1.22)$$

si $u_8 = \pm 0.5$ alors

$$u_i = \begin{cases} -0.5, & \frac{m_i}{2} \leq (j_i - u_8 m_8) \\ 0.5, & -\frac{m_i}{2} \leq (j_i - u_8 m_8) < \frac{m_i}{2}, \quad i = 2, 3, \dots, 7. \\ 1.5, & (j_i - u_8 m_8) < -\frac{m_i}{2} \end{cases} \quad (1.23)$$

Etape 3: Calculer \mathbf{k}' selon les équations suivantes:

$$k'_1 = j_1 + \frac{1}{2} \sum_{i=2}^8 (u_i (m_1 - m_i)) + 3u_8 m_8 + w m_1 \quad (1.24)$$

avec

$$w = \begin{cases} 0, & u_8 = 0, 1 \\ 0.5, & u_8 = \pm 0.5 \end{cases}$$

$$k'_i = j_i + u_i m_i - u_8 m_8, \quad i = 2, 3, \dots, 7 \quad (1.25)$$

et

$$k'_8 = j_8 . \quad (1.26)$$

Etape 4: Calculer $k_i = k'_i \pmod{m_i}$ pour trouver $0 \leq k_i < m_i$ ($i = 1, 2, \dots, 8$) et obtenir le vecteur indice $\mathbf{k} = [k_1, k_2, k_3, k_4, k_5, k_6, k_7, k_8]$.

Etape 5: Fin.

• Décodage (vecteur indice $\mathbf{k} \rightarrow$ mot de code \mathbf{y})

Etape 1: Calculer $\mathbf{y}' = \mathbf{kG}$.

Etape 2: Calculer $\mathbf{z} = [\frac{y'_1}{m_1}, \frac{y'_2}{m_2}, \frac{y'_3}{m_3}, \frac{y'_4}{m_4}, \frac{y'_5}{m_5}, \frac{y'_6}{m_6}, \frac{y'_7}{m_7}, \frac{y'_8}{m_8}]$.

Etape 3: Trouver le plus proche voisin \mathbf{v} de \mathbf{z} dans le réseau E_8 .

Etape 4: Calculer $y_i = y'_i - m_i v_i$ ($i = 1, 2, \dots, 8$) et obtenir le mot de code

$$\mathbf{y} = [y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8].$$

Etape 5: Fin.

• Exemple $\mathbf{m} = [9, 7, 5, 5, 5, 3, 3, 3], \quad \mathbf{y} = [-4, -2, 0, -1, 0, -1, -1, 1].$

$\mathbf{y} \rightarrow \mathbf{k}$:

$$\mathbf{j} = \mathbf{yG}^{-1} = [3, -3, -1, -2, -1, -2, -2, 2],$$

$$\mathbf{u} = [1, 1, 1, 1, 1, 1, 1, 0], \quad w = 0,$$

$$\mathbf{k}' = [16, 4, 4, 3, 4, 1, 1, 2],$$

$$\mathbf{k} = [7, 4, 4, 3, 4, 1, 1, 2].$$

$\mathbf{k} \rightarrow \mathbf{y}$:

$$\mathbf{y}' = \mathbf{kG} = [32, 5, 5, 4, 5, 2, 2, 1],$$

$$\mathbf{z} = [\frac{32}{9}, \frac{5}{7}, \frac{5}{5}, \frac{4}{5}, \frac{5}{5}, \frac{2}{3}, \frac{2}{3}, \frac{1}{3}],$$

$$\mathbf{v} = [4, 1, 1, 1, 1, 1, 1, 0],$$

$$\mathbf{y} = [-4, -2, 0, -1, 0, -1, -1, 1].$$

1.3.3 Résultats expérimentaux

Dans ce paragraphe, nous présentons des résultats de simulation et la comparaison des performances entre différents quantificateurs vectoriels de façon à évaluer le quantificateur vectoriel ellipsoïdal proposé.

La simulation porte sur un signal gaussien transformé par la transformée de KL. L'entrée du quantificateur est un vecteur aléatoire à huit composantes distribuées selon la loi gaussienne à moyenne nulle de variances égales à 18^2 , 13^2 , 9^2 , 6^2 , 4^2 , 3^2 , 2^2 et 2^2 . Evidemment, la distribution du signal à quantifier possède une forme d'ellipsoïde.

Dans la simulation, on utilise un ensemble de 50000 vecteurs comme base d'apprentissage et un autre de 30000 vecteurs comme données de test.

Tout d'abord, le signal est quantifié par deux quantificateurs vectoriels ellipsoïdaux basés sur les réseaux réguliers D_g et E_g . Tenant compte des variances des différentes composantes du signal, on choisit le même vecteur $\mathbf{m} = [18, 14, 10, 6, 4, 4, 2, 2]$ pour ces deux quantificateurs. En conséquence, la taille des dictionnaires est d'environ 20 bits ($\prod_{i=1}^8 m_i = 967680 \approx 2^{19.88}$). A l'aide de la base d'apprentissage, on obtient le meilleur facteur d'échelle $\alpha = 0.21$.

Dans le but de la comparaison, on quantifie aussi le signal en utilisant un quantificateur vectoriel algébrique sphérique [19] et un quantificateur vectoriel statistique fendu (en anglais Split Vector Quantizer [20]).

Le quantificateur vectoriel algébrique sphérique est basé sur le réseau E_g . Son dictionnaire se compose essentiellement des 11 premières sphères de E_g [19] et le nombre de mots de code est de $1048477 \approx 2^{20}$. Comme pour le quantificateur vectoriel algébrique ellipsoïdal, un facteur d'échelle optimal est choisi par apprentissage.

Le dictionnaire du quantificateur vectoriel statistique se divise en deux sous-dictionnaires ayant la même taille de 2^{10} . On fend le vecteur d'entrée en deux parties et l'on construit les sous-dictionnaires en utilisant l'algorithme de K -moyennes. D'après l'expérience obtenue sur l'apprentissage, la meilleure façon de fendre le vecteur d'entrée consiste à former un sous-vecteur avec les deux premières composantes et un autre avec les six dernières composantes.

Le tableau 1.5 donne les résultats obtenus sur les données de test.

TABLEAU 1.5
Performances de divers quantificateurs
pour un signal de distribution ellipsoïdale

Quantificateur	RSB (dB)	Débit (bits/dim.)
Ellipsoïdal (E_8)	16.54	2.4855
Ellipsoïdal (D_8)	15.51	2.4855
Sphérique (E_8)	12.82	2.5000
Statistique	16.26	2.5000

Les résultats démontrent que grâce au "shaping", les quantificateurs vectoriels algébriques ellipsoïdaux améliorent beaucoup les performances de quantification en comparaison du quantificateur vectoriel algébrique sphérique. Le quantificateur vectoriel ellipsoïdal basé sur E_8 apporte un gain de forme de 3.72 dB. Et le quantificateur vectoriel ellipsoïdal basé sur D_8 fournit quand même un gain de forme de 2.69 dB en opérant la compensation pour le gain granulaire.

Comme on le sait, E_8 est le meilleur réseau régulier en huit dimensions. On peut voir dans le tableau 1.5 que le réseau E_8 permet d'obtenir un gain granulaire de 1.03 dB par rapport au réseau D_8 . Bien que quelques calculs de la norme quadratique soient requis pour la recherche du plus proche voisin, cela n'est pas un problème important au sens de

la complexité de la réalisation.

Enfinement, on compare le quantificateur vectoriel ellipsoïdal basé sur E_g avec le quantificateur vectoriel statistique fendu. Les performances du premier sont légèrement meilleures que celles de ce dernier. Cependant, le tableau 1.2 montre que le quantificateur vectoriel statistique conduit à une complexité élevée de calcul. En plus, on a besoin d'un grand espace mémoire pour le stockage du dictionnaire.

En résumé, le quantificateur vectoriel algébrique ellipsoïdal proposé est une technique efficace et simple pour la quantification des signaux de distribution ellipsoïdale.

Chapitre 2

Quantification vectorielle algébrique des paramètres LSF

Les paramètres du codage prédictif linéaire (LPC) sont largement utilisés pour représenter l'enveloppe spectrale à court terme du signal de parole. Pour le codage de la parole à bas débit, il est très important de quantifier ces paramètres avec le moins de bits possible. D'autre part, il est également nécessaire de réduire la complexité de la réalisation pour les systèmes pratiques. Dans ce chapitre, on présente d'abord une représentation efficace des paramètres LPC: paramètres LSF. Ensuite on propose deux approches pour la quantification vectorielle des paramètres LSF.

2.1 Quantification de l'information LPC et paramètres LSF

Le codage prédictif linéaire, connu sous le sigle LPC (Linear Predictive Coding), est une technique populaire du traitement de la parole, et il se fonde sur les connaissances de la production de la parole.

L'étude du mécanisme de la phonation montre que le modèle du conduit vocal se comporte comme un filtre tout pôle dont la fonction de transfert est du type suivant [21]:

$$H(z) = \frac{g}{A(z)} = \frac{g}{1 + \sum_{m=1}^p \alpha_m z^{-m}} \quad (2.1)$$

où p est l'ordre du modèle, les coefficients a_m , appelés aussi les coefficients LPC, sont les paramètres du modèle et g est un facteur de gain. La fonction $H(z)$ est appelée le filtre de synthèse et $A(z)$ est appelée le filtre inverse ou encore le filtre d'analyse.

Le signal de parole est donc modélisé par un signal d'excitation attaquant le filtre de synthèse dont les paramètres varient dans le temps. Bien que la nature du signal d'excitation soit variable, elle présente deux situations extrêmes: une série d'impulsions quasi-périodiques (pour les sons voisés) et un bruit blanc (pour les sons non voisés). La figure 2.1 illustre le modèle de production de la parole.

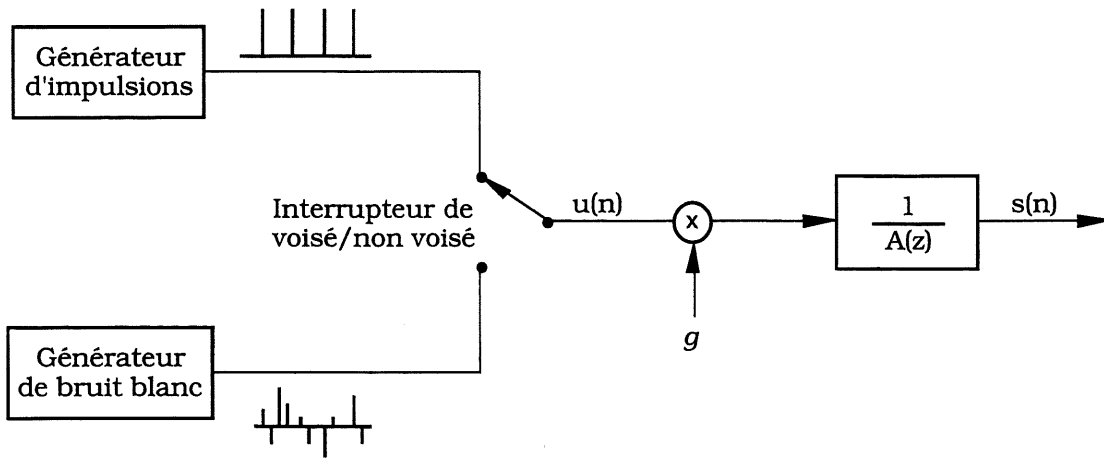


Figure 2.1 Modèle de production de la parole.

Dans le codage prédictif linéaire de la parole, on doit d'abord réaliser une analyse LPC afin de déterminer les coefficients LPC. En général, le signal de parole échantillonné se divise en trames et les coefficients LPC sont estimés sur chaque trame. Il existe de nombreuses méthodes d'estimation: méthode d'autocorrélation, méthode de covariance, méthode de Burg, etc. [22].

Et puis, les coefficients LPC doivent être codés. *La quantification avec qualité transparente* [20] consiste à quantifier l'information LPC sans dégradation perceptuelle.

Pour ce faire, il est habituellement admis qu'il faut satisfaire aux trois conditions suivantes [20]:

- la distorsion spectrale moyenne est d'environ 1 dB;
- la proportion de trames ayant la distorsion spectrale entre 2 et 4 dB est moins que 2%;
- aucune trame n'a une distorsion spectrale supérieure à 4 dB.

Une trame avec la distorsion spectrale supérieure à 2 dB est souvent appelée la trame "outlier" [20].

En pratique, on ne quantifie pas directement les coefficients LPC (les a_m de l'équation (2.1)) parce qu'ils ont de mauvaises propriétés de codage. Dans un grand nombre de systèmes, les coefficients LPC sont transformés en paramètres LSF (Line Spectral Frequency) [23]. Il est prouvé que les paramètres LSF sont une représentation efficace pour la quantification des coefficients LPC [20][24][25][26][27][28][29][30].

Pour un filtre inverse $A(z)$ d'ordre p , deux nouveaux polynômes sont définis comme suit:

$$P(z) = A(z) + z^{-(p+1)}A(z^{-1}) \quad (2.2)$$

et

$$Q(z) = A(z) - z^{-(p+1)}A(z^{-1}). \quad (2.3)$$

Les polynômes $P(z)$ et $Q(z)$ possèdent les principales propriétés suivantes:

- tous les zéros de $P(z)$ et $Q(z)$ sont sur le cercle unité;
- les zéros de $P(z)$ et $Q(z)$ apparaissent de façon alternée sur le cercle unité.

Par conséquent, les racines des polynômes $P(z)$ et $Q(z)$ peuvent s'exprimer par $z = e^{j\omega_m}$ ($m = 1, 2, \dots, p$) et les pulsations ω_m sont appelées les paramètres LSF (Line Spectral Frequency) ou LSP (Line Spectrum Pair).

La transformation de coefficients LPC en paramètres LSF est réversible. Connaissant les paramètres LSF, on peut en déduire les coefficients a_m selon l'équation suivante:

$$A(z) = \frac{P(z) + Q(z)}{2}. \quad (2.4)$$

D'après la deuxième propriété donnée plus haut, les paramètres ω_m ($m = 1, 2, \dots, p$) vérifient les relations

$$0 < \omega_1 < \omega_2 < \dots < \omega_p < \pi. \quad (2.5)$$

L'ordre des paramètres LSF indiqué par l'expression (2.5) est la condition nécessaire et suffisante pour la stabilité du filtre de synthèse $H(z)$ [24], et doit être conservé dans la quantification des paramètres LSF. Par conséquent, si l'on quantifie la différence entre deux paramètres successifs, les relations (2.5) sont automatiquement conservées.

A cause des propriétés des paramètres LSF, ils sont largement utilisés pour la quantification de l'information LPC. Dans ce contexte, diverses approches ont été proposées durant ces dernières années et la quantification avec qualité transparente est obtenue par quelques quantificateurs [20][28][29][30]. Cependant un inconvénient principal de ces techniques est la complexité élevée de la réalisation en matière de stockage et de calcul, car pour la quantification transparente de l'information LPC un quantificateur vectoriel a généralement besoin d'un grand nombre de bits. Ce problème est souvent résolu au prix de la réduction des performances du quantificateur.

2.2 Quantification vectorielle par transformée des paramètres LSF

Nous proposons ici une approche pour la quantification vectorielle des paramètres LSF. Cette approche combine le codage par transformée et la quantification vectorielle par réseaux réguliers. Un quantificateur à 29 bits/trame permet d'obtenir une qualité transparente de la quantification des paramètres LSF.

2.2.1 Codage par transformée

Le codage par transformée est une importante technique de compression [2][16]. Dans cette technique, le signal est représenté par une transformation orthogonale et codé dans le domaine transformé.

Soit $\mathbf{x} = [x_1, x_2, \dots, x_N]^T$ un vecteur signal et \mathbf{A} une matrice orthogonale (en général, unitaire) de dimension $N \times N$. Une transformation orthogonale d'ordre N peut s'exprimer sous forme matricielle:

$$\mathbf{X} = \mathbf{A}\mathbf{x} \quad (2.6)$$

avec

$$\mathbf{A}^{-1} = \mathbf{A}^T. \quad (2.7)$$

La matrice \mathbf{A} s'appelle la matrice de transformée et les éléments du vecteur transformé \mathbf{X} s'appellent les coefficients de transformée.

Dans le codage par transformée, la redondance du signal est exploitée afin de réduire le débit pour une qualité spécifiée de la numérisation. Cette technique présente de grands avantages: chaque coefficient de transformée est codé par des bits différents, de sorte que la haute précision de codage est toujours placée là où elle est nécessaire dans le domaine transformé. En fait, les coefficients dont l'énergie est nulle ou faible ne seront pas codés.

La transformée de Karhunen-Loève (KL) est une transformation optimale pour une distribution gaussienne. Elle maximise le gain de codage par rapport à d'autres

transformations orthogonales comme la transformée de Fourier Discrète [16].

Soit $\mathbf{C}_x = E\{\mathbf{x}\mathbf{x}^T\}$ la matrice de covariance du signal \mathbf{x} , λ_i ($i = 1, 2, \dots, N$) les valeurs propres de \mathbf{C}_x et \mathbf{u}_i ($i = 1, 2, \dots, N$) les vecteurs propres correspondants. La matrice de transformée de KL est alors définie comme

$$\mathbf{A}_{KL} = \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{u}_N \end{bmatrix}, \quad (2.8)$$

c'est-à-dire que les lignes de la matrice de transformée \mathbf{A}_{KL} sont les vecteurs propres de la matrice de covariance \mathbf{C}_x .

La matrice de covariance du vecteur transformée \mathbf{X} est diagonale et elle est calculée par

$$\mathbf{C}_X = E\{\mathbf{X}\mathbf{X}^T\} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_N), \quad (2.9)$$

c'est-à-dire que les variances des coefficients de transformée sont égales aux valeurs propres de la matrice de covariance \mathbf{C}_x .

Si l'on classe les vecteurs propres de façon à ce que les valeurs propres correspondantes apparaissent dans l'ordre décroissant:

$$\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_N, \quad (2.10)$$

on peut voir que la transformée de KL concentre le maximum d'énergie dans ses premiers coefficients de transformée.

Comme on le cite au chapitre précédent, pour une source gaussienne, la

transformée de KL produit des coefficients de transformée dont la densité de probabilité conjointe possèdent une forme d'ellipsoïde [15]. Généralement, ces coefficients sont quantifiés de façon scalaire. Dans l'approche proposée ici, on utilise la quantification vectorielle par réseaux réguliers. Plus précisément, on applique le quantificateur vectoriel algébrique ellipsoïdal proposé au chapitre précédent sur le vecteur transformé.

2.2.2 Quantification vectorielle ellipsoïdale des paramètres LSF

Dans cette étude, la base de données de parole utilisée est basée sur des phrases prononcées dans sept langues différentes. Après passage dans un filtre passe-bande de 200 à 3400 Hz, le signal de parole est échantillonné à 8000 Hz. Une analyse LPC d'ordre 10 est réalisée sur des fenêtres de 24 ms par la méthode d'autocorrélation. Une expansion fréquentielle de 60 Hz est utilisée pour éviter des pics spectraux pointus dans l'analyse LPC. Au total, la base de données se compose de 93500 vecteurs de paramètre LSF. Les 70000 premiers vecteurs sont choisis comme base d'apprentissage et le reste est utilisé pour le test.

Comme le montre le chapitre précédent, la quantification vectorielle par réseaux réguliers est une méthode efficace et simple. Elle permet de réduire très fortement la charge de calcul et le dictionnaire n'a pas besoin d'être stocké. Par conséquent, nous cherchons à utiliser cette technique pour quantifier les paramètres LSF. Mais les paramètres LSF ne possèdent pas de distributions géométriques régulières. Si on les quantifie directement en utilisant un quantificateur vectoriel par réseaux réguliers, on ne peut pas obtenir de résultats satisfaisants.

La transformée de KL permet d'obtenir des coefficients de transformée distribués en forme d'ellipsoïde ramenée selon les axes. La figure 2.2 montre les distributions conjointes des coefficients de transformée de la base de données décrite plus haut. Donc, nous choisissons le quantificateur vectoriel algébrique ellipsoïdal proposé au chapitre précédent pour quantifier efficacement les paramètres LSF dans le domaine transformé.

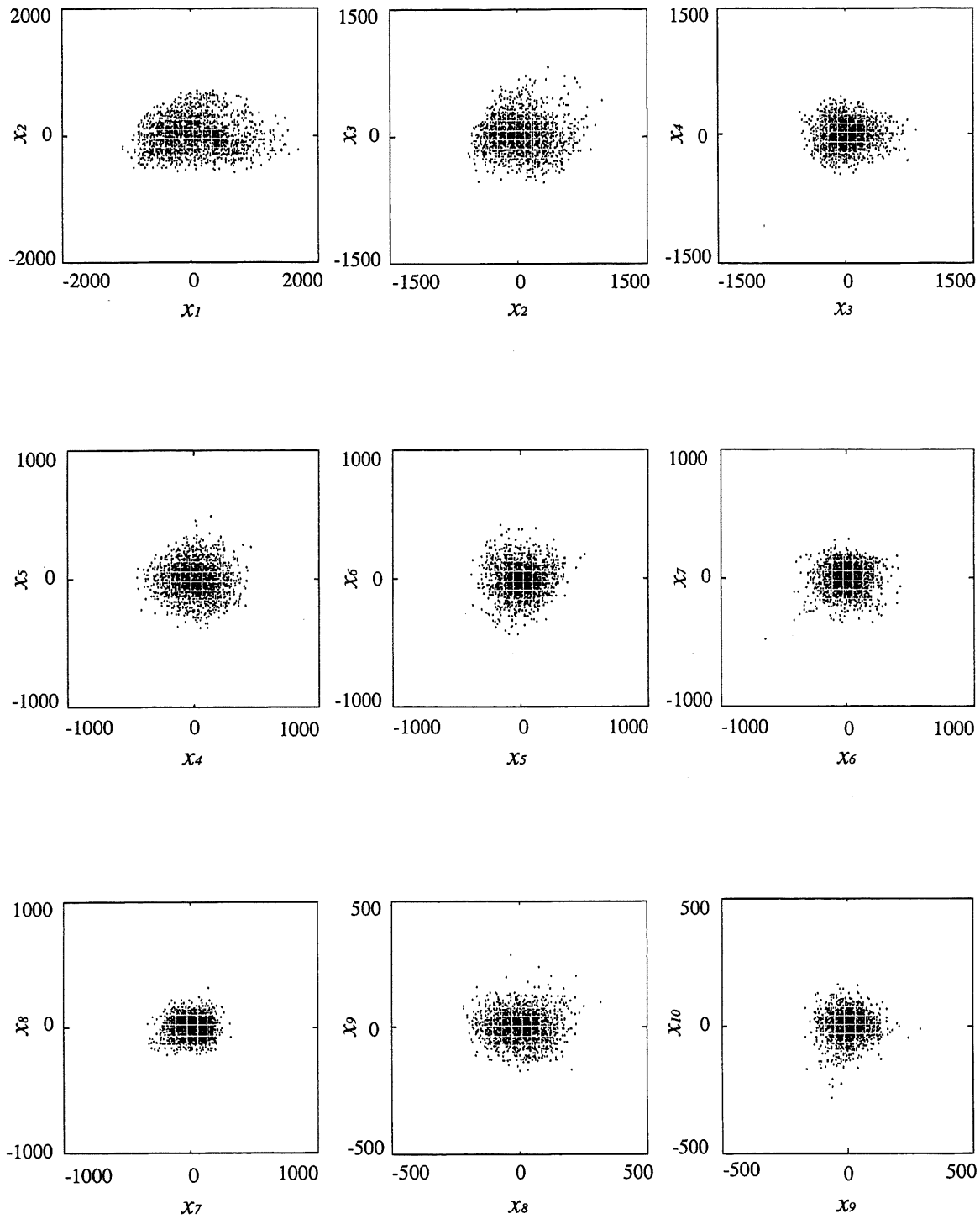


Figure 2.2 Distributions conjointes des coefficients de transformée de KL pour les paramètres LSF.

Etant donné la densité de probabilité non uniforme des coefficients de transformée, on utilise l'union de plusieurs sous-dictionnaires algébriques ellipsoïdaux concentriques afin d'adapter la répartition des mots de code à la densité de probabilité. Plus précisément, à l'aide des facteurs d'échelle on agrandit proportionnellement un dictionnaire ellipsoïdal de base centré sur l'origine de façon à construire des sous-dictionnaires. La densité des points (mots de code) est donc de plus en plus grande à mesure que l'on approche de l'origine. Cette structure du dictionnaire permet de tirer profit du gain de densité non uniforme décrit au chapitre précédent. La figure 2.3 illustre le contour d'un dictionnaire ellipsoïdal avec trois facteurs d'échelle.

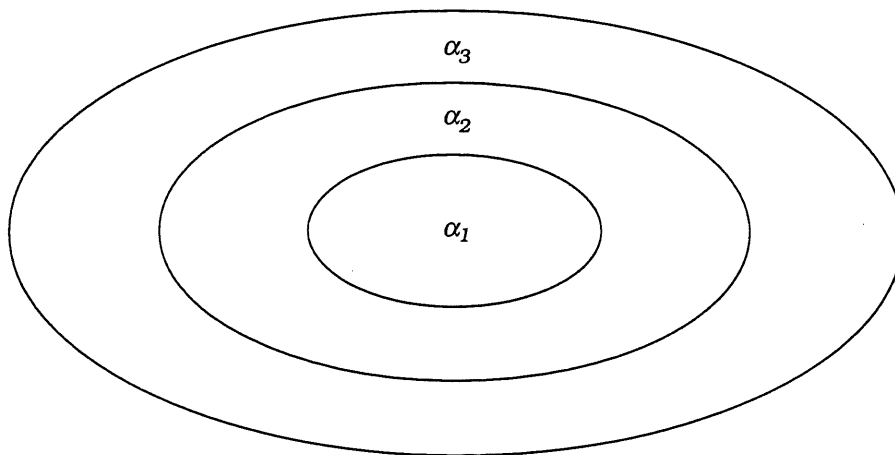


Figure 2.3 Contour d'un dictionnaire ellipsoïdal avec trois facteurs d'échelle.

En pratique, on réduit proportionnellement les coefficients de transformée au lieu d'agrandir le dictionnaire ellipsoïdal de base afin d'exploiter les algorithmes rapides proposé au chapitre précédent. Les meilleurs facteurs d'échelle peuvent être déterminés expérimentalement sur la base d'apprentissage.

Soit $\mathbf{m} = [m_1, m_2, \dots, m_{10}]$ le vecteur spécifiant un sous-quantificateur ellipsoïdal de base et α_j ($j = 1, 2, \dots, K; 0 < \alpha_1 < \alpha_2 < \dots < \alpha_K$) des facteurs d'échelle. Pour chaque vecteur transformé \mathbf{X} , la quantification vectorielle algébrique ellipsoïdale avec plusieurs facteurs d'échelle s'effectue comme suit:

- Réduire \mathbf{X} par le facteur d'échelle α_1 et trouver le plus proche voisin \mathbf{Y} de $\alpha_1 \mathbf{X}$ dans le réseau régulier utilisé;
- Si \mathbf{Y} n'appartient pas au dictionnaire ellipsoïdal de base, réduire successivement \mathbf{X} par les facteurs d'échelle α_j ($j = 2, 3, \dots, K$) jusqu'à ce que le plus proche voisin \mathbf{Y} de $\alpha_j \mathbf{X}$ se trouve dans ce dictionnaire;
- Si \mathbf{Y} de $\alpha_K \mathbf{X}$ est toujours en dehors du dictionnaire de base après l'étape dernière, réduire \mathbf{x} homothétiquement jusqu'à ce que son plus proche voisin appartienne au dictionnaire de base. Dans ce cas, $\alpha_j = \alpha_K$;
- Agrandir proportionnellement \mathbf{Y} par l'inverse du facteur choisi afin d'obtenir le vecteur de reproduction de \mathbf{X} .

Dans cette approche, le dictionnaire du quantificateur vectoriel algébrique ellipsoïdal est basé sur le réseau régulier E_{10} . E_{10} est défini comme

$$E_{10} = (D_{10}) \cup (D_{10} + \{\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}\}). \quad (2.11)$$

Les algorithmes de codage et de décodage pour E_{10} sont essentiellement semblables à ceux pour E_8 , sauf que l'inverse de la matrice génératrice G est défini par

$$G^1 = \begin{bmatrix} \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{1}{2} & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{1}{2} & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{1}{2} & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ -\frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \frac{7}{2} & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 2 \end{bmatrix} \quad (2.12)$$

et l'équation (1.24) est remplacée par

$$k'_1 = j_1 + \frac{1}{2} \sum_{i=2}^{10} (u_i (m_i - m_i)) + 4u_{10}m_{10}. \quad (2.13)$$

Pour construire le dictionnaire ellipsoïdal, on doit d'abord analyser la caractéristique statistique des coefficients de transformée des paramètres LSF. Dans un premier temps, on calcule la matrice de covariance de la base d'apprentissage afin d'obtenir la matrice de transformée et les variances σ_i^2 ($i = 1, 2, \dots, 10$) des coefficients de transformée. Les résultats obtenus sont donnés respectivement dans les tableaux 2.1 et 2.2. La figure 2.4 illustre les écarts types normalisés σ_{Ni} des coefficients de transformée.

D'après l'expérience obtenue sur la base d'apprentissage, on détermine le vecteur $\mathbf{m} = [30, 16, 12, 10, 6, 6, 4, 4, 2, 2]$ qui spécifie la taille du dictionnaire ellipsoïdal. Donc, on obtient un quantificateur vectoriel algébrique ellipsoïdal de base à 27 bits ($\prod_{i=1}^{10} m_i = 132710400 \approx 2^{26.98}$) et il est utilisé pour quantifier les paramètres LSF dans le domaine transformé.

TABLEAU 2.1
Matrice de transformée de KL des paramètres LSF

0.0918	0.2155	0.4240	0.4489	0.4700	0.3974	0.2939	0.2430	0.1712	0.1015
-0.2744	-0.4677	-0.5797	-0.1244	0.3298	0.4049	0.2447	0.1440	0.0318	-0.0065
0.0438	0.1157	0.0752	-0.4431	-0.4536	0.1002	0.3945	0.3869	0.4183	0.2942
-0.1937	-0.3682	-0.0698	0.5581	-0.1858	-0.2796	-0.0906	-0.0756	0.4397	0.4366
0.1050	0.1683	-0.3062	0.1471	0.2161	-0.6592	0.3205	0.4830	-0.1461	-0.0814
-0.1938	-0.3388	0.2693	0.2824	-0.4691	0.0700	0.4307	0.0836	-0.4231	-0.3144
0.4049	0.2805	-0.4179	0.2271	-0.1858	0.1594	0.4145	-0.4960	-0.1166	0.1997
-0.3776	-0.0188	0.2783	-0.2899	0.2856	-0.2473	0.3233	-0.3312	-0.2950	0.5017
0.2356	-0.0910	-0.0529	0.0660	-0.1121	0.1995	-0.3302	0.3990	-0.5417	0.5612
0.6828	-0.5962	0.2289	-0.1838	0.1861	-0.1553	0.1277	-0.0983	0.0829	-0.0226

TABLEAU 2.2
Ecart types des
coefficients de transformée

i	σ_i
1	457.2516
2	253.9828
3	198.5686
4	148.0437
5	122.7929
6	117.2625
7	101.3732
8	77.6448
9	55.0727
10	50.5620

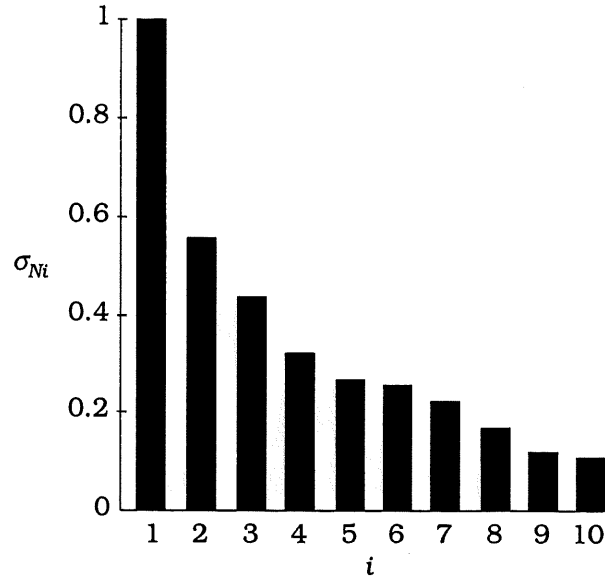


Figure 2.4 Ecarts types normalisés des coefficients de transformée.

La base de test mentionnée au début de ce paragraphe est utilisée pour évaluer les performances du quantificateur. On choisit la distorsion spectrale (SD) comme mesure de distorsion et elle est définie par

$$SD = \left[\frac{1}{2\pi} \int_0^{2\pi} (10 \log H(\omega) - 10 \log \hat{H}(\omega))^2 d\omega \right]^{\frac{1}{2}} \quad (2.14)$$

où $H(\omega)$ et $\hat{H}(\omega)$ sont respectivement le spectre original et le spectre quantifié relatifs à une trame. Dans cette étude, la distorsion spectrale est mesurée sur des spectres en expansion fréquentielle et est calculée dans toute la plage fréquentielle de 0 à 4000 Hz.

Les résultats de test montrent que pour obtenir une qualité transparente de la quantification des paramètres LSF, quatre facteurs d'échelle sont requis. Autrement dit, un quantificateur vectoriel algébrique ellipsoïdal à 29 bits/trame permet d'apporter une distorsion spectrale moyenne d'environ 1 dB, moins de 2% de trames "outliers" dans la

région de 2 à 4 dB et aucune trame "outlier" ayant une distorsion spectrale supérieure à 4 dB. Les quatre facteurs d'échelle sont déjà optimisés sur la base d'apprentissage avant le test et ils sont respectivement égaux à 0.016, 0.020, 0.025 et 0.030. Le tableau 2.3 montre les performances en terme de distorsion spectrale (SD) du quantificateur vectoriel algébrique ellipsoïdal à 29 bits/trame et la figure 2.5 donne l'histogramme des distorsions spectrales pour les résultats de test.

TABLEAU 2.3
Performances (SD) du quantificateur
vectoriel algébrique ellipsoïdal à 29 bits/trame

SD (dB)	"Outliers" (%)	
	2-4 dB	>4 dB
1.03	1.98	0

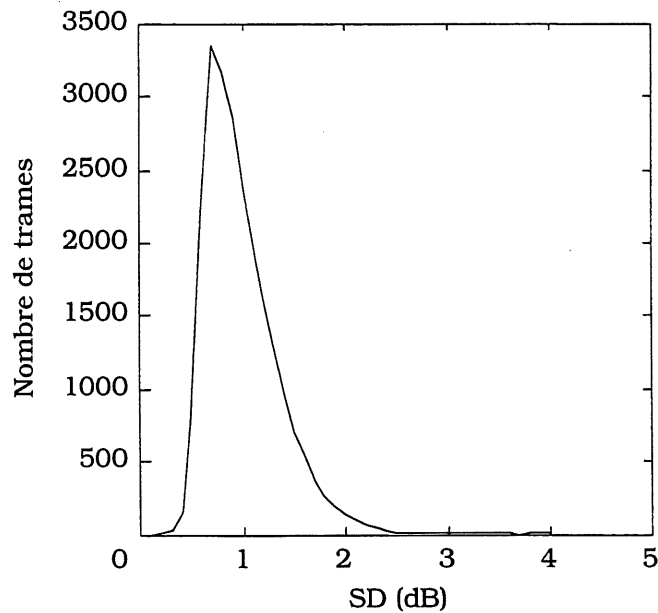


Figure 2.5 Histogramme des distorsions spectrales (SD) pour le quantificateur vectoriel algébrique ellipsoïdal à 29 bits/trame.

Pour faire la comparaison, on conçoit aussi un quantificateur vectoriel fendu à 29 bits avec la même base d'apprentissage. Les paramètres LSF sont quantifiés directement par ce quantificateur. Selon la référence [20], le vecteur de coefficient se fend en trois parties: les trois premiers paramètres LSF, les trois paramètres LSF suivants et les quatre derniers paramètres LSF. Les dictionnaires des trois sous-quantificateurs correspondants contiennent respectivement 1024, 1024 et 512 mots de code. Ce quantificateur vectoriel fendu est évalué sur la même base de test et les résultats obtenus sont donnés dans le tableau 2.4.

TABLEAU 2.4
Performances (SD) du quantificateur
vectoriel fendu en (3, 3, 4) à 29 bits/trame

SD (dB)	"Outliers" (%)	
	2-4 dB	>4 dB
1.01	1.64	0

En comparant les deux quantificateurs vectoriels, on peut voir que bien que les performances du quantificateur vectoriel fendu soient légèrement meilleures que celles du quantificateur vectoriel algébrique ellipsoïdal, le quantificateur vectoriel fendu a besoin d'une mémoire de taille 8.2K pour le stockage du dictionnaire et sa charge de calcul est plus lourde que celle du quantificateur vectoriel algébrique ellipsoïdal. Le quantificateur vectoriel algébrique ellipsoïdal fournit également une qualité transparente de quantification pour le même débit, mais il permet une importante réduction de la complexité de la réalisation.

2.3 Quantification vectorielle par différence des paramètres LSF

Une approche algébrique de la quantification vectorielle des paramètres LSF est présentée dans cette section. Cette technique consiste à appliquer la quantification vectorielle par réseaux réguliers de faibles dimensions au vecteur différentiel des paramètres LSF. Des dictionnaires avec "shaping" sont utilisés pour obtenir de bonnes performances de quantification. Les mots de code sont indexés de façon algébrique. Un quantificateur global à 28 bits/trame garantit une qualité transparente de la quantification des paramètres LSF.

2.3.1 Quantification du vecteur différentiel des paramètres LSF

Cette étude est basée sur la même base de données des paramètres LSF utilisée dans la section précédente.

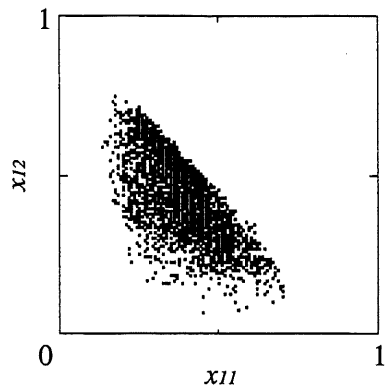
Dans la section 2.1, on montre que les paramètres LSF sont ordonnés dans l'intervalle fini $(0, \pi)$. L'approche proposée ici profite de cette propriété. Deux paramètres LSF particuliers, ω_3 et ω_7 , sont, pour ainsi dire, les "points d'ancrage" de l'approche comme on le verra bientôt. On exploite la propriété particulière de la densité conjointe de ω_3 et ω_7 et l'on quantifie ces deux paramètres LSF en utilisant un quantificateur vectoriel statistique conçu par l'algorithme de *K*-moyennes. Ce quantificateur statistique atteint un double objectif:

- il spécifie les paramètres LSF, ω_3 et ω_7 , avec leurs valeurs quantifiées $\hat{\omega}_3$ et $\hat{\omega}_7$;
- il permet de partitionner la quantification des huit paramètres LSF restants en trois tâches indépendantes de quantification vectorielle afin de réduire considérablement la complexité de la réalisation. Plus précisément, les huit paramètres LSF restants sont divisés en trois groupes: la paire $[\omega_1, \omega_2]$, le triplet $[\omega_4, \omega_5, \omega_6]$ et le triplet $[\omega_8, \omega_9, \omega_{10}]$.

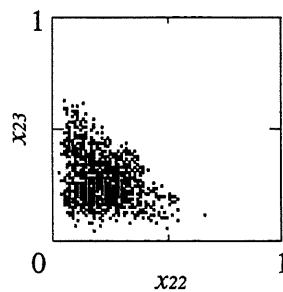
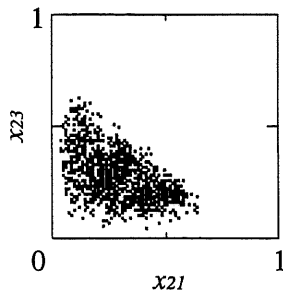
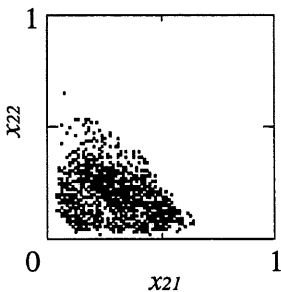
En réalité, grâce à la propriété des paramètres LSF, le décodeur connaît les frontières 0 et ω_3 pour la paire ordonnée $[\omega_1, \omega_2]$, les frontières ω_3 et ω_7 pour le triplet ordonné $[\omega_4, \omega_5, \omega_6]$ et les frontières ω_7 et π pour le triplet ordonné $[\omega_8, \omega_9, \omega_{10}]$. Comme on le sait, quantifier la différence entre deux paramètres successifs permet d'assurer la stabilité du filtre de synthèse. Pour faciliter la conception des quantificateurs vectoriels, on calcule la différence entre deux composantes adjacentes de la paire et des triplets et l'on normalise ces différences par la longueur des intervalles respectifs. D'ailleurs, le vecteur $\mathbf{x}_1 = \left[\frac{\omega_1}{\omega_3}, \frac{\omega_3 - \omega_2}{\omega_3} \right]$, qui est équivalent à la paire originale $[\omega_1, \omega_2]$ au sens de la quantification, se borne à un triangle de taille fixe. D'une façon similaire, les vecteurs $\mathbf{x}_2 = \left[\frac{\omega_4 - \omega_3}{\omega_7 - \omega_3}, \frac{\omega_5 - \omega_4}{\omega_7 - \omega_3}, \frac{\omega_7 - \omega_6}{\omega_7 - \omega_3} \right]$ et $\mathbf{x}_3 = \left[\frac{\omega_8 - \omega_7}{\pi - \omega_7}, \frac{\omega_9 - \omega_8}{\pi - \omega_7}, \frac{\pi - \omega_{10}}{\pi - \omega_7} \right]$ peuvent se borner à une pyramide de taille fixe.

Pour notre base de données de paramètres LSF, la figure 2.6 montre les distributions conjointes des composantes des vecteurs \mathbf{x}_1 , \mathbf{x}_2 et \mathbf{x}_3 . On peut voir que la distribution conjointe des composantes du vecteur \mathbf{x}_1 possède une forme de triangle rectangulaire isocèle et les distributions conjointes des composantes des vecteurs \mathbf{x}_2 et \mathbf{x}_3 occupent une pyramide triangulaire.

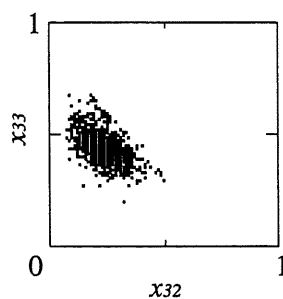
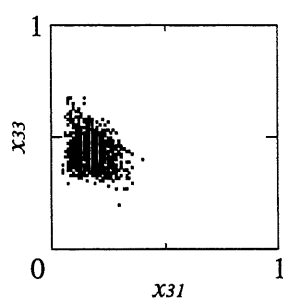
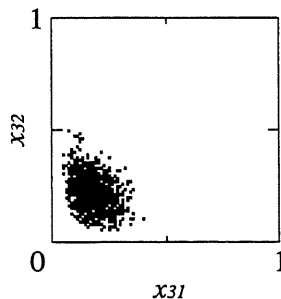
Cette situation est idéale pour l'utilisation de la quantification vectorielle algébrique car la région d'espace occupée par ces vecteurs différentiels normalisés est très simple et régulière. L'indexation des mots de code deviendra donc facile et, ce qui est plus important, la technique du "shaping" peut être utilisée afin d'obtenir le gain de forme décrit au chapitre précédent. Trois quantificateurs vectoriels algébriques sont donc utilisés pour quantifier les vecteurs différentiels normalisés \mathbf{x}_1 , \mathbf{x}_2 et \mathbf{x}_3 . Dans ce contexte, on appelle le quantificateur vectoriel en deux dimensions le *quantificateur vectoriel triangulaire* et le quantificateur vectoriel en trois dimensions le *quantificateur vectoriel pyramidal*.



(a)



(b)



(c)

Figure 2.6 Distributions conjointes des composantes des vecteurs différentiels normalisés.

Afin de minimiser la distorsion totale de quantification, les trois vecteurs à quantifier, $\mathbf{x}_1 = [x_{11}, x_{12}]$, $\mathbf{x}_2 = [x_{21}, x_{22}, x_{23}]$ et $\mathbf{x}_3 = [x_{31}, x_{32}, x_{33}]$, sont définis comme suit à partir des quantités précédemment quantifiées

$$\begin{cases} x_{11} = \frac{\omega_1}{\hat{\omega}_3} \\ x_{12} = \frac{\hat{\omega}_3 - \omega_2}{\hat{\omega}_3} \end{cases}, \quad (2.15)$$

$$\begin{cases} x_{21} = \frac{\omega_4 - \hat{\omega}_3}{\hat{\omega}_7 - \hat{\omega}_3} \\ x_{22} = \frac{\omega_5 - \hat{\omega}_4}{\hat{\omega}_7 - \hat{\omega}_3} \\ x_{23} = \frac{\hat{\omega}_7 - \omega_6}{\hat{\omega}_7 - \hat{\omega}_3} \end{cases} \quad \text{avec} \quad \hat{\omega}_4 = \hat{x}_{21}(\hat{\omega}_7 - \hat{\omega}_3) + \hat{\omega}_3 \quad (2.16)$$

et

$$\begin{cases} x_{31} = \frac{\omega_8 - \hat{\omega}_7}{\pi - \hat{\omega}_7} \\ x_{32} = \frac{\omega_9 - \hat{\omega}_8}{\pi - \hat{\omega}_7} \\ x_{33} = \frac{\pi - \omega_{10}}{\pi - \hat{\omega}_7} \end{cases} \quad \text{avec} \quad \hat{\omega}_8 = \hat{x}_{31}(\pi - \hat{\omega}_7) + \hat{\omega}_7. \quad (2.17)$$

On remarque que les composantes x_{ij} ($i, j = 1, 2, 3$) des vecteurs différentiels normalisés doivent être non négatives au sens de la stabilité. Cependant il est possible que les valeurs négatives de x_{ij} résultent des expressions citées ci-dessus à cause de l'utilisation des paramètres quantifiés. Le cas échéant, on impose $x_{ij} = 0$.

2.3.2 Conception du quantificateur vectoriel triangulaire

Le dictionnaire du quantificateur vectoriel triangulaire est construit par la troncation d'un réseau régulier et ses mots de code forment un triangle rectangulaire isocèle dans le plan. Les composantes de chaque mot de code sont non négatives. La somme de composantes est un nombre entier pair (ou impair) et n'est pas supérieure à une

quantité donnée S . A titre d'exemple la figure 2.7 montre un dictionnaire triangulaire de taille 64 basé sur le réseau D_2 , pour lequel chaque mot de code a une somme paire de composantes et la quantité S est égale à 14.

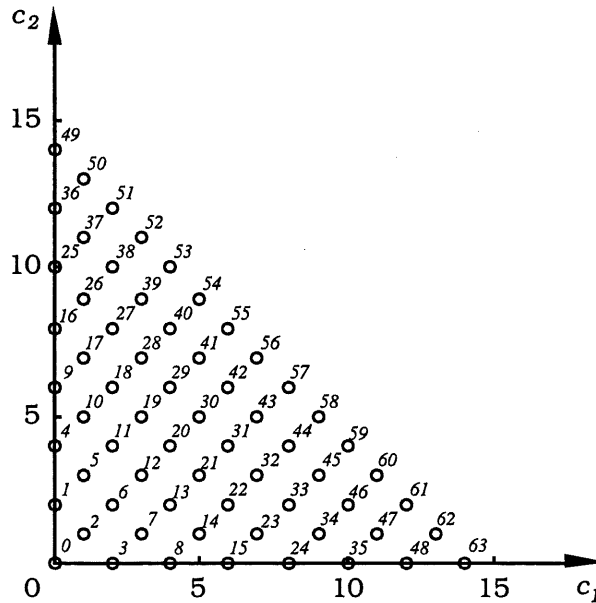


Figure 2.7 Dictionnaire d'un quantificateur vectoriel triangulaire basé sur D_2 ($S = 14$).

Pour adapter ce dictionnaire triangulaire à la distribution conjointe du vecteur différentiel \mathbf{x}_1 , on décale \mathbf{x}_1 par $\boldsymbol{\beta}_1 = [\beta_{11}, \beta_{12}]$ et on l'agrandit par un facteur d'échelle α_1 . Pour un débit donné, les valeurs optimales de α_1 et $\boldsymbol{\beta}_1$ sont obtenues expérimentalement sur la base d'apprentissage.

Le dictionnaire triangulaire n'a pas à être stocké et ses mots de code sont indexés par une règle algébrique. Pour l'exemple illustré à la figure 2.7, les mots de code sont indexés de 0 à 63. Ce dictionnaire est divisé en huit "niveaux" selon la somme de composantes des mots de code. On peut voir à la figure 2.7 que les niveaux 0, 1, 2, ..., 7

contiennent respectivement 1, 3, 5, ..., 15 mots de code. Evidemment, le nombre des mots de code constitue une série arithmétique impaire. A chaque niveau, l'indice k_0 du premier mot de code est égal au nombre de mots de code qui se trouvent aux niveaux inférieurs. Par conséquent, on peut obtenir l'indice de tout mot de code en calculant la somme de k_0 et du rang du mot de code dans son niveau. Comme on le sait, la n ième somme partielle de la série arithmétique impaire vérifie la relation

$$1 + 3 + 5 + 7 + \dots + (2n - 1) = n^2 . \quad (2.18)$$

L'indice k d'un mot de code $\mathbf{c} = [c_1, c_2]$ est donc donné par

$$k = k_0 + c_1 = \left(\frac{c_1 + c_2}{2} \right)^2 + c_1 . \quad (2.19)$$

Il est aussi possible d'utiliser comme mots de code les vecteurs dont la somme de composantes est impaire. Dans ce cas, la méthode d'indexation des mots de code est légèrement différente. Le nombre de mots de code situés aux niveaux successifs forme une série arithmétique paire de sorte que l'indice k d'un mot de code \mathbf{c} est donné par

$$k = k_0 + c_1 = \frac{(c_1 + c_2)^2 - 1}{4} + c_1 . \quad (2.20)$$

Le tableau 2.5 montre les diverses tailles du dictionnaire triangulaire correspondant à la gamme de 4 à 9 bits.

TABLEAU 2.5
Diverses tailles du
dictionnaire triangulaire

Bits	Taille	S
4	16	6
5	30	9
6	64	14
7	121	20
8	256	30
9	506	43

Généralement, on peut trouver le plus proche voisin d'un vecteur arbitraire \mathbf{x} en utilisant les algorithmes décrits dans la section 1.1. En tant que moyen alternatif, une procédure rapide de recherche est présentée maintenant. A l'aide la matrice de transformée suivante

$$T = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}, \quad (2.21)$$

le vecteur \mathbf{x} est tourné d'un angle de $\frac{\pi}{4}$ radian dans le sens contraire des aiguilles d'une montre. Puis, chaque composante du vecteur tourné est arrondie au plus proche nombre entier pair (ou impair) pour obtenir le meilleur "mot de code tourné". Par la transformée inverse, on trouve facilement le plus proche voisin qui satisfait à la condition de parité paire (ou impaire), c'est-à-dire une somme paire (ou impaire) de composantes du mot de code. En réalité, cette transformée de rotation et son inverse sont très simples et requièrent seulement des opérations d'addition. La figure 2.8 montre le schéma de principe du quantificateur vectoriel triangulaire.

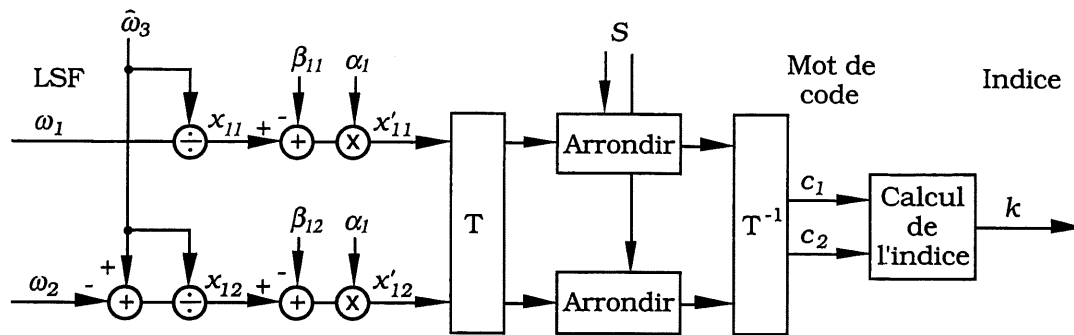


Figure 2.8 Schéma de principe du quantificateur vectoriel triangulaire.

En décodage, on extrait d'abord k_0 de l'indice k reçu et ensuite on en déduit c_1 et c_2 d'après l'expression (2.19) ou (2.20).

2.3.3 Conception du quantificateur vectoriel pyramidal

Le dictionnaire du quantificateur vectoriel pyramidal est un réseau régulier tronqué en trois dimensions et n'a pas besoin d'être stocké comme celui du quantificateur vectoriel triangulaire. Par exemple, la figure 2.9 illustre un dictionnaire pyramidal de taille 125 basé sur le réseau D_3 (dont la somme des coordonnées des points est impaire). Dans ce cas, chaque mot de code du dictionnaire a une somme impaire de composantes et la quantité S est égale à 9. Le tableau 2.6 montre les mots de code de ce dictionnaire et leur indice.

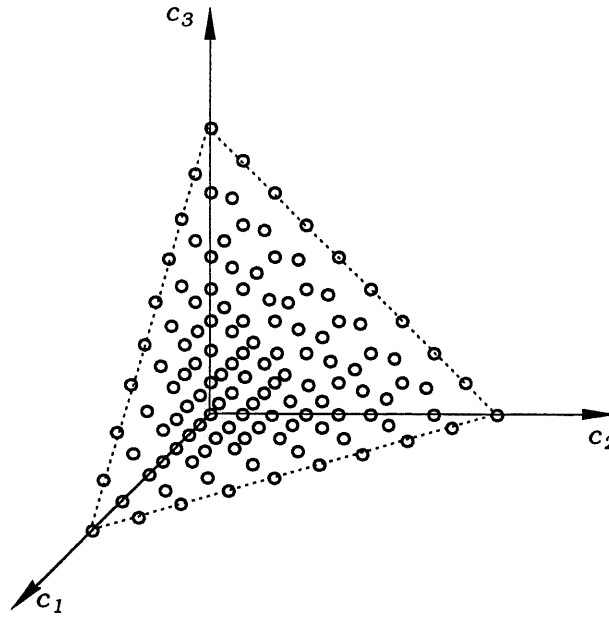


Figure 2.9 Dictionnaire d'un quantificateur vectoriel pyramidal basé sur D'_3 ($S=9$).

TABLEAU 2.6
Mots de code du dictionnaire pyramidal
basé sur D'_3 ($S=9$) et leur indice (en italique)

0	1 0 0	25	1 2 2	50	2 1 4	75	7 2 0	100	2 2 5
1	0 0 1	26	1 3 1	51	2 2 3	76	6 0 3	101	2 3 4
2	0 1 0	27	1 4 0	52	2 3 2	77	6 1 2	102	2 4 3
3	3 0 0	28	0 0 5	53	2 4 1	78	6 2 1	103	2 5 2
4	2 0 1	29	0 1 4	54	2 5 0	79	6 3 0	104	2 6 1
5	2 1 0	30	0 2 3	55	1 0 6	80	5 0 4	105	2 7 0
6	1 0 2	31	0 3 2	56	1 1 5	81	5 1 3	106	1 0 8
7	1 1 1	32	0 4 1	57	1 2 4	82	5 2 2	107	1 1 7
8	1 2 0	33	0 5 0	58	1 3 3	83	5 3 1	108	1 2 6
9	0 0 3	34	7 0 0	59	1 4 2	84	5 4 0	109	1 3 5
10	0 1 2	35	6 0 1	60	1 5 1	85	4 0 5	110	1 4 4
11	0 2 1	36	6 1 0	61	1 6 0	86	4 1 4	111	1 5 3
12	0 3 0	37	5 0 2	62	0 0 7	87	4 2 3	112	1 6 2
13	5 0 0	38	5 1 1	63	0 1 6	88	4 3 2	113	1 7 1
14	4 0 1	39	5 2 0	64	0 2 5	89	4 4 1	114	1 8 0
15	4 1 0	40	4 0 3	65	0 3 4	90	4 5 0	115	0 0 9
16	3 0 2	41	4 1 2	66	0 4 3	91	3 0 6	116	0 1 8
17	3 1 1	42	4 2 1	67	0 5 2	92	3 1 5	117	0 2 7
18	3 2 0	43	4 3 0	68	0 6 1	93	3 2 4	118	0 3 6
19	2 0 3	44	3 0 4	69	0 7 0	94	3 3 3	119	0 4 5
20	2 1 2	45	3 1 3	70	9 0 0	95	3 4 2	120	0 5 4
21	2 2 1	46	3 2 2	71	8 0 1	96	3 5 1	121	0 6 3
22	2 3 0	47	3 3 1	72	8 1 0	97	3 6 0	122	0 7 2
23	1 0 4	48	3 4 0	73	7 0 2	98	2 0 7	123	0 8 1
24	1 1 3	49	2 0 5	74	7 1 1	99	2 1 6	124	0 9 0

Les processus de codage et de décodage du quantificateur vectoriel pyramidal sont semblables à ceux du quantificateur vectoriel triangulaire à bien des égards. Pour l'exemple illustré à la figure 2.9, le dictionnaire pyramidal comprend cinq niveaux qui ont respectivement 3, 10, 21, 36 et 55 mots de code. La série correspondant au nombre de mots de code est $1 \cdot 3 + 2 \cdot 5 + \dots + n(2n+1)$ et la n ième somme partielle de cette série est donnée par

$$1 \cdot 3 + 2 \cdot 5 + \dots + n(2n+1) = \frac{1}{6} n(n+1)(4n+5). \quad (2.22)$$

Soit $\mathbf{c} = [c_1, c_2, c_3]$ un mot de code quelconque et $s = c_1 + c_2 + c_3$ la somme de composantes de \mathbf{c} . A chaque niveau du dictionnaire pyramidal, l'indice k_0 du premier mot de code peut être calculé par l'expression suivante

$$k_0 = \frac{1}{6} m(m+1)(4m+5) \quad (2.23)$$

avec

$$m = \frac{s-1}{2}.$$

Donc, l'indice k du vecteur \mathbf{c} est donné par

$$k = k_0 + \frac{(s-c_1)(s-c_1+1)}{2} + c_2. \quad (2.24)$$

Lorsque le dictionnaire pyramidal se compose des vecteurs dont la somme de composantes est paire, le nombre de mots de code est une série $1 \cdot 1 + 2 \cdot 3 + \dots + n(2n-1)$. La n ième somme partielle de la série s'exprime par

$$1 \cdot 1 + 2 \cdot 3 + \dots + n(2n-1) = \frac{1}{6} n(n+1)(4n-1). \quad (2.25)$$

Donc, on a

$$k_0 = \frac{1}{6} m (m + 1)(4m - 1) \quad (2.26)$$

avec

$$m = \frac{S}{2}.$$

Et l'indice k du vecteur \mathbf{c} est aussi donné par l'expression (2.24).

Le tableau 2.7 montre les diverses tailles du dictionnaire pyramidal pour la gamme de 4 à 9 bits. Remarquez que dans ce tableau on obtient le dictionnaire de taille 503 (9 bits) en enlevant les trois niveaux les plus inférieurs d'un dictionnaire original de taille 525.

TABLEAU 2.7
Diverses tailles du
dictionnaire pyramidal

Bits	Taille	S
4	13	3
5	22	4
6	50	6
7	125	9
8	252	12
9	503	16

Dans la quantification vectorielle pyramidale, on a aussi besoin de décaler les vecteurs d'entrée \mathbf{x}_2 et \mathbf{x}_3 et de modifier leur échelle pour les adapter aux dictionnaires pyramidaux respectifs. Deux valeurs de décalage, $\beta_2 = [\beta_{21}, \beta_{22}, \beta_{23}]$ et $\beta_3 = [\beta_{31}, \beta_{32}, \beta_{33}]$, et deux facteurs d'échelle, α_2 et α_3 , sont introduits et leurs valeurs sont déterminées par apprentissage.

La recherche du plus proche voisin du vecteur d'entrée dans le dictionnaire

pyramidal est un peu plus compliquée que celle du dictionnaire triangulaire. Selon les définitions (2.16) et (2.17), on peut voir que les deux premières composantes de \mathbf{x}_2 (ou \mathbf{x}_3) sont corrélées car la différence x_{22} (ou x_{32}) dépend de la valeur quantifiée \hat{x}_{21} (ou \hat{x}_{31}).

Considérons la recherche du plus proche voisin du vecteur \mathbf{x}_2 dans le dictionnaire illustré à la figure 2.9. Après décalage et agrandissement, on obtient le vecteur \mathbf{x}'_2 . Comme chaque mot de code du dictionnaire a une somme impaire de composantes, la condition de parité exige soit une seule composante impaire, soit toutes les trois composantes impaires. Pour minimiser la distorsion totale de quantification au sens de l'erreur quadratique moyenne, on doit considérer ces deux possibilités dans le codage.

D'abord, la composante x'_{21} est arrondie respectivement au plus proche nombre pair \hat{x}'_{21p} et au plus proche nombre impair \hat{x}'_{21i} . Ensuite, on quantifie conjointement la paire $[x'_{22}, x'_{23}]$ en satisfaisant à la condition de parité impaire sur trois composantes. Plus précisément, pour le nombre impair \hat{x}'_{21i} le plus proche voisin de la paire $[x'_{22}, x'_{23}]$ est trouvé dans le réseau D_2 et pour le nombre pair \hat{x}'_{21p} ce plus proche voisin est trouvé dans le réseau D'_2 (dont la somme des coordonnées des points est impaire). Finalement, on calcule l'erreur quadratique moyenne entre le vecteur \mathbf{x}'_2 et chacun des deux vecteurs de reproduction trouvés pour obtenir le meilleur mot de code. Cette procédure de recherche est illustrée à la figure 2.10.

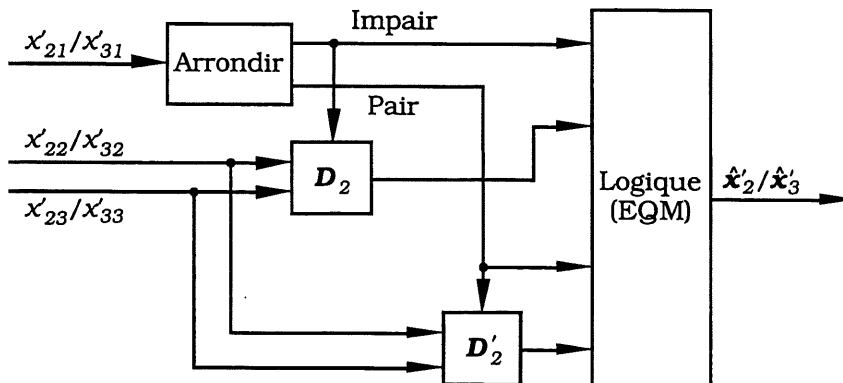


Figure 2.10 Recherche du mot de code dans un dictionnaire pyramidal ($S = 9$).

L'algorithme de décodage du quantificateur vectoriel pyramidal se décompose en quatre étapes suivantes:

Etape 1: Trouver k_0 à partir de l'indice k reçu.

Etape 2: Calculer s par l'opération inverse de l'équation (2.23) ou (2.26).

Etape 3: Supposer $c_2 = 0$ dans l'équation (2.24) et trouver le plus petit nombre entier c_1 ($0 \leq c_1 \leq s$) satisfaisant

$$k \geq k_0 + \frac{(s-c_1)(s-c_1+1)}{2} .$$

Etape 4: Calculer la vraie valeur de c_2 par l'opération inverse de l'équation (2.24) et finalement obtenir c_3 selon $c_3 = s - c_1 - c_2$.

2.3.4 Résultats expérimentaux

Afin d'améliorer les performances de quantification, une stratégie d'allocation non uniforme de bits est employée dans la conception du quantificateur global. Comme les nombres de paramètres LSF correspondant aux vecteurs différentiels normalisés sont différents et généralement les paramètres LSF inférieurs sont plus importants que les supérieurs au sens de la perception, le vecteur à deux dimensions, \mathbf{x}_1 , est quantifié avec moins de bits et les vecteurs à trois dimensions, \mathbf{x}_2 et \mathbf{x}_3 , sont quantifiés avec plus de bits. Cependant, le quantificateur vectoriel statistique pour ω_3 et ω_7 a toujours un dictionnaire de taille 64 (6 bits).

Dénotons QVS le quantificateur vectoriel statistique, QVT le quantificateur vectoriel triangulaire, QVP1 le quantificateur vectoriel pyramidal pour le vecteur \mathbf{x}_2 et QVP2 le quantificateur vectoriel pyramidal pour le vecteur \mathbf{x}_3 . Le tableau 2.8 donne le plan d'allocation de bits pour différents débits dans la région de 26 à 29 bits/trame.

TABLEAU 2.8
Plan d'allocation de bits pour différents débits dans la région de 26 à 29 bits/trame

Débit (bits/trame)	Allocation de bits (bits)			
	QVS	QVT	QVP1	QVP2
26	6	5	8	7
27	6	5	9	7
28	6	5	9	8
29	6	6	9	8

L'évaluation des performances du quantificateur global est basée sur la même base de test et la même mesure de distorsion spectrale utilisées dans la section précédente. Le tableau 2.9 montre les performances en terme de distorsion spectrale (SD) du quantificateur global pour 26 - 29 bits/trame.

TABLEAU 2.9
Performances (SD) du quantificateur vectoriel algébrique pour 26 - 29 bits/trame

Débit (bits/trame)	SD (dB)	"Outliers" (%)	
		2-4 dB	>4 dB
26	1.17	2.97	0
27	1.10	2.22	0
28	1.04	1.78	0
29	0.99	1.49	0

On peut voir dans ce tableau qu'un quantificateur vectoriel algébrique à 28 bits/trame permet d'obtenir une qualité transparente de la quantification des

paramètres LSF, c'est-à-dire une distorsion spectrale moyenne d'environ 1 dB, moins de 2% de trames "outliers" dans la région de 2 à 4 dB et aucune trame "outlier" ayant une distorsion spectrale supérieure à 4 dB. La figure 2.11 illustre l'histogramme des distorsions spectrales correspondant à ce quantificateur à 28 bits/trame.

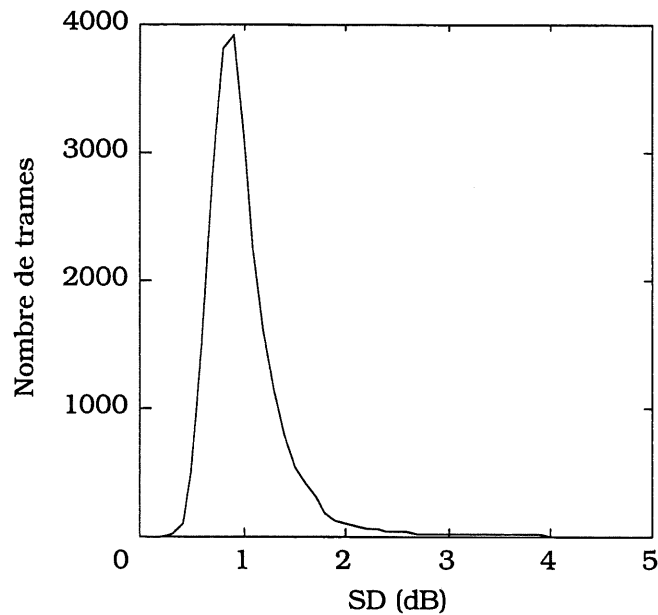


Figure 2.11 Histogramme des distorsions spectrales (SD) correspondant au quantificateur vectoriel algébrique à 28 bits/trame.

Les meilleurs facteurs d'échelle et valeurs de décalage sont obtenus à l'aide de la base d'apprentissage décrite dans la section précédente. Le tableau 2.10 donne les facteurs d'échelle et les valeurs de décalage utilisés respectivement par les quantificateurs QVT, QVP1 et QVP2 dans le cas de 28 bits/trame.

TABLEAU 2.10
Facteurs d'échelle et valeurs de décalage pour
le quantificateur vectoriel algébrique à 28 bits/trame

QV	Facteur d'échelle		Valeur de décalage			
	α_1	21.74	β_1	0.23	0.20	
QVP1	α_2	23.88	β_2	0.07	0.07	0.07
QVP2	α_3	27.27	β_3	0.07	0.13	0.30

On compare aussi les performances du quantificateur vectoriel algébrique avec celles du quantificateur vectoriel fendu en (3, 3, 4) dans la région de 26 à 29 bits/trame. Le tableau 2.11 montre les performances du quantificateur vectoriel fendu.

TABLEAU 2.11
Performances (SD) du quantificateur
vectoriel fendu en (3, 3, 4) pour 26 - 29 bits/trame

Débit (bits/trame)	SD (dB)	"Outliers" (%)	
		2-4 dB	>4 dB
26 (9 - 9 - 8)	1.24	3.48	0
27 (9 - 9 - 9)	1.15	2.27	0
28 (10 - 9 - 9)	1.07	1.86	0
29 (10 - 10 - 9)	1.01	1.64	0

Les résultats obtenus montrent que le quantificateur vectoriel algébrique permet d'obtenir de meilleures performances que le quantificateur vectoriel fendu en (3, 3, 4). Sur

le plan de la complexité, le quantificateur vectoriel algébrique a deux avantages par rapport au quantificateur vectoriel fendu:

- l'espace mémoire pour le stockage des dictionnaires est considérablement économisé (seulement 128 locations de mémoire);
- la recherche des mots de code s'effectue par les opérations très simples, telles que l'arrondi, l'addition et la comparaison, et le calcul de la norme quadratique n'est pas requis.

En comparaison du quantificateur vectoriel algébrique ellipsoïdal présenté dans la section précédente, l'approche proposée ici permet d'économiser 1 bit par trame tout en assurant une qualité transparente de la quantification des paramètres LSF.

Chapitre 3

Application de la quantification vectorielle algébrique au codage de la parole en bande élargie

L'intérêt pour le codage de la parole en bande élargie (50 - 7000 Hz) augmente de plus en plus ces dernières années à cause des applications telles que la conférence audiovisuelle, le visiophone et la téléphonie de haute qualité. Bien que des canaux et réseaux à haut débit deviennent de plus en plus facilement accessibles, le codage à bas débit est toujours nécessaire pour des questions de ressources physiques de stockage et de transport. Dans ce chapitre, nous développons une nouvelle technique de quantification vectorielle algébrique et l'appliquons au codage de la parole en bande élargie pour un débit de 16 kbit/s.

3.1 Codage de la parole en bande élargie

Dans le codage de la parole en bande élargie (50 - 7000 Hz), on exploite le phénomène de masquage perceptuel afin d'obtenir la réduction de débit. Les techniques de codage en sous-bandes et de codage par transformée sont utilisées comme base des méthodes efficaces du codage de la parole en bande élargie. Dans cette section, nous présentons brièvement le signal de parole en bande élargie, le phénomène de masquage et les techniques de codage.

3.1.1 Signal de parole en bande élargie et phénomène de masquage

Le signal de parole en bande élargie (50 - 7000 Hz) diffère beaucoup du signal de parole en bande téléphonique (300 - 3400 Hz). La différence entre ces deux signaux est grande non seulement en terme de bande passante et de plage dynamique, mais aussi en terme de qualité offerte aux auditeurs. L'élargissement de la bande transmise de 300 - 3400 Hz à 50 - 7000 Hz apporte de grandes améliorations subjectives à la qualité de parole. Plus précisément, l'accroissement des basses fréquences (50 - 300 Hz) améliore la netteté de la parole et l'accroissement des hautes fréquences (3400 - 7000 Hz) augmente l'intelligibilité de la parole [31].

Le signal de parole en bande élargie est un signal très complexe possédant de nombreuses caractéristiques. On peut caractériser ce signal comme suit [32]:

- une bande passante de 50 Hz à 7000 Hz;
- les basses fréquences nettement plus énergétiques que les hautes fréquences;
- une dynamique extrêmement forte;
- des attaques très brusques.

Le codage de la parole en bande élargie exploite le phénomène de masquage qui joue un rôle primordial dans la perception auditive. Le phénomène de masquage traduit le fait que lorsqu'on entend deux sons purs de fréquences différentes mais assez proches, l'un des deux, appelé le son masquant, peut rendre totalement inaudible l'autre, appelé le son masqué, selon leurs positions respectives en fréquence et en niveau de pression acoustique. Ce phénomène s'explique par l'inertie temporelle du système d'audition et par le comportement en fréquence de la membrane basilaire de la cochlée [33]. Un seuil de masquage peut être mesuré. Tout signal sous ce seuil ne sera pas audible. Le seuil de masquage dépend de la fréquence et du niveau du son masquant ainsi que des caractéristiques du son masquant et du son masqué.

Il existe deux types de masquage. On parle de masquage fréquentiel si deux sons apparaissent simultanément et de masquage temporel si deux sons apparaissent de façon successive. Nous abordons ici seulement le masquage fréquentiel.

En l'absence d'un signal masquant, un son doit, pour être perçu, avoir un niveau supérieur à un seuil fonction de la fréquence: le seuil d'audition absolu. Ce seuil met en évidence l'étendue du domaine audible 20 - 20000 Hz ainsi qu'une plus forte sensibilité de l'oreille entre 500 Hz et 5000 Hz. Lorsque le système auditif est excité par un son masquant, ce seuil d'audition se modifie. Tout autre son dont la position dans le plan "Niveau - Fréquence" est en dessous de cette courbe est masqué, c'est-à-dire inaudible.

La notion de bandes critiques est un autre concept fondamental en psychoacoustique. Des chercheurs ont suggéré que l'oreille effectue une analyse fréquentielle du signal comme le ferait un banc de filtres, correspondant aux dites bandes critiques, dont la largeur varie avec la fréquence. Lorsqu'on les juxtapose, le domaine audible est divisé en une vingtaine de bandes critiques [32]. Par conséquent, le système auditif peut en général être considéré comme un banc de filtres avec beaucoup de recouvrement.

Pour tirer parti du phénomène de masquage dans l'optimisation d'un codeur, on cherche à rendre le bruit de codage inférieur en tout point du spectre à la courbe de masquage. Une mise en forme spectrale du bruit de codage est alors nécessaire afin de suivre les variations de cette courbe en fonction de la fréquence.

3.1.2 Techniques de codage de la parole en bande élargie

On a vu précédemment qu'un système de codage de la parole en bande élargie atteint son optimum lorsque le bruit de codage se situe en dessous de la courbe de masquage. Une transformation en bloc de type temps - fréquence est souvent utilisée pour réaliser la décomposition spectrale du signal. Le signal original est décomposé en "tranches" d'une durée déterminée, de l'ordre de quelques millisecondes, qui sont ensuite

représentées dans le domaine fréquentiel après une transformation adaptée. Les paramètres de la transformée sont alors quantifiés et codés en tenant compte de la courbe de masquage obtenue à l'aide d'un modèle psychoacoustique: chaque paramètre de la transformée sera codé avec une précision juste suffisante pour que le "bruit de quantification" injecté par le décodeur demeure en dessous du niveau de la courbe de masquage et donc ne soit pas perceptible à l'oreille.

La plupart des codeurs de la parole en bande élargie reposent généralement sur le codage en sous-bandes et le codage par transformée.

Le codage en sous-bandes procède par une découpe préalable du spectre du signal en bandes de fréquences à l'aide d'un banc de filtres particulier, appelé les filtres miroirs en quadrature (QMF, Quadrature Mirror Filters). Des filtres de synthèse assurent la reconstruction parfaite du signal de départ par recombinaison des signaux des sous-bandes. La découpe en sous-bandes permet d'adapter les paramètres du codage aux caractéristiques fréquentielles du signal. Par exemple, le nombre de bits dans chaque sous-bande est alloué en fonction de l'importance relative de l'information à transmettre. D'autre part, le codage en sous-bandes tire un certain profit des propriétés de l'audition. En effet, le bruit de quantification restant confiné dans la bande où il a été créé et son énergie étant proportionnelle à celle du signal, une mise en forme spectrale du bruit est ainsi naturellement réalisée.

Une norme à 64 kbit/s pour le codage de la parole en bande élargie (50 - 7000 Hz), nommée la Recommandation G.722, a été normalisée par l'UIT-T (Union Internationale des Télécommunications — Secteur de la Normalisation des Télécommunications) en 1986. La figure 3.1 illustre le schéma de principe du codeur à 64 kbit/s standardisé par la Recommandation UIT-T G.722 [34].

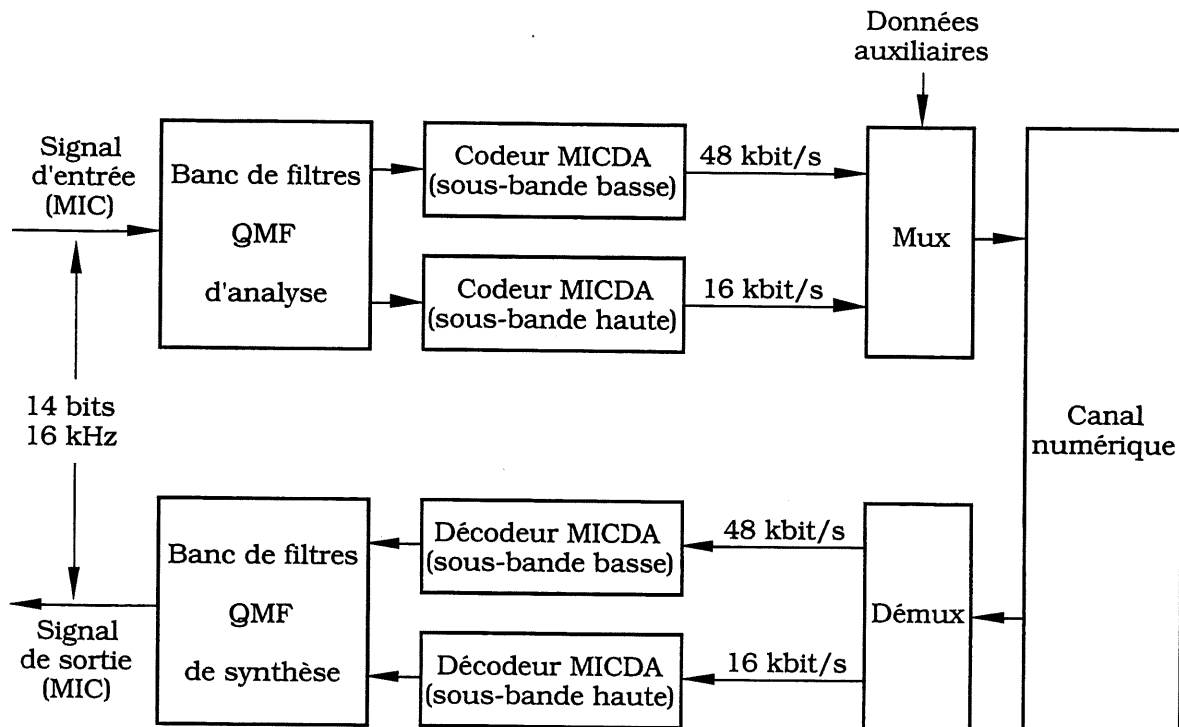


Figure 3.1 Schéma de principe du codeur à 64 kbit/s de la parole en bande élargie standardisé par la Recommandation UIT-T G.722.

Le signal de parole limité à 7000 Hz est échantillonné à 16 kHz. Il est ensuite séparé en deux sous-bandes (0 - 4000 Hz et 4000 - 8000 Hz) par des filtres miroirs en quadrature (QMF). Après sous-échantillonnage à 8 kHz, chaque sous-bande est codée par un codeur MICDA (MIC Différentiel Adaptatif) à 32 kbit/s normalisé par la Recommandation UIT-T G.721 [35]. En réception, les signaux de chaque sous-bande sont décodés, puis le signal est reconstitué par sommation des deux sous-bandes filtrées. Le codage de haute qualité est obtenu par une allocation non uniforme de bits pour laquelle la sous-bande basse et la sous-bande haute sont quantifiées respectivement avec 6 bits et 2 bits par échantillons. Les possibilités de partage du canal à 64 kbit/s entre la parole et les données sont de 64 - 0, 56 - 8 et 48 - 16 kbit/s. Cette norme est considérée comme un point de repère important dans le codage de la parole en bande élargie.

Il est évident que la mise en forme spectrale du bruit de codage sera facilitée si l'on dispose d'une représentation précise du spectre. Ainsi le codage par transformée pour lequel la résolution fréquentielle est élevée tirent plus facilement parti des propriétés perceptuelles que le codage en sous-bandes.

Rappelons le codage par transformée présenté dans le chapitre précédent. Les algorithmes de codage par transformée opèrent par découpe du signal en blocs d'échantillons successifs. Une transformation linéaire, qui délivre un jeu de coefficients représentatifs du spectre du signal, est appliquée sur chaque bloc. Ce sont ces coefficients qui, une fois codés, sont transmis au décodeur. Celui-ci accomplit une transformation inverse des coefficients quantifiés pour revenir dans le domaine temporel de départ.

Pour que l'algorithme de compression de débit soit efficace, la transformation choisie doit concentrer le maximum d'énergie du signal sur le plus petit nombre possible de coefficients. Les transformées orthogonales, telles que la transformée de Karhunen-Loève (KL), la transformée de Fourier discrète et la transformée en cosinus discrète, permettent de satisfaire cette exigence.

Dans le codage par transformée de la parole en bande élargie, on ne transmet pas de composantes spectrales inaudibles, c'est-à-dire des coefficients dont l'énergie est inférieure à un seuil perceptuel (la courbe de masquage). Le masquage fréquentiel est mis à profit pour discriminer les coefficients spectraux inaudibles ainsi que pour assurer le masquage du bruit de quantification. La répartition du débit entre les coefficients est recalculée à chaque bloc d'échantillons de telle façon que le spectre du bruit soit en dessous de la courbe de masquage. En général, plus de débit est accordé aux zones importantes du spectre, par exemple 0 - 4000 Hz pour le signal de parole, et les coefficients dont l'énergie est nulle ou faible ne seront pas codés.

Un nouveau modèle de codage audio, dit TCX (Transform Coded Excitation), a été proposé récemment [36]. Cette technique de codage audio est basée sur le codage par transformée et tire parti du codage CELP (Code-Excited Linear Prediction) [37].

La figure 3.2 montre le schéma de principe du codeur TCX [38].

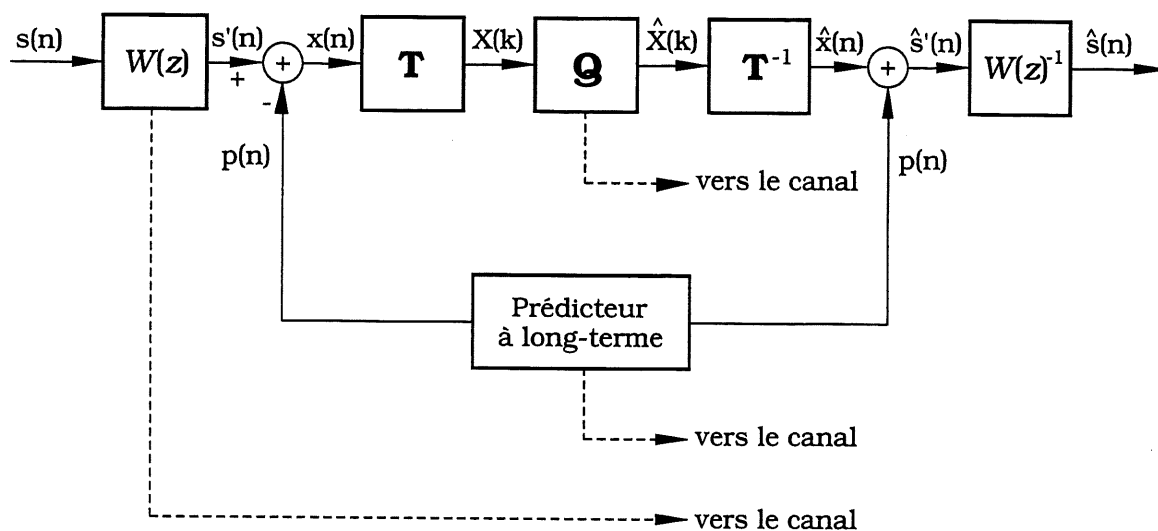


Figure 3.2 Schéma de principe du codeur TCX.

Dans un premier temps, le signal de départ est filtré par un filtre de pondération perceptuelle $W(z)$ afin de générer le signal perceptuel. Ensuite, on retire la contribution du "pitch" et l'on obtient le signal appelé la cible. C'est ce signal cible qui est transformé par la transformée de Fourier discrète T et puis est quantifié de façon vectorielle dans le domaine fréquentiel. Une caractéristique importante du codeur TCX est sa grande réduction de la complexité du codage en comparaison d'autres codeurs de type *analyse par synthèse*.

Le codeur TCX est appliqué au codage de la parole en bande élargie (50 - 7000 Hz) à 16 kbit/s et permet d'obtenir de haute qualité de parole [36].

Finalement, nous présentons l'évaluation de la qualité de la parole codée et synthétique. Il existe deux grands types de mesure: la mesure objective et la mesure subjective [39].

En mesure objective, le rapport signal à bruit (RSB) est un critère simple mais largement utilisé et il permet de mesurer la similarité des formes d'onde entre le signal original et le signal de synthèse. Le RSB est défini comme

$$\text{RSB} = \frac{E\{s^2(n)\}}{E\{|s(n) - \hat{s}(n)|^2\}} \quad (3.1)$$

où $s(n)$ est le signal original et $\hat{s}(n)$ est le signal de synthèse. Et le RSB est souvent représenté en dB par

$$\text{RSB (dB)} = 10 \log_{10} (\text{RSB}) . \quad (3.2)$$

Une des caractéristiques essentielles du signal de parole est sa nature de variation dans le temps. A cause de cette caractéristique, des segments de parole possèdent une énergie importante et d'autres possèdent une énergie très faible. Si l'énergie du bruit est plus ou moins constante, le RSB obtenu ne permet pas de représenter la qualité vraie du codage car l'effet perceptuel du bruit dans la région de signal faible est sévère. En tenant compte de cet effet, on utilise le RSB segmentaire, dénoté RSBseg. On divise le signal de parole en segments de longueur 15 - 20 ms. Ensuite on calcule le RSB de chaque segment et l'on trouve tous les segments de RSB supérieur à un seuil spécifié. Le RSBseg est alors défini comme

$$\text{RSBseg} = \frac{1}{M} \sum_{i=1}^M \text{RSB}(i) \text{ (dB)} \quad (3.3)$$

où M dénote le nombre de segments ayant un RSB supérieur au seuil donné.

Les mesures objectives permettent d'obtenir certaines indications sur la qualité d'un système de codage de la parole. Cependant, une évaluation réelle de la qualité résulte

de tests subjectifs qui mesurent la similarité perceptuelle entre le signal original et le signal reconstruit. La méthode de test subjectif la plus populaire est les notes d'opinion moyennes (MOS, Mean Opinion Score). Le principe général du test MOS consiste à faire écouter des phrases et demander aux auditeurs non experts, à chaque écoute, d'exprimer une opinion. Les notes attribuées sont comprises entre 1 et 5, soit une échelle de qualité croissante. Toutefois, des tests d'écoute informels sont suffisants durant le développement d'un système de codage.

3.2 Quantification vectorielle algébrique à partir du réseau RE_8

Dans le codage par transformée, on fait souvent face à un vecteur transformé de dimension élevée, typiquement de 64 à 512. Pour quantifier efficacement ce vecteur transformé avec un nombre fixe de bits, plusieurs approches sont utilisées [2]. Une méthode générale est la quantification scalaire avec allocation adaptative de bits. Cependant, cette technique offre de mauvaises performances de quantification pour les signaux faibles. En principe la quantification scalaire suivie d'un codage entropique permet de résoudre ce problème, mais il est très difficile de réaliser cette technique dans un système pratique.

Nous proposons ici une solution intermédiaire qui consiste à diviser un vecteur transformé en une série de sous-vecteurs de même dimensions et utiliser un groupe de quantificateurs vectoriels de divers débits. Chaque sous-vecteur est composé de composantes adjacentes qui sont liées en terme d'énergie. On code un sous-vecteur en utilisant l'indice du mot de code ainsi que le numéro du quantificateur utilisé.

Pour construire un ensemble de quantificateurs vectoriels à débit variable, on développe deux approches basées sur le réseau de Gosset tourné RE_8 : *Quantificateurs Vectoriels Algébriques Complémentaires* et *Quantificateurs Vectoriels Algébriques Encastrés*. L'approche complémentaire consiste à utiliser un groupe de sous-

quantificateurs sans recouvrement. Et dans l'approche encadrée, un quantificateur donné comprend les sous-quantificateurs de débits plus faibles.

Le codage entropique, tel que le codage arithmétique, peut être utilisé pour coder les numéros des sous-quantificateurs. Cependant, pour l'approche encadrée, une méthode algébrique que l'on présentera est plus efficace que la technique de codage entropique dans des systèmes pratiques.

3.2.1 Codes sphériques basés sur le réseau RE_8

Rappelons tout d'abord la définition du réseau RE_8 décrite au chapitre 1:

$$RE_8 = (2D_8) \cup (2D_8 + [1, 1, 1, 1, 1, 1, 1, 1]) .$$

Un vecteur \mathbf{x} du réseau RE_8 possède quatre propriétés essentielles:

- \mathbf{x} appartient au réseau Z_8 ;
- la somme des composantes est un multiple de 4, c'est-à-dire $\sum_{i=1}^8 x_i = 0 \pmod{4}$;
- les huit composantes ont la même parité (toutes paires ou toutes impaires);
- la somme $\sum_{i=1}^8 x_i^2$ est un multiple de 8, c'est-à-dire $\sum_{i=1}^8 x_i^2 = 0 \pmod{8}$.

Géométriquement, les points du réseau RE_8 sont répartis sur des sphères concentriques de rayon $2\sqrt{2m}$ et centrées à l'origine [40]. Le paramètre m prend des valeurs entières successives et donc peut devenir un indice naturel pour ordonner les sphères du réseau RE_8 . L'ensemble des points appartenant à une sphère constitue un code sphérique qui peut être utilisé comme dictionnaire d'un quantificateur vectoriel. Alors, les sphères diverses offrent un choix de débits pour les quantificateurs correspondants.

Une propriété très intéressante du réseau RE_8 est que toute permutation des

coordonnées d'un vecteur de RE_8 est encore un de ses vecteurs [7]. A partir de cette propriété, on définit comme faisant partie d'une même classe d'équivalence les vecteurs obtenus par permutation de même composantes. Par exemple, les vecteurs $[2, 0, 0, 0, -2, 0, 0, 0]$ et $[0, 0, -2, 0, 0, 2, 0, 0]$ sont tous obtenus par permutations des huit composantes 2, 0, 0, 0, 0, 0, 0 et -2, et donc ils appartiennent à une même classe d'équivalence.

Le nombre de vecteurs de la classe est donné par la formule dénombrant les permutations avec répétitions. Si tous les vecteurs de la classe comprennent q composantes de valeurs distinctes a_0, a_1, \dots, a_{q-1} et si w^i est le nombre de répétitions de la valeur a_i , alors la cardinalité K de cette classe est calculée par

$$K = \frac{8!}{w^0! w^1! \dots w^{q-1}!} \quad (3.4)$$

Dans chaque classe, on appelle "leader (signé)" de la classe le vecteur dont huit composantes sont ordonnées dans l'ordre décroissant des valeurs. Toute classe d'équivalence est entièrement définie par son leader. En réordonnant les valeurs absolues des composantes du leader signé, on obtient le vecteur appelé "leader absolu".

Tout leader signé peut se déduire du leader absolu correspondant selon la propriété $\sum_{i=1}^8 x_i = 0 \pmod{4}$ [19]. Pour les leaders signés pairs (c'est-à-dire à composantes paires), on

peut les obtenir en changeant simplement les signes du leader absolu correspondant. Et pour générer les leaders signés impairs (c'est-à-dire à composantes impaires), on doit vérifier la contrainte $\sum_{i=1}^8 x_i = 0 \pmod{4}$ en changeant les signes du leader absolu

correspondant. Le tableau 3.1 donne pour chaque sphère m entre 1 et 20 le nombre de leaders absolus ($P_8^A(m)$), le nombre de leaders signés ($P_8^S(m)$), le nombre de points ($N_8(m)$) et le débit correspondant en bit par dimension (Débit/dim.).

TABLEAU 3.1
Inventaire des sphères du réseau RE_g

m	$P_g^A(m)$	$P_g^S(m)$	$N_g(m)$	Débit/dim.
1	2	8	240	0.9884
2	3	15	2160	1.3846
3	3	24	6720	1.5893
4	5	42	17520	1.7621
5	5	54	30240	1.8605
6	5	57	60480	1.9855
7	8	103	82560	2.0416
8	9	119	140400	2.1374
9	9	124	181680	2.1839
10	11	162	272160	2.2568
11	13	220	319680	2.2858
12	11	208	490560	2.3630
13	16	279	527520	2.3761
14	16	305	743040	2.4379
15	16	343	846720	2.4614
16	20	402	1123440	2.5124
17	22	470	1179360	2.5212
18	21	447	1635120	2.5801
19	27	597	1646400	2.5814
20	26	638	2207520	2.6342

On peut voir que pour un quantificateur vectoriel basé sur le réseau RE_g , les mots de code de son dictionnaire se déduisent des leaders par permutation et donc n'ont pas besoin d'être stockés. D'autre part, la recherche du plus proche voisin d'un vecteur quelconque est rapide car ce processus se limite à un petit ensemble de permutations des leaders.

Un vecteur de RE_8 peut être indexé par la somme du cumul H des cardinalités des classes précédentes et du rang t de ce vecteur dans sa classe. Le rang t est calculé par la formule de Schalkwijk [41]. En appelant q le nombre de composantes de valeurs distinctes dans le vecteur, d l'indice particulier et w_j^d le nombre de répétitions de la valeur indexée par d qui restent dans les positions j à 8, on a

$$t = \sum_{j=1}^8 \sum_{d=0}^{d(j)-1} \frac{(8-j)!}{(w_j^d - 1)! \prod_{\substack{i=0 \\ i \neq d}}^{q-1} (w_j^i)!}, \quad (3.5)$$

avec la convention que $0! = 1$ et $(-1)! = \infty$.

Soit $\mathbf{x} = [x_1, x_2, \dots, x_8]$ un vecteur du réseau RE_8 . L'algorithme d'indexation de \mathbf{x} se résume comme suit:

Etape 1: Calculer le rang t de \mathbf{x} dans sa classe à l'aide de l'expression (3.5).

Etape 2: Réordonner les composantes du vecteur \mathbf{x} par ordre décroissant des valeurs et obtenir son leader \mathbf{x}_0 .

Etape 3: Trouver le cumul H des cardinalités associé à \mathbf{x}_0 dans un tableau construit à l'avance.

Etape 4: Déterminer l'indice k de \mathbf{x} par $k = H + t$.

Etape 5: Fin.

3.2.2 Approche complémentaire

Cette approche est une technique de quantification vectorielle à débit variable. Le quantificateur global est composé de l'origine (\mathcal{Q}_0) et cinq sous-quantificateurs sans recouvrement. Les tailles des dictionnaires des sous-quantificateurs sont respectivement

de 16, 256, 4096, 65536 et 1048432, c'est-à-dire 4, 8, 12, 16 et 20 bits. La figure 3.3 illustre la structure des dictionnaires des sous-quantificateurs sans recouvrement.

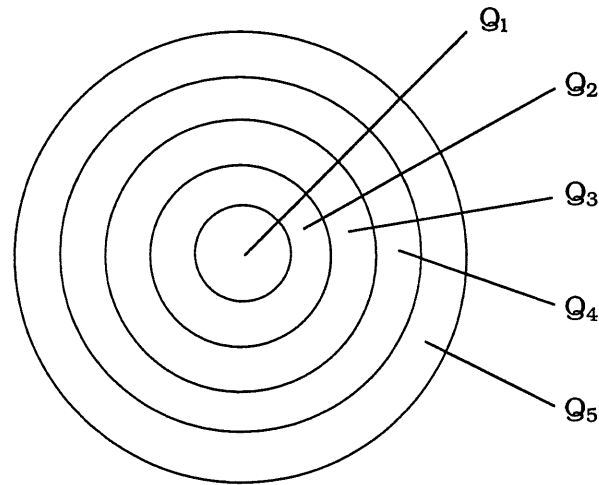


Figure 3.3 Structure des dictionnaires des sous-quantificateurs sans recouvrement.

Le sous-quantificateur Q_1 a un dictionnaire de taille 16 dont les mots de code sont générés par permutations des composantes des vecteurs $[2, 0, 0, 0, 0, 0, 0, 0]$ et $[-2, 0, 0, 0, 0, 0, 0, 0]$. En fait Q_1 est un quantificateur additionnel mais n'est pas indispensable, car il n'appartient pas au réseau RE_8 qui forme les autres quantificateurs. Ce sous-quantificateur est un code sphérique à partir de RE_8 mais translaté par le vecteur $[1, 0, 0, 0, 0, 0, 0, 0]$. Au fond, Q_1 permet de fournir de bonnes performances en terme d'erreur quadratique moyenne pour des signaux faibles de haute probabilité. En l'absence de Q_1 , ces signaux faibles seront quantifiés majoritairement par le vecteur zéro. Les dictionnaires des sous-quantificateurs Q_2, Q_3, Q_4 et Q_5 sont composés principalement des onze premières sphères du réseau RE_8 . Le tableau 3.2 montre pour chaque sous-quantificateur les leaders absolus correspondants et le nombre de mots de code associés à chaque leader absolu.

TABLEAU 3.2
Liste des leaders absolus utilisés dans l'approche complémentaire

Sphère	Leader absolu	Taille	Q ₀	Q ₁	Q ₂	Q ₃	Q ₄	Q ₅
Spéciale	2 0 0 0 0 0 0 0	16		.				
0	0 0 0 0 0 0 0 0	1	.					
1	2 2 0 0 0 0 0 0	112						
1	1 1 1 1 1 1 1 1	128			.			
2	4 0 0 0 0 0 0 0	16			.			
2	2 2 2 2 0 0 0 0	1120				.		
2	3 1 1 1 1 1 1 1	1024				.		
3	4 2 2 0 0 0 0 0	1344				.		
3	2 2 2 2 2 2 0 0	1792					.	
3	3 3 1 1 1 1 1 1	3584					.	
4	4 4 0 0 0 0 0 0	112				.		
4	4 2 2 2 2 0 0 0	8960					.	
4	2 2 2 2 2 2 2 2	256			.			
4	5 1 1 1 1 1 1 1	1024					.	
4	3 3 3 1 1 1 1 1	7168					.	
5	6 2 0 0 0 0 0 0	224			.			
5	4 4 2 2 0 0 0 0	6720					.	
5	4 2 2 2 2 2 2 0	7168					.	
5	5 3 1 1 1 1 1 1	7168					.	
5	3 3 3 3 1 1 1 1	8960					.	
6	6 2 2 2 0 0 0 0	4480					.	
6	4 4 4 0 0 0 0 0	448						.
6	4 4 2 2 2 2 0 0	26880						.
6	5 3 3 1 1 1 1 1	21504						.
6	3 3 3 3 3 1 1 1	7168				.		
7	6 4 2 0 0 0 0 0	2688						.
7	6 2 2 2 2 2 0 0	10752						.
7	4 4 4 2 2 0 0 0	17920						.
7	4 4 2 2 2 2 2 2	7168						.
7	7 1 1 1 1 1 1 1	1024						.
7	5 5 1 1 1 1 1 1	3584						.
7	5 3 3 3 1 1 1 1	35840						.
7	3 3 3 3 3 3 1 1	3584						.
8	8 0 0 0 0 0 0 0	16			.			.
8	6 4 2 2 2 0 0 0	35840						.
8	6 2 2 2 2 2 2 2	2048						.
8	4 4 4 4 0 0 0 0	1120				.		.
8	4 4 4 2 2 2 2 0	35840						.
8	7 3 1 1 1 1 1 1	7168						.
8	5 5 3 1 1 1 1 1	21504						.
8	5 3 3 3 3 1 1 1	35840						.
8	3 3 3 3 3 3 3 1	1024						.
9	8 2 2 0 0 0 0 0	1344						.
9	6 6 0 0 0 0 0 0	112						.
9	6 4 4 2 0 0 0 0	13440						.
9	6 4 2 2 2 2 2 0	43008						.
9	4 4 4 4 2 2 0 0	26880						.
9	7 3 3 1 1 1 1 1	21504						.
9	5 5 3 3 1 1 1 1	53760						.
9	5 3 3 3 3 3 1 1	21504						.
9	3 3 3 3 3 3 3 3	128						.
10	8 4 0 0 0 0 0 0	224					.	.
10	8 2 2 2 2 0 0 0	8960						.
10	6 6 2 2 0 0 0 0	6720						.
10	6 4 4 2 2 2 0 0	107520						.
10	4 4 4 4 4 0 0 0	1792						.
10	4 4 4 4 2 2 2 2	17920						.
10	7 5 1 1 1 1 1 1	7168						.
10	7 3 3 3 1 1 1 1	35840						.
10	5 5 5 1 1 1 1 1	7168						.
10	5 5 3 3 3 1 1 1	71680						.
10	5 3 3 3 3 3 3 1	7168						.
11	8 4 2 2 0 0 0 0	13440						.
11	8 2 2 2 2 2 2 0	7168						.
11	6 6 4 0 0 0 0 0	1344						.
11	6 6 2 2 2 2 0 0	26880						.
11	6 4 4 4 2 0 0 0	35840						.
11	6 4 4 2 2 2 2 2	43008						.
11	4 4 4 4 4 2 2 0	21504						.
11	9 1 1 1 1 1 1 1	1024						.
11	7 5 3 1 1 1 1 1	43008						.
11	7 3 3 3 3 1 1 1	35840						.
11	5 5 5 3 1 1 1 1	35840						.
11	5 5 3 3 3 3 1 1	53760						.
11	5 3 3 3 3 3 3 3	1024						.
12	4 4 4 4 4 4 0 0	1792						.
13	8 6 2 0 0 0 0 0	2688						.

Dans cette approche, seuls ces leaders absolus et certains paramètres sont stockés sous la forme d'un tableau donnant l'ordre lexicographique pour générer et indexer les mots de code des dictionnaires des sous-quantificateurs.

Soit $\mathbf{x} = [x_1, x_2, \dots, x_8]$ un vecteur quelconque à huit dimensions. L'algorithme de quantification pour \mathbf{x} s'effectue par les sept étapes suivantes:

Etape 1: Trouver le plus proche voisin \mathbf{y} de \mathbf{x} dans le réseau RE_8 à l'aide de l'algorithme rapide décrit au chapitre 1.

Etape 2: Calculer le rang t de \mathbf{y} dans sa classe.

Etape 3: Réordonner les composantes de \mathbf{y} par ordre décroissant des valeurs et obtenir son leader \mathbf{y}_0 .

Etape 4: Trouver le cumul H des cardinalités dans le tableau de paramètres.

Etape 5: Dans le cas de \mathbf{x} proche de l'origine, on trouve aussi son plus proche voisin \mathbf{y}' dans le dictionnaire du sous-quantificateur \mathcal{Q}_1 . Ensuite on choisit, entre \mathbf{y} et \mathbf{y}' , le meilleur vecteur au sens de l'erreur quadratique moyenne comme son représentant.

Etape 6: Déterminer l'indice k du représentant selon $k = H + t$ ($H = 0$ pour \mathbf{y}') et le numéro n du sous-quantificateur utilisé à partir de \mathbf{y}_0 (ou \mathbf{y}).

Etape 7: Coder le numéro n par une technique de codage entropique telle que le codage arithmétique.

Si le vecteur \mathbf{x} est situé hors du plus grand sous-quantificateur \mathcal{Q}_5 , le processus de quantification se répète sur une version de \mathbf{x} réduite par un facteur d'échelle jusqu'à ce que un mot de code soit trouvé.

L'algorithme de décodage se décompose en trois étapes suivantes:

Etape 1: Retrouver le leader \mathbf{y}_0 (ou \mathbf{y}') et le cumul H des cardinalités correspondant à partir de l'indice k et le numéro n du sous-quantificateur.

Etape 2: Calculer le rang t de \mathbf{y} dans sa classe par $t = k - H$.

Etape 3: Retrouver \mathbf{y} à partir de \mathbf{y}_0 et t .

3.2.3 Approche encastrée

Dans cette approche, le quantificateur global comprend l'origine (\mathcal{Q}_0) et cinq sous-quantificateurs encastrés. Les dictionnaires des sous-quantificateurs ont respectivement 16, 256, 4096, 65536 et 1048448 mots de code. Les dictionnaires de taille faible sont incorporés dans les dictionnaires de taille élevée. La figure 3.4 montre la structure des dictionnaires des sous-quantificateurs encastrés.

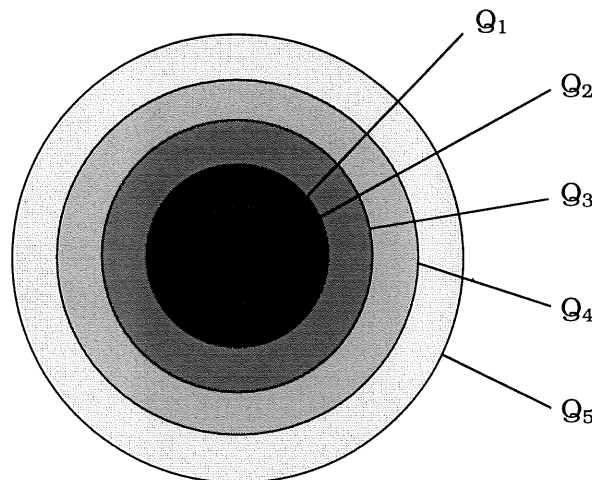


Figure 3.4 Structure des dictionnaires des sous-quantificateurs encastrés.

Comme dans l'approche complémentaire, les 16 mots de code du sous-quantificateur \mathcal{Q}_1 sont obtenus par permutations des composantes des vecteurs $[2, 0, 0, 0, 0, 0, 0, 0]$ et $[-2, 0, 0, 0, 0, 0, 0, 0]$. Essentiellement, les dictionnaires des sous-quantificateurs $\mathcal{Q}_2, \mathcal{Q}_3, \mathcal{Q}_4$ et \mathcal{Q}_5 comprennent respectivement la première sphère, les deux

premières sphères, les cinq premières sphères et les onze premières sphères du réseau RE_8 . Et le dictionnaire de \mathcal{Q}_1 est encastré dans chacun de ces quatre dictionnaires. Le tableau 3.3 donne pour chaque sous-quantificateur les leaders absolus correspondants et leur tailles (le nombre de mots de code).

Dans l'approche encastrée, les procédures de codage et décodage sont identiques à celles utilisées pour l'approche complémentaire et les algorithmes décrits au paragraphe précédent sont employés. En pratique, la structure encastrée des dictionnaires permet de faciliter le processus de codage des numéros des sous-quantificateurs.

Considérons un système de codage dans lequel on quantifie à la fois M vecteurs à huit dimensions avec N_q bits (N_q est un multiple de 4). Les numéros des M sous-quantificateurs utilisés constituent un vecteur \mathbf{z} de composantes non négatives dont la somme est égale à $\frac{N_q}{4}$. Ce nouveau vecteur \mathbf{z} peut également être considéré comme une permutation avec répétitions des numéros des sous-quantificateurs. Par conséquent, on peut utiliser la méthode de codage présentée précédemment pour coder les numéros des sous-quantificateurs.

Tout d'abord on doit trouver tous les leaders en vérifiant la contrainte $\sum_{i=1}^M z_i = \frac{N_q}{4}$

et ensuite on construit avec eux un tableau dans l'ordre lexicographique. L'algorithme décrit au paragraphe 3.2.1 nous permet d'effectuer le codage des numéros des sous-quantificateurs.

En fait, le nombre de bits utilisés pour la quantification n'est pas toujours exactement égal à N_q bits. Autrement dit, les composantes du vecteur \mathbf{z} ne vérifient pas forcément la contrainte $\sum_{i=1}^M z_i = \frac{N_q}{4}$. Si ce nombre de bits est supérieur à N_q bits, on réduit les M vecteurs par un facteur d'échelle. Dans le cas inférieur à N_q bits, la structure encastrée des dictionnaires nous permet de remplacer de petits numéros de sous-quantificateur par des numéros relativement élevés, afin que la somme des M numéros des sous-quantificateurs soit égale à $\frac{N_q}{4}$.

TABLEAU 3.3
Liste des leaders absolus utilisés dans l'approche encadrée

Sphère	Leader absolu	Taille	Q ₀	Q ₁	Q ₂	Q ₃	Q ₄	Q ₅
Spéciale	20000000	16						
0	00000000	1
1	22000000	112		
1	11111111	128		
2	40000000	16		
2	22220000	1120		
2	31111111	1024		
3	42200000	1344		
3	22222200	1792		
3	33111111	3584		
4	44000000	112		
4	42222000	8960		
4	22222222	256		
4	51111111	1024		
4	33311111	7168		
5	62000000	224		
5	44220000	6720		
5	42222220	7168		
5	53111111	7168		
5	33331111	8960		
6	62220000	4480		
6	44400000	448		
6	44222200	26880		
6	53311111	21504		
6	33333111	7168		
7	64200000	2688		
7	62222200	10752		
7	44422000	17920		
7	44222222	7168		
7	71111111	1024		
7	55111111	3584		
7	53331111	35840		
7	33333311	3584		
8	80000000	16		
8	64222000	35840		
8	62222222	2048		
8	44440000	1120		
8	44422220	35840		
8	73111111	7168		
8	55311111	21504		
8	53333111	35840		
8	33333331	1024		
9	82200000	1344		
9	66000000	112		
9	64420000	13440		
9	64222220	43008		
9	44442200	26880		
9	73311111	21504		
9	55331111	53760		
9	53333311	21504		
9	33333333	128		
10	84000000	224		
10	82222000	8960		
10	66220000	6720		
10	64422200	107520		
10	44444000	1792		
10	44442222	17920		
10	75111111	7168		
10	73331111	35840		
10	55511111	7168		
10	55333111	71680		
10	53333331	7168		
11	82222220	7168		
11	66400000	1344		
11	66222200	26880		
11	64442000	35840		
11	64422222	43008		
11	44444220	21504		
11	91111111	1024		
11	75311111	43008		
11	73333111	35840		
11	55531111	35840		
11	53333333	1024		
12	44444400	1792		

Dans un système de codage par transformée, d'une part, cette méthode algébrique permet de faciliter beaucoup le processus de codage par rapport au codage entropique comme on le verra. D'autre part, elle réalise l'allocation adaptative de bits. Pour un débit donné, les sous-vecteurs d'énergie forte sont toujours quantifiés sur un nombre de bits élevé.

3.2.4 Codage entropique

Soit un ensemble d'événements distincts x_1, x_2, \dots, x_L dont les probabilités d'apparition sont respectivement p_1, p_2, \dots, p_L . Shannon [42] a démontré que pour décrire complètement cet ensemble d'événements, le nombre de bits nécessaires en moyenne est donné par

$$H = \sum_{i=1}^L p_i \log_2 \frac{1}{p_i} . \quad (3.6)$$

Cette quantité H est appelée l'entropie.

L'entropie d'une source est positive ou nulle. Elle est nulle lorsque cette source prend certaine valeur avec une probabilité proche de un. Et elle est maximale et vaut $\log_2 L$ lorsque tous les L événements possibles sont équiprobables. Donc, on a

$$0 \leq H \leq \log_2 L . \quad (3.7)$$

En codage entropique, un algorithme optimal permet de coder en principe un événement de probabilité d'apparition p sur $\log_2 \frac{1}{p}$ bits. La méthode de codage de Huffman est une technique très connue [43] et est souvent utilisée pour coder un ensemble de symboles qui décrit l'information à comprimer. L'algorithme de codage de Huffman consiste à construire progressivement un graphe orienté en forme d'arbre binaire afin d'assigner des codes binaires à chaque symbole source. Ainsi, les symboles sont codés

avec des valeurs de longueurs binaires variables. On code les symboles qui ont une occurrence très faible sur une longueur binaire supérieure à la moyenne et les symboles très fréquents sur une longueur binaire très courte. Une propriété du codage de Huffman est que le code est uniquement décodable, car aucun mot de code valide n'est un préfixe d'un autre mot de code valide.

A titre d'exemple la figure 3.5 montre un arbre de codage de Huffman pour un ensemble de six symboles, *a*, *b*, *c*, *d*, *e* et *f*, avec les probabilités respectives 0.5, 0.17, 0.15, 0.08, 0.06 et 0.04. Le tableau 3.4 donne la liste des mots de code pour cet exemple.

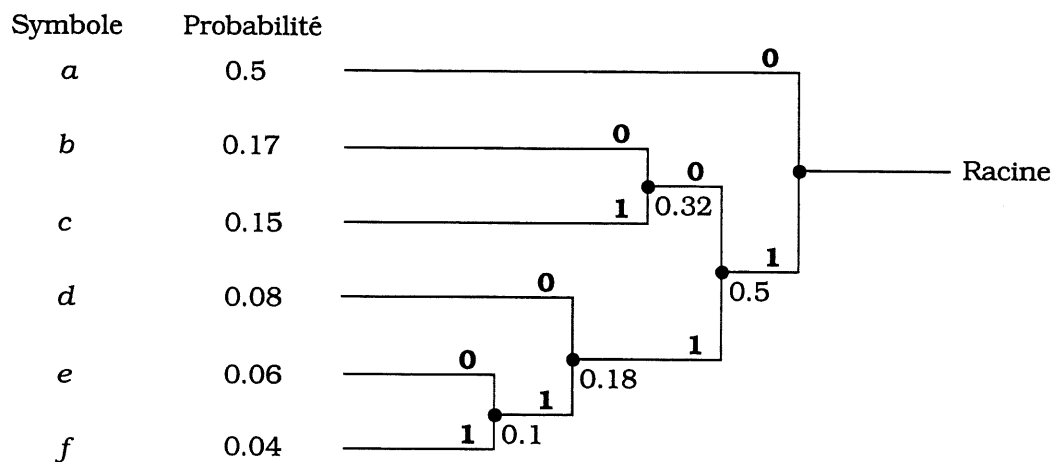


Figure 3.5 Algorithme de codage de Huffman.

TABLEAU 3.4
Liste des mots de code pour l'algorithme de codage de Huffman

Symbole	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
Code	0	100	101	110	1110	1111

Le codage de Huffman permet de fournir la solution optimale dans la cas d'un codage direct. Si toutes les probabilités des symboles d'une source sont les puissances

entières négatives de 2, le nombre de bits nécessaires en moyenne est égal à l'entropie de la source. Mais dans la pratique, cela n'est pas le cas général. Par conséquent, le fait que chaque symbole source doit être codé avec un nombre entier de bits puisse être un important désavantage du codage de Huffman, surtout pour les situations dans lesquelles un symbole possède la probabilité presque égale à unité.

Une méthode alternative, appelée le codage arithmétique, a été développée ces années dernières [44][45][46][47] et est appliquée à la compression de texte [46][48][49] ainsi qu'à la compression d'image [50][51].

Le codage arithmétique est optimal au sens de l'entropie et permet généralement d'apporter les performances supérieures à celles obtenues par le codage de Huffman codant chaque donnée individuelle [46][47]. Les meilleures performances obtenues sont liées au fait que cette méthode de codage ne nécessite pas un nombre entier de bits par symbole. De plus, cette technique se prête aisément à une approche adaptative où la fréquence d'apparition des symboles est estimée et mise à jour de manière continue en cours de codage. Cette propriété permet de séparer la modélisation de la source de la conception de l'algorithme de codage. Ainsi le même algorithme peut s'appliquer à différentes sources.

Le codage arithmétique est une procédure récursive qui code (décode) un symbole à chaque itération. A chaque itération, l'algorithme réduit la largeur d'un intervalle de nombres réels compris initialement entre zéro et un. Cette réduction est fonction de la probabilité du nouveau symbole à coder. Elle est d'autant plus faible que le symbole est probable, d'où une production d'un nombre de bits de spécification qui est une fonction inverse de cette probabilité.

Initialement, à chaque symbole de l'alphabet de la source est attribuée une portion de l'intervalle $[0, 1)$ de largeur équivalente à sa probabilité - connue a priori ou estimée - d'apparition. Le codage du premier symbole du message revient donc à identifier le sous-intervalle qui lui correspond. A l'itération suivante (codage du deuxième symbole du

message), cet intervalle partiel est lui-même subdivisé selon les mêmes règles de partition et un sous-sous-intervalle est identifié. La procédure se répète ainsi jusqu'à la fin du message qui doit être indiquée par un symbole convenu.

Le message est reconstruit, à partir de la séquence binaire correspondante, en cherchant par comparaisons successives à simuler le fonctionnement du codeur qui l'a produite.

L'algorithme de codage arithmétique pour un fichier de symboles se résume comme suit [47]:

Etape 1: On définit initialement l'intervalle $[0, 1)$ comme "l'intervalle actuel".

Etape 2: Pour chaque symbole du fichier, on effectue les trois sous-étapes suivantes:

- (a) On subdivise l'intervalle actuel en sous-intervalles dont chacun correspond à un symbole possible. Les largeurs des sous-intervalles sont proportionnelles aux probabilités des symboles.
- (b) On trouve le sous-intervalle correspondant au prochain symbole et on le choisit comme nouveau "sous-intervalle actuel".
- (c) Si le nouveau sous-intervalle actuel n'est pas situé entièrement dans un des intervalles suivants $[0, \frac{1}{2})$, $[\frac{1}{4}, \frac{3}{4})$ et $[\frac{1}{2}, 1)$, cette sous-étape se termine. Autrement on répète les traitements suivants:
 - (1) Si le nouveau sous-intervalle est situé entièrement dans l'intervalle $[0, \frac{1}{2})$, on transmet "0" et tous les "bits suivants" qui proviennent des symboles précédents et prennent la valeur "1" dans ce cas; ensuite on double la largeur du sous-intervalle en agrandissant l'intervalle $[0, \frac{1}{2})$ jusqu'à $[0, 1)$;
 - (2) Si le nouveau sous-intervalle est situé entièrement dans l'intervalle $[\frac{1}{2}, 1)$, on transmet "1" et tous les "bits suivants" qui proviennent des symboles précédents et prennent la valeur "0" dans

ce cas; ensuite on double la largeur du sous-intervalle en agrandissant l'intervalle $[\frac{1}{2}, 1)$ jusqu'à $[0, 1)$;

- (3) Si le nouveau sous-intervalle est situé entièrement dans l'intervalle $[\frac{1}{4}, \frac{3}{4})$, on retient ce fait à la sortie prochaine en ajoutant un "bit suivant"; ensuite on double la largeur du sous-intervalle en agrandissant l'intervalle $[\frac{1}{4}, \frac{3}{4})$ jusqu'à $[0, 1)$.

La figure 3.6 illustre ce processus d'expansion des intervalles.

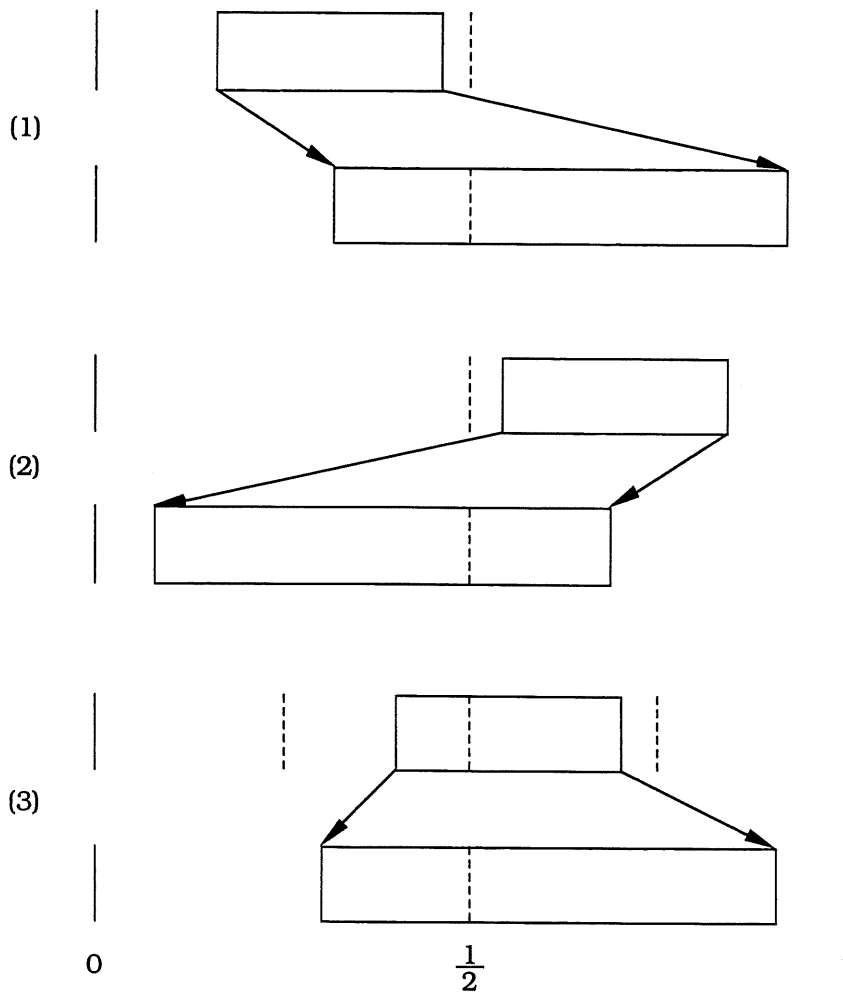


Figure 3.6 Processus d'expansion des intervalles dans le codage arithmétique.

Etape 3: On transmet assez de bits afin de distinguer le "dernier intervalle actuel" d'autres "derniers intervalles" possibles.

Prenons l'exemple d'un fichier composé de quatre symboles "bbbc" et supposons connues les probabilités $p_a = 0.4$, $p_b = 0.5$ et $p_c = 0.1$. Le détail de la procédure de codage est donné dans le tableau 3.5.

TABLEAU 3.5
Détail de la procédure de codage arithmétique

Intervalle actuel	Sortie	Traitement	Sous-intervalle			Entrée
			a	b	c	
[0.00, 1.00)		Subdiviser	[0.00, 0.40)	[0.40, 0.90)	[0.90, 1.00)	b
[0.40, 0.90)		Subdiviser	[0.40, 0.60)	[0.60, 0.85)	[0.85, 0.90)	b
[0.60, 0.85)	1	Agrandir $[\frac{1}{2}, 1)$				
[0.20, 0.70)		Subdiviser	[0.20, 0.40)	[0.40, 0.65)	[0.65, 0.70)	b
[0.40, 0.65)		"Suivre"				
		Agrandir $[\frac{1}{4}, \frac{3}{4})$				
[0.30, 0.80)		Subdiviser	[0.30, 0.50)	[0.50, 0.75)	[0.75, 0.80)	c
[0.75, 0.80)	10	Agrandir $[\frac{1}{2}, 1)$				
[0.50, 0.60)	1	Agrandir $[\frac{1}{2}, 1)$				
[0.00, 0.20)	0	Agrandir $[0, \frac{1}{2})$				
[0.00, 0.40)	0	Agrandir $[0, \frac{1}{2})$				
[0.00, 0.80)	0	Arrêter				

On remarque que le traitement "suivre" consiste à faire suivre le bit de sortie prochain par son opposé.

Dans la pratique, l'intervalle actuel est présenté par des nombres entiers au lieu de décimaux ou fractionnaires afin de faciliter les opérations arithmétiques. On utilise également des entiers pour enregistrer les fréquences d'apparition des symboles avec lesquelles on estime la probabilité des symboles. Le processus de subdivision consiste à choisir des intervalles sans recouvrement dont les largeurs sont approximativement proportionnelles aux nombres d'apparition des symboles.

Reprenons l'exemple montré dans le tableau 3.5 et utilisons un intervalle initial $[0, 1024)$. Les probabilités sont estimées respectivement par $p_a = \frac{410}{1024} \approx 0.4$, $p_b = \frac{512}{1024} = 0.5$ et $p_c = \frac{102}{1024} \approx 0.1$. Le tableau 3.6 montre le détail du codage avec des nombres entiers.

TABLEAU 3.6
Détail du codage arithmétique avec des nombres entiers

Intervalle actuel	Sortie	Traitement	Sous-intervalle			Entrée
			<i>a</i>	<i>b</i>	<i>c</i>	
[0, 1024)		Subdiviser	[0, 410)	[410, 922)	[922, 1024)	<i>b</i>
[410, 922)		Subdiviser	[410, 615)	[615, 871)	[871, 922)	<i>b</i>
[615, 871)	1	Agrandir [512, 1024)				
[206, 718)		Subdiviser	[206, 411)	[411, 667)	[677, 718)	<i>b</i>
[411, 667)		"Suivre"				
		Agrandir [256, 768)				
[310, 822)		Subdiviser	[310, 515)	[515, 771)	[771, 822)	<i>c</i>
[771, 822)	10	Agrandir [512, 1024)				
[518, 620)	1	Agrandir [512, 1024)				
[12, 216)	0	Agrandir [0, 512)				
[24, 432)	0	Agrandir [0, 512)				
[48, 864)	0	Arrêter				

Dans cet exemple, les subdivisions ne sont pas tout à fait pareilles que celles montrées dans le tableau 3.5 parce que les intervalles obtenus sont arrondis aux nombres entiers les plus proches.

3.2.5 Comparaison des performances de quantification

Les deux approches proposées plus haut sont évaluées dans ce paragraphe et leurs performances de quantification sont comparées avec celles d'une approche statistique. On choisit l'erreur quadratique moyenne comme mesure de distorsion.

La simulation est basée sur une source gaussienne à moyenne nulle de variance unité. On fait vingt tirages aléatoires de 10000 vecteurs à huit dimensions pour générer le signal d'entrée des quantificateurs vectoriels. Pour chaque ensemble de 10000 vecteurs, une erreur quadratique moyenne est calculée. Et la distorsion totale de quantification est la moyenne des vingt erreurs quadratiques moyennes obtenues.

La figure 3.7 illustre le schéma de principe de la simulation. Dans un premier temps, un vecteur d'entrée x est quantifié par un des sous-quantificateurs sans recouvrement ou sous-quantificateurs encastrés. Alors on obtient deux quantités: l'indice i du mot de code et le numéro n du sous-quantificateur utilisé. Dans un deuxième temps, on code le numéro n en utilisant le codage arithmétique et l'on obtient le mot de code binaire k . Ce sont l'indice i et le mot de code k qui sont transmis au récepteur.

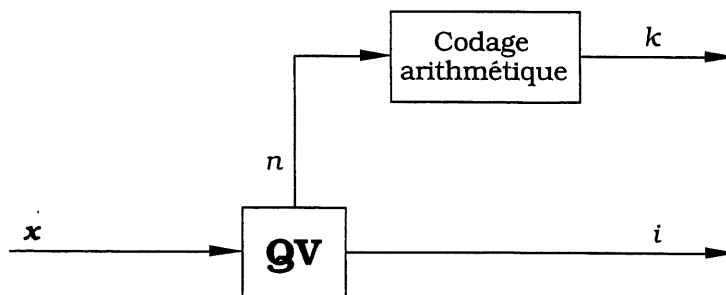


Figure 3.7 Schéma de principe de la simulation sur les approches algébriques proposées.

On remarque que dans la simulation, les vecteurs d'entrée sont ajustés en amplitude par un facteur d'échelle fixé en fonction du débit spécifié. Le tableau 3.7 donne les résultats obtenus pour différents débits.

TABLEAU 3.7
Performances de quantification
des approches algébriques proposées

Débit (bit)	RSB (dB)	
	Approche complémentaire	Approche encastrée
1.01	4.3275	4.3267
1.18	5.3078	5.3046
1.39	6.6416	6.6401
1.70	8.5568	8.5504
1.91	9.8873	9.8811
2.09	11.0581	11.0562
2.34	12.6881	12.6414
2.41	12.9281	12.8403

Une approche statistique est aussi utilisée dans la simulation décrite à la figure 3.7 pour établir une comparaison des performances avec les deux approches algébriques proposées. Cette approche statistique consiste à utiliser un ensemble de six sous-quantificateurs vectoriels à deux dimensions sans recouvrement. Les dictionnaires des sous-quantificateurs comprennent respectivement 1, 2, 4, 8, 16 et 32 mots de code qui sont obtenus par l'emploi de l'algorithme de K -moyennes sur 50000 vecteurs gaussiens. La figure 3.8 montre ces dictionnaires.

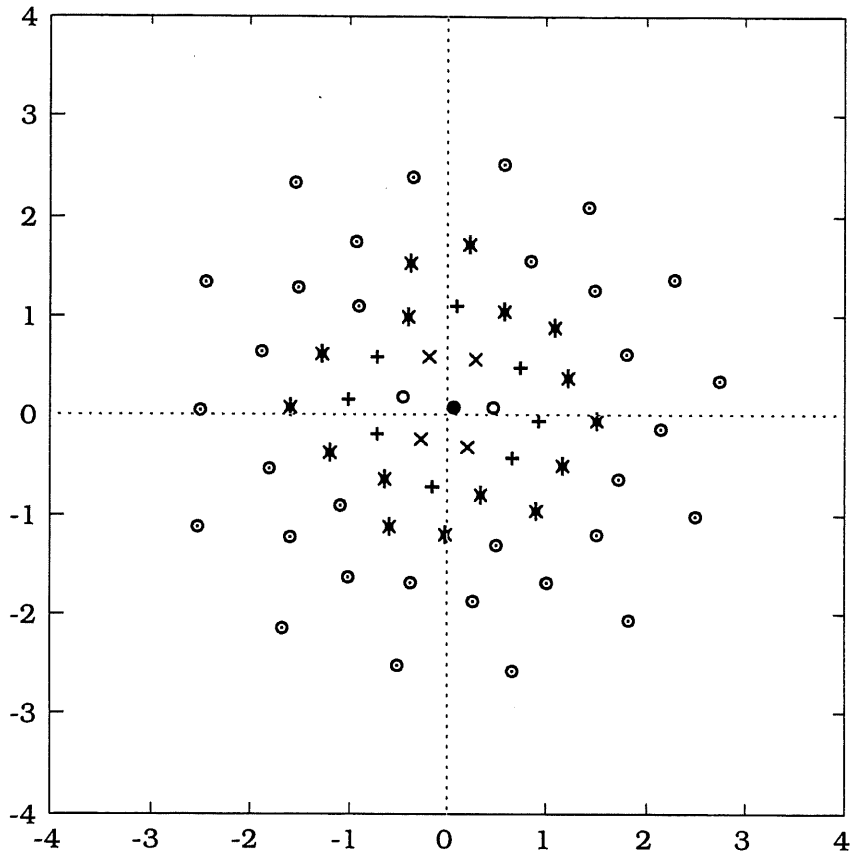


Figure 3.8 Dictionnaires des sous-quantificateurs statistiques.

La figure 3.9 permet de comparer les résultats montrés dans le tableau 3.7 avec ceux obtenus par l'approche statistique ainsi que les limites prédites par la théorie de l'information [1].

En conclusion, on obtient presque les même résultats en terme d'erreur quadratique moyenne dans la simulation en utilisant les deux approches algébriques. De plus, à mesure que le débit augmente, les performances de quantification des approches algébriques sont supérieures à celles de l'approche statistique.

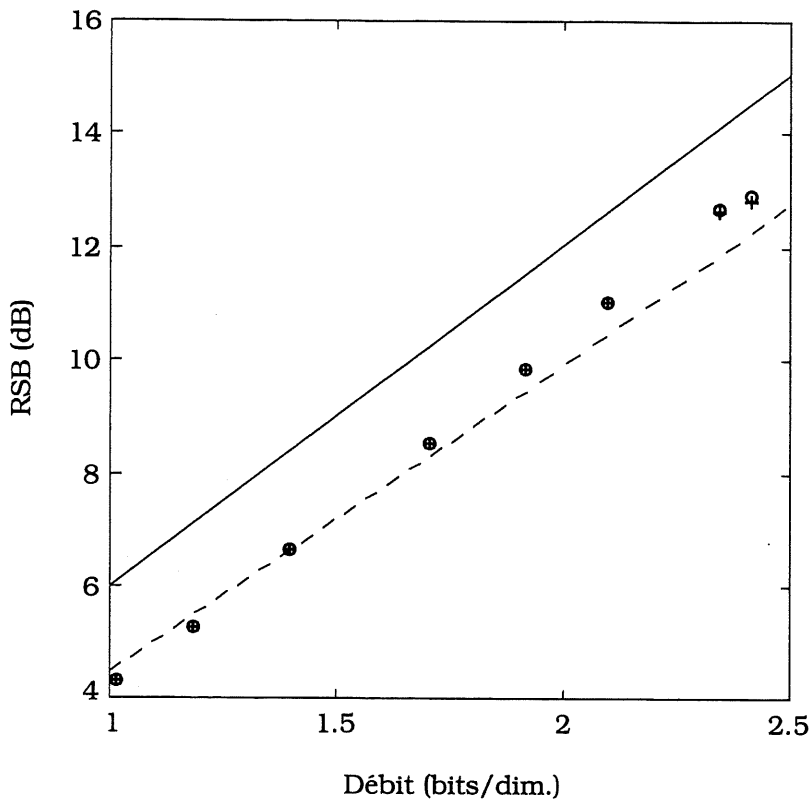


Figure 3.9 Comparaison des performances de quantification entre les approches algébriques et l'approche statistique (l'approche complémentaire: 'o'; l'approche encastrée: '+'; l'approche statistique: '---'; la limite supérieure théorique: '—').

3.3 Application au codage de la parole en bande élargie

A l'aide d'un codeur TCX, nous appliquons l'approche encastrée proposée précédemment au codage de la parole en bande élargie (50 - 7000 Hz) pour un débit de 16 kbit/s. Un groupe de quantificateurs vectoriels algébriques encastrés est utilisé pour quantifier le signal dit "cible" dans le domaine fréquentiel.

3.3.1 Codeur TCX et propriétés de la cible

Le modèle de codage TCX mentionné au paragraphe 3.1.2 est une technique hybride combinant le codage par transformée et le codage prédictif. Cette technique offre l'avantage de permettre la quantification vectorielle sur le domaine fréquentiel dans lequel le masquage fréquentiel est facilement mis à profit pour assurer le masquage du bruit de quantification. Comme on le sait, le quantificateur vectoriel possède les performances meilleures que le quantificateur scalaire dès que la dimension du vecteur à quantifier devient supérieure à une valeur relativement faible. Les résultats donnés dans la référence [36] montrent que le codeur TCX permet d'obtenir une très haute qualité pour la parole en bande élargie (50 - 7000 Hz).

Dans notre application, le signal de la parole en bande élargie (50 - 7000 Hz) échantillonné à 16 kHz est codé par un codeur TCX à 16 kbit/s dont le principe est montré à la figure 3.2. Les valeurs des principaux paramètres de ce codeur TCX sont données dans le tableau 3.8 [38].

TABLEAU 3.8
Paramètres du codeur TCX à 16 kbit/s

Paramètre	Valeur
Longueur de la trame	96 éch. (6 ms)
Ordre du filtre LPC	16
Facteur perceptuel (Filtre $W(z)$)	0.75
Délai minimum du "pitch"	30 éch. (533 Hz)
Délai maximum du "pitch"	285 éch. (56 Hz)
Gain de saturation du "pitch"	1.2
Longueur de la fenêtre d'analyse LPC	320 éch. (20 ms)

Pour chaque trame de 96 échantillons, il y a 96 bits disponibles. Le tableau 3.9 donne le plan de distribution des 96 bits sur une trame [38].

TABLEAU 3.9
Distribution des bits pour une trame de
96 échantillons du codeur TCX à 16 kbit/s

Paramètre	Bit
Délai du "pitch"	8
Gain du "pitch"	4
Filtre LPC	12
Cible	72

On remarque que les 16 coefficients du filtre LPC sont quantifiés sur 48 bits toutes les quatre trames ($6 \text{ ms} \times 4 = 24 \text{ ms}$), c'est-à-dire 12 bits par trame.

On peut voir que la plupart des bits disponibles (jusqu'à 75%) sont utilisés pour quantifier le signal dit "cible" dans le domaine fréquentiel. Par conséquent, la conception du quantificateur pour la cible joue un rôle important dans l'amélioration de la qualité du codage.

Pour quantifier efficacement la cible, il est nécessaire de connaître les caractéristiques statistiques du spectre de ce signal. Une analyse statistique détaillée sur le spectre de la cible est décrite dans les références [38] et [52]. Les résultats de l'analyse se résument comme suit:

- L'enveloppe du spectre d'amplitude de la cible présente une structure de formant et les basses fréquences sont beaucoup plus énergétiques que les hautes fréquences. De plus, les spectres d'amplitude successifs de la cible sont fortement corrélés.

- Les phases de la cible sont approximativement à distribution uniforme entre $-\pi$ et π et ne sont pas corrélées d'une trame à une autre.
- Les coefficients complexes de la transformation normalisés par les amplitudes correspondants, appelés aussi la cible complexe normalisée, possèdent une distribution sphérique qui s'approche de celle des sources gaussiennes.

En examinant l'analyse statistique citée ci-dessus et les résultats expérimentaux obtenus au paragraphe 3.2.5, on voit que l'approche encadrée proposée précédemment est un candidat idéal pour la quantification de la cible dans le domaine fréquentiel.

3.3.2 Quantification de la cible

La figure 3.10 montre le principe de l'approche proposée pour la quantification de la cible dans le domaine fréquentiel et appliquée au codage de la parole en bande élargie (50 - 7000 Hz) à 16 kbit/s.

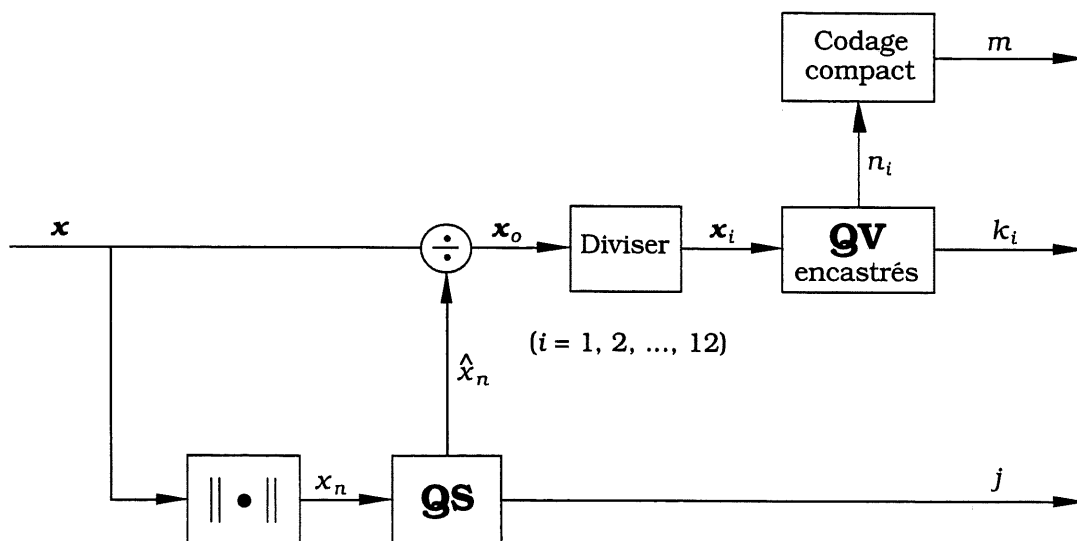


Figure 3.10 Principe de la quantification de la cible dans le domaine fréquentiel pour le codage de la parole en bande élargie (50 - 7000 Hz) à 16 kbit/s.

Après transformée de Fourier discrète, le signal cible est représenté par un vecteur transformé \mathbf{x} de dimension 96 (48 nombres complexes). Le vecteur \mathbf{x} est ensuite décomposé en deux éléments différents:

- la norme x_n (le gain global) de \mathbf{x} ;
- l'orientation \mathbf{x}_o (le vecteur normalisé) de \mathbf{x} .

On quantifie séparément ces deux éléments pour réduire la complexité de la solution.

La norme x_n est calculée par

$$x_n = \sqrt{\sum_{i=1}^{96} x_i^2} \quad (3.8)$$

et quantifiée par un quantificateur scalaire (QS). La version quantifiée de x_n , dénotée \hat{x}_n , est utilisée pour normaliser le vecteur \mathbf{x} . Alors, on obtient l'orientation \mathbf{x}_o par

$$\mathbf{x}_o = \frac{\mathbf{x}}{\hat{x}_n}. \quad (3.9)$$

L'orientation \mathbf{x}_o est divisée en 12 sous-vecteurs à huit dimensions et chaque sous-vecteur est quantifié par un groupe de quantificateurs vectoriels (QV) algébriques encadrés décrit au paragraphe 3.2.3. Un facteur d'échelle permet d'assurer que les sous-vecteurs à quantifier se trouvent tous dans la région de support des dictionnaires des sous-quantificateurs encadrés. Finalement, les numéros des 12 sous-quantificateurs utilisés sont codés de façon compacte.

Pour assurer de bonnes performances de quantification, il est important d'allouer convenablement les bits disponibles entre la norme x_n et l'orientation \mathbf{x}_o . Des expériences montrent qu'une résolution entre 5 et 7 bits pour la quantification de la

norme x_n permet d'atteindre une qualité suffisante, c'est-à-dire qu'aucune dégradation perceptible n'est introduite. Tenant compte des tailles des dictionnaires des sous-quantificateurs algébriques encastrés (les tailles en bits sont des multiples de 4), les 72 bits disponibles pour la quantification de la cible sont alloués comme suit:

- 7 bits pour la quantification de la norme x_n ;
- 65 bits pour la quantification de l'orientation \mathbf{x}_o , y compris un bit d'indication de mode qui sera décrit ultérieurement.

La norme x_n représente la contour d'énergie du signal de parole. Pour des raisons d'ordre physiologique, elle intervient par le biais d'une échelle logarithmique. Plus précisément, l'oreille est très sensible aux signaux de petite énergie. Par conséquent, nous quantifions la norme x_n en utilisant un quantificateur logarithmique construit selon la loi μ [16]. Les petites normes sont quantifiées avec plus de niveaux de quantification et les grandes normes sont quantifiées avec moins de niveaux de quantification. Cela assure un RSB essentiellement indépendant de la variance de x_n dans une gamme de normes aussi large que possible. Donc, la grande robustesse aux modifications de la statistique de la norme x_n est un avantage important de ce quantificateur. On peut aussi utiliser d'autres techniques de codage telles que le codage Δ , le codage MICDA ainsi que la quantification vectorielle pour quantifier la norme x_n . Mais le quantificateur logarithmique est de complexité moindre tout en offrant une qualité suffisante.

L'approche encastrée proposée dans la section précédente est utilisée pour quantifier chaque sous-vecteur de l'orientation \mathbf{x}_o . Il y a deux modes de quantification pour les 65 bits alloués à l'orientation \mathbf{x}_o . Dans le premier mode, les 12 sous-vecteurs de \mathbf{x}_o sont quantifiés sur 44 bits et les 12 numéros des sous-quantificateurs utilisés sont codés sur 20 bits. Le deuxième mode se base sur un résultat de l'analyse statistique concernant la quantification des six derniers sous-vecteurs. Nous avons observé qu'une fois sur deux les six derniers sous-vecteurs sont quantifiés par le vecteur zéro (Q_0). Dans ce cas, on

concentre tous les bits disponibles sur les six premiers sous-vecteurs afin de les quantifier avec une résolution élevée. Dans le deuxième mode, on alloue 52 bits à la quantification des six premiers sous-vecteurs et 12 bits au codage des six numéros des sous-quantificateurs utilisés.

Nous codons les numéros des sous-quantificateurs en utilisant la méthode algébrique décrite au paragraphe 3.2.3 au lieu du codage entropique. Cette méthode réalise un codage compact sans distorsion. Pour un nombre donné N_q de bits ($N_q = 44$ bits dans le premier mode de quantification et $N_q = 52$ bits dans le deuxième mode de quantification), les numéros des sous-quantificateurs utilisés forment un vecteur de composantes non négatives dont la somme est égale à $\frac{N_q}{4}$. Tous les vecteurs possibles peuvent être obtenus par permutation d'un certain nombre de leaders. Les tableaux 3.10 et 3.11 donnent ces leaders correspondant respectivement au premier mode de quantification et au deuxième mode de quantification.

Lorsqu'on quantifie l'orientation \mathbf{x}_o selon un des deux modes, il est possible que le nombre de bits utilisés ne soit pas égal à N_q bits. Si ce nombre de bits est supérieur à N_q bits, on réduit l'orientation \mathbf{x}_o en augmentant successivement la norme quantifiée \hat{x}_n par laquelle le vecteur \mathbf{x} est normalisé. La modification de \hat{x}_n s'arrête aussitôt que le nombre de bits utilisés est égal ou inférieur à N_q bits. Dans le cas inférieur à N_q bits, on représente certains sous-quantificateurs de débit faible en augmentant leurs numéros selon la limite de $\frac{N_q}{4}$, afin de faciliter le codage des numéros des sous-quantificateurs.

TABLEAU 3.10
 Leaders pour le codage des numéros des sous-quantificateurs
 dans le premier mode de quantification de l'orientation α_0

Leader	Taille
5 5 1 0 0 0 0 0 0 0 0	660
5 4 2 0 0 0 0 0 0 0 0	1320
5 4 1 1 0 0 0 0 0 0 0	5940
5 3 3 0 0 0 0 0 0 0 0	660
5 3 2 1 0 0 0 0 0 0 0	11880
5 3 1 1 1 0 0 0 0 0 0	15840
5 2 2 2 0 0 0 0 0 0 0	1980
5 2 2 1 1 0 0 0 0 0 0	23760
5 2 1 1 1 1 0 0 0 0 0	27720
5 1 1 1 1 1 1 0 0 0 0	5544
4 4 3 0 0 0 0 0 0 0 0	660
4 4 2 1 0 0 0 0 0 0 0	5940
4 4 1 1 1 0 0 0 0 0 0	7920
4 3 3 1 0 0 0 0 0 0 0	5940
4 3 2 2 0 0 0 0 0 0 0	5940
4 3 2 1 1 0 0 0 0 0 0	47520
4 3 1 1 1 1 0 0 0 0 0	27720
4 2 2 2 1 0 0 0 0 0 0	15840
4 2 2 1 1 1 0 0 0 0 0	55440
4 2 1 1 1 1 1 0 0 0 0	33264
4 1 1 1 1 1 1 1 0 0 0	3960
3 3 3 2 0 0 0 0 0 0 0	1980
3 3 3 1 1 0 0 0 0 0 0	7920
3 3 2 2 1 0 0 0 0 0 0	23760
3 3 2 1 1 1 0 0 0 0 0	55440
3 3 1 1 1 1 1 0 0 0 0	16632
3 2 2 2 2 0 0 0 0 0 0	3960
3 2 2 2 1 1 0 0 0 0 0	55440
3 2 2 1 1 1 1 0 0 0 0	83160
3 2 1 1 1 1 1 1 0 0 0	27720
3 1 1 1 1 1 1 1 1 0 0	1980
2 2 2 2 2 1 0 0 0 0 0	5544
2 2 2 2 1 1 1 0 0 0 0	27720
2 2 2 1 1 1 1 1 0 0 0	27720
2 2 1 1 1 1 1 1 1 0 0	7920
2 1 1 1 1 1 1 1 1 1 0	660
1 1 1 1 1 1 1 1 1 1 0	12
Taille totale	653016

TABLEAU 3.11
Leaders pour le codage des numéros des sous-quantificateurs
dans le deuxième mode de quantification de l'orientation α_0

Leader						Taille
5	5	3	0	0	0	60
5	5	2	1	0	0	180
5	5	1	1	1	0	60
5	4	4	0	0	0	60
5	4	3	1	0	0	360
5	4	2	2	0	0	180
5	4	2	1	1	0	360
5	4	1	1	1	1	30
5	3	3	2	0	0	180
5	3	3	1	1	0	180
5	3	2	2	1	0	360
5	3	2	1	1	1	120
5	2	2	2	2	0	30
5	2	2	2	1	1	60
4	4	4	1	0	0	60
4	4	3	2	0	0	180
4	4	3	1	1	0	180
4	4	2	2	1	0	180
4	4	2	1	1	1	60
4	3	3	3	0	0	60
4	3	3	2	1	0	360
4	3	3	1	1	1	60
4	3	2	2	2	0	120
4	3	2	2	1	1	180
4	2	2	2	2	1	30
3	3	3	3	1	0	30
3	3	3	2	2	0	60
3	3	3	2	1	1	60
3	3	2	2	2	1	60
3	2	2	2	2	2	6
Taille totale						3906

En résumé, la quantification de l'orientation α_0 procède par les quatre étapes suivantes:

Etape 1: Quantifier l'orientation α_0 selon le deuxième mode de quantification (avec 52 bits).

Etape 2: Vérifier les six derniers sous-vecteurs quantifiés. Si aucun d'entre eux n'est quantifié par le vecteur zéro (Q_0), modifier la norme quantifiée \hat{x}_n et quantifier l'orientation x_o selon le premier mode de quantification (avec 44 bits).

Etape 3: Coder les numéros des sous-quantificateurs utilisés soit avec 20 bits pour le premier mode de quantification (44 bits) soit avec 12 bits pour le deuxième mode de quantification (52 bits).

Etape 4: Indiquer le mode de quantification choisi par le bit d'indication de mode.

3.3.3 Performances et comparaison

Nous avons utilisé 41 phrases-tests échantillonnées à 16 kHz, 20 phrases prononcées par des hommes et 21 phrases prononcées par des femmes, pour évaluer l'approche proposée au paragraphe précédant. Ces phrases-tests sont différentes. Elles sont dénuées de bruit de fond et constituent une très bonne référence.

Dans le but de faire la comparaison, une approche complexe statistique décrite dans la référence [38] est aussi appliquée sur ces 41 phrases-tests avec le même codeur TCX. Dans cette approche statistique, le vecteur transformé représentant la cible est également décomposé en sa norme et son orientation. La norme est quantifiée scalairement sur 7 bits et l'orientation est quantifiée sur 65 bits. L'orientation se compose de 48 nombres complexes. Le spectre d'amplitude des 48 nombres complexes est quantifié vectoriellement sur 8 bits. Les 57 bits restants sont tous alloués aux 48 nombres complexes normalisés par les amplitudes correspondants. Avec une résolution entre 0 et 8 bits, chaque nombre complexe normalisé est quantifié par un quantificateur vectoriel à deux dimensions. Une allocation dynamique de bits est utilisée pour déterminer la résolution accordée à la quantification de chaque nombre complexe normalisé. Tous les

dictionnaires des quantificateurs de cette approche statistique sont construits à l'aide de l'algorithme de K-moyennes décrit au chapitre 1.

Le tableau 3.12 donne les résultats de test obtenus, en terme de RSB total moyen (RSB) et RSB segmentaire moyen (RSBseg), avec l'approche algébrique proposée et l'approche complexe statistique.

TABLEAU 3.12
Comparaison des performances entre l'approche algébrique proposée et l'approche complexe statistique

Approche	RSB (dB)	RSBseg (dB)
Approche algébrique	19.33	16.95
Approche complexe	19.01	16.41

On peut remarquer que l'approche algébrique proposée obtient des résultats objectifs légèrement meilleurs que ceux de l'approche complexe statistique. Les tests subjectifs ont été effectués par des écoutes informelles. Les deux approches permettent de fournir une très bonne qualité de parole et les différences entre les deux approches sont difficilement perceptibles.

Cependant, l'approche algébrique proposée offre des avantages importants en matière de stockage et de complexité de calcul par rapport à l'approche complexe statistique. Dans l'approche complexe, les dictionnaires des quantificateurs statistiques sont stockés dans un espace mémoire de taille 13438. Et dans l'approche algébrique, les mots de code des dictionnaires sont générés par permutation des leaders et le stockage des leaders et paramètres correspondants nécessite une capacité mémoire totale de 3542. Une

économie de 74% est donc obtenue par l'approche algébrique. De plus, la recherche du plus proche voisin est très rapide dans l'approche algébrique car elle s'opère seulement au niveau des leaders. A mesure que la longueur de la trame du codeur TCX augmente, les avantages de l'approche algébrique deviendront de plus en plus remarquables.

Conclusion

De nouvelles techniques de quantification vectorielle algébrique ont été présentées dans cette thèse. Ces techniques efficaces et pratiques ont permis de réduire la complexité (en stockage et en calcul) de la quantification vectorielle et donc ont été appliquées au codage du signal de parole.

Les contributions importantes de cette thèse peuvent se résumer dans les points suivants:

1. Quantification Vectorielle Algébrique Ellipsoïdale

Nous avons proposé des algorithmes rapides de codage pour les réseaux réguliers connus A_2 , D_N et E_8 . En utilisant la technique du "shaping", on a construit algébriquement des dictionnaires ellipsoïdaux basés sur ces réseaux réguliers. Ce quantificateur vectoriel algébrique ellipsoïdal permet de quantifier efficacement des signaux de distribution ellipsoïdale, tels que les coefficients de transformée de Karhunen-Loève et de transformée en ondelettes dans le codage par transformée. En contraste avec les approches de Fischer [15] et de Moureaux [18], nous avons résolu le problème de l'indexation des mots de code.

2. Quantification Vectorielle Algébrique des Paramètres LSF

Une nouvelle méthode de la quantification des paramètres LSF a été proposée. Tenant compte des caractéristiques statistiques des vecteurs à quantifier, nous avons développé les quantificateurs vectoriels algébriques triangulaires et pyramidaux. Leurs dictionnaires n'ont pas besoin d'être mémorisés et la recherche et l'indexation des mots

de code sont très rapides. En assurant une qualité transparente de la quantification des paramètres LSF, cette technique a réalisé une réduction de débit par rapport à la quantification vectorielle statistique. De plus, elle a réduit considérablement la complexité de la quantification.

3. Quantification Vectorielle Algébrique Basée sur des Codes Sphériques

Une nouvelle technique de quantification vectorielle à débit variable a été développée à partir des codes algébriques sphériques. Pour une source gaussienne, elle a permis d'obtenir les performances meilleures que la quantification vectorielle statistique à débit variable. A l'aide d'un codeur TCX, nous avons appliqué cette technique au codage de la parole en bande élargie (50 - 7000 Hz) pour un débit total de 16 kbit/s. Les résultats des tests ont montré qu'elle permet d'offrir une très bonne qualité de parole synthétisée. D'autre part, la complexité de cette technique est moins grande. En comparaison de l'approche complexe statistique décrite dans la référence [38], une économie de 74% en stockage est obtenue par la technique proposée et la recherche du plus proche voisin est très rapide à cause de l'utilisation des algorithmes algébriques.

Des résultats obtenus dans cette étude ont conduit à la publication de trois articles techniques [53][54][55] dans le journal *IEEE Transactions on Speech and Audio Processing* et à la conférence *IEEE International Conference on Acoustics, Speech, and Signal Processing*.

Les travaux présentés dans cette thèse confirment que la quantification vectorielle algébrique est une technique de quantification puissante. Outre le codage de la parole, cette technique a de nombreuses applications dans la compression des signaux sous toutes ses formes.

ANNEXE

A. Algorithme de Conway et Sloane [14]

Soit Λ un réseau régulier dans \mathbf{R}^N , m un nombre entier positif et \mathbf{a} un vecteur dans \mathbf{R}^N . Le code de Voronoï $C_\Lambda(m, \mathbf{a})$ comporte m^N vecteurs de Λ dans la région de Voronoï $m\Lambda$ autour du point \mathbf{a} . Grâce à une propriété de la région de Voronoï $m\Lambda$, les vecteurs indices associés aux m^N vecteurs de cette région sont tous distincts.

Soit G la matrice génératrice de Λ , \mathbf{x} un vecteur appartenant au code $C_\Lambda(m, \mathbf{a})$ et $\mathbf{k} = [k_1, k_2, \dots, k_N]$ ($0 \leq k_i < m$) le vecteur indice associé à \mathbf{x} .

A partir de \mathbf{k} , on peut obtenir \mathbf{x} en effectuant les étapes suivantes:

- $\mathbf{x}' = \mathbf{k}G$.
- $\mathbf{z} = \frac{\mathbf{x}' - \mathbf{a}}{m}$.
- $\mathbf{v} = \mathcal{Q}_\Lambda(\mathbf{z})$, où la fonction $\mathcal{Q}_\Lambda(\cdot)$ effectue la recherche du plus proche voisin dans le réseau Λ .
- $\mathbf{x} = \mathbf{x}' - m\mathbf{v} - \mathbf{a}$.

A partir de \mathbf{x} , le vecteur indice \mathbf{k} est calculé par les étapes suivantes:

- $\mathbf{j} = (\mathbf{x} + \mathbf{a})G^{-1} (= (\mathbf{x}' - m\mathbf{v})G^{-1} = \mathbf{k} - m\mathbf{v}G^{-1})$.
- $\mathbf{k} = \mathbf{j} \pmod{m}$.

B. Algorithme proposé pour A_2

Soit $\mathbf{m} = [m_1, m_2]$ un vecteur composé de deux composantes entières positives et $C_{A_2}(\mathbf{m}, \mathbf{0})$ un ensemble de vecteurs de A_2 dans la région ellipsoïdale centrée sur l'origine $\mathbf{0}$ et

spécifiée par \mathbf{m} . Les composantes m_1 et m_2 ont la même parité (toutes paires ou toutes impaires). On suppose que le vecteur \mathbf{x} appartient à $C_{A_2}(\mathbf{m}, \mathbf{0})$ et son vecteur indice \mathbf{k} vérifie la relation $0 \leq k_i < m_i$ ($i = 1, 2$).

Pour le réseau A_2 , la matrice génératrice G et son inverse G^{-1} sont définis comme suit [14]

$$G = \begin{bmatrix} 1 & 0 \\ \frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix}$$

et

$$G^{-1} = \begin{bmatrix} 1 & 0 \\ -\frac{\sqrt{3}}{3} & \frac{2\sqrt{3}}{3} \end{bmatrix}.$$

$\mathbf{k} \rightarrow \mathbf{x}$:

- $\mathbf{x}' = \mathbf{k}G$.
- $\mathbf{z} = [\frac{x'_1}{m_1}, \frac{x'_2}{m_2}]$.
- $\mathbf{v} = Q_{A_2}(\mathbf{z})$. A cause de $0 \leq k_i < m_i$, la valeur de v_2 est soit égale à 0 soit égale à $\frac{\sqrt{3}}{2}$, et la valeur de v_1 est non négative.
- $\mathbf{x} = \mathbf{x}' - [m_1 v_1, m_2 v_2]$.

$\mathbf{x} \rightarrow \mathbf{k}$:

- $\mathbf{j} = \mathbf{x}G^{-1} = (\mathbf{x}' - [m_1 v_1, m_2 v_2])G^{-1} = \mathbf{k} - [m_1 v_1, m_2 v_2]G^{-1}$
 $= \mathbf{k} - [m_1 v_1 - \frac{\sqrt{3}}{3} m_2 v_2, \frac{2\sqrt{3}}{3} m_2 v_2]$.

- Si $j_2 \geq 0$, alors $v_2 = 0$. Selon la définition du réseau A_2 [3], on a $v_1 = 0, 1, 2, \dots$. Donc, les indices sont calculés par

$$k_1 = j_1 + m_1 v_1 = j_1 \pmod{m_1}$$

et

$$k_2 = j_2 \pmod{m_2}.$$

- Si $j_2 < 0$, alors $v_2 = \frac{\sqrt{3}}{2}$ et $v_1 = \frac{1}{2}, \frac{3}{2}, \frac{5}{2}, \dots$. Donc, on a

$$k_1 = j_1 + m_1 v_1 - \frac{m_2}{2} = j_1 + \frac{m_1 - m_2}{2} \pmod{m_1}$$

et

$$k_2 = j_2 + m_2 = j_2 \pmod{m_2}.$$

C. Algorithme proposé pour D_N

Soit $\mathbf{m} = [m_1, m_2, \dots, m_N]$ un vecteur composé de N composantes entières positives et $C_{D_N}(\mathbf{m}, \mathbf{0})$ un ensemble de vecteurs de D_N dans la région ellipsoïdale centrée sur l'origine $\mathbf{0}$ et spécifiée par \mathbf{m} . Les composantes m_i ($i = 1, 2, \dots, N$) ont la même parité (toutes paires ou toutes impaires). On suppose que le vecteur \mathbf{x} appartient à $C_{D_N}(\mathbf{m}, \mathbf{0})$ et son vecteur indice \mathbf{k} vérifie la relation $0 \leq k_i < m_i$ ($i = 1, 2, \dots, N$).

Pour le réseau D_N , on a [14]

$$G = \begin{bmatrix} 2 & 0 & 0 & \dots & 0 \\ 1 & 1 & 0 & \dots & 0 \\ 1 & 0 & 1 & \dots & 0 \\ \cdot & \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot \\ 1 & 0 & 0 & \dots & 1 \end{bmatrix}$$

et

$$G^{-1} = \begin{bmatrix} \frac{1}{2} & 0 & 0 & \dots & 0 \\ -\frac{1}{2} & 1 & 0 & \dots & 0 \\ -\frac{1}{2} & 0 & 1 & \dots & 0 \\ \cdot & \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot \\ -\frac{1}{2} & 0 & 0 & \dots & 1 \end{bmatrix} .$$

 $\mathbf{k} \rightarrow \mathbf{x}$:

- $\mathbf{x}' = \mathbf{k}G$.
- $\mathbf{z} = [\frac{x'_1}{m_1}, \frac{x'_2}{m_2}, \dots, \frac{x'_N}{m_N}]$.
- $\mathbf{v} = \mathcal{Q}_{D_N}(\mathbf{z})$. A cause de $0 \leq k_i < m_i$, les valeurs de v_i ($i = 2, 3, \dots, N$) sont soit égales à 0 soit égales à 1, et la composante v_1 est un nombre entier non négatif.
- $\mathbf{x} = \mathbf{x}' - [m_1 v_1, m_2 v_2, \dots, m_N v_N]$.

 $\mathbf{x} \rightarrow \mathbf{k}$:

- $\mathbf{j} = \mathbf{x}G^{-1} = (\mathbf{x}' - [m_1 v_1, m_2 v_2, \dots, m_N v_N])G^{-1} = \mathbf{k} - [m_1 v_1, m_2 v_2, \dots, m_N v_N]G^{-1}$
 $= \mathbf{k} - [\frac{1}{2}(m_1 v_1 - \sum_{i=2}^N m_i v_i), m_2 v_2, \dots, m_N v_N]$.
- Si $j_i \geq 0$ ($i \neq 1$), alors $v_i = 0$; si $j_i < 0$ ($i \neq 1$), alors $v_i = 1$.
- $\mathbf{k} = \mathbf{j} + [\frac{1}{2}(m_1 v_1 - \sum_{i=2}^N m_i v_i), m_2 v_2, \dots, m_N v_N]$.

Puisque le vecteur \mathbf{v} appartient au réseau D_N , la somme de composantes de \mathbf{v} est un multiple de 2. Donc, on a

$$\begin{aligned}
k_1 &= j_1 + \frac{1}{2}(m_1 v_1 - \sum_{i=2}^N m_i v_i) = j_1 + \frac{1}{2}(m_1 v_1 - \sum_{i=2}^N m_i v_i) + \frac{1}{2} m_1 \sum_{i=1}^N v_i \\
&= j_1 + \frac{1}{2} \left(\sum_{i=2}^N (m_1 - m_i) v_i \right) + m_1 v_1.
\end{aligned}$$

$$k_1 = j_1 + \frac{1}{2} \left(\sum_{i=2}^N (m_1 - m_i) v_i \right) \pmod{m_1};$$

$$k_i = j_i + m_i v_i = j_i \pmod{m_i}, \quad i = 2, 3, \dots, N.$$

D. Algorithme proposé pour E_8

Soit $\mathbf{m} = [m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8]$ un vecteur composé de huit composantes entières positives et $C_{E_8}(\mathbf{m}, \mathbf{0})$ un ensemble de vecteurs de E_8 dans la région ellipsoïdale centrée sur l'origine $\mathbf{0}$ et spécifiée par \mathbf{m} . Les composantes du vecteur \mathbf{m} satisfont aux trois contraintes suivantes:

- toutes les composantes ont la même parité (toutes paires ou toutes impaires);
- la somme de composantes est un multiple de 4;
- $m_i \geq m_8$ ($i = 1, 2, \dots, 7$).

On suppose que le vecteur \mathbf{x} appartient à $C_{E_8}(\mathbf{m}, \mathbf{0})$ et son vecteur indice \mathbf{k} vérifie la relation $0 \leq k_i < m_i$ ($i = 1, 2, \dots, 8$).

Pour le réseau E_8 , la matrice génératrice G et son inverse G^{-1} sont [14]

$$G = \begin{bmatrix}
2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
\frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2}
\end{bmatrix}$$

et

$$G^{-1} = \begin{bmatrix} \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{1}{2} & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -\frac{1}{2} & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ -\frac{1}{2} & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -\frac{1}{2} & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ -\frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \frac{5}{2} & -1 & -1 & -1 & -1 & -1 & -1 & 2 \end{bmatrix}.$$

 $\mathbf{k} \rightarrow \mathbf{x}$:

- $\mathbf{x}' = \mathbf{k}G$.
- $\mathbf{z} = [\frac{x'_1}{m_1}, \frac{x'_2}{m_2}, \frac{x'_3}{m_3}, \frac{x'_4}{m_4}, \frac{x'_5}{m_5}, \frac{x'_6}{m_6}, \frac{x'_7}{m_7}, \frac{x'_8}{m_8}]$.
- $\mathbf{v} = \mathcal{G}_{E_8}(\mathbf{z})$. A cause de $0 \leq k_i < m_i$, les composantes v_i ($i = 2, 3, \dots, 7$) prennent leurs valeurs dans l'intervalle $[-0.5, 2]$ et la composante v_8 prend sa valeur dans l'intervalle $[-0.5, 1]$, mais la valeur de v_1 ne peut pas être prévue.
- $\mathbf{x} = \mathbf{x}' - [m_1 v_1, m_2 v_2, m_3 v_3, m_4 v_4, m_5 v_5, m_6 v_6, m_7 v_7, m_8 v_8]$.

 $\mathbf{x} \rightarrow \mathbf{k}$:

$$\begin{aligned} \mathbf{j} &= \mathbf{x}G^{-1} = (\mathbf{x}' - [m_1 v_1, m_2 v_2, m_3 v_3, m_4 v_4, m_5 v_5, m_6 v_6, m_7 v_7, m_8 v_8])G^{-1} \\ &= \mathbf{k} - [m_1 v_1, m_2 v_2, m_3 v_3, m_4 v_4, m_5 v_5, m_6 v_6, m_7 v_7, m_8 v_8]G^{-1} \\ &= \mathbf{k} - [\frac{1}{2}(m_1 v_1 - \sum_{i=2}^7 m_i v_i + 5m_8 v_8), m_2 v_2 - m_8 v_8, m_3 v_3 - m_8 v_8, m_4 v_4 - m_8 v_8, \\ &\quad m_5 v_5 - m_8 v_8, m_6 v_6 - m_8 v_8, m_7 v_7 - m_8 v_8, 2m_8 v_8]. \end{aligned}$$

- La valeur de v_8 peut se déduire ci-dessous de la valeur de j_8 :

$$m_8 \leq j_8 \quad \rightarrow \quad v_8 = -0.5,$$

$$0 \leq j_8 < m_8 \quad \rightarrow \quad v_8 = 0,$$

$$-m_8 \leq j_8 < 0 \quad \rightarrow \quad v_8 = 0.5,$$

$$j_8 < -m_8 \quad \rightarrow \quad v_8 = 1.$$

- Si $v_8 = 0, 1$ alors

$$v_i = \begin{cases} 0, & 0 \leq (j_i - v_8 m_8) \\ 1, & -m_i \leq (j_i - v_8 m_8) < 0, \\ 2, & (j_i - v_8 m_8) < -m_i \end{cases} \quad i = 2, 3, \dots, 7;$$

si $v_8 = \pm 0.5$ alors

$$v_i = \begin{cases} -0.5, & \frac{m_i}{2} \leq (j_i - v_8 m_8) \\ 0.5, & -\frac{m_i}{2} \leq (j_i - v_8 m_8) < \frac{m_i}{2}, \\ 1.5, & (j_i - v_8 m_8) < -\frac{m_i}{2} \end{cases} \quad i = 2, 3, \dots, 7.$$

- $\mathbf{k} = \mathbf{j} + [\frac{1}{2}(m_1 v_1 - \sum_{i=2}^7 m_i v_i + 5m_8 v_8), m_2 v_2 - m_8 v_8, m_3 v_3 - m_8 v_8, m_4 v_4 - m_8 v_8, m_5 v_5 - m_8 v_8, m_6 v_6 - m_8 v_8, m_7 v_7 - m_8 v_8, 2m_8 v_8]$.

Puisque le vecteur \mathbf{v} appartient au réseau E_8 , la somme de composantes de \mathbf{v} est un multiple de 2. Donc, on a

$$\begin{aligned} k_1 &= j_1 + \frac{1}{2}(m_1 v_1 - \sum_{i=2}^7 m_i v_i + 5m_8 v_8) = j_1 + \frac{1}{2}(m_1 v_1 - \sum_{i=2}^7 m_i v_i + 5m_8 v_8) + \frac{1}{2} m_1 \sum_{i=1}^8 v_i \\ &= j_1 + \frac{1}{2}(\sum_{i=2}^8 (m_1 - m_i) v_i) + 3m_8 v_8 + m_1 v_1 \\ &= j_1 + \frac{1}{2}(\sum_{i=2}^8 (m_1 - m_i) v_i) + 3m_8 v_8 + w m_1 \end{aligned}$$

$$\text{avec } w = \begin{cases} 0, & v_8 = 0, 1 \\ 0.5, & v_8 = \pm 0.5 \end{cases}.$$

$$k_1 = j_1 + \frac{1}{2} \left(\sum_{i=2}^8 (m_1 - m_i) v_i \right) + 3m_8 v_8 + \omega m_1 \pmod{m_1};$$

$$k_i = j_i + v_i m_i - v_8 m_8 \pmod{m_i}, \quad i = 2, 3, \dots, 7;$$

$$k_8 = j_8 + 2v_8 m_8 = j_8 \pmod{m_8}.$$

BIBLIOGRAPHIE

- [1] T. Berger, "Rate Distortion Theory", Englewood Cliffs, NJ: Prentice-Hall, 1971.
- [2] A. Gersho and R. M. Gray, "Vector Quantization and Signal Compression", Norwell, MA: Kluwer, 1992.
- [3] J. H. Conway and N. J. A. Sloane, "Sphere Packings, Lattices and Groups", Springer-Verlag, New York, 1988.
- [4] J. D. Gibson and K. Sayood, "Lattice quantization", *Advances in Electronics and Electron Physics*, vol. 72, pp.259-330, 1988.
- [5] T. R. Fischer, "A pyramid vector quantizer", *IEEE Trans. Information Theory*, vol. 32, No. 4, pp. 568-583, July 1986.
- [6] J.-P. Adoul, C. Lamblin, and A. Leguyader, "Baseband speech coding at 2400 bps using spherical vector quantization", in *Proc. IEEE ICASSP-84*, pp.1.12.1-1.12.4, Mar. 1984.
- [7] J.-P. Adoul, "La quantification vectorielle des signaux: approche algébrique", *Annales des Télécommunications*, 41, n° 3-4, pp.158-177, 1986.
- [8] J. H. Conway and N. J. A. Sloane, "Fast quantizing and decoding algorithms for lattice quantizers and codes", *IEEE Trans. Information Theory*, vol. 28, No. 2, pp. 227-232, Mar. 1982.
- [9] D. J. Sakrison, "A geometric treatment of source encoding of a Gaussian random variable", *IEEE Trans. Information Theory*, vol. 14, No. 3, pp. 481-486, May 1968.
- [10] M. V. Eyuboglu and G. D. Forney, Jr., "Lattice and trellis quantization with lattice- and trellis-bounded codebooks — High-rate theory for memoryless sources",

- IEEE Trans. Information Theory*, vol. 39, No. 1, pp. 46-59, Jan. 1993.
- [11] R. Laroia and N. Farvardin, "Trellis-based scalar-vector quantizer for memoryless sources", *IEEE Trans. Information Theory*, vol. 40, No. 3, pp. 860-870, May 1994.
- [12] G. D. Forney, Jr., "Trellis shaping", *IEEE Trans. Information Theory*, vol. 38, No. 2, pp. 281-300, Mar. 1992.
- [13] J. N. Livingston, "Shaping using variable-size regions", *IEEE Trans. Information Theory*, vol. 38, No. 4, pp. 1347-1353, July 1992.
- [14] J. H. Conway and N. J. A. Sloane, "A fast encoding method for lattice codes and quantizers", *IEEE Trans. Information Theory*, vol. 29, No. 6, pp. 820-824, Nov. 1983.
- [15] T. R. Fischer, "Geometric source coding and vector quantization", *IEEE Trans. Information Theory*, vol. 35, No. 1, pp. 137-145, Jan. 1989.
- [16] N. S. Jayant and P. Noll, "Digital Coding of Waveforms: Principles and Applications to Speech and Video", Englewood Cliffs, NJ: Prentice-Hall, 1984.
- [17] M. Barlaud, P. Solé, J. M. Moureaux, M. Antonini, and P. Gauthier, "Elliptical codebook for lattice vector quantization", in *Proc. IEEE ICASSP-93*, pp. V.590-593, Apr. 1993.
- [18] J. M. Moureaux, M. Antonini, and M. Barlaud, "Counting lattice points on ellipsoids: Application to image coding", *Electron. Lett.*, vol. 31, No. 15, pp. 1224-1225, July 1995.
- [19] C. Lamblin and J.-P. Adoul, "Algorithme de quantification vectorielle sphérique à partir du réseau de Gosset d'ordre 8", *Annales des Télécommunications*, 43, n° 3-4, pp.172-186, 1988.
- [20] K. K. Paliwal and B. S. Atal, "Efficient vector quantization of LPC parameters at 24 bits/frame", *IEEE Trans. Speech and Audio Processing*, vol. 1, No. 1, pp. 3-14, Jan. 1993.

Bibliographie

- [21] L. R. Rabiner and R. W. Schafer, " Digital Processing of Speech Signals ", Englewood Cliffs, NJ: Prentice-Hall, 1978.
- [22] P. E. Papamichalis, " Practical Approaches to Speech Coding ", Englewood Cliffs, NJ: Prentice-Hall, 1987.
- [23] F. Itakura, " Line spectrum representation of linear predictive coefficients of speech signals ", *J. Acoust. Soc. Amer.*, vol. 57, p. s35(A), Apr. 1975.
- [24] F. K. Soong and B. H. Juang, " Line spectrum pair (LSP) and speech data compression ", in *Proc. IEEE ICASSP-84*, pp. 1.10.1-1.10.4, Mar. 1984.
- [25] G. S. Kang and L. J. Fransen, " Application of line-spectrum pairs to low-bit-rate speech encoders ", in *Proc. IEEE ICASSP-85*, pp. 244-247, Mar. 1985.
- [26] N. Sugamura and N. Farvardin, " Quantizer design in LSP speech analysis and synthesis ", in *Proc. IEEE ICASSP-88*, pp. 398-401, Apr. 1988.
- [27] R. Hagen and P. Hedelin, " Low bit-rate spectral coding in CELP, a new LSP-method ", in *Proc. IEEE ICASSP-90*, pp. 189-192, Apr. 1990.
- [28] R. Laroia, N. Phamdo, and N. Farvardin, " Robust and efficient quantization of speech LSP parameters using structured vector quantizers ", in *Proc. IEEE ICASSP-91*, pp. 641-644, May 1991.
- [29] F. K. Soong and B. H. Juang, " Optimal quantization of LSP parameters ", *IEEE Trans. Speech and Audio Processing*, vol. 1, No. 1, pp. 15-24, Jan. 1993.
- [30] E. Erzin and A. E. Çetin, " Interframe differential vector coding of line spectrum frequencies ", in *Proc. IEEE ICASSP-93*, pp. II.25-28, Apr. 1993.
- [31] N. S. Jayant, J. D. Johnston, and Y. Shoham, " Coding of wideband speech ", *Speech Communication*, vol. 11, No. 2-3, pp. 127-138, June 1992.
- [32] N. Moreau, " Techniques de Compression des Signaux ", Masson, Paris, 1995.
- [33] E. Zwicker and R. Feldtkeller, " Psychoacoustique: l'Oreille, Récepteur d'Information ", Masson, Paris, 1981.
- [34] P. Mermelstein, " G. 722, A new CCITT coding standard for digital transmission of

- wideband audio signals ", *IEEE Communications Magazine*, vol. 26, No. 1, pp. 8-15, Jan. 1988.
- [35] T. Nishitani, I. Kuroda, M. Satoh, T. Katoh, and Y. Aoki, " A CCITT standard 32 kbit/s ADPCM LSI codec ", *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 35, No. 2, pp. 219-225, Feb. 1987.
- [36] R. Lefebvre, R. Salami, C. Laflamme, and J.-P. Adoul, " High quality coding of wideband audio signals using transform coded excitation (TCX) ", in *Proc. ICASSP-94*, pp. I.193-I.196, May 1994.
- [37] M. R. Schroeder and B. S. Atal, " Coded-excited linear prediction (CELP): high-quality speech at very low bit rates ", in *Proc. IEEE ICASSP-85*, pp. 937-940, Mar. 1985.
- [38] R. Lefebvre, " Nouveau modèle de codage audio à prédiction linéaire et transformée orthogonale ", Thèse de doctorat, Université de Sherbrooke, Sherbrooke, Canada, Juin 1995.
- [39] P. Noll, " Wideband speech and audio coding ", *IEEE Communications Magazine*, vol. 31, No. 11, pp. 34-44, Nov. 1993.
- [40] J.-P. Adoul, " Lattice and Trellis Coded Quantizations for Efficient Coding of Speech ", Chapter 57 in " *Speech Recognition and Coding: New Advances and Trends* ", Edited by A. J. R. Ayuso and J. M. L. Soler, Series F, vol. 147, Springer, 1995.
- [41] J. P. M. Schalkwijk, " An algorithm for source coding ", *IEEE Trans. Information Theory*, vol. 18, No. 3, pp. 395-399, May 1972.
- [42] C. E. Shannon, " A mathematical theory of communication ", *Bell Syst. Tech. J.*, vol. 27, pp. 379-423, July 1948.
- [43] D. A. Huffman, " A method for the construction of minimum-redundancy codes ", in *Proc. IRE*, vol. 40, pp. 1098-1011, Sept. 1952.
- [44] J. J. Rissanen, " Generalized Kraft inequality and arithmetic coding ", *IBM J. Res.*

- Develop.*, vol. 20, No. 3, pp. 198-203, May 1976.
- [45] J. J. Rissanen and G. G. Langdon, "Arithmetic coding", *IBM J. Res. Develop.*, vol. 23, No. 2, pp. 149-162, Mar. 1979.
- [46] I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic coding for data compression", *Commun. ACM*, vol. 30, No. 6, pp. 520-540, June 1987.
- [47] P. G. Howard and J. S. Vitter, "Arithmetic coding for data compression", *Proceedings of the IEEE*, vol. 82, No. 6, pp. 857-865, June 1994.
- [48] T. V. Ramabadran and D. L. Cohn, "An adaptive algorithm for the compression of computer data", *IEEE Trans. Communications*, vol. 37, No. 4, pp. 317-324, Apr. 1989.
- [49] A. Moffat, "Implementing the PPM data compression scheme", *IEEE Trans. Communications*, vol. 38, No. 11, pp. 1917-1921, Nov. 1990.
- [50] P. G. Howard and J. S. Vitter, "Parallel lossless image compression using Huffman and arithmetic coding", in *Proc. Data Compression Conference*, pp. 299-308, Mar. 1992.
- [51] P. G. Howard and J. S. Vitter, "Fast and efficient lossless image compression", in *Proc. Data Compression Conference*, pp. 351-360, Mar. 1993.
- [52] J.-P. Adoul and R. Lefebvre, "Wideband Speech Coding", Chapter 8 in "*Speech Coding and Synthesis*", W. B. Kleijn and K. K. Paliwal Editors, ELSEVIER, Amsterdam, 1995.
- [53] M. Xie and J.-P. Adoul, "Fast and low-complexity LSF quantization using algebraic vector quantizer", in *Proc. ICASSP-95*, pp. 716-719, May 1995.
- [54] M. Xie and J.-P. Adoul, "Algebraic vector quantization of LSF parameters with low storage and computational complexity", *IEEE Trans. Speech and Audio Processing*, vol. 4, No. 3, May 1996.
- [55] M. Xie and J.-P. Adoul, "Embedded algebraic vector quantizers (EAVQ) with application to wideband speech coding", in *Proc. ICASSP-96*, May 1996.