

UNIVERSITÉ DE SHERBROOKE

Faculté des sciences appliquées

Département de génie électrique et informatique

**LA CONCEPTION D'UN QUANTIFICATEUR VECTORIEL
ROBUSTE
AUX ERREURS DU CANAL**

**Mémoire de maîtrise ès sciences appliquées
Spécialité : Génie électrique**

Omar AZZAOU

Sherbrooke (Québec), Canada

Septembre 2000

Sommaire

Nous vivons dans l'ère des télécommunications. Les radio-mobiles, le multimédia et les communications globales font déjà partie de notre langage quotidien. La société devient de plus en plus complexe et les besoins de stockage et de communication s'accroissent rapidement. La course à la compression des données est motivée par la théorie du codage de source de Shannon. Lorsque l'information de source est comprimée et codée de façon très efficace, et qu'elle est transmise sur un canal bruité, les performances prévues par le système de codage de source sont très rapidement dégradées. C'est Shannon lui-même qui vient au secours, dans ce cas, en assurant l'existence de schémas de codage de canal capables d'atteindre le modèle de 'canal sans bruit'. Ce résultat fondamental de la théorie de l'information a permis de traiter le problème de codages de source et de canal de façon complètement séparée, sans aucune perte d'optimalité par rapport à un système globalement optimal.

Le travail réalisé au cours de ce projet est motivé par l'augmentation des besoins en codage bas-débit des systèmes basés sur la quantification vectorielle et utilisant des canaux bruités, comme les communications radio-mobiles de la parole. Dans ce mémoire de maîtrise, on présente ainsi les techniques visant à combattre efficacement la dégradation de la performance causée par les canaux bruités.

Remerciements

Ce mémoire de maîtrise est le résultat de deux années de travail au sein du groupe du codage de la parole du Département de Génie Électrique de l'Université de Sherbrooke. Pendant ces deux années, j'ai eu la chance de connaître des personnes qui ont, d'une façon ou d'une autre, contribué à la réalisation de ce travail.

J'adresse dans un premier temps tous mes remerciements à mon directeur de recherche, le Professeur Roch Lefebvre, pour lequel j'exprime le plus profond respect pour ces qualités scientifiques, pour sa grande disponibilité, pour ses encouragements et pour ses précieux conseils tout au long de ce mémoire. Merci Roch!

Je tiens à remercier M. Stéphane Ragot, étudiant en doctorat au sein du groupe de codage de la parole, avec lequel j'ai partagé les bons et les mauvais moments et qui mérite toute ma reconnaissance.

Une place très particulière est réservée à M. Vesa T. Ruoppila, chercheur au sein du groupe de codage, qui m'a appris à améliorer ma méthode scientifique.

Je tiens aussi à remercier M. Jaouad Abdellatif, étudiant en doctorat en microélectronique et M. Hassan Lahdili, étudiant en maîtrise dans le même groupe et mon collègue de travail, pour leur patience, leurs conseils et leur disponibilité pour m'écouter.

Je tiens aussi à remercier M. Milan JELILEK, chercheur au sein du groupe, pour sa disponibilité et sa gentillesse. Mes remerciements vont également à tous mes amis de l'équipe de codage. Merci à tous!

Finalement, je dédie ce travail à la mémoire de ma mère et j'exprime ma plus profonde reconnaissance à mon père qui a terriblement souffert, au-delà de ses capacités physiques

et morales, pour que je continue mes études ainsi que tous les membres de ma famille. Je vous aime beaucoup!

Table des matières

SOMMAIRE	II
REMERCIEMENTS	III
INTRODUCTION	1
CHAPITRE 1 : CODAGE DE LA SOURCE	4
1.1 COMPRESSION SANS PERTE	4
1.2 COMPRESSION AVEC PERTE	5
1.2.1 <i>La quantification vectorielle</i>	6
1.2.1.1 <i>Le quantificateur vectoriel</i>	7
1.2.1.2 <i>Les domaines d'application</i>	10
1.2.2 <i>Distorsion et débit</i>	11
1.3 MODÉLISATION D'UN SIGNAL ET PRÉDICTION LINÉAIRE	13
1.3.1 <i>L'autocorrélation</i>	15
1.3.2 <i>Transformation des coefficients de prédiction : Les LSF (Line Spectrum Frequencies)</i>	16
CHAPITRE 2 : CODAGE DU CANAL	19
2.1 LA DISTORSION DU CANAL	19
2.2 CODAGE DU CANAL AVEC REDONDANCE	20
2.2.1 <i>Les codes linéaires</i>	21
2.3 CODAGE DU CANAL SANS REDONDANCE	22
2.4 CANAL DE TRANSMISSION	23
2.4.1 <i>Canal binaire symétrique</i>	25
2.4.2 <i>La capacité du canal binaire symétrique</i>	26
2.5 CLASSIFICATION DES CODEURS SOURCE-CANAL	28
CHAPITRE 3 : ATTRIBUTION DES INDICES	30
3.1 HISTORIQUE	30
3.2 PROBLÉMATIQUE	30
3.3 LA DISTORSION TOTALE	32
3.4 UN ALGORITHME D'ATTRIBUTION D'INDICES : BINARY SWITCHING	33
3.4.1 <i>La distance de Hamming</i>	36
3.4.2 <i>La description de l'algorithme</i>	38
3.4.3 <i>La complexité de l'algorithme</i>	41
3.4.4 <i>La performance de l'algorithme</i>	42
3.5 CLASSIFICATION PAR PARTITION (CLUSTERING)	45
3.5.1 <i>La description de l'algorithme</i>	47
3.6 L'APPROCHE ADOPTÉE : EXPLOITATION DES 2 ALGORITHMES	49
CHAPITRE 4 : DISCUSSION ET RÉSULTATS	51
4.1 LA DISTORSION SPECTRALE.....	52

<u>4.2 RÉSULTATS</u>	53
<u>4.2.1 Quantificateur vectoriel prédictif à 24 bits par trame</u>	53
<u>4.2.2 Quantificateur vectoriel à 35 bits par trame</u>	54
<u>CONCLUSION</u>	57
<u>Liste des figures</u>	59
<u>Liste des tableaux</u>	60
<u>Bibliographies</u>	61

Introduction

Les communications numériques constituent sans aucun doute l'une des plus importantes révolutions de la science et de la technologie à l'aube de ce millénaire. Propulsé par la théorie de l'information, fondée par C. E. Shannon en 1948, le monde des télécommunications commence lentement à basculer vers le numérique. Les techniques numériques exigent des taux de compression de l'information de plus en plus grands.

La téléphonie numérique repose sur les techniques de compression et de codage de la parole. Des progrès très significatifs ont été faits dans ce domaine. Depuis les premiers schémas à 64 kbits/s, jusqu'aux nouvelles techniques à 2.4kbits/s, le traitement de la parole est devenu un domaine **indispensable** dans la conception et la mise en œuvre d'un système. La production à grande échelle des systèmes radio-mobiles a motivé l'étude de schémas de codage de la parole de plus en plus performants. Les débits de transmission doivent être réduits, tout en conservant la qualité de communication.

Il est clair que la demande de communications mobiles augmente d'une façon exponentielle. En plus, la parole à bande étroite et l'audio sont parmi les services offerts par des systèmes de communication mobiles. Bref, il y a un besoin clair de techniques qui mènent à augmenter l'efficacité et la robustesse dans les communications mobiles. Cela a motivé l'utilisation d'une technique très puissante de codage de source : la quantification vectorielle qui est utilisée fréquemment dans les systèmes actuels.

Avec la transmission des indices du quantificateur vectoriel sur un canal radio, les aspects de robustesse du canal deviennent de plus en plus importants. Dans ce mémoire, on cherche une conception d'un quantificateur vectoriel robuste aux erreurs du canal.

Le but de cette étude consiste à voir comment on peut intégrer au processus d'optimisation du codeur de la source, les effets du bruit du canal. Évidemment, le

problème peut être posé du point de vue du canal : comment construire des codes de canal pour que la reconstruction de la source souffre le moins possible des effets du canal? .

Ce mémoire traite des deux problèmes majeurs de la conception d'un quantificateur vectoriel. Ces derniers consistent à distribuer les vecteurs de reproduction pour couvrir l'espace de signal et à choisir les mots de code ou indices pour que l'effet des erreurs de canal sur les performances soit minimisé.

Organisation du document

Tel que représenté dans la chaîne de communication de la figure 1, le plan du mémoire va suivre les étapes suivantes :

Dans le premier chapitre, nous décrivons des techniques de codage de la source : la quantification vectorielle, la modélisation du signal et le codage prédictif.

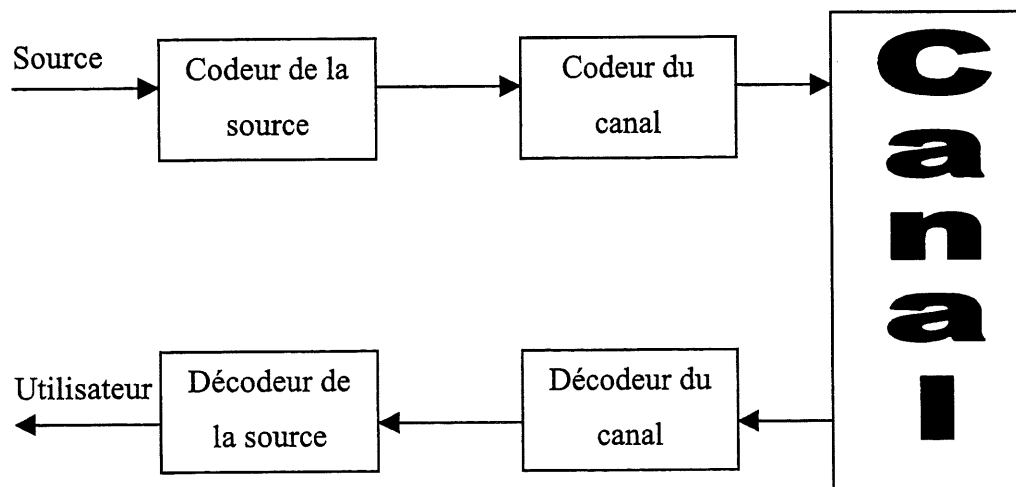


Figure 1.1. Système simple de communication

Le second chapitre porte sur le codage du canal et le milieu de transmission. Le codage du canal est décrit avec et sans redondance. Par la suite, on modélise un canal de transmission avec un exemple simplifié : le canal discret binaire symétrique (BSC).

Le troisième chapitre reprend les techniques de codage du canal. Dans un premier temps, on pose la problématique du codage sans redondance, ainsi que son historique. Dans un deuxième temps, on décrit en détail l'algorithme qui est développé dans cette étude, avec ses performances en présence d'une source gaussienne. Enfin, on trouve une description d'un algorithme de classification.

Le dernier chapitre exploite les algorithmes développés dans le chapitre précédent en les appliquant sur les LSF (Line Spectrum Frequencies). La mesure de la performance est calculée en fonction de la distorsion spectrale.

Chapitre 1 : CODAGE DE LA SOURCE

La compression de l'information, ou le codage de la source est la science (ou l'art) de représenter l'information d'une façon « compacte », en identifiant et en utilisant les structures possibles qui existent dans les données. Ces données peuvent être les caractères d'un texte, les échantillons d'une forme d'onde de la parole ou d'une image ou bien des suites de nombres générés par un autre processus.

Les structures présentes dans les sources sont essentiellement de nature statistique. Dans le cas d'un signal de la parole, les signaux émis sont le résultat de l'excitation de notre conduit vocal. Celui-ci vérifie certaines propriétés physiques et il peut être modélisé d'une façon relativement simple.

Mais la structure dans l'information à coder n'est pas la seule chose que l'on puisse exploiter pour obtenir une certaine compression. On peut par exemple, se servir des caractéristiques de l'utilisateur de l'information. Dans le cas d'une transmission de la parole, puisque l'utilisateur final sera un humain, on peut exploiter certaines limites de l'audition, en particulier les effets de masquage.

1.1 Compression sans perte

Tel que l'indique leur nom, les schémas sans perte ou sans distorsion sont des schémas où il n'y a pas de perte d'informations (figure 1.1). C'est le cas typique des techniques utilisées pour la compression de texte écrit ou de l'information générée par ordinateur. Les données peuvent être reconstruites exactement à partir de leur version comprimée. On trouve ce genre de compression dans l'imagerie médicale dans le but d'obtenir plus d'informations ou bien faire ressortir une certaine propriété du signal.

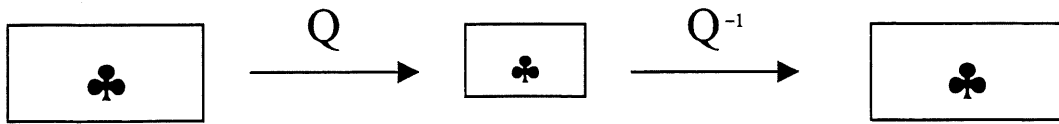


Figure 1.1 : Exemple de compression sans perte

Q est une fonction de quantification de la source

Q^{-1} est la fonction inverse de Q

1.2 Compression avec perte

Dans certaines situations, la contrainte de reconstruction parfaite peut être supprimée. On accepte une certaine distorsion entre l'original et le reconstruit dans le but d'obtenir un taux de compression plus élevé. Comme l'indique la figure 1-2 suivante :

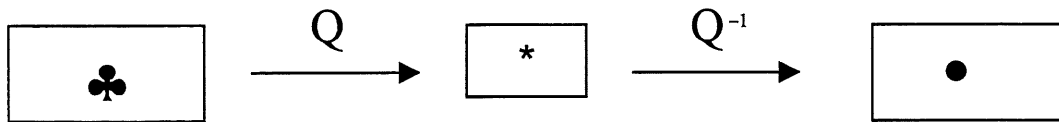


Figure 1.2 : Exemple de compression avec perte

Un exemple d'application de ce type de technique est le codage de la parole. Dans ce cas, les échantillons du signal de parole reconstruits ne sont pas nécessairement les mêmes que ceux du signal émis. On peut tolérer un certain degré de distorsion du moment que la synthèse est assez proche de l'original. Nous présentons maintenant une technique de compression avec perte très répandue en codage de la parole : la quantification vectorielle[5] et [6].

1.2.1 La quantification vectorielle

La quantification vectorielle est un moyen puissant de codage pour les applications de la communication numérique. Elle est largement utilisée pour le codage de la parole, de l'image et de la vidéo.

La procédure de codage consiste à décomposer le signal source x en blocs de N échantillons puis à décider à quelle région appartient le bloc de signal source en lui associant le numéro $i \in \{1 \dots L = 2^b\}$ de la région correspondante. C'est le numéro de la région choisie, qui sera transmis ou stocké. La procédure de décodage consiste à associer au numéro i le représentant correspondant (figure 1.3).

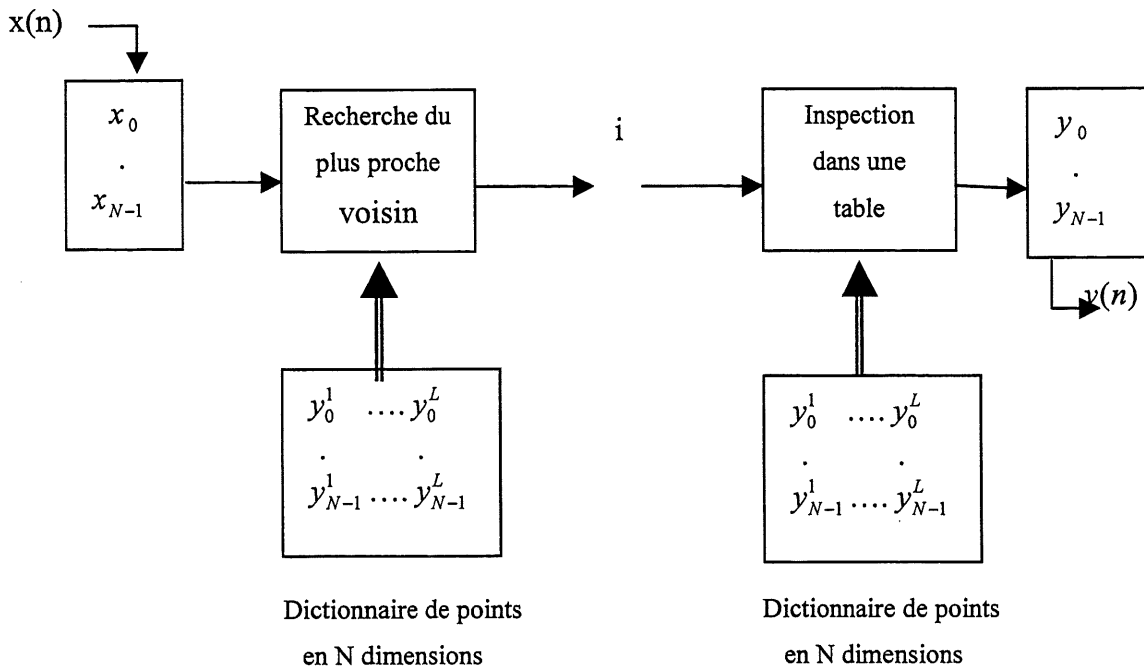


Figure 1.3 : Quantificateur vectoriel

On appelle ' région de voronoï S_i ' la région de l'espace ou tout point x en N dimensions est plus près du point ' i ' du dictionnaire que de tout autre point du dictionnaire.

Tout point de source contenu dans la région de voronoï sera représenté par le vecteur y_i :

$$x \longrightarrow y_i \quad \text{Si et seulement } x \in S_i \text{ avec } i=0..L-1 \quad (1.1)$$

1.2.1.1 Le quantificateur vectoriel

Un quantificateur Q est défini par un dictionnaire $C=\{y_1, y_2, \dots, y_L\}$ et une partition $S=\{S_0, S_1, \dots, S_L\}$ de l'espace du signal.

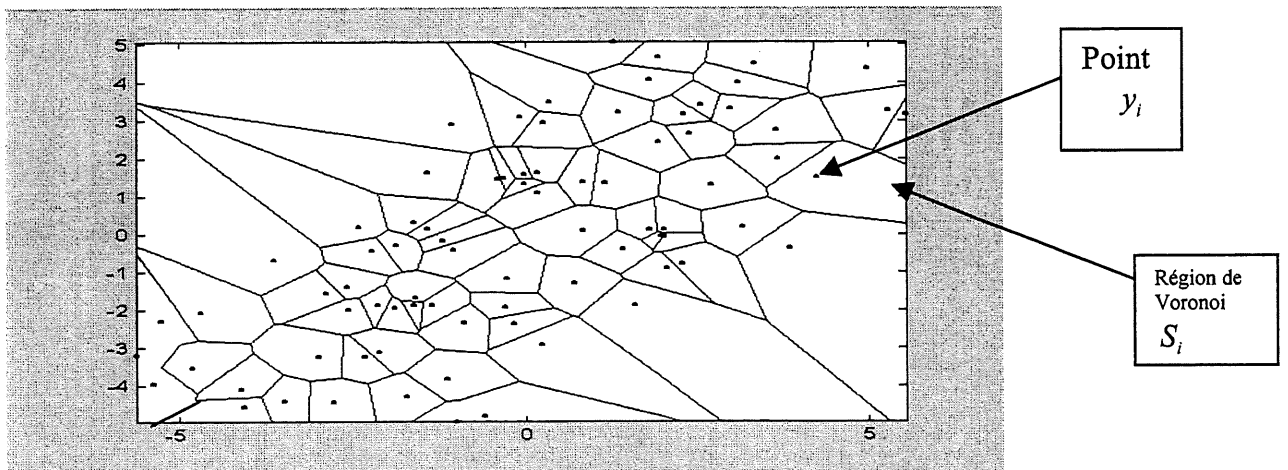


Figure 1.4 : Régions de voronoï d'un dictionnaire à 64 points en 2 dimensions

Un quantificateur optimal est celui qui minimise l'espérance d'erreur entre les vecteurs source x et leur valeur reconstruite $y_i \in \{y_0, y_1, \dots, y_{L-1}\}$ (1.2) :

$$D = E\{x - y_i\}^2 \quad (1.2)$$

Ou alternativement, qui maximise le rapport signal sur bruit :

$$Rsb = \frac{E\{x^2\}}{E\{x - y_i\}^2} \quad (1.3)$$

Ainsi, l'optimalité d'un quantificateur est liée à la position des points y_i dans l'espace de \mathfrak{R}^N .

Pour définir le dictionnaire C, i.e, pour construire les vecteurs de reproduction, on est amené à utiliser une base d'apprentissage. Celle-ci doit être composée d'un grand nombre M de vecteurs représentatifs de la source. Typiquement, on estime que chaque vecteur du dictionnaire doit être construit à partir d'une centaine de vecteurs de la base [7].

Le problème d'optimisation d'un quantificateur scalaire a été traité par Max et Lloyd [26] [27]. Les conditions d'optimalité ont été largement analysées et peuvent être résumées en deux conditions :

- Condition de centroïde
- Condition du voisin le plus proche

La condition du centroïde : on calcule le centroïde optimum pour une classe i donnée.

La condition du voisin du plus proche : on calcule toutes les distances $d(x, y_i)$, et on associe x au centroïde le plus proche.

L'algorithme LBG_VQ [7] est la généralisation de l'algorithme Lloyd - Max à N dimensions. Il est devenu un outil fondamental pour la conception d'un quantificateur vectoriel. L'organigramme de cet algorithme est donné par la suite :

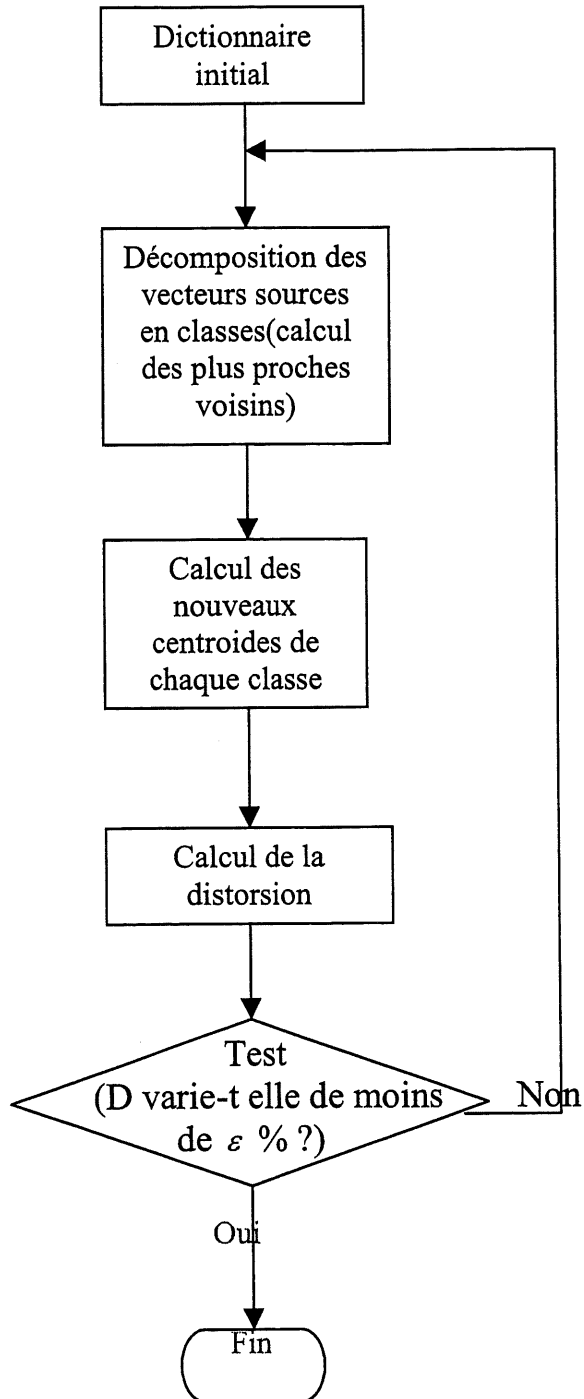
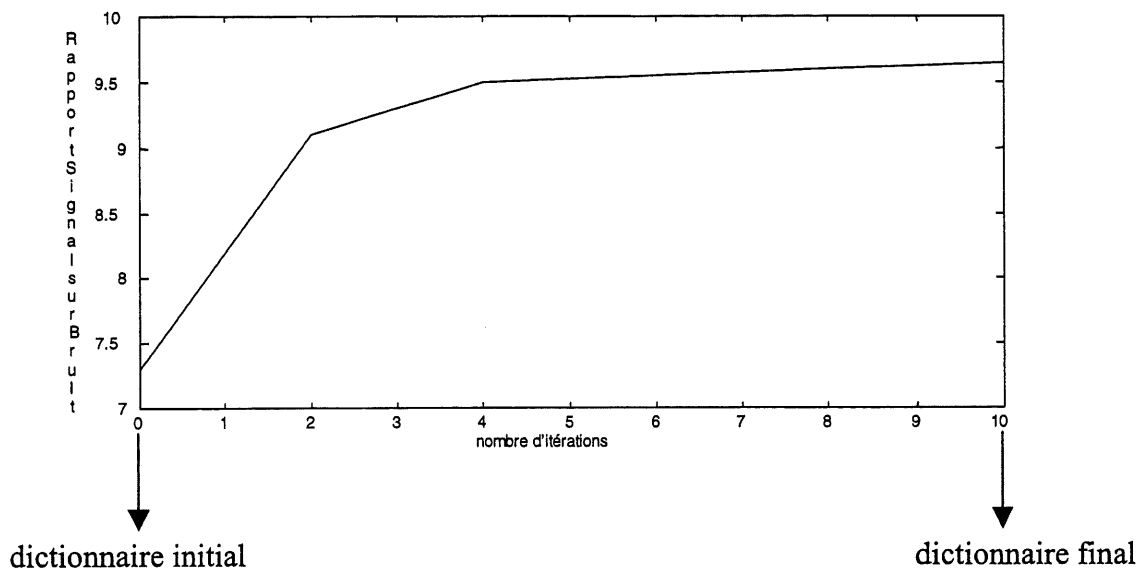


Figure 1.5 : Organigramme de fonctionnement de l'algorithme LBG-VQ

L'algorithme LBG-VQ revient à une amélioration itérative d'un dictionnaire initial qui doit être choisi. À chaque itération, on cherche de nouveaux représentants du dictionnaire qui optimisent le rapport signal sur bruit. D'après le comportement de la figure (1.6), l'algorithme converge déjà après quelques itérations.



*Figure 1.6 : Convergence de l'algorithme LBG
 $b=2$ bit/échantillons ; $N=2$*

Avec : b est le nombre de bits attribués pour chaque échantillon.

N est la dimension de chaque vecteur

1.2.1.2 Les domaines d'application

On peut diviser les applications de ce type de codage en deux grandes classes :

- Codage de la parole,
- Codage des images.

1. Codage de la parole

Le domaine de la parole a des applications très importantes dans le monde des télécoms. Ayant pris la relève de l'analogique, les techniques numériques exigent des taux de compression de plus en plus grands, dans le but d'optimiser la ressource canal. En général, c'est parmi les faibles débits qu'on trouve la plus vaste majorité de ces codeurs.

2. Codage des images

Le domaine des images exige des quantités d'informations assez considérables à gérer. Par conséquent, les codeurs pour ces applications sont du type à débit moyen (très haut dans certaines applications comme la HDTV).

1.2.2 Distorsion et débit

Le but poursuivi dans l'élaboration d'un système de quantification vectorielle est de minimiser la distorsion moyenne du quantificateur pour un nombre donné de classes de quantification lié au débit de transmission, ou inversement de minimiser le nombre de classes pour une distorsion moyenne fixée.

Comme déjà vu à la section (1-2-1), l'opération du codage consiste à désigner à un vecteur d'entrée x , un vecteur dans le quantificateur vectoriel de même dimension choisi parmi une collection finie $C = \{y_1, y_2, \dots, y_L\}$

La distorsion résultante de cette quantification est donnée par la mesure de la différence entre le vecteur d'entrée et celui quantifié (1.2).

D'autre part, la performance d'un quantificateur vectoriel q est mesurée par la distorsion de la quantification moyenne par échantillon $D_s(q)$:

$$D_s(q) = \frac{1}{k} \sum_{i=1}^L \int_{s_i} p(x) d(x, y_i) dx \quad (1.4)$$

Où $p(x)$ est la densité de probabilité de la source.

Le quantificateur vectoriel q^* est optimum si :

$$D_s(q^*) \leq D_s(q) \quad \text{pour tout } q \quad (1.5)$$

Tout comme la distorsion du quantificateur, le taux de distorsion joue un rôle fondamental en ce qui concerne un schéma de codage.

Si on a un dictionnaire C de taille L et que ses vecteurs de reproduction sont de dimension N , l'indice i qui sera transmis sur le canal de transmission nécessite au moins $n = \log_2 L$ bits/échantillon. Donc, pour chaque vecteur de reproduction de dimension N , on associe un indice binaire de n bits, d'où le débit numérique d'un quantificateur est :

$$R = \frac{1}{N} (\log_2 L) \text{ bits/échantillon} \quad (1.6)$$

Le débit d'un quantificateur dépend de la taille de son dictionnaire C et de la dimension de ses vecteurs de reproduction. Par contre, la distorsion du quantificateur dépend évidemment de la position des vecteurs de reproduction dans l'espace. Cette distorsion sera d'autant plus faible que le débit R sera plus élevé, ce qui implique un plus grand nombre de vecteurs de reproduction qui couvrent de mieux en mieux la région pertinente de l'espace du signal à quantifier. La distance entre vecteurs sera plus faible et la distorsion plus réduite.

1.3 Modélisation d'un signal et prédiction linéaire

La modélisation d'un signal $x(n)$ consiste à lui associer un filtre linéaire qui, soumis à une excitation particulière, reproduit ce signal le plus fidèlement possible. L'objectif essentiel de la modélisation d'un signal est de permettre la description de son spectre par un ensemble très limité de paramètres. Le modèle le plus utilisé est le modèle AR(Auto-Regressive). Ses capacités de modélisation (aptitudes à approcher le spectre du signal) sont suffisantes pour beaucoup d'applications. Ainsi que l'on verra dans le paragraphe qui suit, l'estimation du modèle AR qui est basée sur la prédiction linéaire.

En examinant les spectres de puissance à court terme du signal de parole, on peut distinguer quelques résonances fortes. Ces résonances sont reliées à la forme du conduit vocal, qui varie en fonction du temps. L'enveloppe spectrale correspond à la corrélation à court terme entre et la structure des harmoniques correspond à la corrélation à long terme. La corrélation à court terme entre les échantillons adjacents du signal peut être exploitée par un filtre prédictif linéaire variant dans le temps (figure 1.7). Le résultat du filtrage du signal de parole $x(n)$ avec le filtre d'analyse de la prédiction linéaire est appelé le signal résiduel. Si le signal résiduel $r(n)$ est l'entrée de l'inverse du filtre d'analyse c'est-à-dire le filtre de synthèse (figure 1.8), le signal original est récupéré.

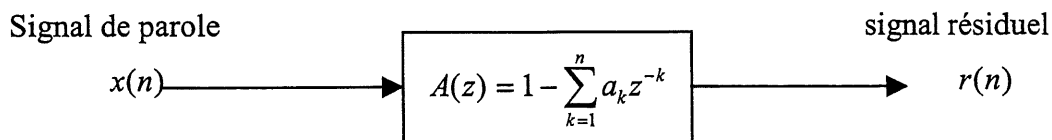


Figure 1.7 : Le filtre d'analyse de prédiction linéaire

Les coefficients linéaires $\{a_k\}$, $k = 1, 2, \dots, N$ sont appelés coefficients de prédiction. Lors de la quantification des a_k , on doit s'assurer de la stabilité du filtre de synthèse (figure 1.8). Autrement dit, les pôles du filtre de synthèse quantifié doivent être dans le cercle unité du plan Z. Cette garantie de stabilité ne peut pas être réalisée de façon générale en quantifiant directement les coefficients a_k . Ainsi, on doit transformer les coefficients à d'autres ensembles de paramètres.

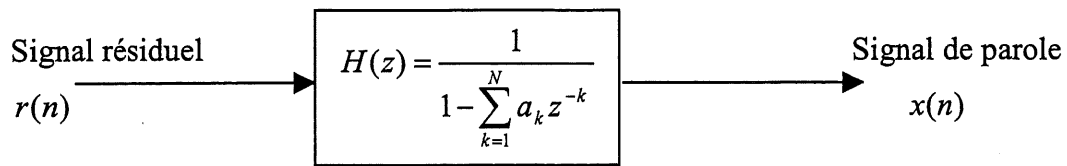


Figure 1.8 : Le filtre de synthèse de la prédiction linéaire

D'après la figure (1.7) ou la figure (1.8), on peut exprimer le rapport entre $x(n)$ et $r(n)$ par l'équation aux différences :

$$r(n) = x(n) - \sum_{k=1}^N a_k x(n-k) \quad (1.7)$$

$$x(n) = \sum_{k=1}^N a_k x(n-k) + r(n) \quad (1.8)$$

Il y a deux approches pour le calcul des coefficients de prédiction LP, la méthode d'autocorrélation et la méthode de covariance. Ces deux approches utilisent la technique des moindres-carrés et choisissent les a_k en réduisant au minimum l'énergie moyenne du signal résiduel. La méthode de l'autocorrélation sera discutée.

1.3.1 L'autocorrélation

Un bloc de n échantillons du signal de parole $x(n)$ est d'abord multiplié par une fenêtre d'analyse $w(n)$ de longueur L_w pour obtenir un segment de parole fenêtré [34].

$$x_w(n) = w(n)x(n) \quad (1.9)$$

On utilise une fenêtre pour réduire les effets de bord (ondulations secondaires dans le spectre). La fenêtre utilisée est typiquement une fenêtre de Hamming définie par :

$$w(n) = \begin{cases} 0.54 - 0.46 \cos\left(\frac{2\pi n}{L_w - 1}\right), & \text{pour } 0 \leq n < L_w \\ 0 & \text{sinon} \end{cases} \quad (1.10)$$

Les coefficients de prédiction a_k optimaux minimisent :

$$E = \sum_{n=-\infty}^{\infty} r^2(n) = \sum_{n=-\infty}^{\infty} \left[x_w(n) - \sum_{k=1}^N a_k x_w(n-k) \right]^2 \quad (1.11)$$

Ce qui s'obtient en posant :

$$\frac{\partial E}{\partial a_k} = 0 \quad \text{pour } k = 1, 2, \dots, M \quad (1.12)$$

On obtient ainsi le système d'équations linéaires suivant:

$$\sum_{n=-\infty}^{\infty} x_w(n)x_w(n-i) = \sum_{k=1}^N a_k \sum_{n=-\infty}^{\infty} x_w(n-i)x_w(n-k) \quad \text{pour } i = 1, 2, \dots, N \quad (1.13)$$

La fonction d'autocorrélation fenêtrée du signal [34] est définie par :

$$R(i) = \sum_{-\infty}^{\infty} x(n)x(n-i) = \sum_{n=i}^{L_w-1} x_w(n)x_w(n-i) \quad (1.14)$$

Le système d'équations linéaires peut s'exprimer sous la forme suivante :

$$\begin{bmatrix} R(0) & R(1) & \dots & R(N-1) \\ R(1) & R(0) & \dots & R(N-2) \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ R(N-1) & R(N-2) & \dots & R(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \cdot \\ \cdot \\ \cdot \\ a_N \end{bmatrix} = \begin{bmatrix} R(1) \\ R(2) \\ \cdot \\ \cdot \\ \cdot \\ R(N) \end{bmatrix} \quad (1.15)$$

Puisque la matrice (1.15) a une structure de Toeplitz, les coefficients a_k peuvent être déterminés efficacement par la récursion de Levinson Durbin [3]. En outre, la matrice de Toeplitz garantit que les pôles résultants du filtre de synthèse LP restent à l'intérieur du cercle unité. La stabilité du filtre est ainsi toujours assurée.

1.3.2 Transformation des coefficients de prédiction : Les LSF (Line Spectrum Frequencies)

Dans la recherche d'une meilleure qualité de la parole synthétisée, plusieurs transformations des coefficients de la prédiction linéaires LPC ont été introduites. Parmi ces transformations les "Line Spectrum Pairs" (LSP) [1] ou les "Line Spectrum Frequencies" (LSF) [2]. D'abord, on a:

$$A(z) = 1 - \sum_{k=1}^N a_k z^{-k} \quad (1.16)$$

À partir de l'équation du filtre d'analyse $A(z)$, on construit deux nouveaux polynômes d'ordre N :

$$P(z) = A(z) + z^{-(N+1)} A(z^{-1}) \quad (1.17)$$

$$Q(z) = A(z) - z^{-(N+1)} A(z^{-1}) \quad (1.18)$$

Ces deux polynômes $P(z)$ et $Q(z)$ sont respectivement antisymétrique et symétrique.

Dans le cas où N (ordre du prédicteur) est pair, on peut écrire les polynômes (1.17) et (1.18) sous la forme :

$$P(z) = (1 + z^{-1}) \prod_{i=2,4,\dots,N} (1 - 2 \cos(w_i)) z^{-1} + z^{-2} \quad (1.19)$$

$$Q(z) = (1 - z^{-1}) \prod_{i=1,3,\dots,N-1} (1 - 2 \cos(w_i)) z^{-1} + z^{-2} \quad (1.20)$$

Les paramètres w_i , $i = 1, \dots, N$ sont les LSF.

Les LSF vérifient les relations d'ordre:

$$0 = w_0 < w_1 < \dots < w_N < w_{N+1} = \pi \quad (1.21)$$

Les fréquences w_0 et w_{N+1} correspondent aux racines $z = -1$ et $z = 1$ des polynômes $P(z)$ et $Q(z)$ respectivement. On a donc n degrés de liberté, i.e. n LSF w_k représentant n coefficients de prédiction a_k .

Les deux polynômes (1.19) et (1.20) ont les propriétés suivantes :

- Toutes les racines de $P(z)$ et $Q(z)$ sont sur le cercle unité.
- Les racines de $P(z)$ et $Q(z)$ apparaissent de façon alternée sur le cercle unité
- La propriété de phase minimum de $A(z)$ est conservée après la quantification des racines de $P(z)$ et $Q(z)$

Les avantages des LSF pour la quantification vectorielle sont nombreux :

- Les LSF des trames adjacentes sont fortement corrélées. Ceci peut être exploité pour réduire le débit en utilisant la quantification prédictive.
- Les LSF ont un bon comportement des propriétés statistiques et la stabilité du filtre de synthèse est assurée par une conservation de la propriété d'ordre. De plus, elles peuvent être utilisées efficacement pour détecter les erreurs de transmission dans les LSF.
- Certaines études ont montré que la concentration des LSF dans certaines bandes de fréquences correspond approximativement à la résonance dans cette bande.

Chapitre 2 : CODAGE DU CANAL

Le potentiel de développement des systèmes de communication sans fil est actuellement considéré comme extrêmement important. Outre la téléphonie mobile dont l'expansion est considérable, les liaisons sans fil seront utilisées dans un avenir proche dans des contextes aussi variés que la diffusion hertzienne, les transmissions de données intra- bâtiments, etc... D'une façon générale, les systèmes sans fil se caractérisent par des besoins accrus en capacité. Ces nouvelles exigences engendrent des problèmes difficiles qui ont relancé l'intérêt des recherches concernant les techniques de transmission par voie hertzienne, et en ont renouvelé les problématiques. À titre d'exemple, citons la mobilité, qui nécessite des canaux de transmission variant très rapidement au cours du temps et où les taux d'erreurs peuvent devenir importants.

Les domaines scientifiques les plus concernés par ces problèmes sont ceux des communications numériques (codage de canal, modulation/ démodulation) et, du traitement du signal appliqué aux transmissions.

2.1 La distorsion du canal

En cours de transmission à travers un canal réel, des perturbations s'ajoutent au signal utile. Ces perturbations constituent le bruit du canal. D'où généralement un message reçu n'est pas le message expédié. Parmi les causes perturbatrices dans un système : l'atténuation, la diaphonie, les influences électromagnétiques...

Dans le cas de la sortie d'un quantificateur vectoriel (figure 2.1), les indices à transmettre $b(y_i)$; $i=1,2,\dots,M$ sont représentés sur n bits. On se réfère à $b(y_i)$ comme l'indice associé à y_i .

$p(j/i)$ est la probabilité que l'indice j soit reçu, sachant que c'est l'indice i qui a été transmis.

La distorsion moyenne par échantillon causée par le bruit du canal est donnée par :

$$D_c(b) = \frac{1}{k} \sum_{i=1}^M \sum_{j=1}^M P(y_i) P(b(y_j) / b(y_i)) d(y_i, y_j) \quad (2.1)$$

avec $P(y_i)$ est la probabilité à priori pour que le mot de code i soit transmis

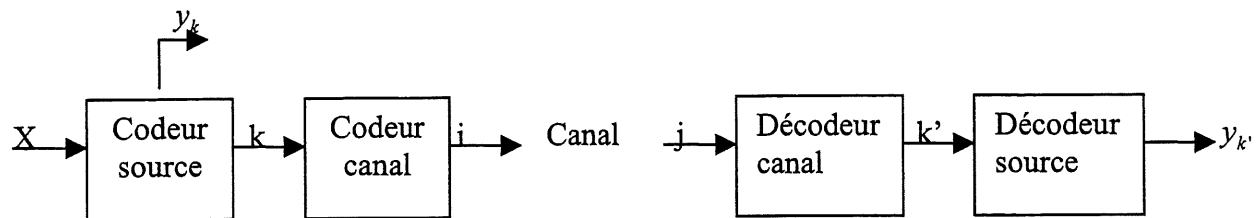


Figure 2-1 : Codage de source et du canal en présence du bruit

2.2 codage du canal avec redondance

Après avoir représenté le vecteur source par k bits dans les schémas des codeurs classiques de la source et du canal (figure 2-2), le code correcteur d'erreurs ajoute $n-k$ bits comme une redondance dans la trame à transmettre. Cette redondance ajoutée permet de protéger les mots binaires contre les erreurs du canal, détecter les erreurs du canal et les corriger sans retransmission. Dans la littérature, il y a plusieurs types de code canal (CRC, BCH, codes de convolution, codes de Hamming, codes de Hadamard, codes de Reed-Solomon.....). Pratiquement tous les systèmes de codage du canal utilisent cette méthode pour la protection.

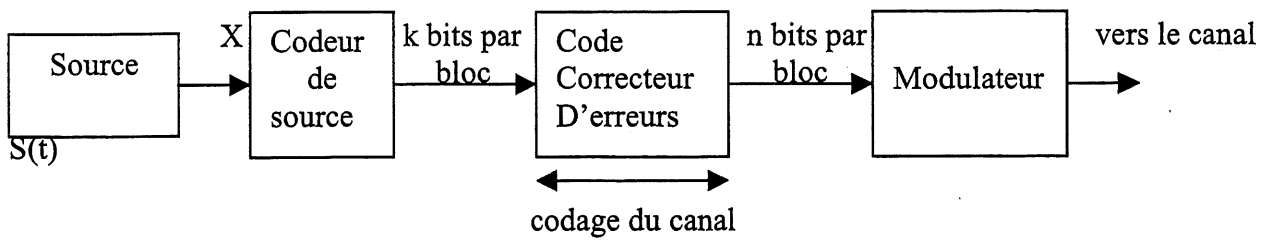


Figure 2-2 : codage de source et de canal

Dans le paragraphe suivant, on verra un exemple de protection des bits avec redondance : les codes linéaires.

2.2.1 Les codes linéaires

On désigne sous ce nom tout code dont chacune des combinaisons contient, en plus des m informations binaires strictement nécessaires, k positions binaires supplémentaires dites de *contrôle*, les k positions de contrôle étant des combinaisons linéaires des m positions d'informations.

Un mot de code s'écrira donc, d'une manière générale, sous la forme :

$$\{s_i\} = \{x_1 x_2 \dots x_j \dots x_m \quad a_1 a_2 \dots a_k\}$$

avec $1 \leq i \leq 2^m$

$$a_j = \sum_{i=1}^{i=m} x_i \cdot a_{ij}$$

$$1 \leq j \leq k$$

Les termes x_i , a_{ij} et a_i pouvant prendre les valeurs 0 ou 1.

L'intérêt de ces codes vient de leurs propriétés détectrices d'erreurs.

Le plus connu est le code de Hamming, dont il existe d'ailleurs différentes classes. C'est ainsi que le code de Hamming (7,4), cette dernière notation signifiant que chaque vecteur de code comprend 7 positions binaires dont 4 positions d'informations, s'écrit sous la forme :

$$\{s_i\} = \underbrace{\{x_1 x_2 x_3 x_4\}}_{\text{information}} \quad \underbrace{\{a_1 a_2 a_3\}}_{\text{contrôle}}$$

dans laquelle :

- Chaque combinaison $\{x_1 x_2 x_3 x_4\}$ représente l'un des 16 niveaux de quantification qu'il est possible de définir en code binaire naturel, les poids de x_1 , x_2 , x_3 et x_4 étant respectivement égaux à 1, 2, 4 et 8 :
- Les relations linéaires de contrôle sont définies par :

$$a_1 = x_4 + x_3 + x_1$$

$$a_2 = x_4 + x_2 + x_1$$

$$a_3 = x_3 + x_2 + x_1$$

2.3 Codage du canal sans redondance

Parmi les raisons pour lesquelles on procède à un codage de canal sans redondance, il y a :

- L'effet des erreurs du canal sur les bits de protection qui peuvent être très importants, si les indices des mots de code ne sont pas bien sélectionnés.
- la réduction du débit binaire.

Dans le cas de la transmission du mot binaire représentant le vecteur quantifié, le quantificateur vectoriel et le système de communication sont habituellement conçus séparément.

La robustesse du codage du canal sans redondance peut être divisée en deux catégories .

1- Les codeurs qui traitent soigneusement l'attribution des indices, prennent la précaution d'atténuer les effets des erreurs du canal. Pour étiqueter les mots de code, on se réfère souvent à l'attribution des indices (un algorithme de ce genre est discuté au chapitre III). Cette conception, connue sous le nom 'Robust vector quantizer' peut être divisée en deux problèmes :

- Problème du codage de la source
- Problème d'attribution d'indices

2 - La deuxième catégorie traite de l'erreur du canal dans la conception et la reconstruction des vecteurs.

Si le problème d'attribution des indices est incorporé dans l'algorithme LBG, alors on parle d'une co-optimisation de codage de la source et du canal. À titre d'exemple, on cite la référence [36].

2.4 canal de transmission

L'information produite par des sources est transmise par des canaux (figure 2-3)

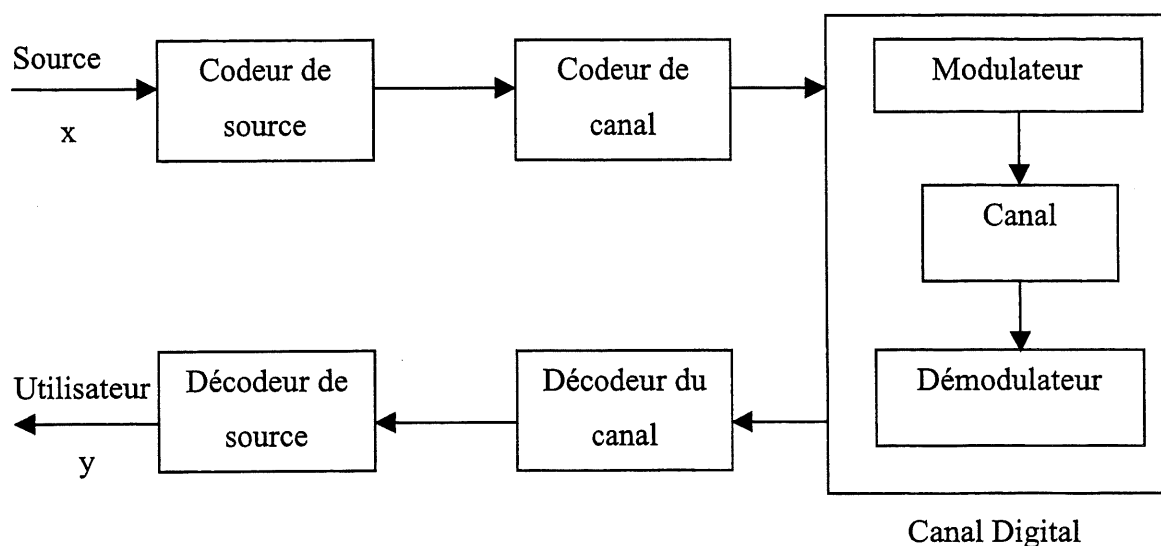


Figure 2.3: Système de communication et canal d'information

Un canal est caractérisé mathématiquement par une distribution de probabilité conditionnelle $p_{y/x}(y/x)$ avec le vecteur x en entrée et le vecteur y en sortie. Autrement dit, la probabilité d'avoir en sortie le vecteur y est donnée par l'entrée du vecteur x .

Le canal peut aussi être défini par la matrice de probabilités de transition suivante, où les tailles des messages d'entrée et de sortie sont respectivement I et J . Cette matrice définit le bruit de canal.

$$[p(y/x)] = \begin{bmatrix} p(y_1/x_1) & p(y_2/x_1) & \dots & p(y_J/x_1) \\ p(y_1/x_2) & p(y_2/x_2) & \dots & p(y_J/x_2) \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ p(y_1/x_I) & p(y_2/x_I) & \dots & p(y_J/x_I) \end{bmatrix}$$

Avec les contraintes suivantes :

$$\sum_{j=1}^J p(y_j/x_i) = 1 \quad \text{pour tout } i \text{ tel que : } 1 \leq i \leq I \quad (2-2)$$

Et $p(y_j/x_i) \geq 0$ pour tout i et pour tout j tels que : $1 \leq i \leq I$ et $1 \leq j \leq J$

Sachant que la distribution d'entrée $p_x(x)$, la distribution de sortie peut être calculée en utilisant la formule :

$$p(y_j) = \sum_{i=1}^I p(y_j/x_i)p(x_i), \quad 1 \leq j \leq J \quad (2-3)$$

Où : $p(x_i)$ est la probabilité que le message x_i soit émis,

$p(y_j/x_i)$ est la probabilité de recevoir y_j sachant que x_i a été émis.

Les trois modèles les plus utilisés actuellement en ce qui concerne le codage du canal demeurent assez simples :

- Le canal binaire symétrique noté (BSC), cas particulier d'un canal discret.
- Le canal à bruit blanc gaussien additif (AWGN, Additive White Gaussian Noise).
- Le canal de Rayleigh

Parmi ces trois modèles, le modèle choisi est le canal binaire symétrique qui reste facile à traiter, mais il n'est pas toujours adapté aux réseaux réels comme les réseaux sans fil. Une étude détaillée sur le canal binaire symétrique est donnée dans le paragraphe suivant.

2.4.1 Canal binaire symétrique

Les entrées de ce canal sont les deux symboles binaires 0 et 1. Chaque symbole binaire présenté au canal est transmis correctement avec la probabilité $1-p$ et incorrectement avec la probabilité p .

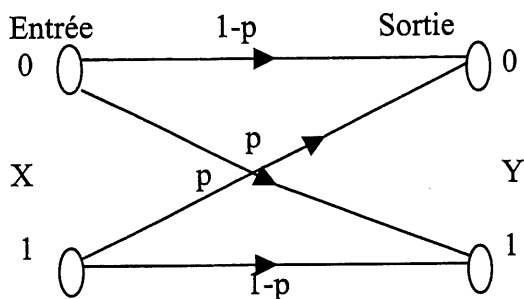


Figure 2.4 : Arbre de transition d'un canal binaire symétrique

		Sortie	
		0	1
Entrée	0	1-p	p
	1	p	1-p

Figure 2.5 : Matrice de transition

Les transformations qui peuvent être introduites par le bruit du canal sont des probabilités conditionnelles :

$$P(Y = 0 / X = 0) = P(Y = 1 / X = 1) = 1 - p \quad (2-3)$$

$$P(Y = 0 / X = 1) = P(Y = 1 / X = 0) = p \quad (2-4)$$

À chaque instant $t = n$, le canal binaire symétrique peut être dans l'un ou l'autre des deux états : " bon " ou " mauvais". Si le canal est dans l'état bon avec une probabilité $1-p$, il n'y aura aucun changement sur le bit transmis. Par contre, si le canal est dans l'état mauvais, le bit transmis sera modifié.

2.4.2 La capacité du canal binaire symétrique

Pour un canal donné, il existe un code de canal qui permet la transmission sans erreur à travers le canal à un taux C , où C est la capacité du canal.

Personne n'a encore trouvé un code qui permet l'utilisation d'un canal à sa capacité. Cependant, Shannon a déjà prouvé que le code existe [11].

Pour un canal binaire symétrique avec une certaine probabilité de transition $1-p$, la capacité est donnée par :

$$C = 1 + p \log_2(p) + (1-p) \log_2(1-p) \quad (2-5)$$

Dans la figure (2.6), on représente la capacité d'un canal binaire symétrique en fonction de la probabilité p . Si la probabilité $p = 0$, $C=1$. Par contre, s'il y'a une probabilité égale de recevoir 1 ou 0 ($p = 0.5$) sans tenir compte du signal envoyé, le canal est complètement incertain et aucun message ne peut être envoyé à travers ce canal. Dans ce cas, la capacité du canal est nulle ($C = 0$).

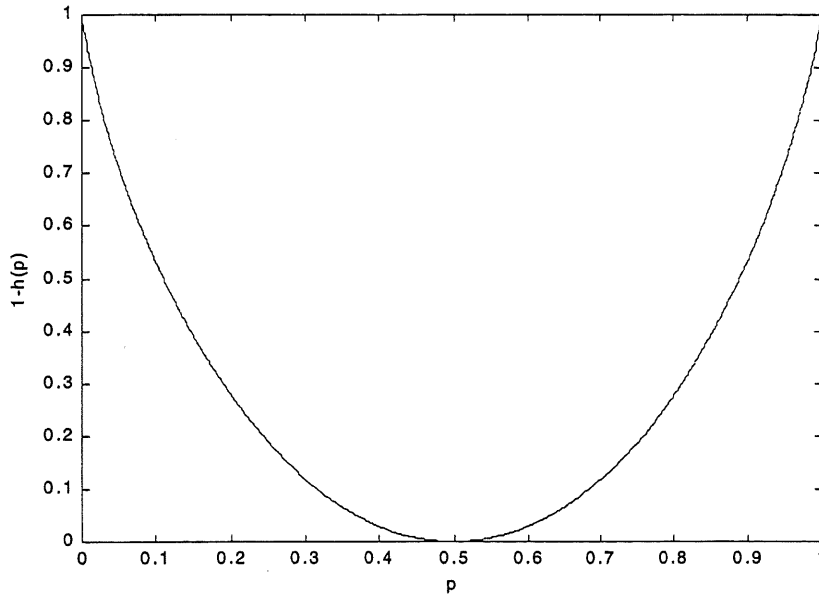


Figure 2.6 : La capacité d'un canal binaire symétrique

Pour un canal de bande passante B , nous pouvons transmettre en principe $2B$ changements par seconde. En effet, la capacité en termes bits/secondes est de $2BC$. Le cas d'un canal binaire symétrique, en remplaçant C dans l'équation (2-5), sa capacité sera :

$$C = 2B(1 + p \log_2(p) + (1-p) \log_2(1-p)) \text{ bits/sec} \quad (2-6)$$

Le débit maximum W pour un canal binaire symétrique est ainsi $W=2B$. Nous notons que $C < W$, c'est-à-dire que la capacité est toujours moins que le débit. Un autre terme à définir est le taux de transfert des bits D .

Dans la théorie, on a :

$$W \geq C \geq D \quad (2-7)$$

Et pratiquement :

$$W > C > D \quad (2-8)$$

Le théorème de Shannon pour le codage du canal ne s'applique pas sur la source mais juste sur le canal. Si la source est codée d'une façon optimale, nous pouvons reformuler le théorème de codage de canal : une source d'informations avec l'entropie $H(x)$ peut être transmise sans erreurs à travers un canal avec :

$$H(x) \leq C \quad (2-9)$$

Pour plus de détails sur ce chapitre , le lecteur pourra consulter la référence [28] .

2.5 Classification des codeurs source-canal

Dans cette perspective, on peut classer les différents codeurs de source et de canal en 4 classes :

- Classe 1 : Codeurs concaténés Source-Canal

Dans ce cas, on concatène un codeur de source connu et un codeur du canal connu et on détermine les taux binaires du codeur de source et du codeur du canal, afin de maximiser la performance globale du système.

- Classe 2 : Codeurs conjoints Source-Canal

Dans cette classe, il y a tous les schémas de codage pour lesquels les opérations de codage de source et de canal sont intégrées dans une seule entité.

- Classe 3 : Codeurs conjoints avec contraintes

Dans cette classe, on retrouve les schémas de codage de source et de canal pour lesquels, soit le codeur, soit le décodeur ou les deux, sont modifiés pour tenir compte des erreurs du canal. Ici, un codeur de source optimisé pour un modèle sans bruit est réoptimisé en tenant compte d'une certaine statistique de bruit dans le canal

- Classe 4 : Codeurs adaptés

Ce genre de codeur consiste à chercher la façon optimale de rattacher un mot de code de source sur un mot de code de canal selon les conditions du canal [35]. Dans cette catégorie, on trouve la technique sous nom AMR (adaptive multi-rate).

Chapitre 3 : ATTRIBUTION DES INDICES

3.1 Historique

L'importance de l'attribution des indices était discutée en 1976 et la première référence qui a donné l'importance à ce problème pour le cas d'un quantificateur scalaire est traitée par Rydbeck et Sundberg dans [12]. L'extension de cette étude d'attribution d'indices pour le cas du quantificateur vectoriel est introduite par Gray et Gercho dans [13] et [14]. Dans la littérature, on trouve Farvardin qui a implémenté le recuit simulé [16], Potter et Chiang [15] en 1995 ont proposé une méthode utilisant le critère minmax et Knagenhjelm et Agrell (1996) qui ont formulé le problème d'attribution des indices en utilisant la transformation Hadamard et en introduisant le concept de linéarité de dictionnaire. Ces derniers ont développé un algorithme connu sous le nom "linearity increasing Swap algorithm" (LISA)[17].

3.2 Problématique

Lorsqu'un canal est présent entre les fonctions Q et Q^{-1} , l'indice transmis par la fonction Q et l'indice reçu par la fonction Q^{-1} peuvent être différents si une erreur de transmission s'est produite (figure 3.1).

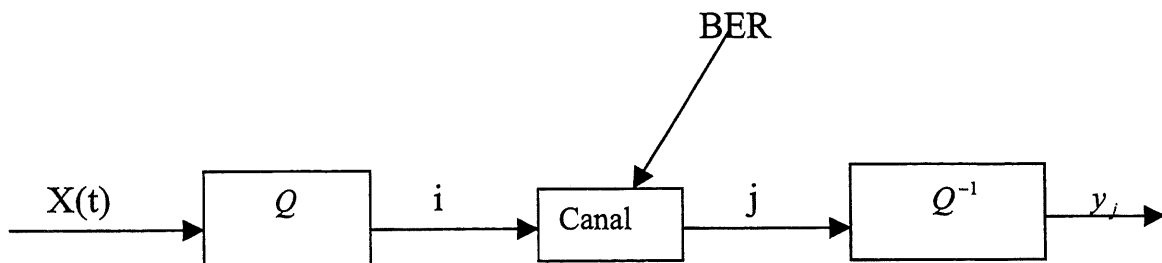


Figure 3.1 : Une chaîne de communication en présence du bruit de canal

Exemple d'un quantificateur à 2 dimensions :

Dans cet exemple, on va montrer l'impact d'une erreur de canal sur un mot de code i , présenté sur 3 bits, à transmettre sur un canal bruité :

Tel qu'illustré à la figure 3.2, le vecteur d'entrée au quantificateur est x . En appliquant la règle du plus proche voisin, sa quantification sera par le vecteur y_1 , donc le mot de code à transmettre est : $i = 001$

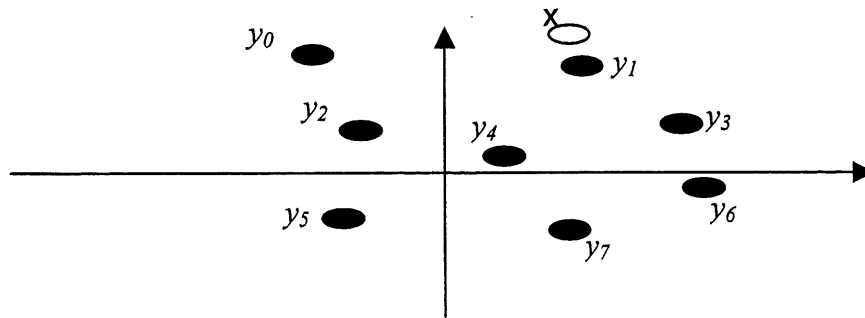


Figure 3.2 : Un quantificateur vectoriel à 2 dimensions de 8 vecteurs

S'il y'a une erreur sur le premier bit du i , la valeur reçue sera $j = 101$.

Au décodage, la fonction Q^{-1} donnera le vecteur y_5 en sortie et non le vecteur y_1 . Ce qui résultera d'augmenter l'erreur totale entre le point x et le point décodé y_5 .

Comment pourrait-on diminuer cet impact du canal?

Le problème revient à déterminer l'allocation optimale des indices i à tous les points y_i , afin de minimiser la distorsion encourue par des erreurs de canal, en particulier la probabilité de recevoir l'indice j sachant que c'est l'indice i qui a été transmis. Dans la littérature, les algorithmes qui tentent de déterminer l'allocation optimale des indices se retrouvent, sous le nom 'Index assignment'.

Un quantificateur vectoriel conçu par l'algorithme LBG-VQ peut être sensible aux erreurs de canal dues à sa tendance d'organiser aléatoirement les indices. Les mots de code qui sont inter-changés peuvent être très distants dans l'espace de signal, ce qui contribue largement à la distorsion totale quand les erreurs se manifestent. Donc, le problème d'attribution des indices est une partie importante de la conception d'un système de communication.

Comme il y a $N!$ façons différentes de permuter N objets (dans le cas présent ce sont N chiffres), une recherche exhaustive de la gestion optimale des indices n'est pas réalisable en pratique et les algorithmes sont en conséquence sous-optimaux.

Par exemple, prenons le cas d'un quantificateur à 8 valeurs ($\log_2(8)=3$ bits). Le nombre de permutations possibles des indices i parmi les 8 points est :

$$P = 8! = 8 * 7 * 6 * 5 * 4 * 3 * 2 * 1 = 40320$$

Pour optimiser l'allocation des indices en essayant toutes ces possibilités, il faudrait :

- 1) Allouer les indices selon une permutation donnée
- 2) Calculer l'erreur moyenne due au canal pour un grand nombre de vecteurs x
- 3) Allouer les indices selon une nouvelle permutation
- 4) refaire les étapes 2) et 3) $8! - 1 = 40319$ fois

3.3 La distorsion totale

Par définition, la distorsion totale est la distorsion entre le signal d'entrée et le signal de sortie. Puisqu'on a vu 2 types de distorsion : distorsion de quantification et distorsion du canal, donc cette distorsion totale est la somme des deux distorsions.

Totty et Clark ont montré [20] que quand le bruit du canal est ajouté à un quantificateur optimal, l'erreur quadratique moyenne peut être séparée en : l'erreur quadratique de

quantification et l'erreur quadratique due au bruit du canal. Elle est donnée ici par le Lemme suivant :

Lemme 1 : La distorsion moyenne d'un quantificateur qui satisfait la condition du centroïde en présence du bruit du canal peut être écrite comme :

$$D(q; b) = D_s(q) + D_c(b) \quad (3.1)$$

Avec $D_s(q)$: est la distorsion causée par le quantificateur,

$D_c(b)$: est la distorsion causée par le bruit du canal

D'après (1.5) et (2.1), la distorsion totale est :

$$D(q; b) = \frac{1}{k} \sum_{i=1}^M \int_{s_i} p(x) d(x, y_i) dx + \frac{1}{k} \sum_{i=1}^M \sum_{j=1}^M P(y_i) P(b(y_j) / b(y_i)) d(y_i, y_j) \quad (3.2)$$

Si on prend comme critère de mesure l'erreur quadratique, la distorsion totale devient :

$$D(q; b) = \frac{1}{k} \sum_{i=1}^M \int_{s_i} p(x) \|x - y_i\|^2 dx + \frac{1}{k} \sum_{i=1}^M \sum_{j=1}^M P(y_i) P(b(y_j) / b(y_i)) \|y_i - y_j\|^2 \quad (3.3)$$

3.4 Un algorithme d'attribution d'indices : Binary Switching

L'algorithme choisi pour cette étude est un algorithme sous-optimal et qui a comme nom 'Binary Switching' [13]. Ce choix est justifié par une étude déjà faite par Knagenhjelm et Agrell [17] qui ont montré que la performance de cet algorithme est meilleure que les algorithmes qui sont dans [17] et [16], ainsi que la complexité de cet algorithme est minimale par rapport à la méthode LISA.

On définit une opération d'addition au niveau des bits, notée \oplus de la façon suivante :

$$\oplus : \{0,1\}^b * \{0,1\}^b \longrightarrow \{0,1\}^b \quad (3.4)$$

Si $i = i_1, i_2, \dots, i_b \in \{0,1\}^b$ et $j = j_1, j_2, \dots, j_b \in \{0,1\}^b$ où $i_k, j_k \in \{0,1\}$ pour $1 \leq k \leq b$, alors $i \oplus j$ est un élément de $\{0,1\}^b$ de représentation binaire : c_1, c_2, \dots, c_b où $c_k = i_k + j_k \pmod{2}$ pour $1 \leq k \leq b$. En effet, chaque bit binaire c_k est le résultat d'un ou-exclusif de i_k et j_k ($c_k = i_k \text{ XOR } j_k$).

Si l'indice à transmettre à travers un canal bruité appartient à l'ensemble $\{0,1\}^b$, ses valeurs en général vont être reçues sous différentes formes dans l'ensemble $\{0,1\}^b$. En effet, un canal avec bruit sans mémoire peut être représenté par un mappage :

$$\tau : \{0,1\}^b \longrightarrow \{0,1\}^b :$$

$$\text{avec } \tau(z) = z \oplus \eta \quad (z \in \{0,1\}^b) \quad (3.5)$$

et η est un vecteur aléatoire à valeurs dans $\{0,1\}^b$.

Ainsi τ est une fonction de vecteur aléatoire η qui décrit les effets des erreurs du canal sur un indice binaire transmis.

On note par S_N l'ensemble des permutations π de $\{0,1\}^b$ dans lui-même.

Chaque permutation $\pi \in S_N$ est appelée la fonction d'attribution des indices du quantificateur.

Définition : Pour chaque quantificateur vectoriel $Q = D \circ C$ et pour chaque permutation π de $\{0,1\}^b$, le quantificateur vectoriel pour un canal bruité est un mappage de R^p vers un ensemble de vecteur Y qui est donné par :

$$Q_\pi = D \circ \pi^{-1} \circ \tau \circ \pi \circ C \quad (3.6)$$

Où : C est le codeur de source,

D est le décodeur,

τ est le mappage d'un canal bruité,

π^{-1} se désigne par la permutation inverse de π .

Donc, un canal Q_π est représenté par (C, D, π) (figure 3.3) :

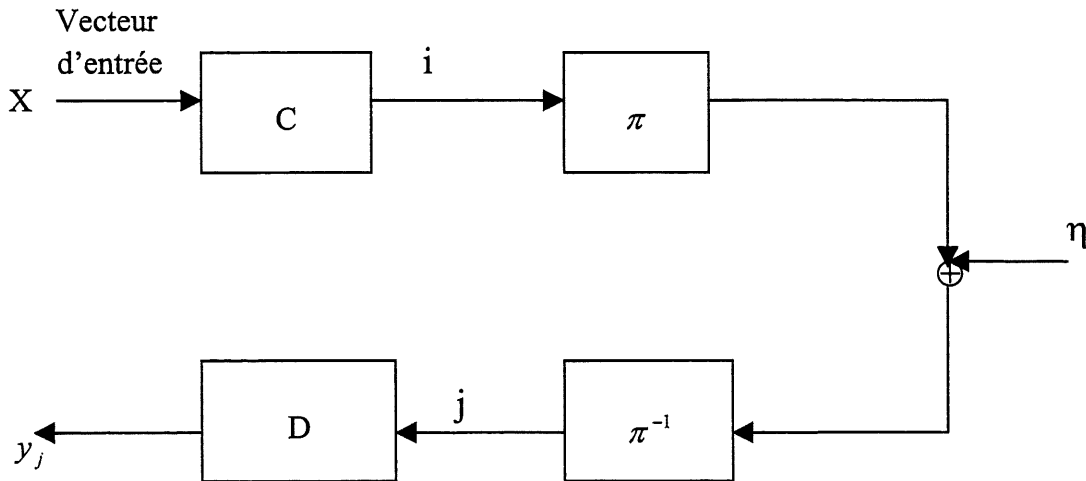


Figure 3.3 : Diagramme d'un QV avec un canal discret sans mémoire.

Dans le cas d'un canal sans bruit : La variable aléatoire η est égale à 0, τ est le mappage identité dans $\{0,1\}^b$, $Q_\pi = Q$ et le choix de la fonction d'attribution des indices π n'a aucun effet sur la performance du quantificateur.

Pour un quantificateur vectoriel donné Q et un canal donné, on cherche à trouver le quantificateur vectoriel Q_π qui minimise la distorsion moyenne donnée par :

$$e_\pi = E[d(X, Q_\pi(X))] \quad (3.7)$$

L'objectif de cette approche est de trouver la meilleure permutation qui minimise (3.7) parmi les S_N permutations possibles.

$$e_{\min} = \min\{e_{\pi} : \pi \in S_N\} \quad (3.8)$$

Donc, la meilleure permutation est celle qui minimise la quantité suivante :

$$e_{\pi} = E[d(x, y_j)] \quad (3.9)$$

L'indice binaire représentant le vecteur x transmis à travers un canal est :

$$\pi(C(X)) = \pi(i) \quad (3.10)$$

L'indice reçu est $\tau(\pi(i))$ et l'indice du vecteur utilisé pour quantifier X à la réception est $\pi^{-1}(\tau(\pi(i)))$. On définit l'indice j de la manière suivante :

$$j = \pi^{-1}(\tau(\pi(i))) = \pi^{-1}(\pi(i \oplus \eta)) \quad (3.11)$$

3.4.1 La distance de Hamming

Pour chaque indice binaire $q \in \{0,1\}^b$ et pour chaque entier m avec $0 \leq m \leq b$, on définit l'ensemble des voisins de cet indice binaire par :

$$N^m(q) = \{r \in \{0,1\}^b : H(q,r) = m\} \quad (3.12)$$

$H(.,.)$ est la fonction de la distance de Hamming.

$N^m(q)$ est l'ensemble de tous les entiers qui appartiennent à $\{0,1\}^b$ et dont la distance de Hamming avec q est égale à m .

En effet, $N^m(q)$ est l'ensemble des indices résultants des bits erronés [14] et Il est calculé comme suit :

$$|N^m(q)| = \binom{b}{m} \quad (3.13)$$

Exemple :

On prend : $b = 4$ et $q=2=0010$

L'ensemble des voisins de 2 est donné par une présentation décimale :

$$N^0(2) = \{2\},$$

$$N^1(2) = \{3, 0, 6, 10\}$$

$$N^2(2) = \{1, 4, 8, 7, 11, 14\}$$

$$N^3(2) = \{12, 15, 9, 5\}$$

$$N^4(2) = \{13\}$$

Définissons le coût de chaque vecteur de sortie y_k :

$$C_{\pi}^{(m)}(y_k) = p(y_k) \sum_{w \in N^m(\pi(k))} d(y_k, y_{\pi^{-1}(w)}) \quad (3.14)$$

avec $p(y_k) = \Pr[y \in R_k] \quad (3.15)$

Pr est la probabilité d'un événement aléatoire.

L'équation (3.14) mesure la contribution relative aux erreurs de bits attendues quand m bits sont erronés et y_k est sélectionné par le codeur.

On définit q_m la probabilité de recevoir le mot binaire de $\{0,1\}^b$ transmis par :

$$q_m = \varepsilon^m (1 - \varepsilon)^{b-m} \quad (3.16)$$

Avec ε est la probabilité de changement de bit.

D'autre part, le coût total $C_\pi(y)$ d'un vecteur y est :

$$C_\pi(y) = \sum_{m=1}^b q_m C_\pi^{(m)}(y) \quad (3.17)$$

qui mesure la contribution totale de la distorsion totale causée par les erreurs possibles du canal.

La distorsion totale de tout le dictionnaire est donnée par :

$$D_\pi = \sum_{k=0}^{N-1} C_\pi(y_k) \quad (3.18)$$

3.4.2 La description de l'algorithme

La description du canal (notamment la probabilité de changement de bit (ε)) et le dictionnaire du quantificateur vectoriel sont donnés en premier. L'algorithme s'arrête éventuellement en produisant une attribution des indices aux vecteurs du dictionnaire.

Pour chaque permutation des indices π , la sélection des vecteurs y_i , qui est une approximation des vecteurs d'entrée x , cause la transmission des indices $\pi(i)$. C'est conceptuellement équivalent à ce que pour chaque indice i le vecteur y_i change son emplacement selon $\pi(i)$.

L'algorithme binary switching réarrange le dictionnaire pour que la sommation (3.18) soit un minimum local. L'idée générale est d'appliquer itérativement la commutation des positions de deux vecteurs pour réduire le terme D_π après chaque commutation. La diminution de D_π apporte la progression de l'algorithme. Chaque commutation constitue un changement dans la permutation π . Le choix de paires des vecteurs à commuter pour chaque itération est déterminée par un ordre des vecteurs. Pour chaque vecteur y , on attribue un coût $C_\pi(y)$ (3.17). Les vecteurs sont triés en ordre décroissant selon leurs coûts. Le vecteur avec le coût le plus élevé y_0 est sélectionné en premier. On cherche parmi tous les vecteurs, le vecteur qui sera commuté avec y_0 et qui aura le plus d'impact sur la diminution de la distorsion totale du dictionnaire D_π (3.18). S'il n'y a aucun vecteur à switcher avec y_0 , on prend le vecteur qui a le deuxième plus grand coût pour switcher avec un autre vecteur. S'il ne reste aucun vecteur à switcher qui peut diminuer la distorsion totale, l'algorithme est donc arrivé à l'état du minimum local.

Par l'ordonnancement des vecteurs selon leurs contributions dans la diminution de la distorsion totale, l'algorithme donne un sens de direction afin de savoir quel est le vecteur le plus important à switcher en premier. Le vecteur avec un grand coût contribue largement à la distorsion du quantificateur vectoriel.

L'algorithme du binary switching fonctionne selon les étapes suivantes :

Algorithme du Binary Switching

- (1) Choisir aléatoirement une permutation initiale $\pi_{initiale}$ pour le dictionnaire d'entrée.

$$i = 0$$

- (2) Pour $k=1$ jusqu'à $nb_vecteur$
 - Calculer $C_\pi(y_k)$ (3.17)Fin

- (3) Trier les vecteurs selon leurs coûts en mode décroissant

$$\delta^* = 0$$

- (4) Pour $j=1$ jusqu'à $nb_vecteur$
 - Calcul D_π (3.18).
 - Switcher le vecteur y_i avec y_j
 - Calcul $D_{\pi'}$ (3.18)
 - $\delta = D_{\pi'} - D_\pi$
 - Switcher le vecteur y_j avec y_i
 - Si ($\delta > 0$) alors retourner à (4)
 - Si ($|\delta| > \delta^*$) alors $\delta^* = |\delta|$ et $j^* = j$Fin

- (5) Si ($|\delta^*| > 0$) alors
 - Switcher le vecteur y_i avec y_{j^*} .
 - Retourner à (2)

- (6) Si ($|\delta^*| = 0$) alors
 - Si ($i = nb_vecteur$) alors arrêt
 - Sinon
 - $i++$
 - Retourner à (2)

3.4.3 La complexité de l'algorithme

Comme déjà vu précédemment, la solution optimale consiste à chercher la meilleure permutation parmi les $N!$ permutations possibles. Admettons que l'on ait un dictionnaire d'une taille de 32 vecteurs, alors le nombre de permutations possibles est supérieur à 10^{35} .

L'algorithme réduit cette complexité de calcul en obtenant une bonne permutation qui augmente la qualité d'un signal envoyé à travers un canal bruyant.

On décomposera la complexité selon les procédures de l'algorithme les plus importantes :

- L'algorithme de tri
- Le calcul de la quantité $C_{\pi}(y_i)$

Cette complexité est divisée entre la première itération (initialisation) et la mise à jour des valeurs.

Pour l'initialisation (étape 1), la complexité du calcul de la quantité $C_{\pi}(y_i)$ est de l'ordre de $O(n \log(n))$ et l'algorithme de tri (l'algorithme choisi est le tri Bull) est de l'ordre de $O(n \log(n))$.

Après la première itération, le calcul de $C_{\pi}(y_i)$ devient de l'ordre de $O(\log(n))$ et la complexité de l'algorithme du tri se réduit à l'ordre de $O(\log(n))$.

Pendant chaque itération de l'algorithme, $C_{\pi}(y_i)$ doit être calculé pour chaque couple de vecteurs choisis avant de déterminer le vecteur qui donne la meilleure réduction de D_{π} . Alors, le nombre de calcul de $C_{\pi}(y_i)$ est limité par le nombre total des couples des vecteurs qui est de l'ordre de $O(n^2)$.

Donc la complexité totale de l'algorithme est de l'ordre de : $O(n^2 \log(n))$

3.4.4 La performance de l'algorithme

Après la première itération de l'algorithme, la distorsion D_π (3.18) diminue de façon remarquable (voir figure 3.4). Après la 7ème itération, la distorsion commence à se stabiliser. Cette figure est obtenue par des simulations de l'algorithme sur une source gaussienne. En effet, l'algorithme cherche à atteindre le local optimum.

On obtient une itération par une recherche des meilleurs indices à attribuer pour chaque vecteur du quantificateur.

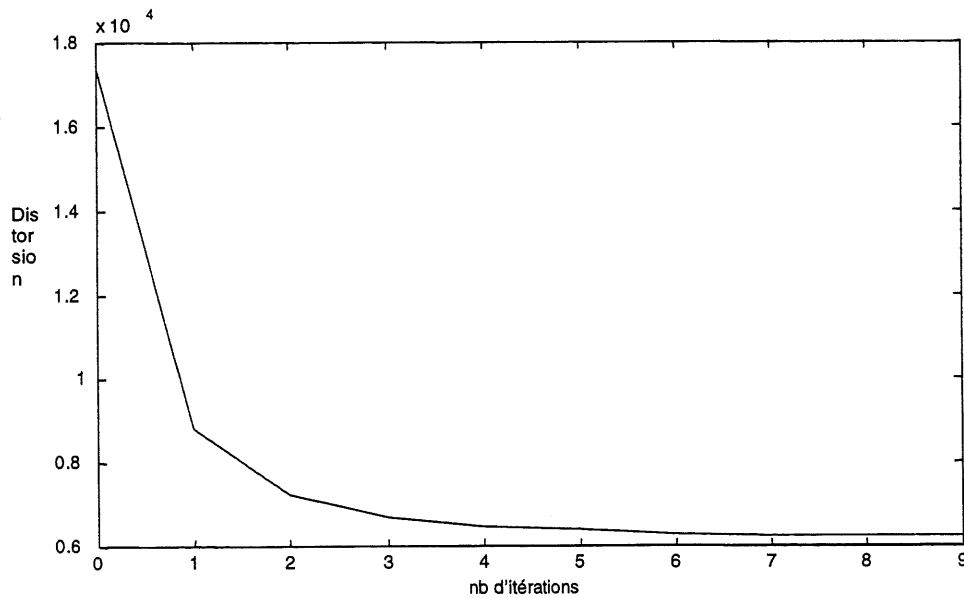


Figure 3.4: Convergence de l'algorithme

Afin de réduire le temps de l'exécution de l'algorithme, le coût $C_\pi(y_i)$ (3.17) de chaque vecteur de la source est calculé en fonction d'un changement d'un seul bit du mot de code y_i . En effet, le concept de changement d'un seul bit a prouvé une satisfaction égale au concept de changement de tous les bits du mot de code.

Une autre façon de voir la performance de l'algorithme : dans la figure suivante (3.5), on compare la performance d'une permutation produite après l'exécution de l'algorithme

(dite : meilleure permutation), une autre permutation obtenue juste après la construction du quantificateur (dite : permutation initiale) et la troisième permutation (dite : la mauvaise permutation) produite par le même algorithme en maximisant la distorsion D_π (3.18). La mesure de performance entre les 3 permutations est prise en fonction du rapport-signal-sur-bruit (SNR) avec différents niveaux de bruit entre 10^{-5} et 10^{-1} .

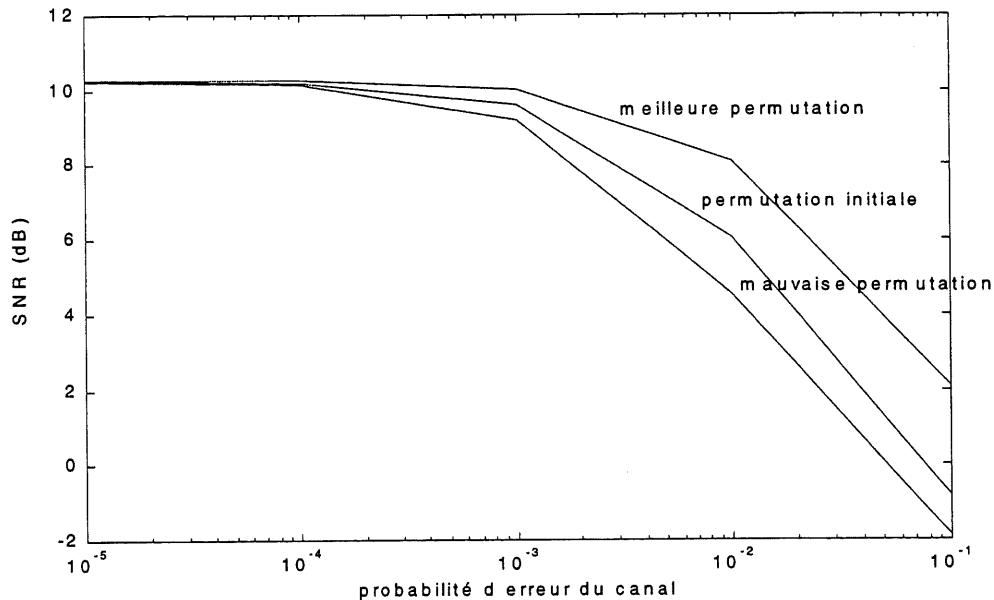


Figure 3.5 : Performance du BS dans un canal BSC en fonction de la probabilité d'erreur ε . Source gaussienne, taille 256, dimension 4

Une mauvaise permutation nous permet de voir les limites de la distorsion quand une permutation aléatoire est utilisée dans la conception d'un quantificateur vectoriel sans prendre en compte les erreurs du canal.

Donc la différence du SNR entre une mauvaise permutation et une permutation produite par le BS nous permet de voir les performances possibles que l'on puisse apporter en employant le BS.

D'autre part, la performance du quantificateur vectoriel a plus de sens quand le niveau du bruit du canal est élevé. Autrement dit, le gain devient plus élevé.

Une autre façon de voir cette performance, consiste à faire une comparaison entre les deux figures suivantes(3.6 et 3.7). La première figure (3.6) est une permutation initiale des indices où on ne prend pas en considération les effets des erreurs du canal. Par contre, dans la permutation de la figure (3.7), l'algorithme a essayé de rapprocher les indices qui ont une petite distance de Hamming.

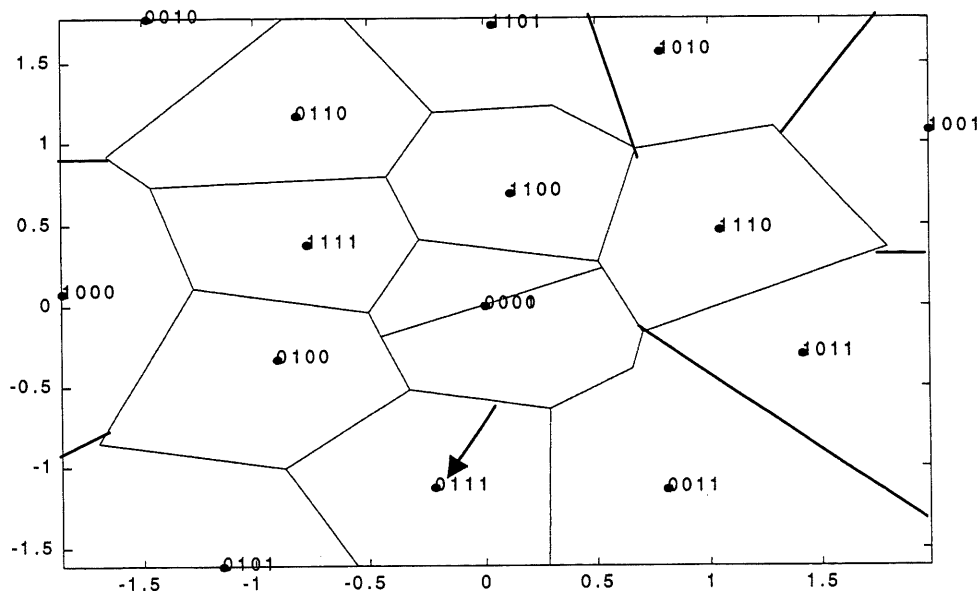


Figure 3.6: Les régions de voronoï d'un quantificateur à 16 vecteurs de dimension 2 avec une attribution initiale des indices

Exemple :

Le point 0111 dans la figure (3.6) était remplacé après l'exécution de l'algorithme par le mot binaire 1101 dans la figure (3.7). Ce changement est fait pour que le mot binaire 1101 soit proches des mots binaires qui ont des distances de Hamming petites. On donne par les suites les voisins de ce mot binaire avec leur distance de Hamming.

$$- 1001 \quad \longrightarrow \quad H(1101, 1001) = 1$$

- 0100 \longrightarrow $H(1101, 0100) = 1$
- 0101 \longrightarrow $H(1101, 0101) = 1$
- 1111 \longrightarrow $H(1101, 1111) = 1$
- 0001 \longrightarrow $H(1101, 0001) = 2$

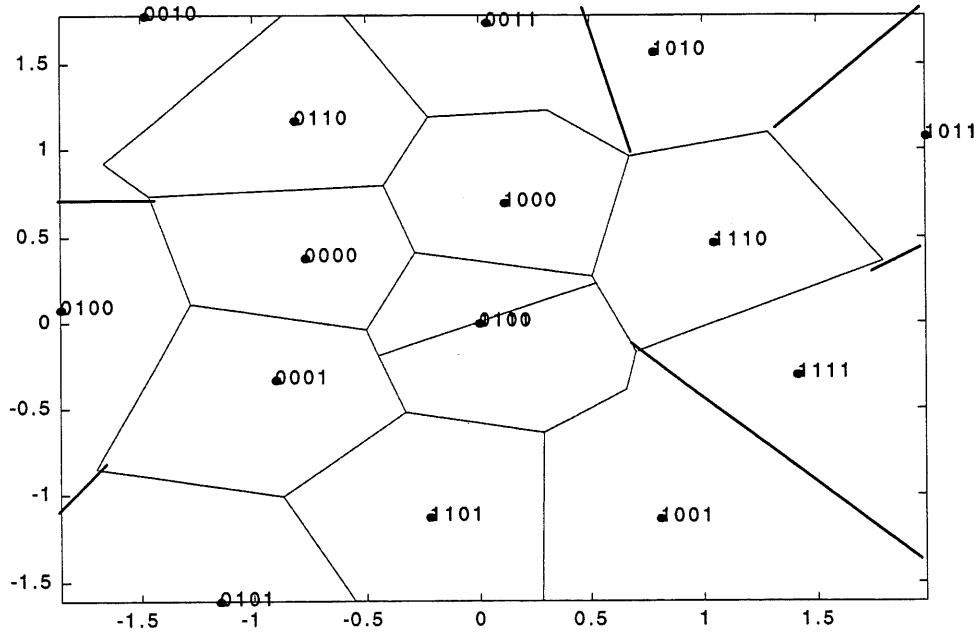


Figure 3.7: La région de voronoï d'un quantificateur à 16 vecteurs de dimension 2 avec une attribution des indices après l'algorithme

3.5 Classification par partition (Clustering)

Dans la même perspective, qui consiste à limiter l'impact des erreurs du canal, on présente ci-dessous une nouvelle approche plus connue sous le nom 'Clustering'.

Il s'agit de regrouper r vecteurs en k classes de telle sorte que les vecteurs d'une même classe soient les plus proches possibles et que les classes soient bien séparées.

Dans le cas d'un quantificateur de N vecteurs, il s'agit de diviser la région de Voronoï en sous-régions (classes) (figure 3.8). Chaque classe va contenir r vecteurs proches (3.19).

$$r = \frac{N}{k} \quad (3-19)$$

Nous cherchons désormais à obtenir une partition en k classes ou k a été fixé à priori. La plupart des techniques procèdent par améliorations successives d'une partition de départ : nous décrivons par la suite un de ces algorithmes.

Après le regroupement final des vecteurs par classe, on numérote les classes.

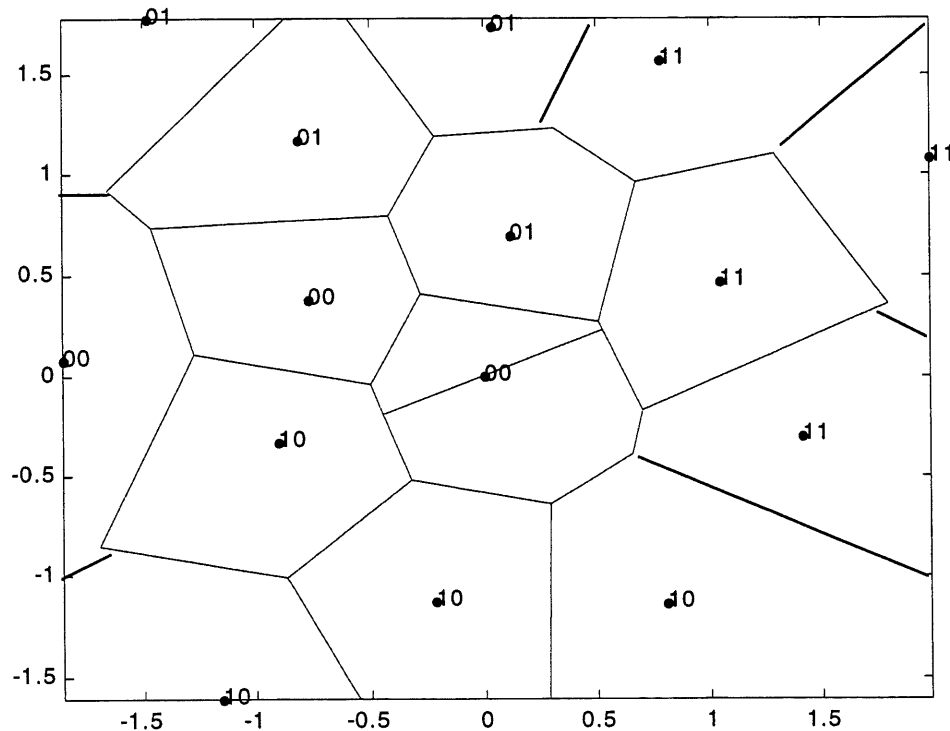


Figure 3.8 : une région de voronoï répartie en 4 sous-régions (classes). Chaque vecteur est étiqueté par le numéro du cluster.

Dans la figure suivante, on montre l'espace réservé par chaque sous-région(cluster)

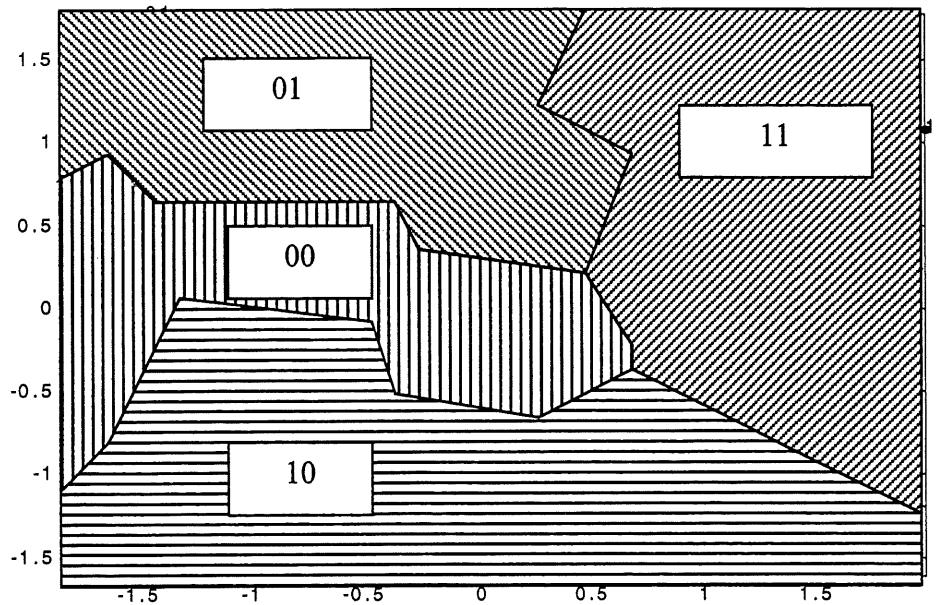


Figure 3.9 : une région de voronoï répartie en 4 sous-régions (classes) avec 4 vecteurs dans chaque classe

3.5.1 La description de l'algorithme

L'algorithme dit 'clustering' est un algorithme qui essaye de répartir une région de voronoï en sous-régions (classe). Cette répartition est exécutée à condition de regrouper dans chacune des classes les vecteurs voisins.

En premier lieu, l'algorithme essaye d'initialiser le centre de chaque classe aléatoirement. Une fois que l'initialisation est faite, il va continuer à attribuer pour chaque classe r vecteurs.

À l'aide de l'itération de Lloyd, il va trouver d'autres centres des classes. Autrement dit, il va chercher des nouveaux centres pour chaque classe .

L'algorithme s'arrête quand la distorsion globale du quantificateur est atteinte. Avec l'arrêt de l'algorithme, on aura k centres de classes et chaque classe contient r vecteurs.

L'algorithme adopté pour cette étude fonctionne selon l'organigramme suivant :

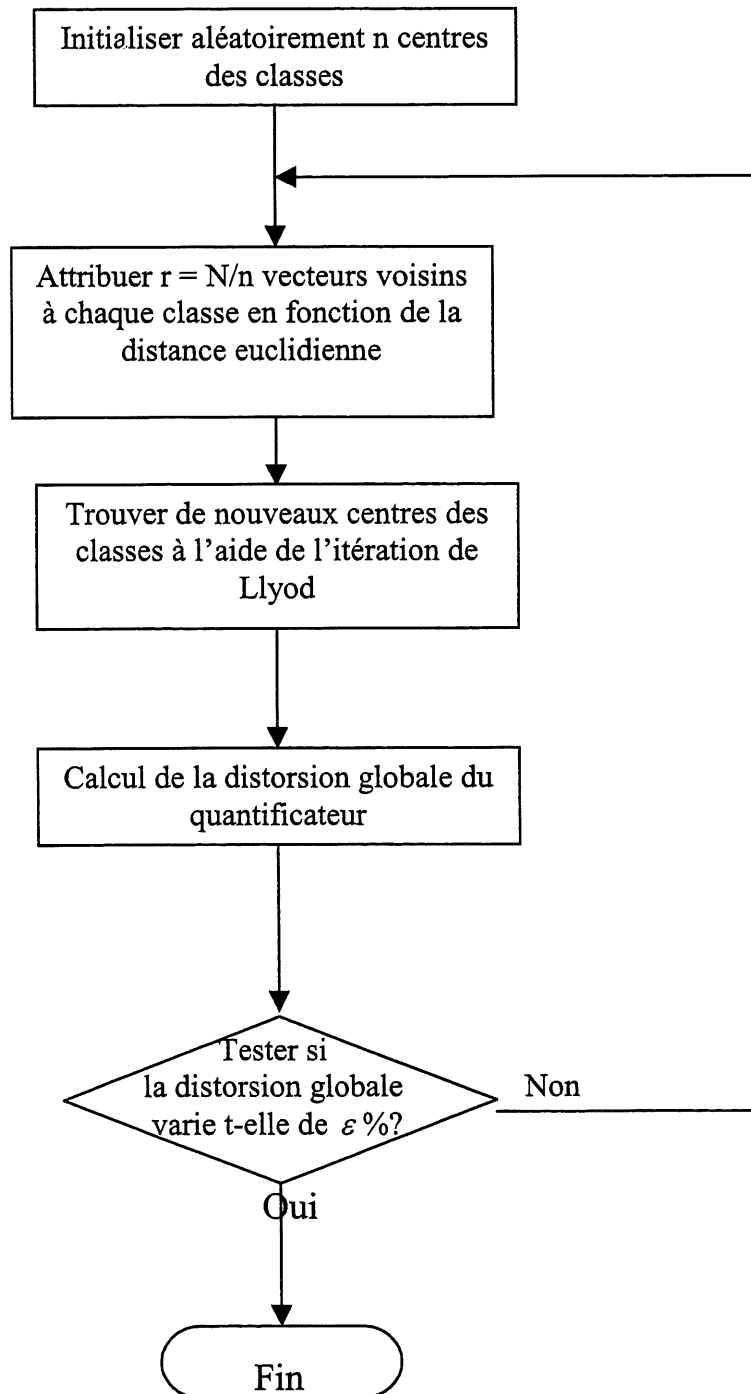


Figure 3.10 : Organigramme du fonctionnement de l'algorithme

3.6 L'approche adoptée : Exploitation des 2 algorithmes

L'approche de protection des mots binaires contre les erreurs du canal dans cette étude consiste à faire une combinaison des 2 algorithmes cités au-dessus.

En premier lieu, on divise le quantificateur en sous-région à l'aide de l'algorithme de classification. Une fois les centres des classes sont obtenus avec les vecteurs qui appartiennent pour chaque classe, l'algorithme de binary switching sera appliqué pour chaque classe.

Exemple :

Taille d'un quantificateur : 128

Nombre de classes : 8

les vecteurs du quantificateur seront codés sur 7 bits ($2^7 = 128$)

Les numéros des classes seront codés sur 3 bits $2^3 = 8$

Chaque classe contient $2^4 = 16$ vecteurs. Pour chaque classe on applique l'attribution des indices.

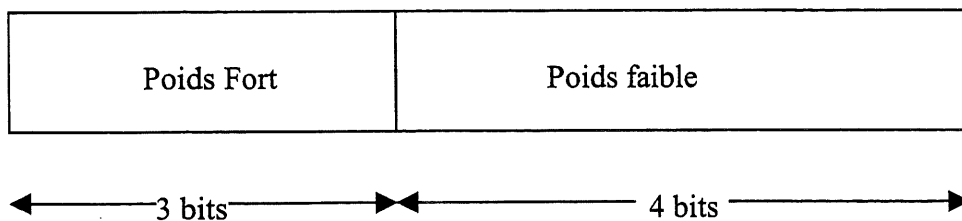


Figure 3.11 : Schéma d'un mot binaire représentant un vecteur LSF.

Donc pour ce quantificateur, on procédera 8 fois à l'attribution des indices et ce pour chaque classe.

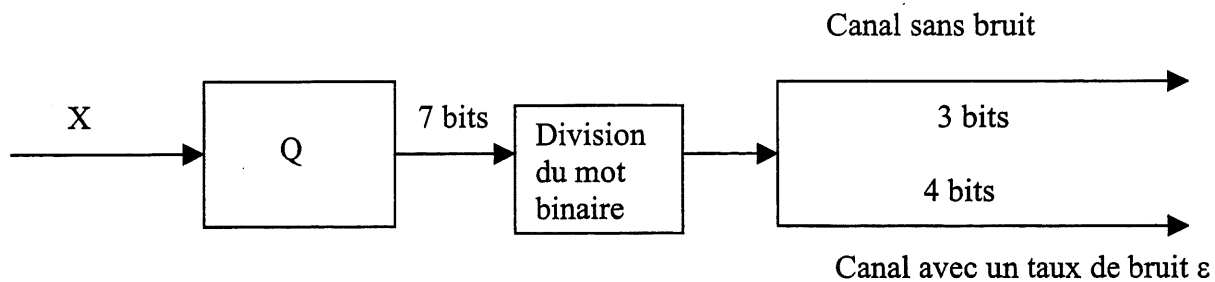


Figure 3.12: Acheminement du mot binaire

Tous les vecteurs qui appartiennent à une même classe auront les mêmes bits du poids fort de leurs représentations binaires.

À l'émission, le mot binaire sera envoyé en deux parties, le numéro de la classe à qui il appartient et son numéro dans la classe.

Le numéro de la classe, qui représente le poids fort du mot binaire, sera protégé avec une certaine redondance. Cette dernière rend le canal parfait, c'est-à-dire sans erreurs du canal. Par contre, le reste du mot binaire, qui représente le poids faible du mot binaire, ne sera pas protégé, donc son acheminement sera sur un canal bruité (figure 3.12).

S'il y a une erreur sur un mot, la protection du numéro de la classe, réduira l'impact de cette erreur. On verra par la suite la performance d'un quantificateur vectoriel avec une variation des bits réservés pour les classes ainsi que la réduction du bruit en utilisant l'attribution des indices.

Dans le prochain chapitre, on appliquera cette approche sur un quantificateur vectoriel des LSF.

CHAPITRE 4 : DISCUSSION ET RÉSULTATS

Dans la recherche d'une meilleure qualité de la parole synthétisée, on donnera une différence entre une attribution aléatoire et celle produite avec les algorithmes discutés dans le chapitre précédent. Notre objectif principal est de quantifier les paramètres de l'enveloppe spectrale (LSF) avec une distorsion spectrale aux alentours de 1 dB [18] en prenant en considération les effets des erreurs du canal.

Une autre étude sera faite sur la réduction des bits et l'introduction des bits de protection en fonction des différents taux d'erreurs.

L'approche sera appliquée sur deux types de quantificateurs :

- Le premier est un quantificateur vectoriel prédictif à 2 étages avec un splitting (5,5) au 2^{ème} étage. Le quantificateur vectoriel du premier étage contient 512 vecteurs, soit un débit de 9 bits par trame. Au deuxième étage, Le premier quantificateur contient 256 vecteurs, soit 8 bits par trame et le deuxième quantificateur a comme taille 128 vecteurs, soit 7 bits par trame. Donc le débit est de 24 bits par trame. Aux lecteurs qui veulent en savoir plus sur cette structure, ils trouveront une large explication dans [19].
- Le deuxième est composé de 5 quantificateurs. Chaque quantificateur est de taille 128 vecteurs, chaque vecteur est codé sur 7 bits ($2^7 = 128$) et chaque couple de LSF est quantifié indépendamment, soit un débit de $7*5 = 35$ bits par trame.

Ce chapitre se décompose en 2 parties : d'abord, la mesure utilisée pour cette étude qui est la distorsion spectrale, ensuite la comparaison des résultats des permutations aléatoires avec des permutations robustes, et enfin la réduction des bits que l'on peut obtenir en gardant la même qualité perceptuelle du signal de la parole.

4.1 La distorsion spectrale

Malheureusement, on n'est pas encore arrivé à exprimer explicitement la distorsion en termes des paramètres LSF. Généralement, dans la conception, on utilise soit l'erreur quadratique, soit l'erreur quadratique pondérée par une fonction choisie convenablement pour exploiter certaines caractéristiques du spectre de la parole.

La mesure de la distorsion spectrale [2] notée (SD) est donnée par l'équation (4-1) :

$$D_n = \frac{1}{2\pi} \int_{-\pi}^{\pi} [10 \log S_n(\omega) - 10 \log \hat{S}_n(\omega)]^2 d\omega \quad (4-1)$$

où $S_n(\omega)$ est le spectre original et $\hat{S}_n(\omega)$ est la version construite.

Si on choisit une fonction de pondération qui dépend des paramètres LSF adjacents, on peut exploiter le fait que si deux paramètres sont proches, ceci implique la présence d'un pic du spectre à cette fréquence particulière. Comme la localisation d'un pic de spectre est importante pour la qualité de la parole codée alors les deux paramètres en question doivent être soigneusement codés.

La mesure de distorsion définie par l'erreur quadratique pondérée est donnée par :

$$d(x, \hat{x}) = \sum_{m=1}^p w_m(x) (\hat{x}_m - x_m)^2 \quad (4-2)$$

Où $\{w_m(x)\}$, $m = 1, \dots, p$ est une fonction de pondération qui dépend du vecteur d'entrée x , avec $x = (x_1, x_2, \dots, x_p)$.

Si $w_m(x) = 1$, on retrouve la mesure de distorsion définie par l'erreur quadratique.

La distorsion spectrale moyenne est calculée [32] par :

$$D_{\text{moy}}[\text{dB}] = \frac{1}{N_t} \sum_{n=1}^{N_t} (D_n)^{1/2} \quad (4-3)$$

N_t est le nombre de trames.

Pour tester la qualité du signal reconstruit de la parole, on impose les contraintes suivantes[18]:

- 1- La distorsion spectrale moyenne doit être moins de 1 dB
- 2- Moins de 2% de trames ayant la distorsion spectrale plus que 2dB
- 3- Aucune trame avec une distorsion spectrale plus grande de 4 dB.

Pour les quantificateurs vectoriels LBG-VQ, la réduction de cette distorsion à moins de 1 dB, demande des dictionnaires très larges.

4.2 Résultats

Les algorithmes de simulation sont identiques à ceux présentés au chapitre 3.

4.2.1 Quantificateur vectoriel prédictif à 24 bits par trame

La performance d'un quantificateur vectoriel prédictif d'un débit de 24 bits par trame est étudiée pour différents taux d'erreur entre 0.001% et 10% . Le tableau suivant résume cette performance.

BER	SD en dB	Points aberrants (Outliers)	
		2 – 4 dB	> 4 dB
0.001 %	0.91	0.34	0
0.01%	0.92	0.86	0
0.1%	1.01	2.57	0.86
1%	1.88	17.03	5.58
10%	4.95	30.36	62.99

Tableau 4-1 : Les effets des erreurs du canal en fonction de la distorsion spectrale (SD) avec un quantificateur vectoriel prédictif de 24 bits par trame

Ce qu'on peut tirer comme remarque du tableau 4-1, est que les erreurs du canal ont beaucoup d'effet sur les trames à partir d'un taux d'erreur 0.1% . Cet effet se résulte dans le nombre de trames qui ont une distorsion spectrale supérieure à 4 dB. Ainsi que la distorsion spectrale qui dépasse 1 dB à partir d'un taux d'erreur plus élevé de 0.1%. Pour cette raison, on fera une autre étude sur un quantificateur vectoriel avec un dictionnaire plus large.

4.2.2 Quantificateur vectoriel à 35 bits par trame

En ce qui concerne la deuxième structure du quantificateur vectoriel de 35 bits par trame, dans la figure (4.1), la distorsion spectrale est calculée avant et après l'attribution des indices pour un canal binaire symétrique en variant le taux d'erreur du canal entre 10^{-5} et 10^{-1} .

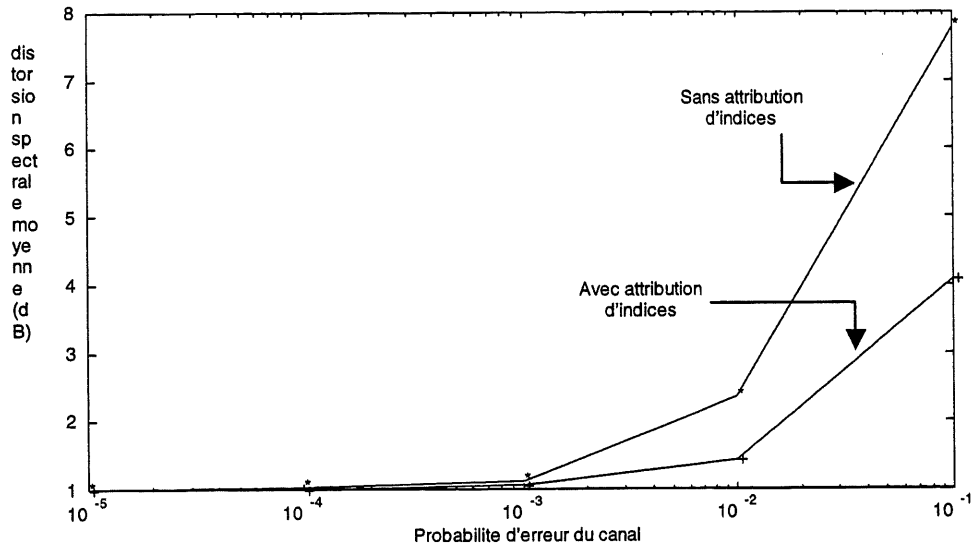


Figure 4.1 : Variation de la distorsion spectrale des deux permutations en fonction des Taux de probabilité d'erreur du canal.

D'après la figure précédente, la diminution de la distorsion spectrale entre la permutation initiale et la meilleure permutation est très importante dans le cas où la probabilité d'erreur du canal est très élevée. On peut remarquer en outre que lorsque le taux de la probabilité d'erreur du canal est inférieur à 10^{-3} , la distorsion spectrale est aux alentours de 1 dB, ce qui implique que le binary switching est un algorithme robuste pour un canal faiblement bruité. Par contre, si le canal est fortement bruité, la distorsion spectrale est supérieure de 1 dB.

Pour remédier à ce problème, des méthodes correctives sont nécessaires dont le but est d'assurer une bonne qualité perceptuelle de la parole. D'où l'intérêt de l'algorithme de classification présenté dans le chapitre précédent.

Par la suite, on discutera le nombre de bits à protéger en fonction de la probabilité des erreurs du canal.

NB : Les bits protégés sont toujours les premiers bits du mot binaire à transmettre.

Dans le tableau suivant [4.2], on présente combien de bits doivent être protégés pour garder une distorsion spectrale aux alentours de 1 dB au cas où le taux de la probabilité d'erreur du canal est supérieur à $0.1\% = 10^{-3}$.

BER	Nombre de bits protégés	SD en dB	Points aberrants (Outliers) in %	
			2-4 dB	> 4 dB
0.1%	0	0.88	1.30	0.00
0.2%	2	0.90	1.72	0.01
0.3%	4	0.86	1.39	0.00
0.4%	5	0.86	1.69	0.01
0.5%	5	0.87	1.91	0.01
1%	6	0.83	0.90	0.00

Tableau 4.2 : Le nombre de bits nécessaires à protéger avec les différentes variations du BER

D'après le tableau, le nombre de bits à protéger est toujours proportionnel au taux d'erreur du canal, on est toujours capable de garder une distorsion spectrale aux alentours de 1dB sans protéger tous les bits. À partir d'un taux d'erreur supérieur à 1 %, il faut absolument que tous les bits des vecteurs représentant les LSF soient protégés afin de garantir une bonne qualité perceptuelle de la parole contre les canaux fortement bruités.

À partir des simulations, on observe que les deux structures étudiées sont suffisamment robustes pour un canal faiblement bruité. Une protection des bits est obligatoire quand le milieu de transmission est fortement bruité.

CONCLUSION

Il y a un besoin continu d'un codage robuste des signaux numériques de la parole avec des taux de codage différents. Les codeurs de parole à bas débit sont utilisés dans le cadre de communications radio-mobile en raison de la difficulté à transmettre sur ce type de canal. Dans cette catégorie de codeurs, on trouve les codeurs à prédiction linéaire qui sont généralement utilisés pour modéliser les trames de la parole et extraire les paramètres significatifs.

Dans ce mémoire, on a présenté une étude sur les deux problèmes : codage de la source et codage du canal. Ce codage conjoint devient rapidement un outil nécessaire pour la conception et le développement des systèmes de communication notamment les systèmes sans fil. En fait, ce type de codage nous donne une réponse adéquate aux deux problèmes majeurs liés à la conception d'un quantificateur vectoriel :

- Le problème d'apprentissage : comment distribuer les vecteurs de reproduction sur l'espace du signal?
- Le problème de robustesse : comment choisir les indices à transmettre pour que les effets de canal soient minimales sur la performance générale d'un tel système ?

Nous avons montré le rôle de la quantification vectorielle dans les transmissions digitales. Cette technique est un moyen très puissant de codage de la parole, de l'image et de la vidéo.

La deuxième partie de ce mémoire est consacrée au codage du canal. Ce codage est divisé en deux parties : codage sans redondance et codage avec redondance. Le problème du codage sans redondance consiste à trouver une attribution optimale des indices. Dans ce contexte, les algorithmes connus en pratique sont des algorithmes sous optimaux. Alors que

l'algorithme discuté est très robuste pour les applications où le canal est faiblement bruité. Dans le cas où le canal est fortement bruité, une certaine redondance s'impose.

Dans ce but, on a discuté un algorithme de classification, qui sert à déterminer le nombre de bits à protéger avec une redondance dans le cas où le canal est fortement bruité.

Pour conclure, deux idées nous semblent très intéressantes dans ce type de codage et qui méritent une grande considération :

- Comme le bruit du canal dépend de plusieurs facteurs et que le taux d'erreur d'un canal varie dans le temps, une étude sur le milieu de transmission pour estimer le niveau du bruit est très intéressante.
- De concevoir une structure adaptative qui permet de connaître l'état du canal (c'est-à-dire le taux d'erreur du canal) avant la transmission des données.

Liste des figures

Figure 1.1 : Système simple de communication.....	2
Figure 1.1 : Exemple de compression sans perte.....	5
Figure 1.2 : Exemple de compression avec perte.....	5
Figure 1.3 : Quantificateur vectoriel.....	6
Figure 1.4 : Régions de voronoï d'un dictionnaire à 64 points en 2 dimensions.....	7
Figure 1.5 : Organigramme de fonctionnement de l'algorithme LBG-VQ.....	9
Figure 1.6 : Convergence de l'algorithme LBG $b=2$ bit/échantillons ; $N=2$	10
Figure 1.7 : Le filtre d'analyse de prédiction linéaire.....	13
Figure 1.8 : Le filtre de synthèse de la prédiction linéaire.....	14
Figure 2-1 : Codage de source et du canal en présence du bruit	20
Figure 2-2 : codage de source et de canal.....	21
Figure 2.3: Système de communication et canal d'information.....	23
Figure 2.4 : Arbre de transition d'un canal binaire symétrique.....	25
Figure 2.5 : Matrice de transition	26
Figure 2.6 : La capacité d'un canal binaire symétrique.....	27
Figure 3.1 : Une chaîne de communication en présence du bruit du canal.....	30
Figure 3.2 : Un quantificateur vectoriel à 2 dimensions de 8 vecteurs.....	31
Figure 3.3 : Diagramme d'un QV avec un canal discret sans mémoire.....	35
Figure 3.4: Convergence de l'algorithme.....	42
Figure 3.5 : Performance du BS dans un canal BSC en fonction de la probabilité d'erreur ε . Source gaussienne, taille 256, dimension 4.....	43
Figure 3.6: Les régions de voronoï d'un quantificateur à 16 vecteurs de dimension 2 avec une attribution initiale des indices	44
Figure 3.7: La région de voronoï d'un quantificateur à 16 vecteurs de dimension 2 avec une attribution des indices après l'algorithme.....	45
Figure 3.8 : une région de voronoï répartie en 4 sous-régions (classes). Chaque vecteur est étiqueté par le numéro du cluster.....	46
Figure 3.9 : une région de voronoï répartie en 4 sous-régions (classes) avec 4 vecteurs dans chaque classe.....	47
Figure 3.10 : Organigramme du fonctionnement de l'algorithme.....	48
Figure 3.11 : Schéma d'un mot binaire représentant un vecteur LSF.....	49
Figure 3.12: Acheminement du mot binaire.....	50
Figure 4.1 : Variation de la distorsion spectrale des deux permutations en fonction des Taux de probabilité d'erreur du canal.....	58

Liste des tableaux

<i>Tableau 4-1 : Les effets des erreurs du canal en fonction de la distorsion spectrale (SD) avec un quantificateur vectoriel prédictif de 24 bits par trame.....</i>	<i>54</i>
<i>Tableau 4.2 : Le nombre de bits nécessaires à protéger avec les différentes variations du BER.....</i>	<i>56</i>

Bibliographies

- [1] Soong, F. K, Juang B.H (1984) 'Line Spectrum Pair (LSP) and Speech Data Compression'. IEEE Transactions on acoustics, Speech and Signal Processing, p 1.101 – 1.104
- [2] Sugamura N, Itakura F. (juin 1986) 'Speech Analysis and Synthesis Methods' Developed at ECL in NTT- From LPC to LSP, Speech Communication, North Holland, vol. 5, N. 2 p 199-215
- [3] G.H. Golub and C. F. V. 'Loan, Matrix Computations'. The Johns Hpkins University Press second ed., 1989
- [4] F. Itakura, 'Line spectrum representation of linear prediction coefficients of speech signals,' Journal Acoustical Society America, vol. 57, p 535, 1975.
- [5] Gray, R. M (avril 1984) 'Vector Quantization,' IEEE ASSP MAGAZINE, Vol 1, N° 2, p 4-29
- [6] A.Gercho and R. Gray, 'Vector quantization and signal compression.' Norwell,MA: Kluwer Academic Publisher
- [7] y. Linde, A. Buzo, and R. M. Gray, 'An algorithm for vector quantizer design,' IEEE Trans. On Communications, Vol. COM 28, p84-95, Jan 1980
- [8] B .S. Atal and M.R. Schroeder. 'Predictive coding of speech signals'. Bell System Tech. J., pp 1973-1986, October 1970

- [9] S. Singhal and B.S. Atal. 'Improving performance of multi-pulse LPC coders at low bit rates'. Proc. ICASSP 84. pp.11.3.1-11.3.4, 1984
- [10] P. Kroon, E.F. Deprettre and R.J. Sluyter. 'Regular-pulse excitation – A novel approach to efficient multipulse coding of speech'. IEEE Trans" ASSP, Vol. 34, N° 5, pp 1054-1063, October 1986
- [11] C.E. Shannon and W.Weaver, The mathematical Theory of Communication, University of Illinois Press, Urban, 1949.
- [12] N. Rydbeck and C. Sundberg, " Analysis of Digital Errors in Non Linear PCM Systems, " IEEE Trans. On Communications, Vol. 24, 1976.
- [13] K. Zeger and A. Gercho, "Zero Redundancy Channel Coding in Vector Quantization, " Electron. Lett., Vol. 23, N°. 12, pp 654-656, June 1987
- [14] K. Zeger and A. Gercho, "Pseudo Gray Coding", IEEE Trans on Communications, Vol. 23, pp. 2147-2158, Dec 1990
- [15] D.M. Chiang and L" Potter, 'Minimax non-redundant channel coding for vector quantization ' in Proc. IEEE Int. Conf. Acoust. Speech Sign. Process., (Minneapolis), pp. 617-620, Apr 1993.
- [16] Farvardin, N., 'A study of vector quantization for noisy channels', Systems Research Center, University of Maryland, (Technical Research Report) 35p
- [17]. Petter Knagenhjelm, Eric Agrell, 'The Hadamard Transform – A Tool for Index Assignment'. IEEE Trans On Communication Theory, Vol. 42 , N° 4 , July 1996

[18] Kuldip K. Paliwal, and Bishnu S. Atal , 'Efficient Vector Quantization of LPC Parameters at 24 Bits/Frame ' IEEE Transactions on Speech and Audio Processing. Vol 1, N° 1. January 1993.

[19] W. P. Leblanc, S. A. Mahmoud. 'Efficient Search and Design Procedures for Robust Multi-Stage VQ of LPC Parameters for 4 Kb/s Speech Coding' Transactions on Speech Coding and Audio Processing, Vol 1. N° 4, October 1993.

[20] R.E.Totty and G.C. Clark, 'Reconstruction error in waveform transmission,' IEEE Trans. Inform. Theory, vol. IT-13, pp. 336-338, Apr 1967.

[21] Philip A. Chou, Michelle Effros, Robert M.Gray 'A Vector Quantization Approach to Universal Noiseless Coding and Quantization' IEEE Transactions on Information Theory. Submitted 17 June 1994, Revised 28 September 1995, Finalized 16 February 1996.

[22] N. Farvardin and V. Vaishampayan, 'Optimal Quantizer Design for Noisy Channels : an approach to Combined Source-Channel Coding', IEEE Trans. On Information Theory, vol. IT 33, pp. 827-833, Nov.1987.

[23] K.K. Paliwal and B.S.Atal. 'Efficient Vector Quantization of LPC Parameters at 24 BITS/Frame' IEEE Transactions on Speech and Audio Processing, Vol.1, N°.1, January 1993.

[24] W.P.Lebanc, B. Bhattacharya.S.A. Mahmoud and V. Cuperman. 'Efficient Search and Design Procedures for Robust Multi-Stage VQ of LPC Parameters for 4 Kb/s Speech Coding ' IEEE Transactions on Speech and Audio Processing, Vol.1. N°.4.October 1993.

[25] L .P. Hyvarinen 'Information Theory for Systems Engineers' Springer Verlag New York Heidelberg Berlin 1970

- [26] J. Max . 'Quantizing of Minimum Distorsion,' I. R. E Trans. Inf. Theory, vol. IT-6, pp 7-12, Mar.1960
- [27] S.P. Lloyd, 'Least Squares Quantization in Pcm ', IEEE. On Information Theory, vol. 28, pp. 129-137, Mar. 1982.
- [28] J. B. Anderson, Seshadri Mohan, 'Source and Channel Coding An Algorithm approach ' Kluwer Academic Publishers 1991.
- [29] N. Moreau, ' Compression de l'information'. Masson 1995.
- [30] R, Lefebvre. Unpublished work.
- [31] R. Salami, C.Laflamme, J-P. Adoul. ' 8 kb/s ACELP coding of speech with 10 ms speech-frame. Proc. ICASSP 94 pp II-97_II-100, 1995.
- [32] R. MATMTI, Quantification vectorielle robuste aux erreurs de transmission . Mémoire de maîtrise. Université de Sherbrooke. 1995
- [33] Jorge Rodriguez Guisantes. 'Codage Conjoint de source et de canal' Mémoire de thèse. L'école nationale supérieure des télécommunications. 1997.
- [34] W. B. Kleijin, K. K. Paliwal 'Speech Coding and Synthesis' Editeurs Elsevier, 1995.
- [35] Ekudden, E.; Hagen, R.; Johansson, I.; Svedberg, J. ' The Adaptive Multi-Rate Speech Coder'. Speech Coding Proceedings, 1999 IEEE Workshop on, 1999 Pages : 117-119.
- [36] T.Taniguchi, F. Amano, and S. Unagami, 'Combined source and channel coding based on multimode coding,' in Proc. Int. Conf. Acoust. Speech Sign. Process., (Albuquerque), pp. 447-480, 1990.