

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur
et de la Recherche Scientifique
Université Akli Mohand Oulhadj - Bouira -
Tasdawit Akli Muḥend Ulḥağ - Tubirett -



وزارة التعليم العالي والبحث العلمي
جامعة أكلي محمد أولحاج
- البويرة -

Faculté des Sciences et des Sciences Appliquées

كلية العلوم والعلوم التطبيقية

Référence :/MM/2021

المرجع :/م/ 2021

Mémoire de Master

Présenté au

Département : Génie Électrique

Domaine : Sciences et Technologies

Filière : Télécommunications

Spécialité : Systèmes des Télécommunications

Réalisé par :

HAMOUCHE yasmine

AISSAT djamila

Thème

**Mise au point d'un banc logiciel de manipulation de
codeurs de parole**

Devant le jury composé de :

Mr :	KABLA Aida	M.A.A	Univ. Bouira	Président
	SAIDI Mohammed	M.A.A	Univ. Bouira	Rapporteur
	CHELBI Salim	M.C.A	Univ. Bouira	Examineur

Année Universitaire : 2021-2022

TABLE DES MATIERES

	<i>Page</i>
Table des matières	i
Liste des figures.....	iii
Liste des tableaux	v
Liste des abréviations	vi
Introduction générale.....	1
 <i>Chapitre 1 : Mise au point de quantification scalaire et de quantification vectorielle</i>	
Introduction	4
La quantification scalaire	4
Principe de la quantification scalaire.....	5
Le quantificateur scalaire optimal	5
L'algorithme de Lloyd-Max	6
La quantification vectorielle.....	6
Principe de la quantification vectorielle	6
Conception d un quantificateur optimal	8
Algorithme LBG	9
Méthodes d'initialisation Du Dictionnaire.....	11
Méthode aléatoire	11
Un algorithme a seuil.....	11
Méthode dichotomie vectorielle	11
1.11 Conclusion.....	12
 <i>Chapitre 2 : Mise au point de codeurs temporels : MIC, MICD, MICDA et DELTA</i>	
Introduction	15
Le codeur MIC (modulation par impulsion codée)	15
L'échantillonnage	16
Principe	16
Aspect temporel	16
Aspect fréquentiel	17
Théorème de Shannon	17
La quantification.....	17
la quantification uniforme	18

La quantification non uniforme	19
Le codage	19
Le MIC pour la téléphonie	20
Caractéristique de la compression selon la loi A	20
Comparaison avec la loi μ	21
Le codage utilisé dans le MIC	21
Modulation d'impulsions codées différentielles (MICD)	22
Principe de codeur MICDA	23
Conversion du format MIC d'entrée	23
Calcul du signal de différence	23
Quantificateur adaptatif	23
Quantificateur adaptatif inverse	23
Adaptation du facteur d'échelle	23
Contrôle de la vitesse d'adaptation	24
Prédicteur adaptatif et calcul du signal reconstitué	24
Détecteur de tonalité et de transition	24
Modulation delta	24
Conclusion	25
<i>Chapitre 3 : Mise au point d'un ensemble de codeurs fonctionnent à différentes débits</i>	
Introduction	27
Codage prédictif linéaire	27
Présentation de logiciel MATLAB	28
Structure du logiciel	28
Résultats de simulation	29
Codeur PCM	29
Codeur DPCM (MICD)	31
Codeur ADPCM	32
Codeur LPC	33
Conclusion	40
Bibliographie	42

REMERCIEMENTS

Nous commençons par remercier Allah tout puissant de nous avoir donné la volonté, l'amour du savoir et surtout le courage et la patience pour effectuer ce modeste travail

Nous exprimons notre profonde gratitude à notre promoteur Mr Mohammed SAIDI d'avoir accepté de nous encadrer pour notre projet de fin d'études, ainsi que pour son soutien, ses remarques pertinentes et son encouragement.

Nous remercions également les membres de jury pour l'intérêt qu'il ont bien voulu porter à notre travail en acceptant en acceptant de l'examiner et de le juger.

Enfin nos remerciements s'adressent plus particulièrement à nos familles, ainsi qu'à toutes les personnes qui ont participé de près ou de loin à l'élaboration de ce travail.



DEDICACE

Je dédie ce modeste travail à :

*Mes très chers parents qui m'ont aidé et soutenu tout au long
demes Études et mon succès :*

*A la source de mon bonheur, la flamme de mon cœur, celle qui
s'estToujours sacrifié pour me voir réussir : Maman Hadda que
j'adore*

*A l'homme de ma vie, mon exemple éternel, mon soutien moral
etma Source de joie : Mon père Aissa*

*Mes chères frères Sami et Youcef et ma sœur Yousra pour leur
amour et leur présence
dans ma vie*

*A mes profs surtout à ceux de l'université de
Bouira et mes chers amis, en particulier*

« Abd el basset »

*A mon binôme Djamila avec qui j'ai partagé tous les moments
destress de fatigue*

YASMINE



DEDICACE

A la lumière de mes jours, la source de mes efforts, a ma mère djegdjiga, qui a œuvré pour ma réussite, par son amour, son soutien, tous les sacrifices consentis et ses précieux conseils, pour toute son assistance et sa présence dans ma vie, reçois à travers ce travail aussi modeste soit-il, l'expression de mes sentiments et de mon éternelle gratitude.

*A mon père Rabeh, A mes frères et sœurs
a mes neveux et nièces*

Merci de m'avoir soutenu et témoigné votre affection durant tout ce temps. Je vous aime

A toute ma famille, mes amis en particulier « Nabila ,Meriem », a chaque personne qui était la pour moi un jour

*A tous mes enseignants qui m'ont
marqués Je vous dédie ce travail*

DJAMILA

Liste des figures

<i>Figure</i>	<i>page</i>
: Quantification scalaire	4
: quantificateur scalaire	5
: Schéma présentant les régions de voronoï associées aux vecteurs représentants	7
: Schéma général d'un quantificateur vectoriel	8
: les étapes du codage MIC	16
: l'échantillonnage d'un signal analogique.....	16
: spectre d'un signal analogique.....	17
: spectre d'un signal analogique échantillonné	17
: quantification linéaire sur 4 bits [PFE]	18
: quantification non linéaire sur 4 bits.....	19
: tracé des deux lois de quantification logarithmique.....	21
: schéma de codeur MICD.....	22
: schéma de codeur MICDA	24
: modulateur DELTA	25
: principe du codage LPC	28
: fenêtre principal de logiciel.....	28
: fenêtre de programme principale.....	29
: signaux (original, reconstitué, erreur) de codeur PCM à 4 bits (A-law).....	29
: signaux (original, reconstitué, erreur) de codeur PCM à 4 bits (Mu-law)	30
: signaux (original, reconstitué, erreur) de codeur DPCM à 4 bits.....	31
: signaux (original, reconstitué et erreur) de codeur ADPCM à 4 bits.....	32
: Forme d'onde de 'a.wav' et sa forme reconstituée pour P=4.....	33
: Forme d'onde de 'a.wav' et sa forme reconstituée pour P=10	33
: Forme d'onde de 's.wav' et sa forme reconstituée pour P=4.....	33
: Forme d'onde de 's.wav' et sa forme reconstituée pour P=10.....	33

Liste des figures

: Spectre de l'enveloppe du son <i>a.wav</i> (en bleu) et leur modèle (en rouge)	34
: Spectre de l'enveloppe du son <i>s.wav</i> (en bleu) et leur modèle (en rouge).....	34
: Codage LPC d'un signal voisé « <i>a.wav</i> »	35
: Codage LPC du signal <i>a.wav</i> pondéré par une fenêtre rectangulaire avec chevauchement	36
: Codage LPC du signal <i>a.wav</i> pondéré par une fenêtre de Hamming.....	36
: Codage LPC de „ <i>a.wav</i> ’ pondéré par une fenêtre de Hamming avec chevauchement	36
: Spectre de la 3 ^{ème} trame du signal ‘ <i>a.wav</i> ’ et le spectre du modèle	37
: Codage LPC d'un signal non voisé « <i>s.wav</i> »	38
: Codage LPC du signal <i>a.wav</i> pondéré par une fenêtre rectangulaire avec chevauchement	38
: Codage LPC du signal <i>a.wav</i> pondéré par une fenêtre de Hamming.....	38
: Codage LPC de „ <i>a.wav</i> ’ pondéré par une fenêtre de Hamming avec chevauchement	39
: Codage LPC du signal non voisé « <i>s.wav</i> » pour $P = 10$	39
: Codage LPC d'un signal de parole « <i>phrase_f.wav</i> » pour $P = 10$	40

Liste des tableaux

<i>Tableau</i>	<i>page</i>
Tableau 1.1 : Algorithme LBG.....	10
Tableau 1.2 : Méthode dichotomie vectorielle	12
Tableau 2.1: représentation binaire sur 3 bits.....	20
Tableau 3.1 : distorsion du codeur MIC pour différents ordre de nombre de bits	30
Tableau 3.2 : distorsion du codeur MICD pour différents nombre de bits.....	31
Tableau 3.3 : distorsion du codeur MICDA pour différents nombre de bits.....	32
Tableau 3.4 : RSB du codeur LPC pour différentes fenêtres de pondération et différentes ordres P du filtre.....	37
Tableau 3.5 : RSB du codeur LPC pour différentes fenêtres de pondération et différentes ordres P du filtre.....	40

Liste d'abréviations

A

ADPCM : Adaptive différentiel pulse code modulation

APC : adaptive predictive coding

D

DAC : Digital analog converter

DM : Delta modulation

DPCM : Différentiel pulse code modulation

DVD : Digital video Disk

F

FSK : Frequency shift keying

H

HD : High definition

I

ISDN : Integrated services digital network

L

LBG : Linde-Buzo-Gray

LPC : Linear predictive coding

LPCM : Linear pulse code modulation

N

NRZ : No return to zero

Liste d'abréviations

P

PCM : Pulse code modulation

R

RLL : Run-length limited

RZ : Return to zero

S

SNR : Signal-to-noise ratio

SQ : Scalar quantization

VQ : Vector quantization

STN : Switch telephone network

V

VoIP : Voice over IP

W

WAV : Broadcast Wave Format



Introduction
générale

Introduction générale

Parmi les moyens de communication d'un système, la parole se révèle l'un des plus intéressants. Du point de vue de l'utilisateur et du système, l'usage de la parole avec celui naturellement associé de la langue naturelle facilite la manipulation des informations fournies ou reçues du système. D'autre part, la parole s'est avérée être le plus performant des médias lors de nombreuses expériences effectuées en laboratoire ou en situation réelle.

Aujourd'hui les communications s'intéressent aux moyens de transmettre une information depuis la source jusqu'à un utilisateur à travers un canal. La nature de la source peut être très variée. Il peut s'agir par exemple d'une voix, d'un signal électromagnétique ou d'une séquence de symboles binaires, canal peut être une ligne téléphonique, une liaison radio, un support magnétique ou optique. La transmission peut se faire dans l'espace ou dans le temps. Le codeur représente l'ensemble des opérations effectuées sur la sortie de la source avant la transmission. Ces opérations peuvent être par exemple la modulation, la compression, le brouillage, l'ajout de redondance pour combattre les effets du bruit, ou encore l'adaptation à des contraintes de spectre. Elles ont pour but de rendre la sortie de la source compatible avec le canal. Enfin le décodeur doit être capable, à partir de la sortie du canal, de restituer de façon acceptable l'information fournie par la source. Pour que cette transmission soit établie il faut tout d'abord convertir le signal analogique signal audio vers un signal numérique passant par plusieurs étapes en site la compression pour réduire le flux de données nécessaires à une reconstruction de bonne qualité du signal de parole d'origine. Le terme numérisation correspond à l'opération d'échantillonnage du signal de parole continu suivi d'une quantification, ce qui représente respectivement une discrétisation des axes des temps et des amplitudes

Dans notre projet de fin d'étude on a décomposé notre travail sur 3 chapitres :

Le premier chapitre on a entamé le sujet d'une façon générale dans lequel on a parlé sur le codage de parole la quantification scalaire et optimale qui fait partie de la quantification vectorielle sa procédure et plus profondément sur la quantification vectorielle.

Dans ce chapitre on précise quelques algorithmes utilisés tels que l'algorithme LBG et les méthodes d'initialisation au de là on a fait une interprétation de résultats de comparaison entre les deux types de quantification scalaire et vectorielle.

Pour que la parole soit transmise doit passer sur la partie de codage utilisant plusieurs types de codeurs qu'on a mentionnés dans le deuxième chapitre à titre d'exemple le

Introduction générale

codeur MIC qui passant vers le MIC pour la téléphonie ajoutant a ces méthodes on trouve le MICD, MICDA et la modulation DELTA. Dans cette partie on traiter profondément sur chaque type de codeur commençant par le principe ,la quantification , échantillonnage, le codage et la compression pour la MIC suivant on a mentionner les caractéristiques de loi a et les comparer par la loi u pour la MIC de la téléphonie, pour le MICD et MICDA on a discuter sur son principe , quantification adaptative et prédiction adaptative on a passe a modulation DELTA ses effets de saturation de pente en modulation delta et la modulation delta adaptative on a terminer avec une interprétation et résultat .

Dans le chapitre 3 on a présenté la partie pratique utilisant logiciel de simulation MATLAB. Dans cette partie on a fait une simulation sur plusieurs type de codeurs a fin d'avoir le meilleur type pour l'obtention de bonne qualité de codage de la parole.

Enfin, nous terminerons notre mémoire par une conclusion générale sur ce travail et nous présenterons ses perspectives.

Chapitre 1

*Mise au point de quantification scalaire
(QS) et de quantification vectorielle (QV)*

Introduction

En général, la quantification apparaît dans un processus de compression, pour réduire la quantité d'information, de façon souvent irréversible, car elle introduit une distorsion et il faudra minimiser cette dernière lors de la conception d'un quantificateur. Dans ce chapitre nous décrivons brièvement les deux types de quantification suivis par ces principes, puis on a parlé sur l'algorithme de Lloyd-Max et l'algorithme de LBG pour l'élaboration de dictionnaire.

La quantification scalaire

La quantification scalaire étant un cas particulier de la quantification vectorielle où la dimension des vecteurs est égale à un [1].

Soit $s(n)$ le paramètre à quantifier. Notons $f_s(s)$ la densité de probabilité de la variable $s(n)$ et b sa résolution ou le nombre de bits nécessaires pour la représenter. L'opération de quantification scalaire consiste à discrétiser le paramètre s pour obtenir une représentation numérique $i(n)$ de l'information qu'il représente. Son domaine de définition est alors partitionné en $L=2^b$ intervalles distincts et un représentant est défini pour chacun de ces intervalles est fixe [2].

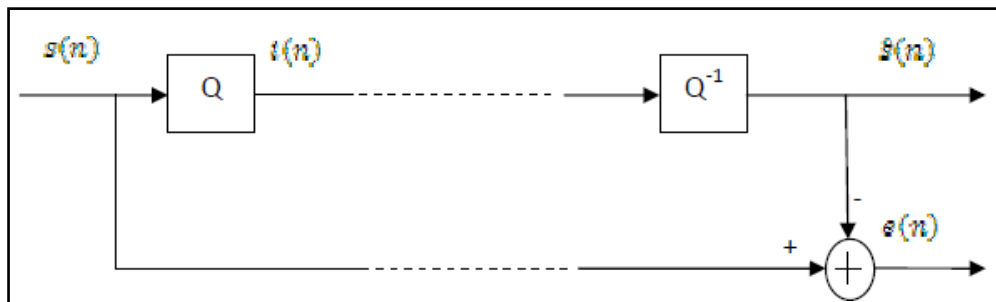


Figure 1.1: Quantification scalaire [2].

Le procédé d'encodage Q décide à quel intervalle appartient le paramètre $s(n)$ et lui associe le numéro $i(n) \in \{1 \dots L\}$ correspondant (Fig.). C'est ce numéro d'intervalle qu'il faudra transmettre au $\hat{s}(n)$ processus de décodage Q^{-1} qui effectuera la procédure inverse en associant à $i(n)$ son représentant). L'opération de quantification apportera toujours des dégradations irréversibles par rapport au signal d'origine qui se traduisent par une erreur, ou bruit, de quantification $e(n)$:

$$e(n) = s(n) - \hat{s}(n) \quad (1.1)$$

Où $\{\hat{s}(n)\}_{1 \leq i \leq L}$ est l'ensemble de représentants, appelé dictionnaire [2].

Principe de la quantification scalaire

La quantification scalaire dans le cas général passe sur 3 étapes suivantes :

1. Diviser l'intervalle d'entrées $[a, b]$ en N_d sous-intervalles $\{[d_i, d_{i+1}[], i \in \{0 \dots N_d - 1\}\}$.
 d_i =seuils de décision.
2. Associer à chaque sous-intervalle $[d_i, d_{i+1}[$, une valeur r_i (niveau de reconstruction).
3. On code une donnée d de S par r_i si $d \in [d_i, d_{i+1}[$. En pratique, on envoie l'indice de l'intervalle. Le décodeur doit connaître le niveau de reconstruction.

La quantification scalaire est dite uniforme sinon est dite non uniforme.

On peut définir un quantificateur scalaire optimal si on connaît la statistique de la source.

Pour cela, on cherche l'ensemble [seuil de décision, niveau de reconstruction] qui minimise l'erreur quadratique moyenne du signal reconstruit [3].

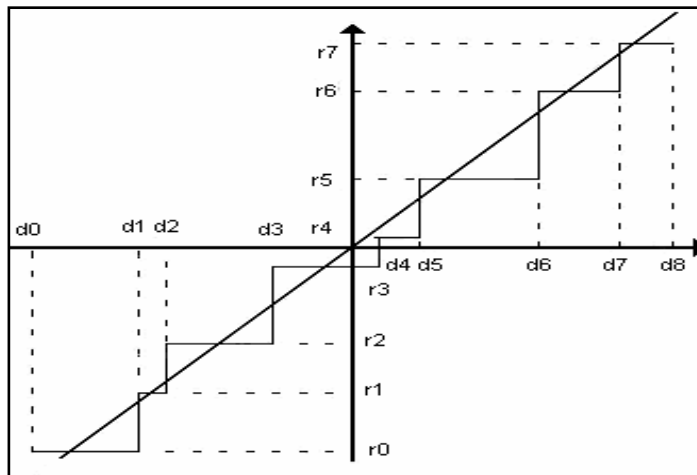


Figure 1.2 : quantificateur scalaire [3].

Le quantificateur scalaire optimal

Le quantificateur scalaire optimal est celui qui minimise, pour une source donnée et un débit maximal fixé, l'erreur moyenne de reconstruction due aux bruits de quantification et de surcharge. Les niveaux de reconstruction sont donc répartis en tenant compte de la densité de probabilité de la variable à quantifier. La concentration des niveaux de reconstruction est plus importante dans la zone de l'espace où la densité de probabilité des valeurs à quantifier est plus élevée. Dans la pratique on ne connaît pas la densité de probabilité des valeurs à quantifier. On utilise donc une base de données d'apprentissage composée d'un grand nombre d'échantillons représentatifs de la source. Pour construire un quantificateur optimal, on utilise l'algorithme (très connu) de Lloyd-Max [4, 5] sur la base de données d'apprentissage.

L'algorithme de Lloyd-Max

L'algorithme de Lloyd-Max est un algorithme utilisé pour construire un quantificateur scalaire optimal, il s'agit donc une méthode pour quantifier un signal en une dimension pour réduire la distorsion mesurée par l'erreur quadratique moyenne.

Les travaux de Lloyd et de Max [4,5] ont permis d'établir les deux conditions nécessaires d'optimalité pour construire un quantificateur adapté à des sources quelconques. Pour caractériser le quantificateur scalaire optimal il faut trouver les meilleures partitions P_i du signal et les meilleurs représentants y_i associés (les représentants du dictionnaire C) afin de minimiser la distorsion, cette minimisation conjointe n'avoir pas de solution simple, les deux conditions d'optimalités sont données comme suit :

- 1- connaissant le dictionnaire, il est possible de trouver le meilleur partitionnement, étant donné un dictionnaire $\{y_0, \dots, y_{L-1}\}$, on peut classer les valeurs de x dans chaque partition P_i , c'est la règle dite **du plus proche voisin** soit vérifiée [6]:

$$P_i = \{x \mid d(x, y_i) \leq d(x, y_j) \forall i \neq j, j \in \{0, \dots, L-1\}\} \quad (1.2)$$

- 2- connaissant le partitionnement, il est possible de trouver le meilleur dictionnaire, étant donné un partitionnement $P = \{P_0, \dots, P_{L-1}\}$, on peut trouver les meilleures représentants y_i pour chaque partition P_i sont obtenus par la condition dite de **centroïde ou centre de gravité** donnée comme suite [6]:

$$y_i = E[x \mid x \in P_i] = \frac{\int_{x \in P_i} x f_x(x) dx}{\int_{x \in P_i} f_x(x) dx} \quad (1.3)$$

La quantification vectorielle

Par extension à la quantification scalaire qui traite chaque échantillon indépendamment des précédents, le quantificateur vectoriel permet de prendre en compte la dépendance entre les différentes composantes du signal [7]. La propriété fondamentale de ce quantificateur est la recherche de la corrélation qui peut exister entre les échantillons successifs du signal [8].

Principe de la quantification vectorielle

La quantification vectorielle VQ (vector quantization) consiste à représenter tout vecteur x de dimension k par un vecteur y de même dimension appartenant à un ensemble fini appelé dictionnaire Y ou les catalogue des formes [9].

Les vecteurs $y_i \in Y$ sont appelés les vecteurs représentants, les vecteurs de reproduction ou code-vecteurs.

Nous pouvons être définir un quantificateur vectoriel de dimension k et de taille N comme une application Q de R^k vers le dictionnaire Y de R^k :

$$Q : R^k \rightarrow Y , \quad (1.4)$$

$$x \rightarrow Q(x) = y_i$$

avec $Y = \{y_0, y_1, \dots, y_{N-1}\}$ et $y_i \in R^k$ pour $0 \leq i < N$. Cette application Q détermine implicitement une partition de l'espace source R^k en N régions R_i , $i = 0, \dots, N-1$ ou la région R_i , $i = 0, \dots, N-1$ ou la région est associées au vecteur y_i . Ces régions sont appelées classes, cellules ou régions de voronoï sont déterminées par :

$$R_i = \{x \in R^k \mid Q(x) = y_i \text{ si } (d(x, y_i) \leq (d(x, y_j), \forall j \neq i) \} \quad (1.5)$$

$$\text{Avec : } R_i \cap R_j = \emptyset, i \neq j, \forall i, j = 0, 1, \dots, N-1$$

En général, le nombre de vecteurs-code du dictionnaire Y ou la taille du dictionnaire est de la forme $N=2^n$, avec n le nombre de bits utilisé pour représenter chaque indice binaire d'un vecteur-code y_i , et l'indice binaire i écrit comme un entier positif [1].

La figure suivante (figure 1.3) monte un ensemble des vecteurs dans l'espace, A chaque groupe de vecteurs est associé un vecteur de code représentatif. Chaque vecteur de code réside dans sa propre région Voronoï.

Ces régions sont séparées avec des lignes imaginaires. Etant donné un vecteur d'entrée, le vecteur de code qui est choisi pour le représenter est celui dans la même région Voronoï [3].

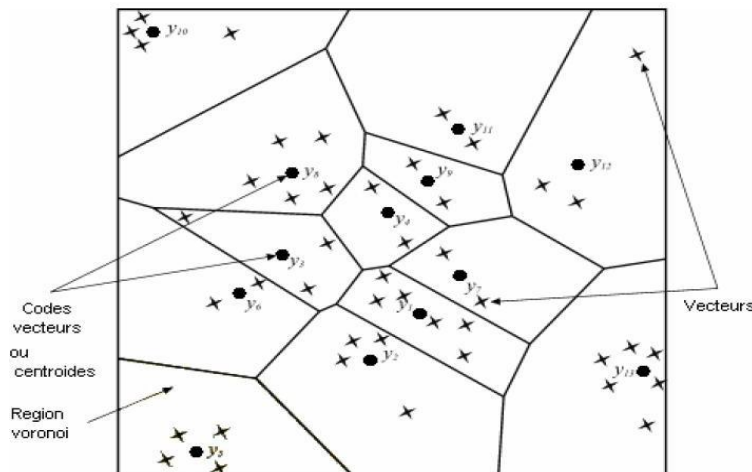


Figure 1.3 : Schéma présentant les régions de voronoï associées aux vecteurs représentants [3].

Généralement, un quantificateur vectoriel se décompose par deux fonctions : l'encodeur et le décodeur

a) l'encodeur

La procédure d'encodage consiste, pour tout vecteur x du signal d'entrée, à rechercher dans le dictionnaire Y le vecteur-code y_i « le plus proche » de x . La notion de proximité est modélisée dans notre mise œuvre par la distance de la Euclidienne entre les deux vecteurs x et y_i :

$$D(x, y_i) = \sum_{j=0}^{k-1} (x_j - y_{i,j})^2 \quad (1.6)$$

C'est juste l'adresse du code-vecteur y_i ainsi sélectionné qui sera transmise, c'est donc à ce niveau que s'effectue la compression. Le dictionnaire est tout simplement représenté par un tableau et l'adresse du code-vecteur correspond à son indice dans le tableau [6].

b) Le décodeur

Le décodeur est considéré comme un récepteur chargé de la reconstruction du signal, pour cela, le décodeur dispose d'une copie identique du dictionnaire qu'il consulte afin de restituer le code vecteur source correspondant à indice reçue. Donc le décodeur réalise l'opération de décompression [1].

Un quantificateur vectoriel peut est représenté par la figure (1.4) cette figure offre la combinaison des opérations d'encodage et de décodage.

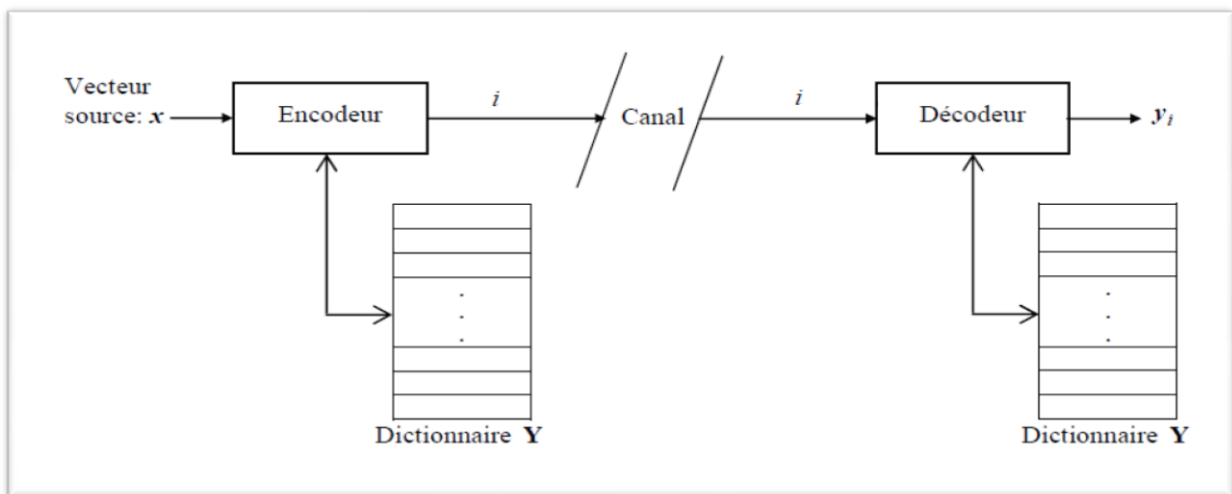


Figure 1.4 : Schéma général d'un quantificateur vectoriel [1].

Conception d un quantificateur optimal

Le mot quantifier est répartir dans un espace de dimension fixée un nombre détermine de représentants, ce nombre étant fonction du débit alloué au quantificateur.

Pour une distribution statistique donnée de la source et un débit donné, le quantificateur optimal est celui qui minimise la distorsion moyenne [10] :

$$D = E[d(X, Q(X))] = \int_{x \in R^k} d(x, Q(x)) f_x(x) dx \quad (1.7)$$

Où X est un vecteur source aléatoire dans R^k , $f_x(x)$ est la fonction densité de probabilité (pdf). Pour un quantificateur vectoriel donné un dictionnaire $Y = \{y_i, i = 0 \text{ à } N - 1\}$ et pour partition les régions de voronoi $R_i, i=0 \text{ à } N-1$, donc la distorsion est donnée comme suite :

$$D = \sum_{i=0}^{N-1} \int_{R_i} d(x, y_i) f_x(x) dx \quad / \quad x \in R_i \quad (1.8)$$

La recherche de l'optimum global est très compliqué (cela consiste à trouver en même temps la meilleur partition et les meilleurs vecteurs-code qui minimisent la distorsion moyenne D). Dans la pratique la conception d'un quantificateur optimal est réalisée de façon répétitive en vérifiant les deux conditions d'optimalités suivantes [10] :

1. Pour un dictionnaire fixé (l'encodeur optimal) $Y = \{y_0, y_1, \dots, y_{N-1}\}$, la partition optimal est décrite par les régions de voronoi :

$$R_i = \{x \in R^k \mid Q(x) = y_i \text{ si } (d(x, y_i) \leq (d(x, y_j), \forall j \neq i) \quad (1.9)$$

C'est la règle du plus proche voisin.

2. Pour une partition donnée (décodeur optimal) le vecteur représentant y_i doit minimiser la distance à la région de voronoi R_i , le vecteur y_i est donc le centroïde de cette cellule $y_i = Cent(R_i)$.

L'élaboration du dictionnaire d'un quantificateur vectoriel se fait à partir d'une séquence d'apprentissage. La méthode la plus populaire est l'algorithme de Lloyd-Max généralisé, appelé aussi algorithme LBG [9].

Algorithme LBG

L'algorithme LBG proposé par Linde, Buzo, Gray [9], correspond à une extension de l'algorithme de Lloyd-Max utilisé pour l'élaboration de dictionnaires dans le cas de quantification scalaire [4, 5, 10] cet algorithme est une application des deux conditions d'optimalité et le dictionnaire met à jour itérativement le rôle de l'algorithme est pour un dictionnaire initial donné d'essayer d'optimiser le codeur et le décodeur [11, 12]. L'algorithme est donné au tableau 1.1 [1].

Déterminez la taille de dictionnaire N (le nombre de vecteur-code).

Sélectionner N vecteurs-code aléatoires et laissez-le être le dictionnaire initial. Les vecteurs code initiaux peuvent être aléatoirement choisis du jeu de vecteurs d'entrée.

Fixer : - Une séquence d'apprentissage $x_i : i=0, 1, \dots, N_t - 1$.

-Un dictionnaire initial $Y^{(0)}$ de taille N .

-Un seuil $a > 0$.

-Un nombre maximum d'itération p .

Etape 01 : initialisation

- $D^{(-1)} = \infty$.

- $m = 0$.

Etape 02 : classification

L'utilisation de la distance Euclidien groupe les vecteurs autour de chaque vecteur-code.

Ceci est fait en prenant chaque vecteur d'entrée et en essaye de trouver la distance Euclidien entre ce dernier et chaque vecteur-code. Le vecteur d'entrée appartient au groupe du vecteur-code qui rapport la distance minimal.

pour le dictionnaire courant : $Y^{(m)} = \{y_i^{(m)}, i=0, 1, \dots, N-1\}$, trouver la partition optimale $R^{(m)} = \{R_i^{(m)}\} \{i=1, \dots, N\}$ qui minimise la distorsion moyenne soit :

$$x \in R_i^{(m)} \text{ si } d(x, y_i^{(m)}) < d(x, y_j^{(m)}), \forall j \neq i$$

calcul de la distorsion moyenne :

$$D^{(m)} = \sum_{i=0}^{N-1} [\sum_{x \in R_i^{(m)}} d(x, y_i^{(m)})]$$

Etape 03 : test de fin :

- Si $[\frac{D^{(m-1)} - D^{(m)}}{D^{(m)}} \leq a]$ ou $m=p$

-Stopper : le QV final est décrit par la partition $R^{(m)}$ et le dictionnaire $Y^{(m)}$ (arrêter le traitement).

- Sinon

-Continuer

Etape 04 : réactualiser le dictionnaire

- Pour la partition $R^{(m)}$, calculer le dictionnaire optimal

$$Y^{(m+1)} = \{y_i^{(m+1)}, i = 0, 1, \dots, N - 1\}$$

Avec $y_i^{(m+1)} = \text{cent}(R_i^{(m)})$

- $M \leftarrow m+1$

- Retourne à **l'étape 02**

Tableau 1.1 : Algorithme LBG

Cet algorithme d'optimisation interactif travaille à partir d'un dictionnaire initial. Donc le problème c'est le choix de ce dictionnaire initial ce qui a été prouvé l'expérience pour contribuer fortement à la performance de l'algorithme LBG.

En effet chaque itération ne provoquant qu'un changement local de dictionnaire, l'algorithme converge vers le minimum local le plus proche de dictionnaire initial.

Méthodes d'initialisation Du Dictionnaire

Un certain nombre de méthodes ont été proposées pour l'initialisation du dictionnaire :

Méthode aléatoire : le dictionnaire le plus simple est celui qui contient les premiers vecteurs de la suite d'apprentissage ou le vecteur extraits aléatoirement de cette suite ces vecteurs peuvent ne peuvent pas être représentatifs de la séquence d'apprentissage et conduire à des résultats médiocre [6].

Un algorithme à seuil : on fixe une distance minimale entre les éléments au lieu de prendre le vecteur aléatoirement cette méthode permet d'obtenir une meilleure représentativité que dans ces précédents mais pas toujours satisfaisante, le seuil est souvent difficile à déterminer puisque il dépend de la complexité de la séquence d'apprentissage.

Méthode dichotomie vectorielle : (méthode de division récursive) : cette méthode a été proposée dans la version initial de l'algorithme LBG sous le nom de "splitting" [13], nous procédons comme suit:

- Le premier dictionnaire contient un seul vecteur-code y_0 , qui est le centroïde de toute la séquence d'apprentissage, le vecteur-code est calculé.
- Ensuite, le vecteur résultant est divisé en deux vecteurs-code $y_0 - \varepsilon$ et $y_0 + \varepsilon$, avec ε : est un vecteur de perturbation fixe. L'algorithme LBG est ensuite appliqué sur ce dictionnaire (de deux vecteurs) pour construire un dictionnaire optimal de taille 2.
- Le processus de fractionnement est appliqué à chaque vecteur-code du dictionnaire de taille 2. Ceci génère un dictionnaire de taille 4 et l'algorithme LBG est appliqué pour optimiser ce dictionnaire. Le processus est répété jusqu'à obtenir un dictionnaire de taille désirée N .

Un débit $R = \log_2(N)$ entier est alors exigé, dans ce cas succession de QV ayant de débit croissants de 0 à R est générée [3].

Le procédé de construction est donné au tableau [1.2] suivante :

Etape 01: initialisation

- Initialisation le dictionnaire Y en prenant $Y[0] = y_0$ ou y_0 est le centroïde de la séquence d'apprentissage pris dans son ensemble.
- Noter U le vecteur de R^k ayant toutes ses composantes égale à 1.
- Fixer une perturbation scalaire ε .
- Fixer le débit R.
- Fixer un seuil de distorsion ∇ .
- Prendre $r=0$.

Etape 02 :

Perturbation le dictionnaire courant $Y = \{y_i\} \{i = 0, \dots, 2^r - 1\}$

$$\forall i = 0, \dots, 2^r - 1 \left\{ \begin{array}{l} Y[2i + 1] = Y[i] - \varepsilon U \\ Y[2i] = Y[i] + \varepsilon U \end{array} \right\}$$

Incrémenter r.

Etape 03 :

- Considérer le nouveau dictionnaire $Y = \{y_i\} \{i = 0, \dots, 2^r - 1\}$.
- Exécuter l'algorithme de LBG sur le dictionnaire Y.
- Si $(r|k = R$ ou $\Delta^{(r)} < \nabla)$.
 - Arrêter en retenant $y^{(r)}$.
- sinon.
- Retourner à l'étape 02.

Tableau 1.2: Méthode dichotomie vectorielle

Cet algorithme à été implémenté lors de la conception de notre quantification vectorielle. La charge en calculs engendrée par la construction du dictionnaire est très importante [6], mais n'est pas considérée comme un handicap prépondérant étant donné qu'un dictionnaire est construit une fois pour toutes.

1.11 Conclusion

L'opération de codage par quantification permet de réduire le nombre de bit nécessaires à la représentation d'un signal. Dans ce premier chapitre, nous avons présenté les deux types de scalaire (QS) et vectorielle (QV).

La première opérer sur des échantillons individuels, l'algorithme de Lloyd-max permet d'obtenir des niveaux de reproduction produisent une distance minimale, les

performances d'un quantificateur scalaire peuvent être améliorées en quantifiant des vecteurs plutôt que des scalaires [1].

La deuxième permet un codage à des débits (en bit par échantillon) fractionnaire, pour un quantificateur vectoriel utilisant un dictionnaire-non structuré de N vecteurs-code, l'algorithme LBG est appliqué pour l'élaboration du dictionnaire.

Le quantificateur scalaire non uniforme donne un bon résultat par rapport au quantificateur scalaire uniforme.

Chapitre 2

Mise au point de codeurs temporels :

MIC, MICD, MICDA et DELTA

Introduction

Le codage en formes d'onde peut être effectué dans deux domaines : temporel et fréquentiel, dans ce chapitre nous parlons que des codeurs temporels, ces derniers réalisent le processus de codage sur des échantillons temporels du signal, nous précisons en détail les méthodes de codage suivi par leurs principes qui sont : le codage PCM (pulse code modulation), le codage DPCM (différentiel pulse code modulation), le codage ADPCM (adaptive différentiel pulse code modulation), le codage DM (delta modulation).

Le codeur MIC (modulation par impulsion codée)

Principe

La modulation par impulsion et codage (MIC) ou pulse code modulation (PCM) est une méthode utilisée pour une représentation numérique d'un signal audio analogique [14]. Ce codage est réalisé un quantificateur de chaque échantillon du signal sur un certain nombre de bits, le nombre de quantification est une puissance de 2 de sorte que toutes les valeurs sont codées par un mot binaire [3]. Le traitement indépendant de chaque échantillon indique qu'il y'a pas de codage ou de compression de données.

Cette modulation est utilisée presque partout en télécommunication, en effet tous les téléphones numériques l'utilisent pour transmettre la voix (RTC ou VoIP) et le son [15].

Notamment pour les disques compact audio ; et l'enregistrement sur bandes DAT et Mini disc, les disques optiques à haute capacité (blu-ray et HD DVD), ainsi que pour les fichiers WAV et BWF standards [16]. Elle est implantée sur les cartes sons des ordinateurs avec une quantification sur 8 et 16 bits. C'est une numérisation brute, très fidèle au signal analogique, sans compression. Il est possible, par le biais des modulations numériques, de transmettre un signal analogique sous forme numérique et d'utiliser ainsi les mêmes canaux de transmission que les équipements de transmission de données. Le réseau Numérique à Intégration de Services (RNIS), est ainsi rendu possible. Il s'agit là d'une véritable révolution dans les télécommunications, dont toutes les implications n'ont pas encore été réalisées [17].

La modulation par impulsion codée (MIC) nécessite trois opérations données comme suite :

- L'échantillonnage
- La quantification
- Le codage

La figure (2.1) suivante représente les étapes du codage MIC

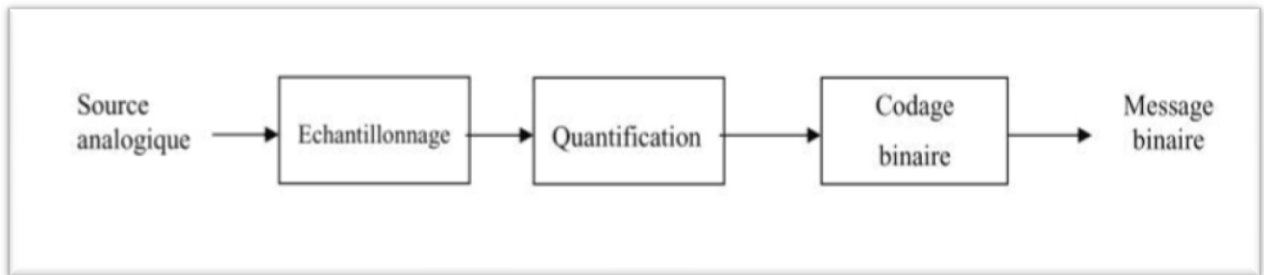


Figure 2.1 : les étapes du codage MIC [18].

L'échantillonnage

Principe

Soit un signal monodimensionnel $S(t)$. L'échantillonnage est l'opération qui consiste à prélever des valeurs du signal à intervalles de temps de façon régulière, le nombre d'échantillons par seconde est appelée fréquence d'échantillonnage F_e , la grandeur $F_e = \frac{1}{T_e}$, avec T_e la période d'échantillonnage. Cette opération consiste à obtenir un nuage de point (les échantillons) qu'il existe une seule courbe reliant tous ces point et ayant un contenu fréquentiel entre 0 Hz et la moitié de la fréquence d'échantillons appelée fréquence de Nyquist [19].

Aspect temporel

L'échantillonnage convertit les signaux analogiques $S(t)$, en une suite numérique $S(nT_e)_{n \in \mathbb{Z}}$. En général, nous notons les échantillons $S[n] = S(nT_e)$, la figure (2.2) représente l'échantillonnage d'un signal analogique [17].

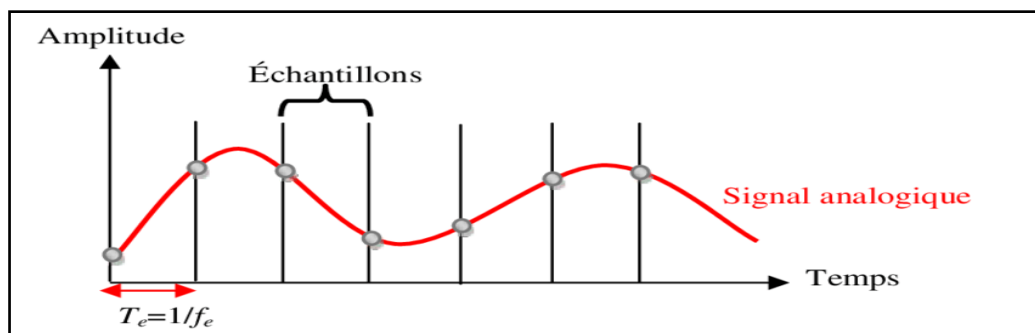


Figure 2.2: l'échantillonnage d'un signal analogique [17].

Aspect fréquentiel

Comme c'est montré sur la figure (2.3), l'échantillonnage d'un signal analogique duplique le spectre des fréquences de ce même signal avec une période F_e . Cette duplication du spectre ne pose pas de problème si le spectre du signal analogique est borné, c'est-à-dire si toute l'énergie du signal est comprise entre deux fréquences F_{min} et F_{max} comme sur la figure (2.4) [17].

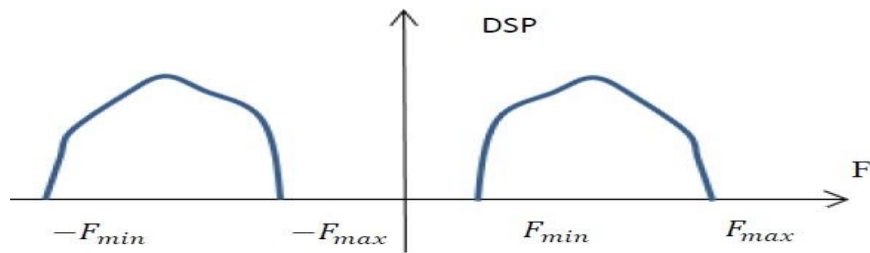


Figure 2.3 : spectre d'un signal analogique [17].

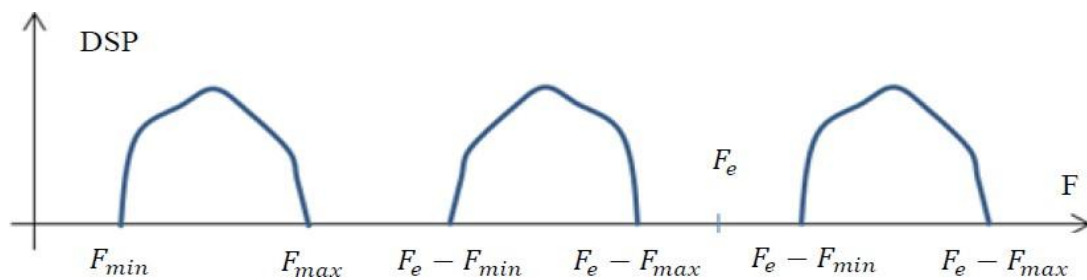


Figure 2.4 : spectre d'un signal analogique échantillonné [17].

Théorème de Shannon

Pour qu'il n'y ait pas interférence entre le spectre du signal analogique et le spectre du signal échantillonné, il faut que $F_e - F_{max} > F_{max}$, ce qui donne le théorème de Shannon : $F_e \geq 2 \times F_{max}$ (la fréquence d'échantillonnage doit être égale ou supérieure au double de la fréquence maximale), quand on respecte cette condition il n'y a pas de chevauchement de spectre et donc pas de dégradation de l'information. Pour assurer qu'un signal que l'on veut échantillonner possède un spectre borné, nous utilisons un filtre appelé filtre anti repliement.

La quantification

En principe chacun des échantillons prélevés peut prendre une infinité de valeurs du fait du signal analogique [20], l'opération de quantification consiste à remplacer la valeur exacte analogique de l'échantillon par la plus proche valeur approximative extraite d'un

ensemble fini. L'application qui réalise cette opération est la CAN (conversion analogique numérique).

Les éléments binaires étant en nombre fini de bits, avec N bits, nous ne disposerons donc que de 2^n combinaisons pour quantifier des échantillons qui peuvent prendre une infinité de valeurs [21].

La quantification a un effet arrondi l'amplitude de chaque échantillon à l'une de ces 2^n valeurs. Donc on peut dire que le nombre de bits de quantification détermine la précision en amplitude ou la dynamique de la conversion, tandis que la fréquence d'échantillonnage détermine la précision temporelle de la conversion [6].

la quantification uniforme

La quantification uniforme consiste à choisir les niveaux de quantification donc toute la gamme des échantillons sont quantifiable avec un pas de quantification constant, ce dernier on peut définir comme l'intervalle de tension entre deux valeurs numérique binaires successives. Pour le déterminer on divise l'intervalle de tension (V_{min} , V_{max}) par le nombre N de niveaux de quantification [17], la forme de pas de quantification et donne comme suite :

$$S_0 = \frac{(V_{max} - V_{min})}{2^N} \quad (2.1)$$

Pour répartir les niveaux de quantification par rapport aux niveaux de tension, nous traçons le graphe entre les échantillons et les niveaux de quantification la figure (2.5) montre un exemple de graphe pour une quantification sur 4 bits.

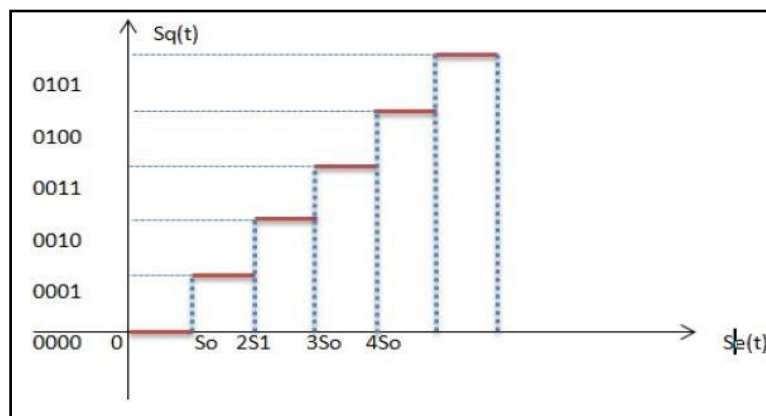


Figure 2.5 : quantification linéaire sur 4 bits [17].

Dans le cas d'une quantification uniforme à n bits, le rapport signal sur bruit de quantification est donné par $(\frac{S}{N})_n = 4.8 + 6n$, n est le nombre de bits.

La quantification non uniforme

Il faut utiliser un pas de quantification faible pour les petites amplitudes du signal et un pas de quantification plus important pour les grandes amplitudes pour obtenir un (SNR) constant quelle que soit l'amplitude du signal [17], la figure (2.6) montre un exemple de graphe pour une quantification sur 4 bits .

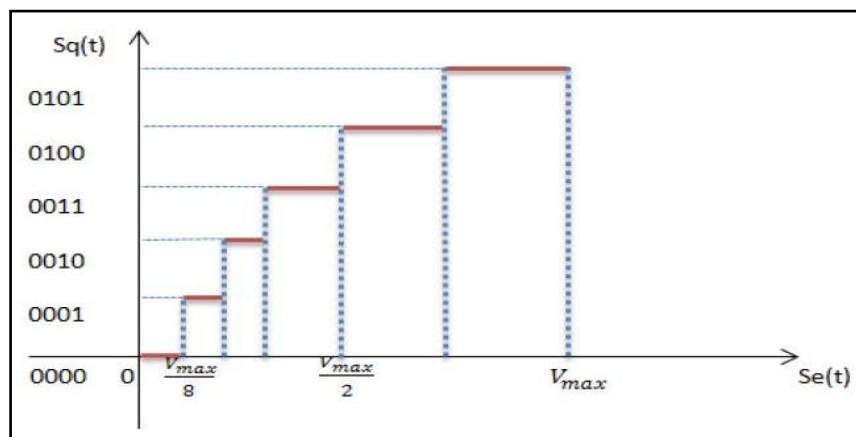


Figure 2.6 : quantification non linéaire sur 4 bits [17].

Les lois non linéaires qui sont utilisées essentiellement sont la loi A (utilisée en Europe) et la loi μ (utilisée en Japon, USA) [22]. On a parlé de ces lois en détail dans ce chapitre.

Le codage

Fait correspondre à un niveau de quantification donné déterminé par l'approximation de la valeur exacte de l'échantillon par la loi de quantification, une expression numérique généralement binaire. Il existe plusieurs façons d'établir la correspondance entre l'ensemble des amplitudes quantifiées et l'ensemble des nombres binaires qui doivent les représenter [17].

En général, les signaux codés ayant des amplitudes positives ou négatives, les représentations préférées sont celles qui retiennent l'information de signe. Les plus courants pour le codage sont les suivantes :

- Signe et valeur absolue.
- Complément à 1.

Nombre	Signe et valeur absolue	Complément à 1
+3	011	011
+2	010	010
+1	001	001
0	000	000
-1	101	110
-2	110	101
-3	111	100

Tableau 2.1: représentation binaire sur 3 bits.

Le codage MIC utilise 8 bits pour la représentation binaire. Donc le débit binaire sera :

$$D = 8 \times 8000 = 64000 \text{ bits/s} = 64 \text{ Kbits/s.}$$

Le MIC pour la téléphonie

Deux variantes (incluses dans la norme CCITT G.711) se sont répandues dans la téléphonie : la norme américaine (μ -law) et la norme européenne (A-law) utilisé dans le reste du monde et dans les communications internationales, elles sont toutes deux des variations d'une correspondance exponentielle [23].

Caractéristique de la compression selon la loi A

La loi A (A-law) est une fonction typique de la quantification non linéaire, utilisé principalement à des fins de compression pour les applications de traitement de la voix humaine aussi est un algorithme standardisé, défini dans le standard ITU-T G.711 [24], ce système est utilisé en Europe et qui est définie comme suit :

$$f(x) = \left\{ \begin{array}{l} \text{sgn}(x_n) \frac{A|x_n|}{1+\ln(A)} \text{ si } |x_n| \leq \frac{1}{A} \\ \text{sgn}(x_n) \frac{1+\ln(A|x_n|)}{1+\ln(A)} \text{ si } \frac{1}{A} \leq |x_n| \leq 1 \end{array} \right\} \quad (2.1)$$

sgn est la fonction signe de x.

x_n est la valeur normalisé de x (division par la valeur maximale) du signal x.

A est une constante qui détermine le rapport de la compression (en pratique on peut souvent $A=87,6$).

La fonction inverse de la loi A (expansion) est donnée par :

$$f^{-1}(y) = \text{sgn}(x) \left\{ \begin{array}{l} |y| \frac{1+\ln(A)}{A} \text{ si } |y| < \frac{1}{1+\ln(A)} \\ \frac{\exp(|y|(1+\ln(A)))-1}{A} \text{ si } \frac{1}{1+\ln(A)} \leq |y| \leq 1 \end{array} \right\} \quad (2.3)$$

Comparaison avec la loi μ

La loi μ (Mu-law) est un système de quantification logarithme d'un signal audio, il est principalement utilisé pour traiter la voix humaine, cette loi est utilisée aux USA et au Japon qui est donnée par l'expression suivante [25]:

$$y = \text{sgn}(x) \frac{\ln(1 + \mu \frac{|x|}{x_{\max}})}{\ln(1 + \mu)} \quad (2.4)$$

μ est une constante qui contrôle le rapport de la compression. Pour $\mu=255$, la loi μ est très proche de la loi A.

La loi A et μ sont très voisines mais la loi μ a un meilleur rapport signal à bruit pour les faibles niveaux [3].

La figure (2.6) représente les deux lois de quantification logarithmique A-law et Mu-law

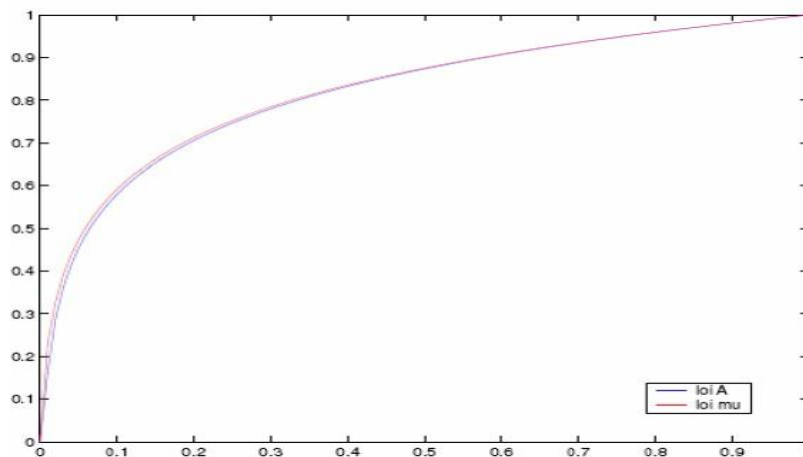


Figure 2.7 : tracé des deux lois de quantification logarithmique [3].

Le codage utilisé dans le MIC

La modulation d'impulsion codée peut utiliser le codage retour à zéro (RZ) ou non retour à zéro (NRZ), pour qu'un système NRZ soit synchronisé, on utilise uniquement l'information émise, il ne doit pas y avoir de longues séquences de symboles identiques,

Principe de codeur MICDA (MIC différentiel adaptatif)

Conversion du format MIC d'entrée

Ce bloc convertit le signal d'entrée $s(k)$ (signal MIC selon la loi A ou la loi μ en un signal MIC Uniforme $s_l(k)$ [26].

Calcul du signal de différence

Ce bloc calcule le signal de différence $d(k)$ d'après le signal MIC uniforme et le signal $s_l(k)$ estimé $s_e(k)$:

$$d(k) = s_l(k) - s_e(k) \quad (2.5)$$

Quantificateur adaptatif

On utilise un quantificateur adaptatif non uniforme à 31, 15, 7 ou 4 niveaux pour quantifier le signal de différence $d(k)$ en vue d'un fonctionnement à 40, 32, 24 ou 16 kbit/s respectivement. Avant la quantification, $d(k)$ est converti en une représentation logarithmique en base 2 avec le facteur d'échelle $y(k)$, qui est calculé au moyen du bloc d'adaptation du facteur d'échelle [26].

Quantificateur adaptatif inverse

Une version quantifiée $d_q(k)$ du signal de différence est obtenue en utilisant le facteur d'échelle $y(k)$ pour les valeurs spécifiques de la caractéristique de quantification normalisé.

Adaptation du facteur d'échelle

Ce bloc calcule $y(k)$, facteur d'échelle du quantificateur et du quantificateur inverse. Les autres correspondantes sont le signal de sortie du quantificateur à 5,4,3,2 bits $I(k)$ et le paramètre de contrôle de la vitesse d'adaptation $a_1(k)$

Le principe de base appliqué pour déterminer le facteur d'échelle du quantificateur est l'adaptation bimodale :

- Rapide, pour les signaux qui produisent des signaux de différence à grandes fluctuations (par ex : signaux vocaux).
- Lente, pour les signaux qui produisent des signaux de différence à faibles fluctuations (par ex : données transmises dans la bande vocale, tonalités).

La vitesse d'adaptation est déterminée par une combinaison de facteurs d'échelle à adaptation lente et rapide [26].

Le facteur d'échelle (non verrouillé) à adaptation rapide $y_u(k)$ se calcule par récurrence dans le domaine logarithmique en base 2, à partir du facteur d'échelle logarithmique $y(k)$:

$$y_u(k) = (1 - 2^{-5})y(k) + 2^{-5} W[I_c(k)] \quad (2.2)$$

Contrôle de la vitesse d'adaptation

Le paramètre de contrôle $al(k)$ peut prendre une valeur comprise entre 0 et 1. Sa valeur tend vers l'unité pour les signaux vocaux et vers zéro pour les signaux de données transmis dans la bande vocale. Il se déduit de la mesure du taux de variation du signal de différence [26].

Prédicteur adaptatif et calcul du signal reconstitué

Le prédicteur adaptatif a pour fonction essentielle de calculer la valeur estimée du signal $se(k)$ à partir du signal de différence quantifié $dq(k)$. On a recours à deux structures de prédicteur adaptatif: une structure du sixième ordre qui simule des zéros et une structure du deuxième ordre qui simule des pôles modélisant le signal d'entrée. Cette double structure permet de traiter efficacement les divers types de signaux que l'on peut rencontrer [26].

Détecteur de tonalité et de transition

Pour améliorer le fonctionnement dans le cas de signaux provenant de modems à modulation par déplacement de fréquence (MDF) en mode caractère, un processus de détection en deux temps a été défini. Tout d'abord, la détection du signal à bande étroite (tonalité, par exemple) provoque le passage du quantificateur en mode d'adaptation rapide. De plus, on définit une transition de signal à bande étroite telle que les coefficients du prédicteur puissent être mis à zéro, et que le quantificateur puisse être forcé au mode d'adaptation rapide [26].

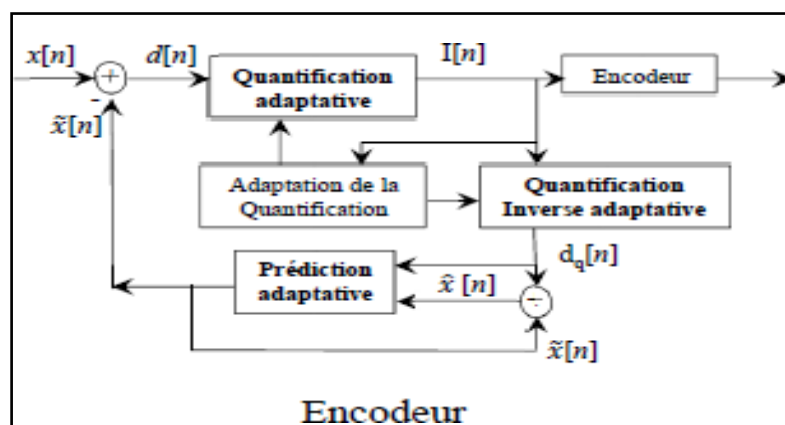


Figure 2.9 : schéma de codeur MICDA [26].

Modulation delta

La modulation différentielle (ou delta) d'impulsion-code (DPCM), à l'inverse de la modulation linéaire LPCM, code les valeurs PCM comme différences entre la valeur courante et la valeur précédente. Pour l'audio, ce type de codage réduit le nombre de bits exigé de 25 % environ par rapport au PCM [18].

Comme la DPCM mais la table de quantification à seulement 2 valeurs :

$$d_q(n) = \begin{cases} +A & d_p(n) \geq 0, \text{ bit de sortie} = 1 \\ -A & d_p(n) < 0, \text{ bit de sortie} = 0 \end{cases}$$

– On peut rendre la valeur de A adaptative:

- Longue séquence de valeurs identiques di signal : A ↑
- Les valeurs alternent: A ↓
- Erreur de quantification plus grande, mais peut être compensée par une plus grande fréquence d'échantillonnage et filtrage du signal modulé

– Modulation sigma-delta.

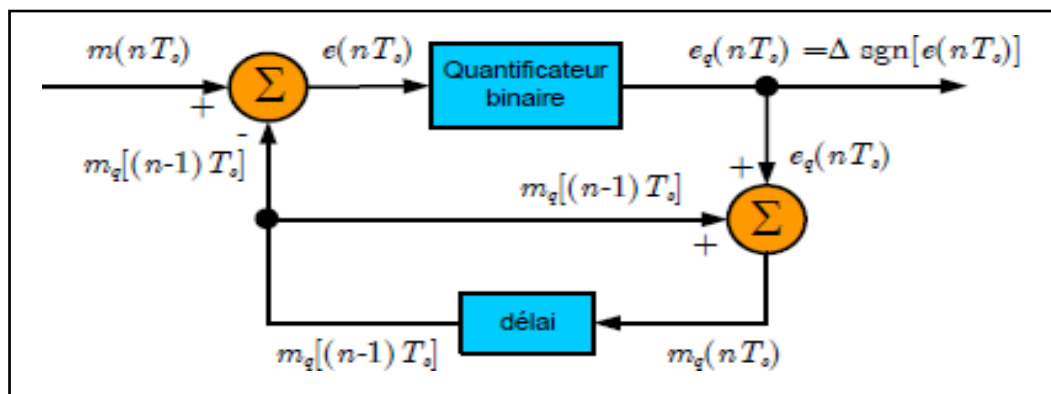


Figure 2.10: modulateur DELTA.

Conclusion

Dans le chapitre 2 nous avons étudié les différents types de codeurs de la parole MIC, MICD, MICDA, DELTA, pour le premier codeur (MIC) on a cité les trois étapes telles que l'échantillonnage, la quantification et le codage, ce qui nous a permis de comprendre la MIC uniforme et de la comparer avec les deux techniques de numérisation la loi A et μ , d'autre part pour les autres codeur (MICD, MICDA et DELTA) nous avons présenté profondément leur principe.

Chapitre 3

*Mise au point d'un ensemble de codeurs
fonctionnent à différentes débits*

Introduction

La parole est considéré le premier lien de communication pour cela il faut d'avoir une bonne qualité pour établir de bons résultats de conversion de donnés.

Pour la représentation ou l'obtention de signal de parole utilisant les déférentes types de codeurs qu'on a traite dans le chapitre précédant, dans ce chapitre on a parler généralement sur MATLAB qui est un logiciel de programmation et d'exécution a vaste bibliothèque a partir de ce dernier on a obtenue des figures donnent des résultats de simulation, puis on fait une comparaison, par conséquence on obtient le but de ce pratique est chiffrer la qualité des codeur PCM, DPCM, ADPCM et LPC.

Codage prédictif linéaire

Le codage prédictif linéaire (LPC) est une méthode de codage et de représentation de la parole elle repose principalement sur l'hypothèse que la parole peut être modélisée par un processus linéaire .il s'agit donc de prédire le signal à un instant n a partir des p précédents. La parole n'étant cependant pas un processus parfaitement linéaire la moyenne que constitue la somme pondérée du signal sur p pas de temps introduit une erreur qu'il est nécessaire de corriger par l'introduction du $e(n)$. Le codage par prédiction linéaire consiste donc à déterminer les coefficients a_k qui minimisent l'erreur $e(n)$, ceci en fonction d'un ensemble de signaux constituant un corpus d'apprentissage [27].

$$S(n)=\sum_{k=1}^p a_k \cdot s(n - k) + e(n) \quad (3.1)$$

La methode du codage par prediction lineaire est tout autant utilisee en RAP qu'en compression pour le transfert de la voix par telephone ou radio .Elle n'est cependant pas parfait puisque l'erreur de prediction peut etre importante sans qu'il soit possible par cette methode de la courriger [27].

Le processus de synthèse a donc deux phases la première phase c'est l'identification de la fonction de transfert et la deuxième les diamètres des cylindre .on obtient donc le schéma de la figure ci-dessous .le gain permet d'ajuster l'énergie du signal synthétisé à celle du signal original.

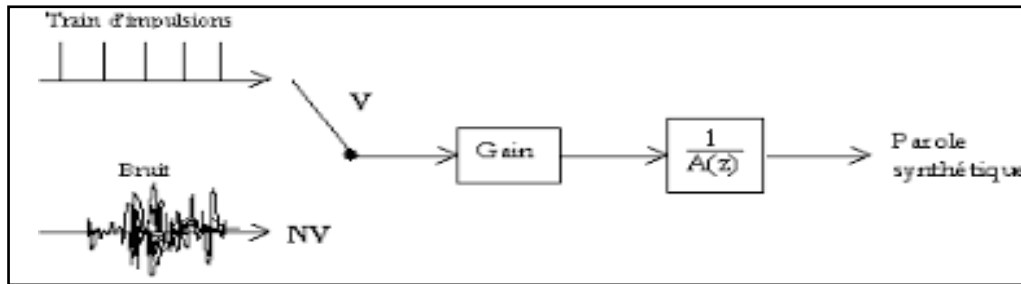


Figure 3.1 : principe du codage LPC [27].

$A(z)$: fonction de transfert.

Présentation de logiciel MATLAB

MATLAB est un langage de script [28] émulé par un environnement de développement du même nom, il est utilisé à des fins de calcul numérique, développé par la société the mathworks, MATLAB permet de manipuler des matrices, d'afficher des courbes et des données, de mettre en œuvre des algorithmes, de crée des interfaces utilisateurs, et peut s'interfacer avec d'autres langages comme le C, C++, java, et fortan, Les utilisateurs de MATLAB sont de milieux très différents tels que l'ingénieur, les sciences et l'économie, dans un contexte industriel que pour la recherche [29].

Structure du logiciel

Le logiciel réalisé est composé de plusieurs bibliothèques et de programmes exécutables, notamment pour les codeurs normalisé modifiés ainsi que les outils de prétraitement et d'évaluation de la qualité de la parole synthétisée. Un programme principal composé d'une fenêtre principale et menu principal se charge de la gestion des appels (création des objets, appel des fonctions membres, appel des programmes exécutable). [1]

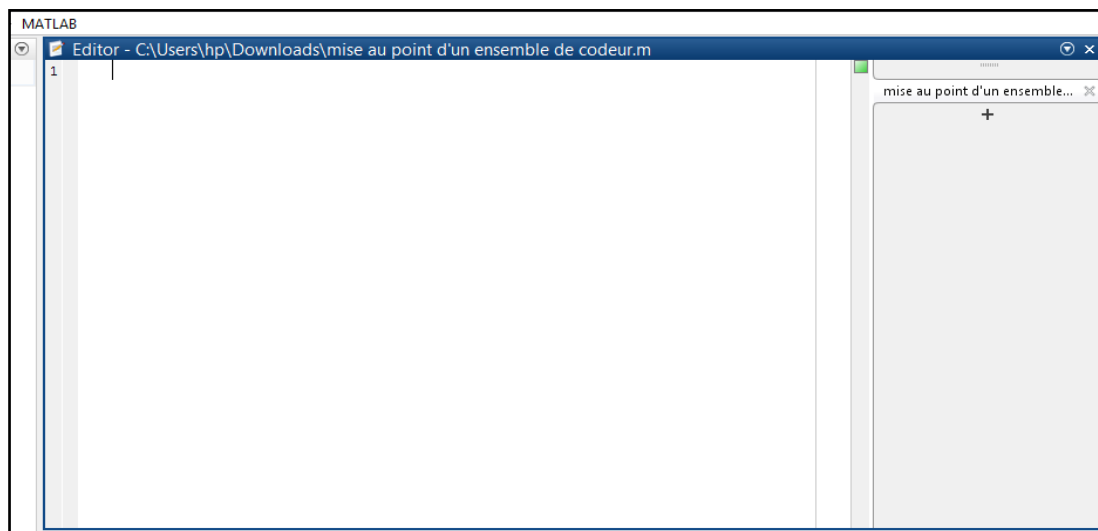


Figure 3.2: fenêtre principal de logiciel.

Résultats de simulation

Dans la partie de simulation on a quatre types de codeur a traiter a partir d'un programme, principal utilisant ce dernier pour choisir le codeur à étudier : PCM, DPCM, ADPCM et LPC (figure 3.3), précisé pour chaque codeur ces propres paramètres (le nombre de bits, la taille prédicteur P), a fin d'obtenir les figures représentant les signaux (original et reconstitué et synthétique).

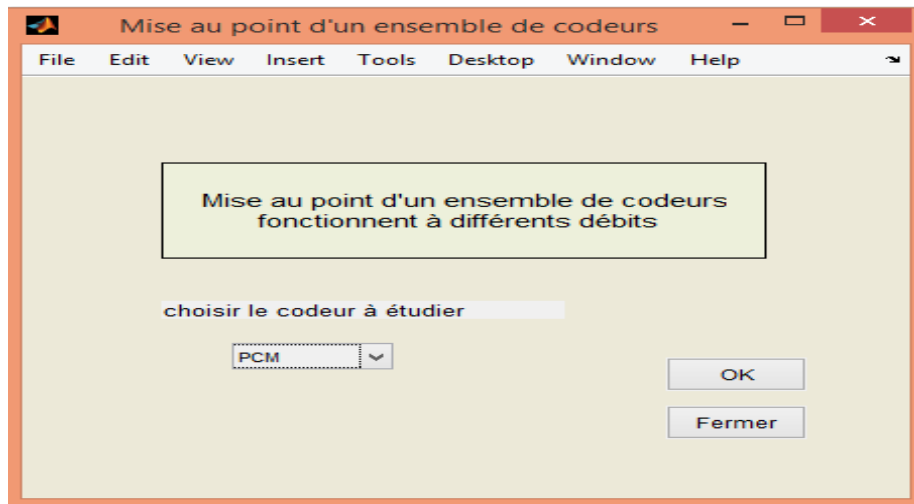


Figure 3.3 : fenêtre de programme principale.

Codeur PCM

Nous commençons de premier principe par le codeur PCM (MIC) utilisant les deux lois de quantification logarithmique : la loi A et μ , et chaque loi on a changé le nombre de bits (4bits, 8bits), les figures suivant sont des résultats de l'exécution.

➤ Codeur PCM à 4 bits (la loi A)

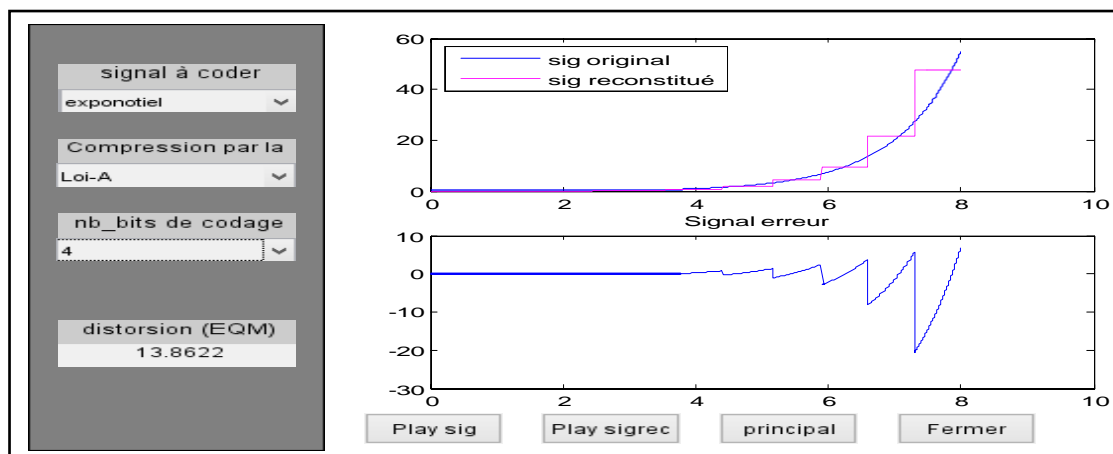


Figure 3.4 : signaux (original, reconstitué, erreur) de codeur PCM à 4 bits (A-law).

➤ Codeur PCM à 4 bits (la loi μ)

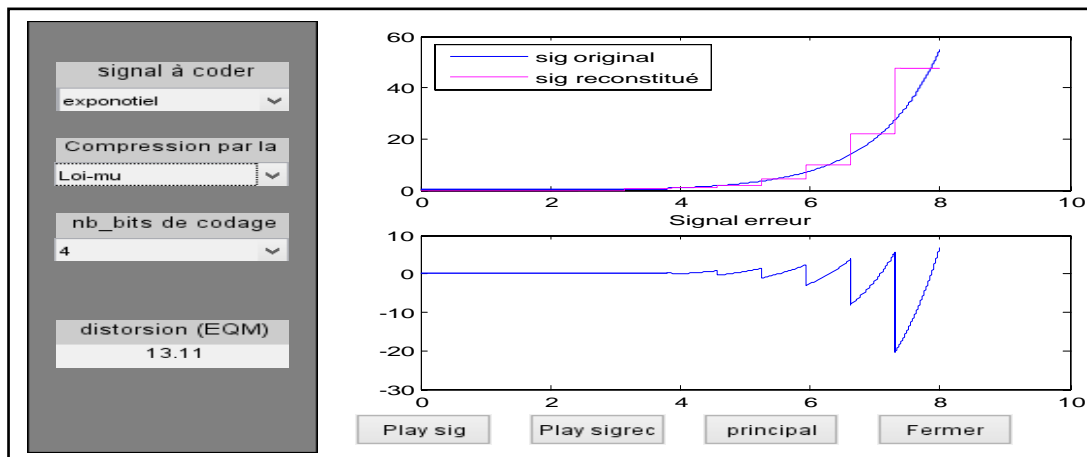


Figure 3.5: signaux (original, reconstitué, erreur) de codeur PCM à 4 bits (Mu-law).

- Le tableau ci-dessous résume les résultats (distorsion) obtenus selon différents nombre de bits.

	La loi A	La loi μ
Le nombre de bits de codage	la valeur de la distorsion (EQM)	la valeur de la distorsion (EQM)
4	13.8622	13.11
8	0.051696	0.049145
16	$7.7976e^{-7}$	$7.451e^{-7}$

Tableau 3.1 : distorsion du codeur MIC pour différents ordre de nombre de bits.

Les figures précédentes (3.4), (3.5), représentent trois signaux : signal original, signal reconstitué et signal erreur, les deux signaux précédents changent avec le changement de valeurs (le nombre de bits de codage).

A partir de tableau (3.1), on observe que quand le nombre de bits de codage augmente la valeur de la distorsion va diminuer dans les deux cas (la loi A et μ).

- **Comparaison entre les résultats**

Pour éviter la distorsion et obtenir une bonne qualité de signal de la parole, il faut que le nombre de bits soit augmenté.

La loi A et μ sont très voisines (proche) mais la loi μ est meilleur rapport signal à bruit pour les faibles niveaux.

Codeur DPCM (MICD)

Le changement qu'on a fait dans ce type est le nombre de bits de codage, la figure suivant est de résultat de l'exécution.

➤ Codeur DPCM (n=4 bits)

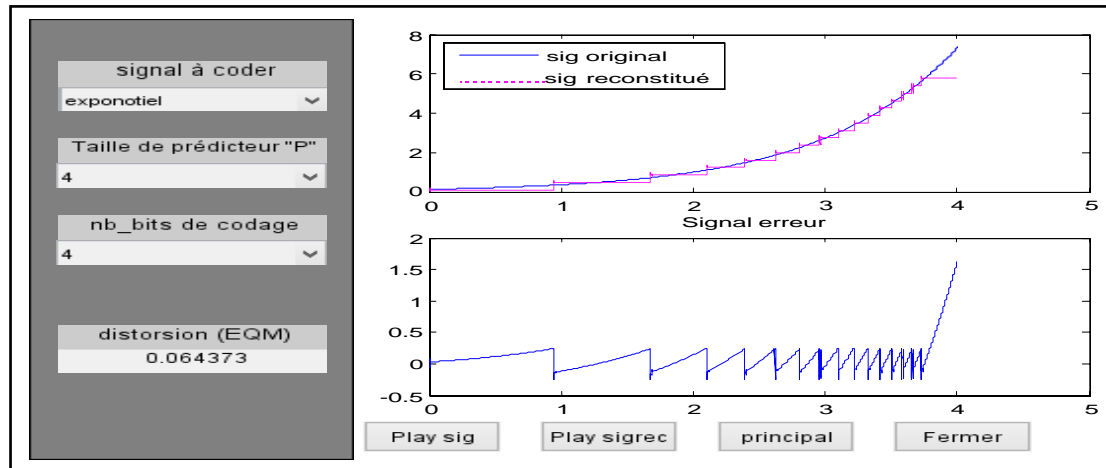


Figure 3.6: signaux (original, reconstitué, erreur) de codeur DPCM à 4 bits.

- Le tableau ci-dessous résume les résultats obtenus (distorsion) selon différentes nombre de bits.

Le nombre de bit de codage	La taille de prédicteur P	La distorsion (EQM)
4	4	0.064373
8	4	0.050577
16	4	0.050526

Tableau 3.2 : distorsion du codeur MICD pour différents nombre de bits.

Codeur ADPCM

Le changement qu'on a fait dans ce type est le nombre de bits de codage, la figure suivant est de résultat de l'exécution.

➤ Codeur ADPCM (n=4 bits)

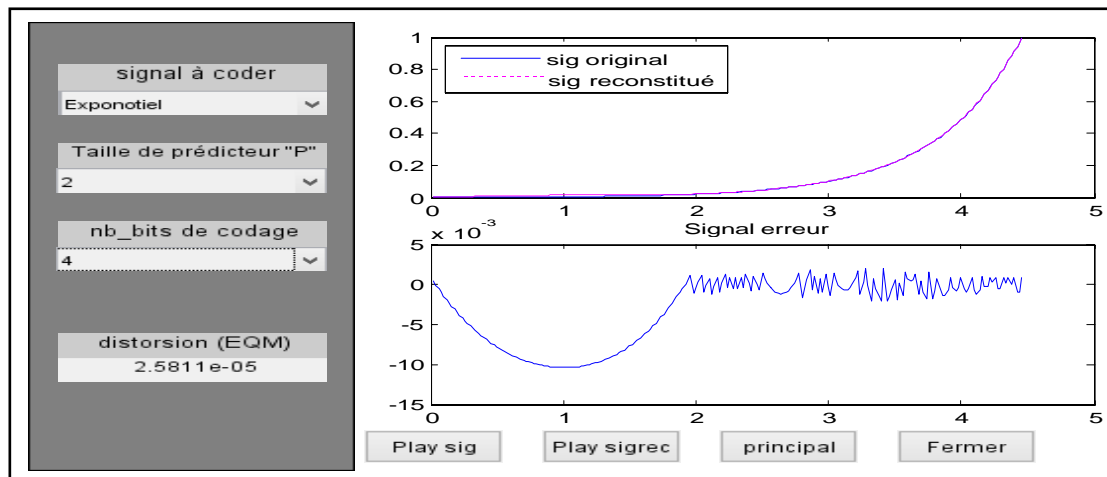


Figure 3.7 : signaux (original, reconstitué et erreur) de codeur ADPCM à 4 bits.

- Le tableau ci-dessous résume les résultats obtenus (distorsion) selon différentes nombre de bits.

Le nombre de bits de Codage	La taille de prédicteur P	La distorsion (EQM)
4	2	$2.5811e^{-5}$
16	2	$2.9598e^{-14}$

Tableau 3.3 : distorsion du codeur MICDA pour différents nombre de bits.

Discussion :

Pour les deux codeurs DPCM et ADPCM

On constate que les deux formes du signal original et reconstitué presque identique juste il y a un peu d'erreur.

A partir des résultats obtenus nous notons que les valeurs de la distorsion est très petites.

Lorsque le nombre de bits est augmente les valeurs de la distorsion n'ont pas changé (il y a un peu de changement).

Codeur LPC

Cette partie sera consacrée à l'étude générale du codage par la prédiction linéaire. On a codé deux signaux déférents « a.wav » et « s.wav ».

- Les figures suivantes sont des résultats de l'exécution du script codec.m avec défférents valeurs de P.

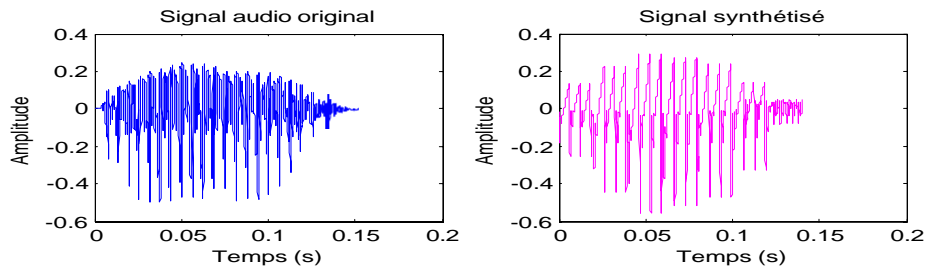


Figure 3.8 : Forme d'onde de 'a.wav' et sa forme reconstituée pour P=4.

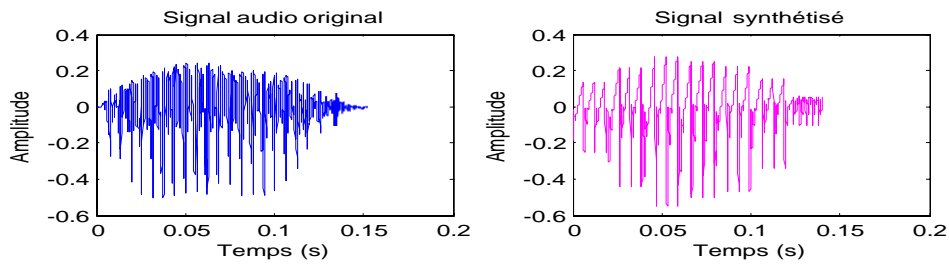


Figure 3.9 : Forme d'onde de 'a.wav' et sa forme reconstituée pour P=10.

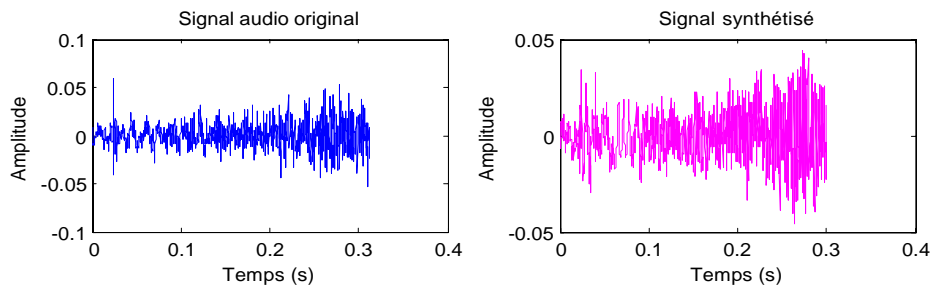


Figure 3.10 : Forme d'onde de 's.wav' et sa forme reconstituée pour P=4.

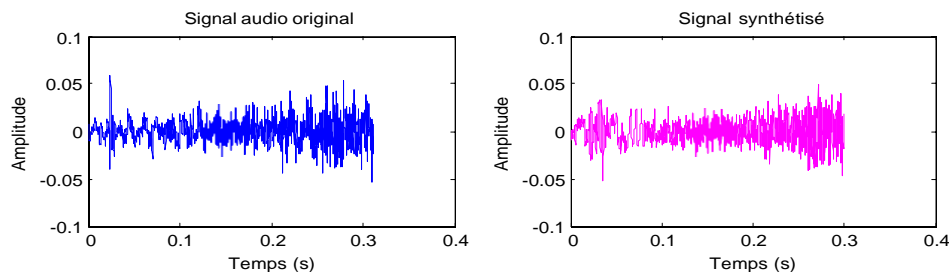


Figure 3.11 : Forme d'onde de 's.wav' et sa forme reconstituée pour P=10.

On constate que les formes des ondes des deux sons *a.wav* et *s.wav* diffèrent de leurs formes reconstituées selon les différentes valeurs de P , notamment pour le son voisé (*a.wav*). Comme on peut le constater pour ce dernier, que le signal reconstitué est moins riche en fréquences.

La figure 3.10 représente le spectre de la 3^{ème} trame de "*a.wav*" (en bleu) et les spectres du modèle (en rouge) avec différentes valeurs de P . La figure 3.11 représente le script de la 3^{ème} trame du signal "*s.wav*".

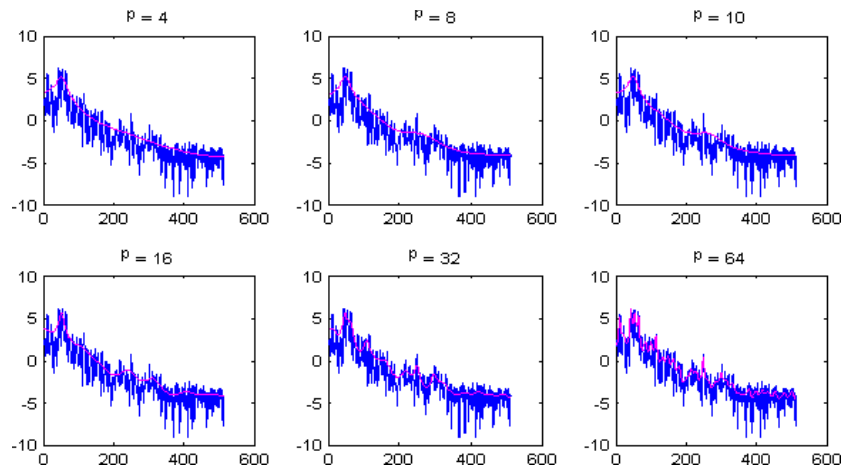


Figure 3.12 : Spectre de l'enveloppe du son *a.wav* (en bleu) et leur modèle (en rouge).

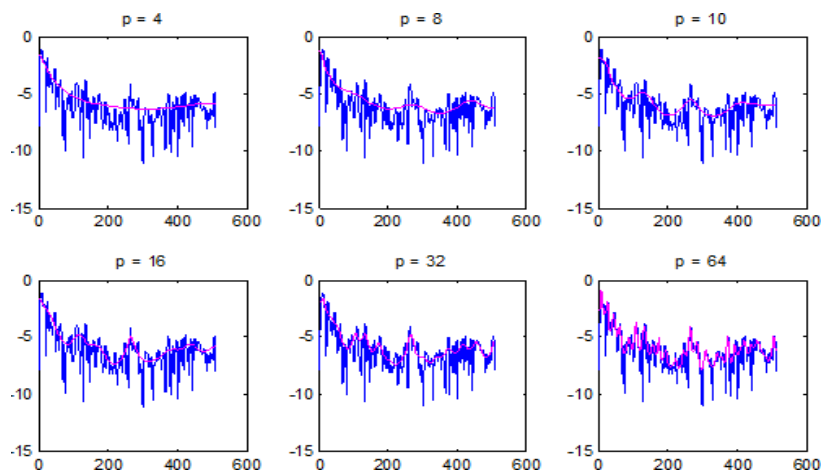


Figure 3.13 : Spectre de l'enveloppe du son *s.wav* (en bleu) et leur modèle (en rouge).

On constate que l'enveloppe spectrale du modèle devient de plus en plus proche de l'enveloppe spectrale du signal de départ lorsque P augmente. On pourra écouter le signal parole reconstitué quelque soit la valeur P et juger sa qualité perceptuelle.

Discussion

Plus l'ordre de prédiction est élevé, plus la fonction de transfert du filtre du conduit vocal s'approche des formants du spectre du signal original. Cependant, les signaux synthétiques ne s'améliorent pas (figures 3.8, 3.9, 3.10 et 3.11 précédentes). Ça montre que le problème est issu du signal d'excitation du modèle ce qui influence sur la reconstruction du signal original. Une bonne reconstruction du signal original dépend ou effet de la qualité du signal d'excitation.

On conclut que le système est assez performant dans la mesure où on peut dire que la méthode utilisée permet effectivement de reconstituer d'une façon satisfaisante un signal

vocal avec un nombre de paramètres très restreint (compression des données). Le but n'étant pas nécessairement de reconstruire le signal original à 100 %, mais de pouvoir comprendre son contenu.

Dans cette partie, on va présenter quelques résultats obtenus par le codeur LPC.

- La figure suivante illustre l'interface de ce codeur et le résultat obtenu avec un son voisé « *a.wav* ».

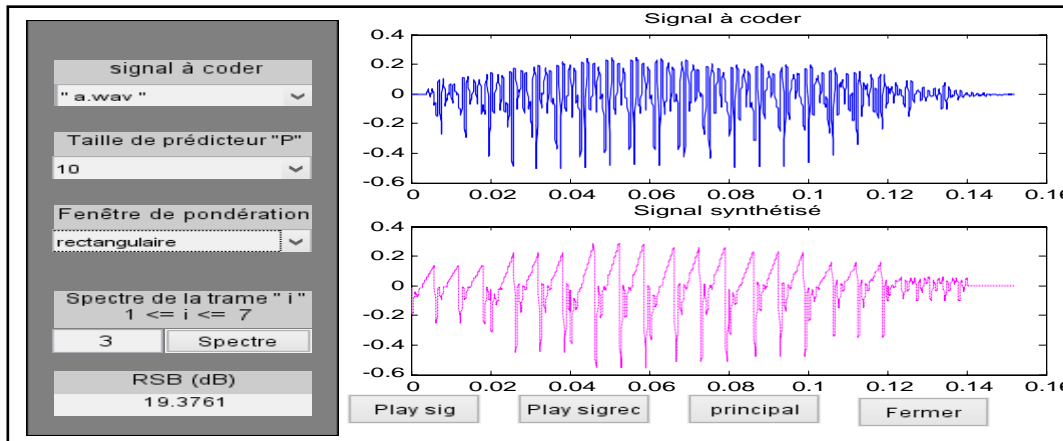


Figure 3.14 : Codage LPC d'un signal voisé « *a.wav* ».

On constate que la forme d'onde du signal synthétisé n'est pas identique au signal original quelque soit la taille du prédicteur P . On constate aussi que le signal synthétisé est moins riche en fréquences. Cependant, une pondération du signal original par une fenêtre rectangulaire avec recouvrement, on améliore le signal synthétisé Figure (3.15).

L'écoute du signal synthétique assure la compréhension du message reçu mais avec une qualité moindre par rapport au signal original. À cet effet, le RSB obtenu est égal à 19.3761dB.

- On présente ci dessous les résultats obtenus par l'utilisation d'une fenêtre de rectangulaire avec chevauchement pour $P=10$.

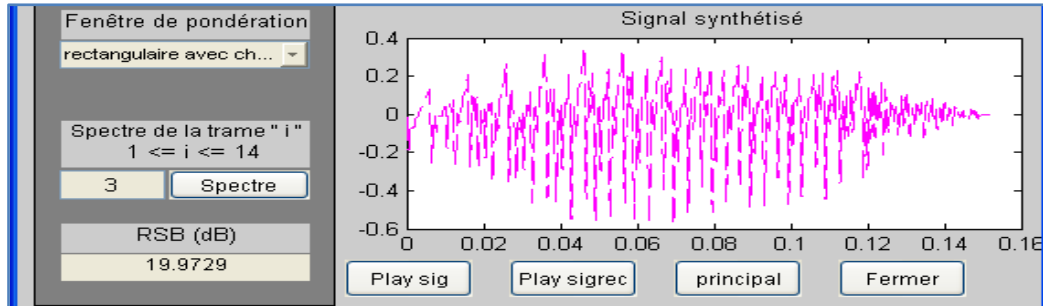


Figure 3.15: Codage LPC du signal a.wav pondéré par une fenêtre rectangulaire avec chevauchement.

- On présente ci dessous les résultats obtenus par l'utilisation d'une fenêtre de Hamming pour $P=10$.

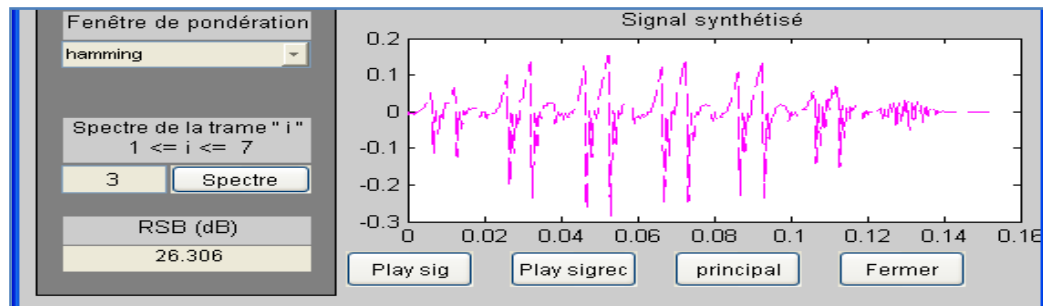


Figure 3.16 : Codage LPC du signal a.wav pondéré par une fenêtre de Hamming.

- On présente ci dessous les résultats obtenus par l'utilisation d'une fenêtre de Hamming avec chevauchement pour $P=10$.

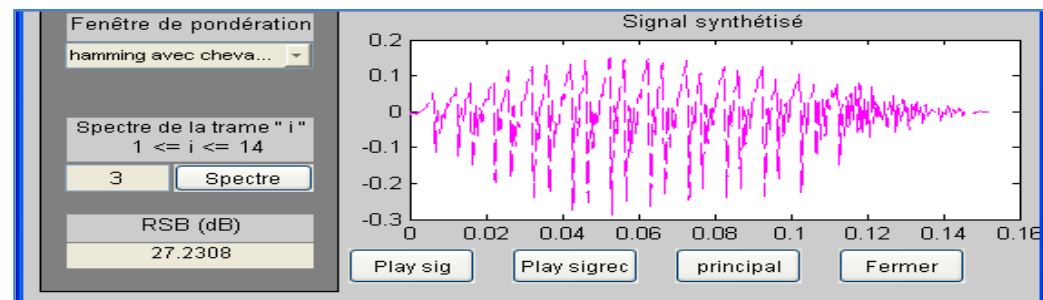


Figure 3.17 : Codage LPC de „a.wav’ pondéré par une fenêtre de Hamming avec chevauchement.

Donc l'utilisation d'une fenêtre de Hamming avec chevauchement permet d'augmenter le RSB et donc la performance du système. La fenêtre de Hamming est caractérisée par son grand lobe principal où se concentre la plus grande de l'information comparativement aux lobes secondaires.

On peut observer le spectre d'une trame et le spectre du modèle via le bouton « spectre » (figure 3.18).

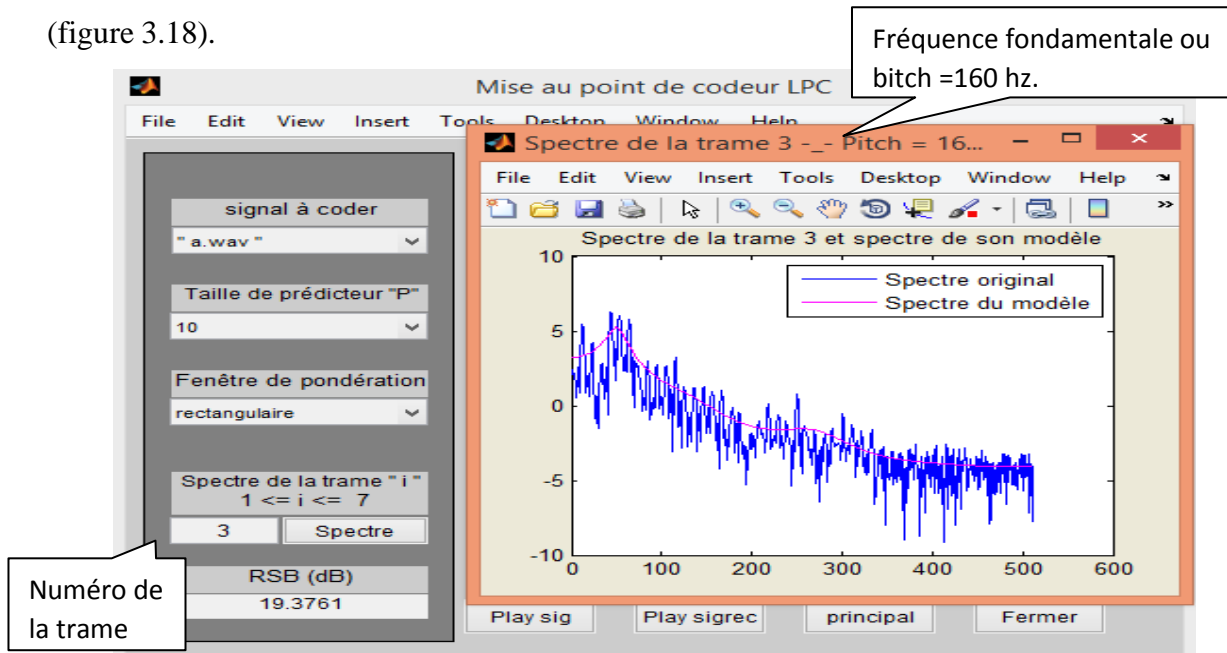


Figure 3.18 : Spectre de la 3^{ème} trame du signal 'a.wav' et le spectre du modèle.

Le tableau suivant résume les résultats (RSB) obtenus selon différentes fenêtres de pondération et selon différentes tailles du prédicteur P d'un son voisé „a.wav”.

Fenêtre / taille de prédicteur	P=4	P=8	P=10	P=14
rectangulaire	19.1851	19.3492	19.3761	19.4986
Rectangulaire avec chevauchement	19.7395	19.9196	19.9729	20.037
hamming	25.3182	26.1974	26.306	26.5572
Hamming avec chevauchement	26.299	27.1196	27.2308	27.4524

Tableau 3.4 : RSB du codeur LPC pour différentes fenêtres de pondération et différentes ordres P du filtre.

- La figure suivante illustre l'interface de ce codeur et le résultat obtenu avec un son non voisé « s.wav ».

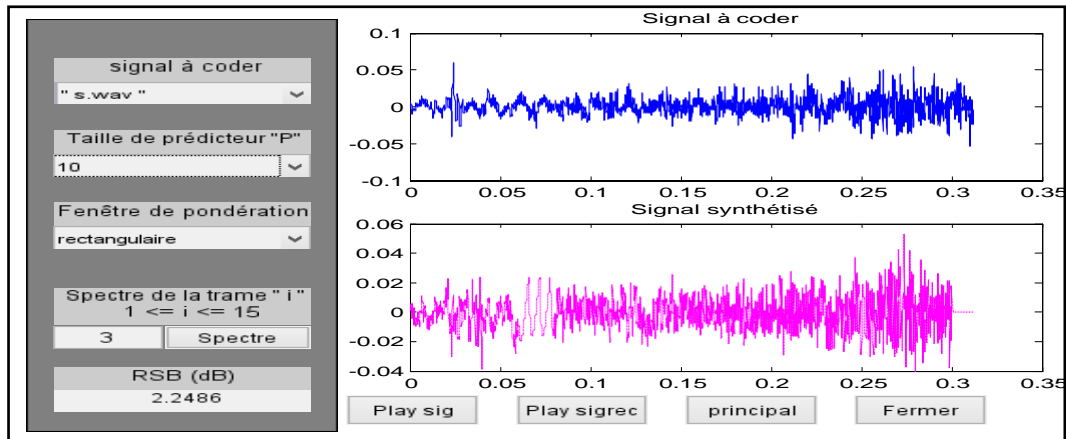


Figure 3.19: Codage LPC d'un signal non voisé « s.wav ».

- On présente ci dessous les résultats obtenus par l'utilisation d'une fenêtre de rectangulaire avec chevauchement, hamming et hamming avec chevauchement pour P=10.

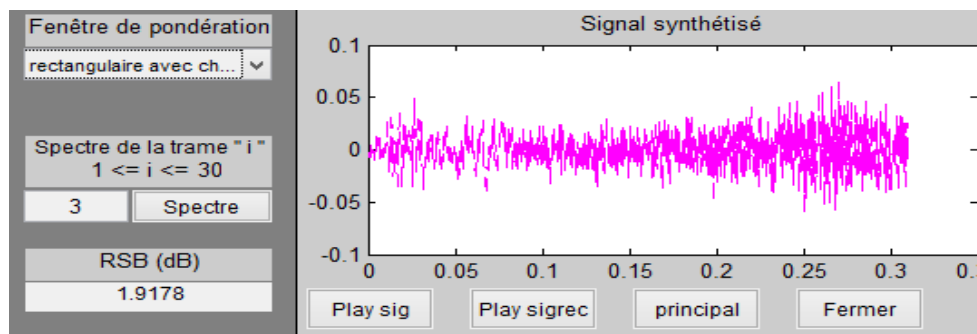


Figure 3.20 : Codage LPC du signal a.wav pondéré par une fenêtre rectangulaire avec chevauchement.

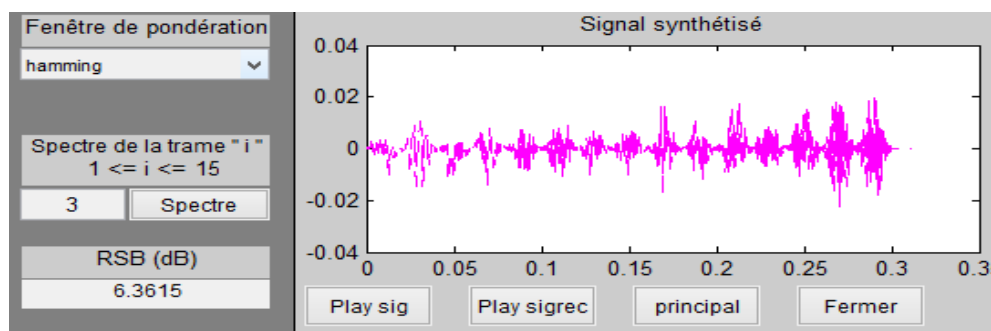


Figure 3.21 : Codage LPC du signal a.wav pondéré par une fenêtre de Hamming.

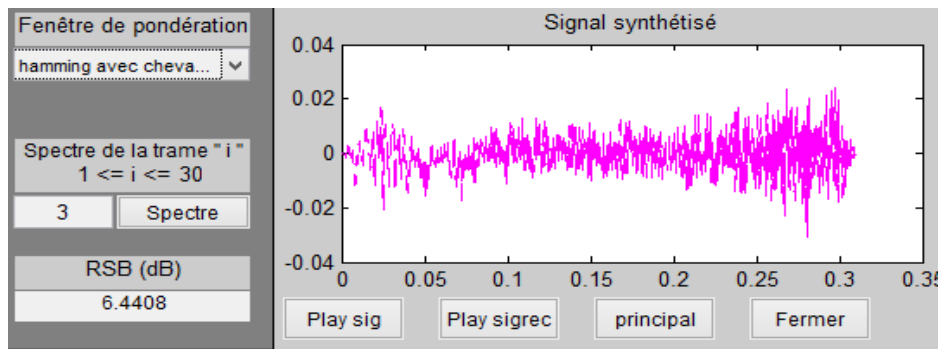


Figure 3.22 : Codage LPC de „*a.wav*’ pondéré par une fenêtre de Hamming avec chevauchement.

On peut observer le spectre d’une trame et le spectre du modèle via le bouton « spectre » (figure 3.23).

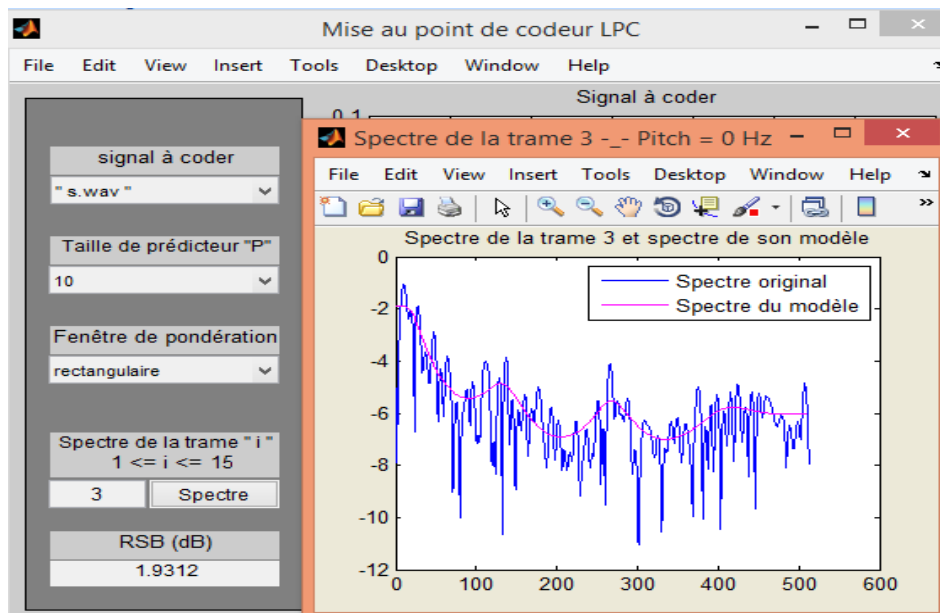


Figure 3.23: Codage LPC du signal non voisé « *s.wav* » pour $P = 10$.

On observe que l’évolution de l’enveloppe spectrale du modèle suit la variation du spectre du signal à synthétiser. Comme peut le constater, la 3^{ème} trame est bien non voisée (fréquence fondamentale nulle). Donc, on peut dire que la taille du prédicteur $P=10$ est suffisante pour réaliser un codage LPC de qualité.

Le tableau suivant résume les résultats (RSB) obtenus selon différentes fenêtres de pondération et selon différentes tailles du prédicteur P d’un son non voisé „*s.wav*”.

Fenêtre / taille de prédicteur	P=4	P=8	P=10	P=14
rectangulaire	1.6406	1.7974	2.2486	2.0412
Rectangulaire avec chevauchement	1.437	1.8432	1.9178	2.2425

hamming	6.272	6.1321	6.3615	6.7498
Hamming avec chevauchement	5.8679	6.084	6.4408	6.8475

Tableau 3.5 : RSB du codeur LPC pour différentes fenêtres de pondération et différentes ordres P du filtre.

La figure 3.24 représente un autre exemple de codage LPC pour un signal de parole.

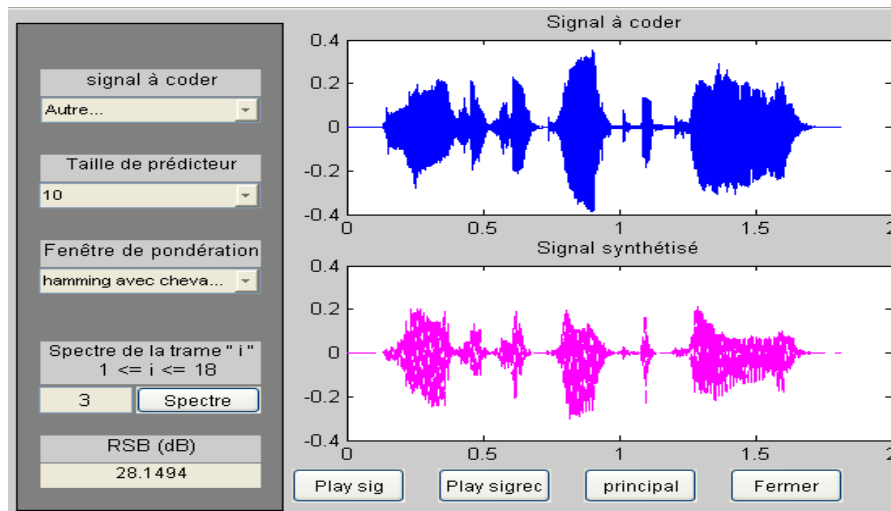


Figure 3.24 : Codage LPC d'un signal de parole « *phrase_f.wav* » pour $P = 10$.

On constate que la forme d'onde du signal synthétisé est moins proche de celle du signal original. Par contre, l'écoute du signal synthétisé permet de comprendre son contenu malgré la mauvaise qualité perçue.

Rappelons que, le but principal du codage LPC n'est pas de reconstruire le signal original à 100% mais plutôt de comprendre son contenu.


Conclusion

Dans ce chapitre nous avons étudié les codeurs PCM, DPCM, ADPCM et le codage prédictif linéaire (LPC), ce dernier on a présenté au début de chapitre trois.

A partir de la simulation on a obtenu des résultats pour chaque codeur :

- le codeur PCM : pour réduire la distorsion et obtenir une bonne qualité de signal de la parole il faut que le nombre de bits de codage doivent être augmenté.
- le codeur PCM donne de meilleurs résultats comparativement aux deux codeurs DPCM et ADPCM.

- Pour le codeur LPC, les résultats obtenus montre que la méthode qu'elle à été présentée, nous a permis d'obtenir une parole de synthèse qui n'a pas été satisfaisante à l'écoute. Un codeur LPC permet de diminuer de manière significative le débit binaire associé, au prix d'une dégradation de la qualité de la parole. Le tout étant de ne pas diminuer trop ce débit afin de conserver l'intelligibilité du signal.



***Conclusion
générale***

Conclusion générale

Notre travail consiste à mettre au point d'une plate forme de codeur .A fin de réaliser cet ensemble de codeur ,nous avons suivi une démarche qui consiste à faire une brève étude théorique sur le principe de ces codeurs .Ainsi l'approche analytique fut utilisée pour déterminer les principales caractéristiques de ce dernier mais quelque soit le degré de précision de résultats de l'analyse ,les résultats des mesures expérimentales sont toujours nécessaires pour valider les données théoriques .ceci nous a mené à réaliser sous forme didactique(travaux pratiques) , un ensemble de codeurs ,à l'aide du logiciel MATLAB .

Nous avons présentée dans le premier chapitre la quantification scalaire et vectoriel ; ces deux méthodes nous ont permis de fournir une représentation compacte et efficace des données source (données échantillonnées) tout en préservant l'information essentielle qu'elles portent.

L'étudiant constatera que le quantificateur scalaire non uniforme donne un bon résultat par rapport au quantificateur scalaire uniforme .il verra aussi que la performance s'augmente en fonction du nombre des niveaux de quantification „L“. D'autre part la performance du quantificateur vectoriel devient meilleure s'il dispose d'une bonne base d'apprentissages représentative de tous les signaux possibles à quantifier. En plus la performance de la QV dépend de la taille du dictionnaire en nombre de vecteur, de la dimension de ces vecteur et enfin du seuil « ϵ » utilisé dans l'algorithme LBG.

De ce qui concerne le deuxième chapitre on a approfondi beaucoup plus sur les types des codeurs le MIC, MICD, MICDA et la modulation DELTA citons le principe de fonctionnement la quantification l'échantillonnage et la compression de chaque codeur à fin de les introduire pour la partie pratique de chapitre suivant.

Le troisième chapitre l'étudiant constatera que le codeur PCM donne de meilleurs résultats comparativement aux deux codeurs DPCM et ADPCM. Ainsi, l'utilisation d'une quantification et d'une prédiction adaptative dans le cas d'ADPCM permet d'améliorer les performances du codeur DPCM. Par ailleurs le codeur ADPCM donne des résultats assez proches comparativement au codeur PCM. Mais son intérêt réside dans la diminution de débits (de moitié) pour une qualité assez proche.

Conclusion générale

Dans ce chapitre a l'objectif aussi de présenter les codeurs LPC qui permet de la transmission de la parole a faible débit. Plus l'ordre de prédiction est élevés plus la fonction de transfert du filtre de modèle reste très différent du signal d'excitation original, ce qui influence la reconstruction du signal du signal original dépend du signal d'excitation.

On peut dire comme perspective, que ce travail représenté un premier banc de manipulation de codeur l'objectif final du projet n'est certes pas achevé cependant ce travail ouvre plusieurs chantiers pour d'abord l'achever puis ensuite lui apporter des améliorations.

Bibliographie

- [1] : M. Djamah, « Codage échelonnable à granularité fine de la parole utilisant la quantification vectorielle arborescente », Thèse de Doctorat en Télécommunications, INRS (Centre Énergie Matériaux Télécommunications), Université du Québec.
- [2] : M.Farah Faris, « codage de la parole pour des transmissions très bas débit », Thèse de magistère en électronique, Université Des Sciences et de la Technologie Houari Boumediene, Faculté d'électronique et d'informatique, Spécialité : Communications parlée.
- [3] : Mr Lachachi Nour-Eddine, « codage paramétrique de la parole en vus de sa transmission sur internet », Thèse de magistère, Spécialité : informatique, Université D'Oran –ES-Senia-, Octobre 2006.
- [4] : S.P. Lloyd, « Least squares quantization in PCM », IEEE Trans. Inform. Theory, Vol. IT-28, No.2, pp. 129-137, March 1982.
- [5] : J. Max, « Quantizing for minimum distortion », IEEE Transactions on Information Theory, Vol. IT-6, pp.7-12, March 1960.
- [6] : Teddy Voinson, « Quantification vectorielle algébrique avec zone morte : application à la compression d'image à bas débit et au tatouage d'image », Université Henri Poincaré – Nancy1, 2003.
- [7] : N.Moreau, « technique des compressions des données », Masson, France, 1995.
- [8] : R.Goldberg , L.Rick , « A practical handbook of speech coders », Ed CRC press LCC, USA, 2000.
- [9] : Y. Linde, A. Buzo, and R.M. Gray, « An algorithm for vector quantization design », IEEE Trans. Comm. Vol. 28, No. 1, pp. 84-95, Jan. 1980.
- [10] : A. Gersho and R.M. Gray, Vector Quantization and Signal Compression, Kluwer Academic Publishers, Boston, 1992.
- [11] : I. Katsavounidis, C. Kuo, and Z. Zhang, “A new initialisation technique for generalized Lloyd iteration”, IEEE Signal Processing Letters, Vol. 1, pp. 144- 146, October 1994.
- [12] : M. Bouzid, “Codage Conjoint de Source et de Canal pour des transmissions par Canaux Bruités”, Thèse de doctorat, Université USTHB, Algérie, Avril 2006.

Bibliographie

- [13] : N.Farvardin , J.Modestino, « Optimum quantizer performance for a class of non-gaussian memorless sources », IEEE Tran, On Information Theory Vol 30 mai 1984.
- [14] : « modulation par impulsions et codage », Le grand dictionnaire terminologique, office québécois de la langue française, 19 juillet 2020.
- [15] : Michel Joindot, « Alain Glavieux », „ introduction aux communications numériques“, Dunod, 2007.
- [16] : „ [http:// www.licm.fr/IMG/pdf/cours-Strategies_de_codage_canal-pdf](http://www.licm.fr/IMG/pdf/cours-Strategies_de_codage_canal-pdf)“, „stratégie de codage canal“, 2003/2004.
- [17] : O.Manel khadidja, S.Amina, « Etude et simulation d’une chaine de transmission MIC, Thèse de Master, Université SAAD DAHLAB, Spécialité : Réseau et télécommunication, Blida, 2011/2012.
- [18] : https://fr.wikipedia.org/wiki/Modulation_par_impulsions_et_codage.
- [19] : J. Hernández, «Algorithmes d’acquisition, compression et restitution de la parole à vitesse variable. Etude et mise en place», Projet de fin d’études, Spécialité : Informatique, École Nationale Supérieure de l’Électronique et de ses Applications (ENSEA) Cergy Pontoise, Paris, Avril 1995.
- [20] : M.Antonini, Vincent Ricordel, « Quantification », Compression et codage des images et des vidéos : traité IC2, Série traitement du signal et de l’image, Paris, janvier 2002.
- [21] „http://public.iutenligne.net/telecommunication/Berthet/Module_numérisation/BruitDeQuantif/index.html“, « Numérisation des signaux pour l’audiovisuel et la transmission ».
- [22] : Yvon Mori, « Filtrage Numérique », „ Hermes Science“,2007.
- [23] : ITU-T Recommendation G.711, « Pulse code modulation (PCM) of voice frequencies», November1988.
- [24] : [https://fr.wikipedia.org/wiki/Loi_A#:~:text=La%20loi%20A%20\(en%20anglais,711](https://fr.wikipedia.org/wiki/Loi_A#:~:text=La%20loi%20A%20(en%20anglais,711).
- [25] : https://fr.wikipedia.org/wiki/Loi_%C2%B5.
- [26] : « Aspects généraux des systèmes de transmission numérique ; équipements terminaux », Union Internationale Des Télécommunications, Genève, 1990.

Bibliographie

[27] : <https://reussirlem2info.files.wordpress.com/2012/12/codage-de-la-parole.pdf>.

[28] : « Scripting Language vs Programming Language », (consulté le 31 août 2019).

[29] : [https:// fe.mathworks.com/company/aboutus.html](https://fe.mathworks.com/company/aboutus.html).

Résumé

Le codage de parole devenue plus en plus une question de recherche très important dans le domaine de la télécommunication, la transmission dans le cas générale passe sur plusieurs opération la quantification, l'échantillonnage et la compression quelque soit le message transmis (image, voix...) dans notre thème de recherche on intéressé sur la partie de codage qui est base sur les types de codeurs.

Dans ce mémoire on présente le codage de la parole a bas débit d'une façon générale qui consiste comme une technologie informatique permettant a un logiciel d'interpréter une langue naturelle humaine ainsi que ce dernier considérer un axe de recherche très important dans le domaine de compression de la parole qui tend à réduire le débit d'information à condition que la qualité de signal ne soit pas dégradé, Ce manuscrit aborde la mise au point d'un ensemble de codeurs fonctionnent à différents débits MIC, MICD, MICDA et LPC, ces codeurs son évalués en termes de qualités perceptuelle et leurs performances.

Mots clé : le codage de parole, la transmission, la quantification, l'échantillonnage, la compression, le codage a bas débit, un codeur, MIC, MICD, MICDA, LPC.

Abstact

Speech coding has become more and more a very important research question in the field of telecommunications, the transmission in the general case passes on several operations quantification, sampling and compression whatever the message transmitted (image, voice ..) in our research theme we are interested in the coding part which is based on the of coders.

In this thesis we present the coding of speech at low speed in a general way which consists of a computer technology allowing software to interpret a natural human language as well as the latter considering a very important research axis in the field of speech compression which tends to reduce the information rate provided the signal quality is not degraded, This manuscript addresses the devlopment of a set coders operating at different PCM, DPCM, ADPCM and LPC bitrates, these coders are evaluated in terms of perceptual qualities and their performance.

Keywords : coding of speech, the transsmission, the quantification, the sampling, compression, the codeur, MIC, MICD, MICDA, LPC.