

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
École Nationale Polytechnique



Département d'Électronique  
Laboratoire Signal et Communication

---

**Mémoire**

---

Mémoire de projet de fin d'étude pour l'obtention du DIPLÔME D'INGÉNIEUR  
D'ÉTAT EN ÉLECTRONIQUE

par

KHANFRI Zakaria  
&  
MOUSSAOUI Fares

<p><b>ALGORITHME DE CODAGE DE LA PAROLE</b></p>
---

Soutenu le 20/06/2016 :  
devant le jury

---

Président de jury:	Pr. MEHENNI	ENP.
Rapporteur:	Pr. BERKANI	ENP.
Examineur:	MAA A.MOUSSAOUI	ENP.

Promotion 2016  
E.N.P, 10, Avenue Hassen BADI, EL HARRACH, ALGER  
ENP 2016

# DÉDICACE

I Dedicate this work to:  
My Parents & My Brothers,  
MY Family,  
MY Friends,

*Zaki*

JE dédie ce travail,  
À mes Parents, à mes frères & Soeurs,  
À ma famille,  
mes amis,  
*Fares*

# REMERCIEMENTS

Nous remercions le bon Dieu de nous avoir donné la volonté et la patience qui nous ont permis de mener à bien ce travail.

Nous remercions vivement notre promoteur, Pr. D.BERKANI, pour sa gentillesse, ses précieux conseils et pour la confiance qu'il nous a témoignée en dirigeant ce travail tout en nous laissant travailler très librement.

On tient à adresser nos sincères remerciements à tous les membres du Jury chargé d'examiner la soutenance de notre projet de fin d'études.

Nous remercions tous ceux qui ont de près ou de loin contribué à l'aboutissement et l'amélioration de ce travail.

# RÉSUMÉ

ملخص:

في هذا العمل قُمنّا باقتراح خوارزمية لتشفير الكلام من خلال إستعمال طرق جديدة حول التكميم الشعاعي المقسم (SVQ) و ذلك من أجل تخفيض معدّل البت المخصّص للمعاملات LSF . و هذا يتطلّب دراسة إنتاج الكلام بالإضافة إلى التعرف الخطّي من أجل حساب المعاملات LPC .

كلمات مفتاحية: التكميم الشعاعي المقسم ، CELP ، التعرف الخطّي ، خطوط التواترات الطيفية LSF ، قاموس .

## Abstract:

In this work we have proposed a speech coding algorithm by using a new methodes of quantization more efficient (SVQ) and this in order to reduce the bit-rate allocated to these LSF coefficients . This requires studing speech production and also the linear prediction to compute LPC parameters.

**Key Words:** Split vector quantization, CELP, linear Prediction, line spectral frequencies LSF, codebook.

## Résumé:

Dans ce travail on a proposé un algorithme de codage de la parole en utilisant de nouvelles méthodes de quantification plus efficaces (SVQ) et ceci dans le but de réduire le débit alloué à ces coefficients LSF. Ce qui nécessite une étude de la production de la parole ainsi que la prédiction linéaire pour calculer les paramètres LPC.

**Mots Clés:** Quantification vectorielle divisée, CELP, Prédiction linéaire, lignes de fréquences spectrales LSF, dictionnaire.

# Table des Matières

**Table des matières**

**Liste des figures**

**Liste des tableaux**

<b>Introduction Générale</b>	<b>11</b>
<b>1 Principes de Codage de la Parole</b>	<b>12</b>
1.1 Introduction . . . . .	12
1.2 La Production de la voix . . . . .	12
1.2.1 Son voisé et non voisé . . . . .	13
1.2.2 Propriétés de Perception . . . . .	14
1.3 Modélisation de Processus de la Production de la Parole . . . . .	15
1.4 La prédiction Linéaire . . . . .	16
1.4.1 Minimisation d'Erreur de Prédiction . . . . .	16
1.4.2 Algorithme de Levinson-Durbin . . . . .	17
1.5 Représentation des Coefficients de Prédiction . . . . .	19
1.5.1 Les Coefficients de Réflexion . . . . .	19
1.5.2 Les Coefficients (LARs) et (ASRC) . . . . .	20
1.5.3 Les Fréquences de Raies Spectrales (LSF) . . . . .	20
1.6 Conversion LSF-LPC . . . . .	22
1.7 Détecteur de Voisement . . . . .	23
1.7.1 Énergie et Taux de Passage par Zéro . . . . .	23
1.7.2 Détection du Pitch . . . . .	24

<b>2</b>	<b>Techniques de Codage de la Parole</b>	<b>25</b>
2.1	Introduction . . . . .	25
2.2	Techniques de Codage . . . . .	25
2.2.1	Quantification Scalaire . . . . .	26
2.2.2	Quantification Vectorielle . . . . .	28
2.2.3	L'encodeur . . . . .	29
2.2.4	Le décodeur . . . . .	30
2.2.5	Codage de forme d'onde . . . . .	32
2.2.6	Codage paramétrique . . . . .	35
2.2.7	Codage hybride . . . . .	36
2.3	Mesure de la Distorsion . . . . .	36
2.3.1	Débit de Transmission . . . . .	36
2.3.2	Délai de Codage . . . . .	36
2.3.3	Qualité de Parole . . . . .	37
2.3.4	Mesures Subjective de La Qualité de Parole . . . . .	37
<b>3</b>	<b>Codage CELP</b>	<b>40</b>
3.1	Introduction . . . . .	40
3.2	Principe du Codage CELP . . . . .	40
3.2.1	L'encodeur . . . . .	41
3.2.2	Analyse par prédiction linéaire . . . . .	42
3.2.3	Analyse par synthèse . . . . .	43
3.2.4	Modélisation du signal d'excitation . . . . .	44
3.2.5	Pondération perceptive . . . . .	44
3.2.6	Le décodeur . . . . .	45
<b>4</b>	<b>Implémentation sur MATLAB</b>	<b>47</b>
4.1	introduction . . . . .	47
4.2	Calcul des coefficient LPC . . . . .	47
4.2.1	Fenêtrage et segmentation . . . . .	48
4.2.2	Calcul des Coefficients d'Auto-corrélation . . . . .	50
4.2.3	LEVINSON-DURBIN - LPC . . . . .	51
4.2.4	Filtre perceptuel . . . . .	52
4.2.5	L'inverse de Levinson-Durbin . . . . .	54
4.3	Calcul et Quantification des LSFs . . . . .	54
4.3.1	Construction des Polynômes LSFs . . . . .	55
4.3.2	Zéros du Polynôme Symétrique . . . . .	56
4.3.3	Zéros du Polynôme Antisymétrique . . . . .	57

4.3.4	Test des cas de Mal-conditionnement . . . . .	59
4.3.5	Quantification des LSFs . . . . .	60
4.3.6	Correction des Quantifications . . . . .	61
4.4	Estimation du pitch en boucle ouverte . . . . .	65
4.5	Codebook Stochastique . . . . .	66
	<b>Conclusion Générale</b>	<b>68</b>
	<b>Bibliographie</b>	<b>68</b>

# Liste des figures

1.1	Coupe de l'appareil phonatoire . . . . .	13
1.2	Le spectre d'un son voisé . . . . .	13
1.3	Le spectre d'un son non voisé . . . . .	14
1.4	Modèle simplifié de production de la parole . . . . .	15
1.5	Filtre d'analyse . . . . .	17
1.6	La séquence du signal résiduel . . . . .	18
1.7	Filtre de synthèse . . . . .	18
1.8	spectre d'amplitude du filtre LPC . . . . .	19
2.1	Quantification sclaire uniforme avec cinq niveau $L=5$ . . . . .	28
2.2	Schéma général d'un quantificateur vectoriel . . . . .	30
2.3	Encodeur et décodeur d'une quantification SVQ . . . . .	32
2.4	Codage PCM . . . . .	33
2.5	Structure première DPCM . . . . .	34
2.6	DPCM avec Quantificateur à l'intérieur du feedback . . . . .	35
3.1	Principe du Codeur LPAS . . . . .	41
3.2	(a) Encodeur . . . . .	42
3.3	Spectre LPC du signal de la parole (en pointillé) et réponse fréquentielle du filtre de pondération perceptive $A(z)/A(z/\gamma$ (trait plein) avec $\gamma=0.8$ . . . . .	45
3.4	(b) Décodeur . . . . .	46
4.1	Prédiction linéaire et auto-corrélation . . . . .	48
4.2	Exemple d'un signal engrestré avec le mot "ONE" . . . . .	49
4.3	Fenêtre de Hamming à 240 points . . . . .	49
4.4	L'effet de la fenêtre de Hamming . . . . .	50
4.5	L'effet de la fenêtre de Hamming dans le domaine spectrale . . . . .	50
4.6	Réponse fréquentielle du filtre de synthèse LP . . . . .	52

4.7	Plan en $z$ des coefficients LP de la 25 <sup>ème</sup> trame . . . . .	52
4.8	Enveloppe Spectrale Avant et Après $W(z)$ . . . . .	53
4.9	Calcul Polynomial et Quantification des LSFs . . . . .	55
4.10	Racines du Polynôme LSP Symétrique . . . . .	57
4.11	Racines du Polynôme LSP Antisymétrique . . . . .	58
4.12	Racines da la trame 25 dans le plan $z$ . . . . .	59
4.13	Racines des polynômes LSPs pour un filtre à phase non-minimales . . . . .	60
4.14	Analyse courte et long terme sur la 30 <sup>ème</sup> trame . . . . .	66

# Liste des tableaux

2.1	MOS: Points Moyens d'Opinion . . . . .	38
4.1	Valeurs d'auto-corrélation $r(25,;)$ et $r(30,;)$ . . . . .	51
4.2	Analyse LP des deux trame 25 et 30 . . . . .	51
4.3	Coefficients du filtre LP avant-après expansion . . . . .	53
4.4	RCs de la trame 25 . . . . .	54
4.5	Construction des Polynômes $P(z)$ et $Q(z)$ pour la trame 25 . . . . .	55
4.6	Zéros de du polynôme $P(z)$ correspond à la trame 25 . . . . .	57
4.7	Zéros du polynôme $Q(z)$ correspond à la trame 25 . . . . .	58
4.8	Matrice de Quantification des LSF . . . . .	61
4.9	LSFs Quantifier (trame 25) . . . . .	61
4.10	LSFs quantifier par SVQ [3;3;4] (trame 25) . . . . .	62
4.11	LSFs quantifiers par SVQ [3;3;4] (trame 30) . . . . .	62
4.12	LSFs quantifiers par SVQ [3;4;3] (trame 25) . . . . .	63
4.13	LSFs quantifiers par SVQ [3;4;3] (trame 30) . . . . .	63
4.14	Comparaison entre QVs [3;3;4] & [3;4;3], & SQ . . . . .	64
4.15	Pitch & Gain . . . . .	66
4.16	Indice d'excitation et cible calculé pour le mot "One". . . . .	67

# Introduction Générale

L'objectif du travail présenté dans ce projet de fin d'études est de proposer un algorithme de codage de la parole en s'inspirant des codeurs déjà existant et en utilisant la SVQ (Quantification vectorielle par split), et en cas de succès adapter l'algorithme résultant pour certaines applications spécifiques.

Cette tâche a pu aboutir grâce à une étude brève des différents codeurs.

- Le premier chapitre reposait sur l'étude de la technique de prédiction linéaire qui est présente pratiquement dans tous les codeurs de la parole moderne.
- Le deuxième chapitre consistait à étudier différentes techniques de codages, ce chapitre comprenait aussi une description des différents moyens d'évaluation de la qualité de la parole fournie par un codeur afin de les appliquer dans le chapitre de l'implémentation.
- Le troisième chapitre était une étude de l'algorithme CELP.
- Dans le chapitre quatre. Nous avons réalisé l'implémentation de l'intégralité des blocs du codeur, cette implémentation visait l'extraction de tous les paramètres caractérisant le signal au biais de cet algorithme.

# Principes de Codage de la Parole

## 1.1 Introduction

En raison des caractéristiques du conduit vocal humain, le signal de parole est fortement redondant. Ces redondances permettent aux algorithmes de codage de compresser le signal en enlevant l'information non pertinente contenue dans le signal. Donc la connaissance du système vocal et des propriétés du signal de parole est essentielle pour concevoir des codeurs efficaces.

Les propriétés du système auditif humain peuvent également être exploitées, pour améliorer la qualité perceptuelle du signal codé. Avant d'aborder le problème de codage de parole plus précisément, quelques caractéristiques du signal vocal sont présentées, et permettront de mieux apprécier les différentes techniques de codage présentées par la suite. Une partie simple de la théorie acoustique est présentée et les notions de phonème, de formant, de son voisé, non voisé et de pitch sont définis.

## 1.2 La Production de la voix

La parole apparaît physiquement comme une variation de la pression de l'air causée et émise par le système articulatoire. Quand une personne parle, sous le contrôle du système nerveux central qui reçoit en permanence des informations, les poumons jouent le rôle d'un générateur du système de production de la parole, ils fournissent l'énergie nécessaire à la production du son, en poussant de l'air à travers la trachée-artère, au sommet de celle-ci se trouve le larynx où la pression de l'air est modulée avant d'être appliquée au conduit vocal, composé des cavités nasales et buccale pour la plupart des sons. La glotte fournit l'entrée avec certaine fréquence du pitch (fondamentale) ( $F_0$ ). Le conduit vocal travail comme

## 1.2. La Production de la voix

---

un instrument de musique produisant un son. En effet, les différentes formes du conduit vocal produisent des sons différents. La cavité buccale joue le rôle majeur pour former les différentes formes du conduit vocal [5]. Pour produire des sons nasaux, la cavité nasale est souvent incluse dans le conduit vocal, elle est connectée en parallèle avec la cavité buccale. Une coupe du conduit vocal est montré dans la Figure 1.1:

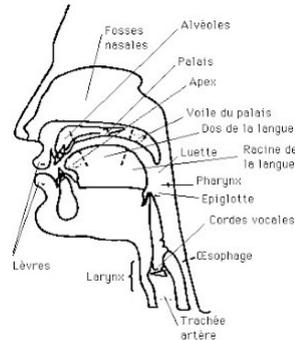


Fig 1.1: Coupe de l'appareil phonatoire

### 1.2.1 Son voisé et non voisé

Quand le conduit vocal est excité par des impulsions périodiques de pression, résultantes des oscillations des cordes vocales, la pression accumulée puis libérée inopinément par l'ouverture de la glotte, crée des sons appelés voisé, ce sont des sons qui constituent entre autres les voyelles. Le spectre d'un tel son est esquissé à la Figure 1.2

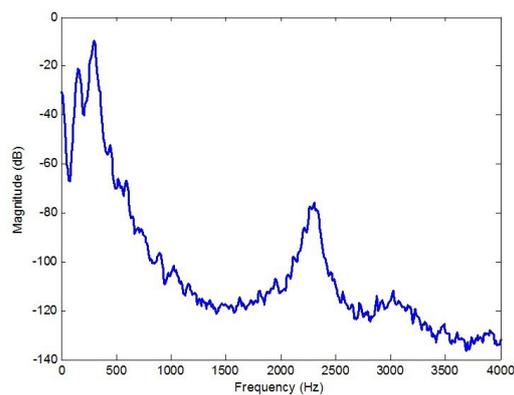


Fig 1.2: Le spectre d'un son voisé

Il est caractérisé par des pics épars, le premier correspond à la fréquence fondamentale, les autres à des fréquences appelées formants. Les trois premiers formants sont nécessaires

## 1.2. La Production de la voix

---

pour décrire un spectre vocal, les formants d'ordres supérieurs ont des applications diverses telles que la reconnaissance de voix.[6]

Le resserrement du conduit vocal entraîne des sons semblables à des consonnes, en plus de ce resserrement, les cordes vocales n'entrent pas en vibrations, elles restent écartées, c'est la raison pour laquelle ces sons sont aperiodiques, ils sont généralement assimilés à un bruit blanc à la sortie d'un filtre constitué par la partie du conduit vocal sise entre la constriction et les lèvres.[6]

Le spectre d'un son non voisé ne possède pas de pitch à la différence du voisé, mais il garde quelques attributs de ce dernier, en ce qui concerne sa structure formantique. Le spectre d'un tel son est donné par la Figure 1.3.

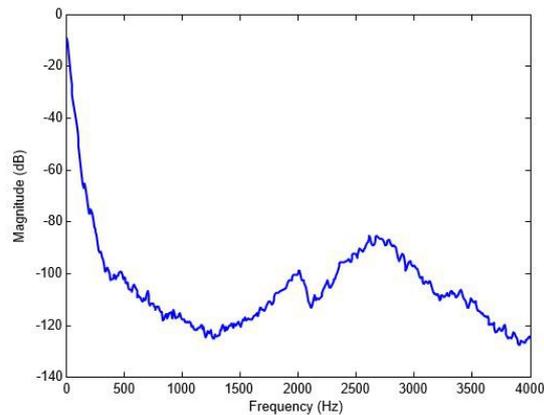


Fig 1.3: Le spectre d'un son non voisé

### 1.2.2 Propriétés de Perception

Lors de la transmission d'un signal de parole, la compression de celui-ci entraîne une perte d'information irréversible, mais l'oreille humaine reste sensible au signal de parole synthétisé après la réception, en effet, la gamme de fréquence entre 200-3400 Hz est la plus importante pour l'intelligibilité de la parole, et c'est à cette bande que l'oreille est plus sensible. Lors de la réalisation d'algorithmes de codages des signaux de parole, il est indispensable de prendre en compte les caractéristiques du système auditif, parmi les caractéristiques exploitables pour la conception de codeurs de parole, on évoque :

- La sensibilité de phase : elle peut être négligée, car l'oreille est surtout sensible à la parole par le biais des informations du spectre d'amplitude.

- La perception de la forme spectrale : pour cette caractéristique la sensibilité aux pics est plus importante que celle des vallées, c'est pour cette raison, qu'il faut modéliser en priorité les formants du signal.
- Masquage de fréquences : un masque de fréquence dont la forme est analogue à l'enveloppe spectrale du signal de parole est utilisée pour éliminer tout bruit inférieur à un certain seuil, cette caractéristique est exploitée par certains algorithmes pour compresser le bruit en fonction du seuil dans le but de réduire la distorsion perceptive audible.

### 1.3 Modélisation de Processus de la Production de la Parole

Le processus de production de la parole peut être modélisé par le système montré dans la Figure 1.4 Un signal voisé peut être modélisé par le passage d'un train d'impulsions  $e(n)$  à travers un filtre numérique récuratif de type tous pôles. On montre que cette modélisation reste valable dans le cas du son non-voisé, à condition que  $e(n)$  soit cette fois un bruit blanc. Il est souvent appelé modèle autorégressif. Les paramètres du modèle AR sont : la période du train d'impulsions (sons voisés uniquement), la décision Voisé/Non Voisé (V/NV), le gain  $G$ , et les coefficients du filtre  $1/A(z)$ , appelé filtre de synthèse [7].

Dans les deux cas (sons voisés ou non voisés), le facteur du gain contrôle l'intensité de l'excitation. Dans le domaine temporel, le signal de parole est la convolution de l'excitation et la réponse impulsionnelle du système linéaire de génération de la parole. Dans le domaine spectral, le spectre de la parole généré, est le produit simple des spectres de l'excitation et du système linéaire. Les différents sons sont produits par ce modèle, en changeant la source de l'excitation et les configurations du système linéaire.

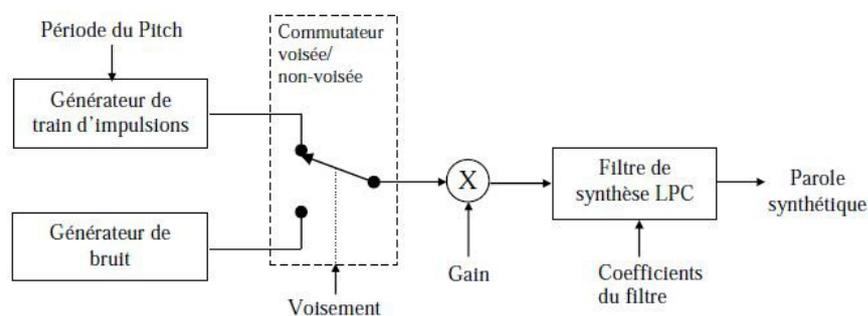


Fig 1.4: Modèle simplifié de production de la parole

## 1.4 La prédiction Linéaire

La prédiction linéaire, est l'une des techniques fondamentales pour enlever la redondance d'un signal. Elle consiste à prédire la valeur d'un échantillon  $s(n)$  à partir d'une combinaison linéaire de  $M$  échantillons passés  $s(n-i)$ . La séquence estimée  $\hat{s}(n)$  est:

$$\hat{s}(n) = a_1s(n-1) + a_2s(n-2) + \dots + a_Ms(n-M) = \sum_{i=1}^M a_i s(n-i) \quad (1.1)$$

La différence entre l'échantillon actuel et l'échantillon prédit est appelée erreur de prédiction, exprimée par :

$$e(n) = s(n) - \hat{s}(n) = s(n) - \sum_{i=1}^M a_i s(n-i) \quad (1.2)$$

Le problème revient à déterminer les coefficients de prédiction ( $a_i$ ).

### 1.4.1 Minimisation d'Erreur de Prédiction

Les meilleurs coefficients de prédiction sont obtenus par la minimisation de la valeur quadratique moyenne de l'erreur de prédiction. On cherche donc les coefficients  $a_i$  qui minimisent la puissance de l'erreur de prédiction définie par:

$$E = \sum_n e^2(n) = \left( s(n) - \sum_{i=1}^M a_i s(n-i) \right)^2 \quad (1.3)$$

Ici le signal  $s(n)$  est supposé égal à zéro pour  $n < 0$  et  $n > N$  (par exemple, en le multipliant par une fenêtre d'une durée finie), la minimisation de l'erreur ( $\frac{\partial E}{\partial a_i} = 0$ ,  $1 \leq i \leq M$ ), conduit au système d'équations linéaires de Yule-Walker [8]:

$$\sum_{k=1}^M a_k R(|i-k|) = R(i) \quad 1 \leq i \leq M \quad M: \text{Ordre de prédiction} \quad (1.4)$$

Où

$$R(k) = \sum_{n=0}^{N-1-k} s(n)s(n+k), \text{ est la fonction d'autocorrélation du signal de parole } s(n).$$

L'équation (1.4) correspond à l'équation matricielle:

$$\begin{bmatrix} R[0] & R[1] & R[2] & \dots & R[M-1] \\ R[1] & R[0] & R[1] & \dots & R[m-2] \\ R[2] & R[1] & R[0] & \dots & R[m-3] \\ \vdots & \dots & \ddots & \dots & \vdots \\ R[M-1] & R[M-2] & R[M-3] & \dots & R[0] \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_m \end{bmatrix} = \begin{bmatrix} R[1] \\ R[2] \\ R[3] \\ \dots \\ R[M] \end{bmatrix} \quad (1.5)$$

La matrice d'autocorrélation est une matrice dite Toeplitz symétrique. La résolution de ce système est couramment réalisée par l'algorithme de Levinson-Durbin. C'est un algorithme qui résout le système en un nombre restreint d'opérations sans calculer l'inverse de la matrice d'autocorrélation [5].

### 1.4.2 Algorithme de Levinson-Durbin

$a_k^{(i)}$  c'est la  $k^{\text{ème}}$  valeur du coefficient de LPC après une  $i^{\text{ème}}$  itération  
 $E^{(i)}$  c'est l'erreur résiduel après une  $i^{\text{ème}}$  itération

#### Étapes

1. Initialiser:

$$E^{(0)} = R(0)$$

2. Calcule des coefficients  $k_i$

$$k_i = \frac{[R(i) - \sum_{j=1}^{i-1} a_j^{(i-1)} R(i-j)]}{E^{(i-1)}}$$

3. mettre  $a_i^{(i)} = k_i$  et  $a_j^{(i)} = a_j^{(i-1)} - k_i a_{i-j}^{(i-1)}$  pour  $1 \leq j \leq i$

4. Calculer  $E^{(i)} = (1 - k_i^2) E^{(i-1)}$

5. Répéter 2,3 et 4 jusqu'à ce que  $i=p$  (l'ordre de la prédiction).

Une fois les coefficients de prédiction sont calculés, l'équation 1.1 peut être utilisée pour trouver la séquence d'erreurs Figure 1.5

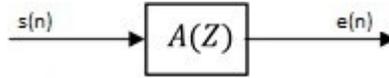


Fig 1.5: Filtre d'analyse

La fonction de transfert du filtre d'analyse (Filtre inverse) est donnée par :

$$A(z) = 1 - \sum_{i=1}^M a_i z^{-i} \quad (1.6)$$

La Figure 1.6 représente un exemple d'erreur de prédiction, aussi nommée excitation ou signal résiduel calculée à partir du filtre d'analyse.

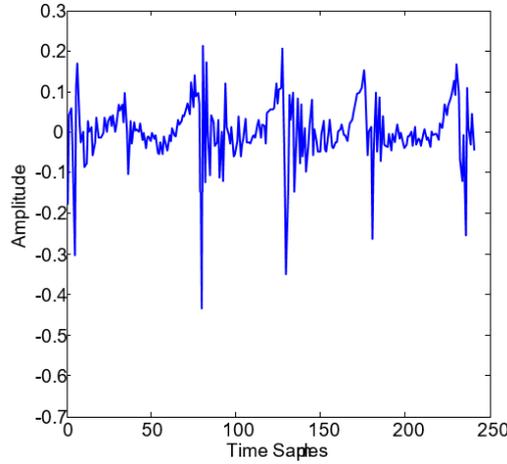


Fig 1.6: La séquence du signal résiduel

Les échantillons  $e(n)$  ayant une variance plus faible que le signal de parole, ils nécessitent moins de bits pour être représentés.

Le signal de parole peut être reconstitué à partir des coefficients de prédiction et du signal résiduel en utilisant le filtre de synthèse qui modélise le conduit vocal du locuteur Figure 1.7.

On peut réécrire l'équation 1.2 sous la forme :

$$s(n) = \sum_{i=1}^M a_i s(n-i) + e(n) \quad (1.7)$$

L'entrée et la sortie de ce filtre sont respectivement  $e(n)$  et  $s(n)$ :

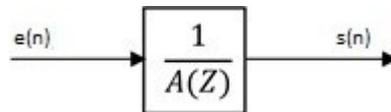


Fig 1.7: Filtre de synthèse

La fonction de transfert du filtre de synthèse (Filtre LPC) est donnée par :

$$H(z) = \frac{1}{(1 - \sum_{i=1}^M a_i z^{-i})} \quad (1.8)$$

La Figure 1.8 représente l'enveloppe spectrale de 30ms d'une voyelle , calculée par l'utilisation de la méthode d'autocorrélation avec  $M=10$ , la fenêtre de Hamming est utilisée.

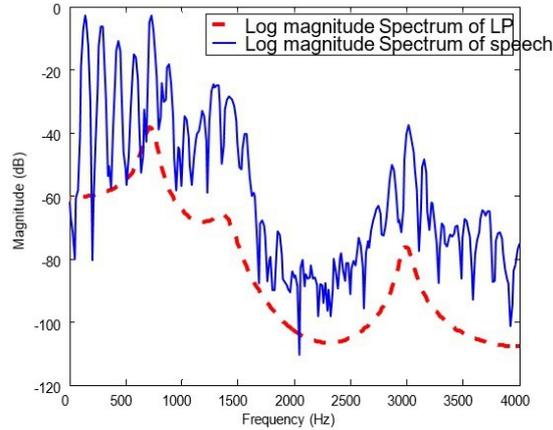


Fig 1.8: spectre d'amplitude du filtre LPC

Le choix de l'ordre de prédiction  $M$ , résulte d'un compromis. Il doit être suffisamment élevé pour reproduire correctement la structure formantique du signal de parole ; un ordre de 8 est nécessaire pour créer 4 pics dans le spectre et on a vu que le signal de parole comporte généralement 4 formants. Inversement, l'ordre doit être le plus faible possible pour économiser le débit. On choisit donc  $M$  compris entre 8 et 16 (10 dans notre cas).

## 1.5 Représentation des Coefficients de Prédiction

Dans les applications de codage de parole, il est nécessaire de quantifier les paramètres LPC avec un minimum de distorsion. Aussi, il est exigé que le filtre de synthèse reste stable après la quantification de ces paramètres. La quantification directe des coefficients LPC n'est pas conseillée car les petites erreurs de quantification dans ces coefficients peuvent produire des erreurs spectrales relativement grandes, et peuvent causer l'instabilité du filtre  $H(z)$ . Par conséquent, il est nécessaire d'utiliser un grand nombre de bits pour accomplir une bonne quantification des paramètres LPC eux-mêmes. Par l'utilisation de 6 bits par coefficient (c-à-d 60 bits par trame) pour une quantification scalaire de coefficients LPC dans la base de données FM, 25.5% des filtres sont instables, et la distorsion spectrale moyenne est de 1.83 dB [9]. Afin de pallier ces difficultés, des paramètres mathématiquement équivalents aux coefficients LPC d'ordre  $M$  sont calculés. Ces nouveaux coefficients préservent toute l'information des coefficients LPC mais possèdent des propriétés particulières.

### 1.5.1 Les Coefficients de Réflexion

Les coefficients de réflexion (RCs) peuvent être obtenus à partir des coefficients LPC par l'utilisation de l'algorithme de Levinson -Durbin. Ces coefficients ont deux avantages majeurs par rapport aux coefficients LPC :

- Ils sont moins sensibles spectralement à la quantification.
- La stabilité du filtre tout pôles peut être assurée en gardant chaque coefficient dans la gamme de -1 à +1 pendant le processus de quantification [6].

### 1.5.2 Les Coefficients (LARs) et (ASRC)

La distribution statistique des coefficients de réflexion est très loin d'une loi uniforme, il est plus simple de quantifier un scalaire présentant une distribution relativement uniforme. Cependant, cet inconvénient peut être vaincu par l'usage d'une transformation non-linéaire appropriée pour uniformiser leur distribution.

- LAR (Log Area Ratio), ou rapport d'aires logarithmiques, sont déduits des coefficients RCs après transformation non-linéaire, et sont utilisés pour leurs bonnes propriétés de quantification linéaire, ils ont un rapport avec les fonctions des surfaces du tube vocal.

$$LAR_i = \ln \left( \frac{1 + k_i}{1 - k_i} \right) \quad 1 \leq i \leq M \quad (1.9)$$

- Une deuxième transformation, est la transformation en sinus inverse (ASRC)

$$J_i = \sin^{-1}(k_i) \quad (1.10)$$

Les représentations LAR et ASRC exigent approximativement 32 bits par trame pour fournir une distorsion spectrale moyenne de 1 dB [9].

### 1.5.3 Les Fréquences de Raies Spectrales (LSF)

Parmi les représentations équivalentes aux coefficients de prédiction, les Fréquences de Raies Spectrales (Paramètres LSF) ont les meilleures caractéristiques de quantification. Une interprétation physique de ces paramètres permet d'établir un lien avec les formants.

A partir du polynôme  $A(z)$  d'ordre  $M$ , on construit deux nouveaux polynômes d'ordre  $M+1$  :

$$\begin{aligned} P(Z) &= A(Z) + Z^{-(M+1)}A(Z^{-1}) \\ &= 1 - (a_1 + a_M)Z^{-1} - (a_2 + a_{M-1})Z^{-2} - \dots - (a_M + a_1)Z^{-M} + Z^{-(M+1)} \end{aligned} \quad (1.11)$$

$$\begin{aligned} Q(Z) &= A(Z) - Z^{-(M+1)}A(Z^{-1}) \\ &= 1 - (a_1 - a_M)Z^{-1} - (a_2 - a_{M-1})Z^{-2} - \dots - (a_M - a_1)Z^{-M} - Z^{-(M+1)} \end{aligned} \quad (1.12)$$

Ce qui donne:

$$A(Z) = \frac{[P(Z) + Q(Z)]}{2} \quad (1.13)$$

On peut montrer que ces deux polynômes ont les propriétés suivantes:

- Le polynôme  $P(z)$  est un polynôme symétrique. Le polynôme  $Q(z)$  est un polynôme antisymétrique.
- Si toutes les racines de  $A(z)$  sont à l'intérieur du cercle unité toutes les racines de  $P(z)$  et de  $Q(z)$  sont sur le cercle unité.
- Les racines de  $P(z)$  et de  $Q(z)$  apparaissent de façon alternée sur le cercle unité. Si  $M$  est paire,  $P(z)$  a pour racine évidente  $-1$  et  $Q(z)$  a pour racine évidente  $+1$ .

L'expression des paires de racines complexes conjuguées des polynômes  $P(z)$  et  $Q(z)$  est donnée par :

$$(1 - e^{j\omega_i} Z^{-1})(1 - e^{-j\omega_i} Z^{-1}) = 1 - 2Z^{-1} \cos(\omega_i) + Z^{-2} \quad (1.14)$$

Avec  $\omega_i$  étant la fréquence de la raie spectrale.  $P(z)$  et  $Q(z)$  peuvent être factorisés comme suit [10] :

$$\begin{aligned} P(Z) &= (1 + Z^{-1}P'(Z)) \\ &= (1 + Z^{-1}) \prod_{i=1,3,\dots,\frac{M}{2}-1} (1 - 2Z^{-1} \cos(\omega_i) + Z^{-2}) \end{aligned} \quad (1.15)$$

et

$$\begin{aligned} Q(Z) &= (1 - Z^{-1})Q'(Z) \\ &= (1 - Z^{-1}) \prod_{i=2,4,\dots,\frac{M}{2}} (1 - 2Z^{-1} \cos(\omega_i) + Z^{-2}) \end{aligned} \quad (1.16)$$

Les paramètres  $\omega_1 \dots \omega_M$  sont rangés en ordre décroissant:

$$0 < \omega_1 < \omega_2 < \dots < \omega_{M-1} < \omega_M < \pi$$

Le carré de la réponse en amplitude du filtre de synthèse est donné par:

$$\begin{aligned}
 |H(\theta)|^2 &= \frac{1}{|A(\omega)|^2} \\
 &= 2^2 |P(\omega) + Q(\omega)|^{-2} \\
 &= 2^{-M} \left[ \cos^2\left(\frac{\omega}{2}\right) \prod_{i=1,3,\dots,\frac{M}{2}-1} (\cos(\omega) - \cos(\omega_i))^2 + \sin^2\left(\frac{\omega}{2}\right) \prod_{j=2,4,\dots,\frac{M}{2}} (\cos(\omega) - \cos(\omega_j))^2 \right]^{-1}
 \end{aligned} \tag{1.17}$$

Le premier terme à l'intérieur des crochets dans l'équation (1.17) se rapproche de 0 quand  $\omega$  se rapproche de  $\pi$  ou une des  $\omega_i$  ( $i = 1, 3, \dots, \frac{M}{2} - 1$ ), et le deuxième terme se rapproche de 0 quand  $\omega$  se rapproche de  $\frac{\pi}{2}$  ou une des  $\omega_j$  ( $j = 2, 4, \dots, \frac{M}{2}$ ) se rapproche de 0. Par conséquent, lorsque deux paramètres LSP  $\omega_i$  et  $\omega_j$  se rapprochent, le gain de  $H(z)$  devient grand et une résonance se produit. C'est pourquoi, le spectre de parole est relié directement avec les paramètres LSP [10]. S'il y'a une erreur causée par la quantification d'un paramètre LSP, l'erreur est localisée.

On peut transformer les LSP (positions angulaires) en LSF dans le domaine des fréquences normalisées  $f_i$  ( $0 \leq f_i \leq 0.5$ ),  $f_i = \frac{\omega_i}{2\pi}$ , ou en fréquence réelles  $F = \frac{\omega_i}{2\pi} F_e$ , où  $F_e$  est la fréquence d'échantillonnage ( $0 \leq F \leq F_e/2$ ),  $\frac{\omega_i}{2\pi}$ .

### Algorithme de Calcul

Les paramètres LSF peuvent être calculés par la procédure suivante [10] :

1. Trouver les coefficients de prédiction linéaire LPC.
2. Former  $P(z)$  et  $Q(z)$ .
3. Calculer les zéros du polynôme symétrique " $P(z)$ " et asymétrique " $Q(z)$ ". la règle de Descartes peut être utilisé pour ceci.
4. Test des cas de mal-conditionnement.

La propriété :  $0 < \omega_1 < \omega_2 < \dots < \omega_M < \pi$ , permet de réduire considérablement la complexité de recherche des paramètres LSF.

## 1.6 Conversion LSF-LPC

Une fois quantifiés, les coefficients LSF sont reconvertis en coefficients LPC,  $a_i$ . Pour cela, on détermine les coefficients des polynômes  $P'(z)$  et  $Q'(z)$  par expansion des équations (1.19), sur la base des coefficients LSF quantifiés [11].

$$P' = \frac{P(Z)}{(1 + Z^{-1})}, \quad Q' = \frac{Q(Z)}{(1 - Z^{-1})} \quad (1.18)$$

A partir des coefficients  $q_i (q_i = \cos(\omega_i))$ , on calcule les coefficients  $p'(i), i = 1, \dots, M/2$ , par la relation de récurrence suivante :

$$\begin{aligned} &\text{For } i = 1 \text{ to } 5 \\ &\quad P' = -2q_{i-1}p'(i-1) + 2p'^{(i-2)} \\ &\text{For } j = i-1 \text{ to } 1 \\ &\quad p'(j) = p'(j) - 2q_{2i-1}p'(j-1) + p'(j-1) \\ &\text{End } i \\ &\text{End } j \end{aligned} \quad (1.19)$$

avec les valeurs initiales  $p'(0) = 1$  et  $p'(-1) = 0$ . Les coefficients  $q'(i)$  sont calculés de la même manière, avec remplacement des  $q_{2i-1}$  par  $q_{2i}$ .

Une fois les coefficients  $p'(i)$  et  $q'(i)$  calculés, on multiplie les polynômes  $P'(z)$  et  $Q'(z)$  par  $(1 + Z^{-1})(1 - Z^{-1})$  respectivement, pour obtenir les polynômes  $P(z)$  et  $Q(z)$ , qui donnent les coefficients suivants :

$$\begin{aligned} p(i) &= p'(i) + p'(i-1) \quad i = 1, \dots, 5. \\ q(i) &= q'(i) + q'(i-1) \quad i = 1, \dots, 5. \end{aligned} \quad (1.20)$$

Finalement, on retrouve les coefficients LPC à partir des coefficients  $p(i)$  et  $q(i)$  comme suit :

$$a_i = \begin{cases} 0.5p(i) + 0.5q(i) \\ 0.5p(11-i) + 0.5q(11-i) \end{cases} \quad (1.21)$$

## 1.7 Détecteur de Voisement

Le rôle du détecteur de voisement est de classer les trames de parole voisée /non voisée. La mesure de, l'énergie, et le taux de passage par zéro sont les méthodes les plus utilisées.

### 1.7.1 Énergie et Taux de Passage par Zéro

L'énergie du  $j^{\text{ème}}$  trame du signal de parole est :  $E = \sum_{n=0}^{N-1} s_j^2(n)$  où  $s_j(n)$  et l'échantillon  $n$  dans la trame  $j$ . Habituellement l'énergie d'une trame voisée est plus grande que celle

d'une trame non voisée. Le taux de passage par zéro est obtenu en comptant le changement de signe dans les échantillons successifs de la parole. Le ZCR (Zero-Crossing Rate) du son voisé est inférieur à celui du son non voisé

$$ZCR = \frac{1}{2} \sum_{n=0}^{N-1} |sgn(s(n)) - sgn(s(n-1))| \quad (1.22)$$

### 1.7.2 Détection du Pitch

La détection du pitch a été pour longtemps une partie active de recherche à cause de son importance dans la synthèse de la parole. Il y'a plusieurs algorithmes de détection de pitch, qui ont leurs propres avantages et inconvénients. En général, les méthodes qui sont utilisées communément. Elles sont: méthode par AMDF, méthode d'auto-corrélation, méthodes temporelles et méthodes fréquentielles [5].

## Conclusion

Dans ce chapitre nous avons présenté les fondements théoriques et pratiques de la modélisation du système de production de la parole, et principalement l'estimation des paramètres LPC et les différentes représentations de ces derniers. Les techniques de codage de la parole font l'objet du chapitre suivant.

# Techniques de Codage de la Parole

## 2.1 Introduction

Dans ce chapitre nous allons présenter brièvement les notions de base et les techniques du codage de la parole relatives au codage source. Le nombre de codeurs différents étant très vaste, nous ne tenterons pas de récapituler tous les codeurs existants ou de les présenter en détail. Nous allons nous borner aux techniques les plus répandues. On conclura le chapitre par les différents critères couramment utilisés pour juger et classer les méthodes de codage.

## 2.2 Techniques de Codage

Le codage de la parole est une issue incontournable pour représenter le signal de parole numérisé en un nombre minimal de bits, en maintenant une assez bonne qualité de ce signal.

Avec l'émergence des nouvelles techniques de numérisation à la fin des années soixante et le besoin croissant d'adapter le débit de transmission aux capacités du canal et d'améliorer les systèmes de sécurité et d'archivage. Le codage de la parole a connu un essor remarquable, en effet de nombreux travaux ont été effectués afin de maximiser le compromis entre l'efficacité, le coût et la qualité des systèmes de communication.

Dans cette section, on décrira brièvement les différentes techniques de codage, et cela par ordre chronologique, en évoquant quelques notions sur les quantifications scalaire et vectorielle, qui jouent un rôle rudimentaire dans les systèmes de communication numérique.

### 2.2.1 Quantification Scalaire

La quantification est la procédure qui a pour but de représenter un ensemble d'éléments avec un autre ensemble plus petits, donc c'est un système qui arrondit la valeur d'une entrée en lui associant une autre parmi un nombre fini de valeurs.

Un quantificateur scalaire, d'un point de vue mathématique, établit une application subjective entre un nombre réel  $x \in R$  et un nombre  $y_i \in Y$  où  $Y$  est un ensemble fini à  $N$  éléments :

$$Q : R \rightarrow Y, \quad (y_1, y_2, \dots, y_N) \in Y, \quad \text{Donc } Q(x) = y_i : x \in R; \quad i = 1, \dots, N$$

En traitement du signal, l'ensemble  $Y$  est appelé dictionnaire de  $N$  niveaux. On peut assimiler la quantification scalaire à un système, qui à une entrée qui peut prendre une valeur quelconque, et suivant son mode de fonctionnement, présente à sa sortie, une autre valeur adéquate parmi  $N$  valeurs disponibles.

Un quantificateur scalaire peut être considéré comme la combinaison de deux applications successives,  $E$  qui joue le rôle d'un codeur et  $D$  qui joue celui d'un décodeur[11] :

$$E : R \rightarrow I \quad , \quad D : I \rightarrow R$$

Avec  $I$  l'ensemble des indices du dictionnaire  $\{1, 2, \dots, N\}$ , ainsi  $Q(x) = y_i$  alors  $E(x) = i$  et  $D(i) = y_i$

De plus:

$$\hat{x} = Q(x) = D(E(x)) = y_i \tag{2.1}$$

Soit  $Q$  une opération de quantification et  $\hat{x}$  la valeur quantifiée, lors d'une telle opération, l'introduction d'une erreur est inévitable, celle-ci vaut :

$$x - \hat{x} = x - Q(x) = e \tag{2.2}$$

$e$  est appelée erreur ou bruit de quantification.

Le critère de choix de la valeur de  $\hat{x} = Q(x)$  est la minimisation de la distance  $d(x, Q(x))$ , telle que la moyenne de distorsion est donnée par [13] :

$$D = \sum_{i=1}^N \int_{q_{i-1}}^{q_i} d(x, Q(x)) p_x(x) dx \tag{2.3}$$

Où:

$x$  est la valeur à quantifier,  $Q(x)$  et la valeur quantifier.

$d(x, Q(x))$  est la distance euclidienne.

$p_x(x)$  la densité de probabilité de  $x$ .

$N$  et le nombre de niveaux du quantificateur et  $q_i$  sont les bornes du quantificateur.

La résolution d'un quantificateur scalaire définit le nombre de bits nécessaire pour représenter la valeur à quantifier, elle est donnée par :

$$r = \log_2 N \quad (2.4)$$

Les performances d'un quantificateur scalaire peuvent être mesurées par le rapport signal sur bruit SNR défini par :

$$SNR = 10 \log_{10} \frac{\sigma_x^2}{D} \quad (2.5)$$

Où  $\sigma_x^2$  représente la variance de  $x$ .

### La Quantification scalaire uniforme

Le quantificateur scalaire uniforme à débit fixe est entièrement déterminé par :

- Les  $1 + L$  niveaux de décisions :  $x_0, x_1, \dots, x_L$ , qui partitionnent en  $L$  intervalles égaux l'axe des réels  $R$  et déterminent le pas de quantification.
- Les  $L$  valeurs de reproduction :  $y_1, y_2, \dots, y_L$ , qui sont les centres de masse de chacun des intervalles de décision.

Il apparait deux sortes d'erreurs ou bruits de quantification :

- \* Le bruit granulaire qui se produit lorsque la valeur d'entrée  $x$  se situe dans l'un des intervalles  $[x_i, x_{i+1}]$ , l'erreur résultante qui est la différence entre  $x$  et la valeur quantifiée  $Q(x)$  ne peut être supérieure à un demi pas de quantification. Le pas de quantification étant défini par:

$$\Delta = y_{i+1} - y_i, \quad i = 1, 2, \dots, L - 1 \quad (2.6)$$

- \* Le bruit de surcharge ou de dépassement qui se produit lorsque la valeur d'entrée se situe hors de l'intervalle  $[x_0, x_L]$ . La valeur de reproduction est alors :  $y_1$  ou  $y_L$ , et l'erreur de quantification peut être supérieure à un demi pas de quantification.

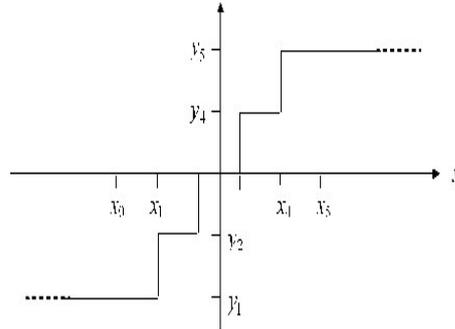


Fig 2.1: Quantification scalaire uniforme avec cinq niveau L=5

### 2.2.2 Quantification Vectorielle

La quantification vectorielle est une extension de la quantification scalaire à un espace multidimensionnel, en termes moins hermétiques, il s'agit de quantifier conjointement des échantillons du signal d'entrée, cela diminuera à la fois la distorsion et le nombre de bits nécessaires pour quantifier et coder le signal. Un tel quantificateur est amené donc à considérer l'entrée non pas comme une valeur individuelle comme le cas du quantificateur scalaire mais comme un vecteur d'une certaine dimension, dont les composants ne sont rien d'autre que les échantillons du signal en question. La dimension de ce vecteur d'entrée détermine la dimension du quantificateur vectorielle.

#### Principes

Mathématiquement, un quantificateur vectoriel est une application qui assigne à chaque vecteur  $x$  appartenant à un espace euclidien à  $M$  dimensions un vecteur à  $M$  dimensions appartenant à un ensemble de  $N$  vecteurs de  $M$  dimensions.

$$Q : R^M \rightarrow Y$$

Avec

$$\begin{aligned} x &= [x_1, x_2, \dots, x_M]^T \\ (y_1, y_2, \dots, y_n) &\in Y \\ y_i &= [y_{i1}, y_{i2}, \dots, y_{iM}]^T; \quad i = 1, 2, \dots, N. \end{aligned}$$

En traitement du signal, l'ensemble  $Y$  est appelé dictionnaire du quantificateur vectoriel, d'après la définition mathématique, on peut écrire :

$$Q(x) = y_i; \quad i = 1, 2, \dots, N \tag{2.7}$$

La résolution d'un quantificateur vectoriel mesure le nombre de bits requis pour représenter un vecteur du dictionnaire, elle est donnée par :

$$r = \log_2(N) \quad (2.8)$$

Pour le choix du code vecteur le plus adéquat, un critère de décision est utilisé, celui-ci diffère d'un type de quantificateur à un autre, selon le besoin en précision et dimensions.

On définit la mesure de distorsion :

$$d(x, Q(x)) = \begin{cases} 0 & , Q(x) = x \\ > 0 & , \text{ailleurs} \end{cases} \quad (2.9)$$

En général, la mesure de distorsion est donnée par l'erreur quadratique :

$$d(x, Q(x)) = \|x - Q(x)\|^2 = \sum_{j=1}^M (x_j - \hat{x}_j)^2 \quad (2.10)$$

La distorsion moyenne vaut alors :

$$D = E[d(X, Q(X))] = \int_{R^M} d(x, Q(x)) f_x(x) dx \quad (2.11)$$

Avec :

X est un vecteur de distribution aléatoire à M dimension

$f_x(x)$  est la fonction de densité de probabilité de X.

On définit le quantificateur vectoriel du plus proche voisin (Nearest Neighbor Quantizer): comme celui qui définit la région de répartition donnée par :

$$R_i = \{x : d(x, y_i) \leq d(x, y_j); \quad \forall j \in I; i \in I\} \quad (2.12)$$

D'après cette définition, la région  $R_i$  contient les vecteurs x qui donne la plus petite distorsion quand ils sont quantifiée au code vecteur par rapport aux autres code vecteurs.

### 2.2.3 L'encodeur

Le rôle de l'encodeur 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 par une fonction de distance entre les deux vecteurs x,  $y_i: d(x, y_i)$ .

Dans le cas de la distance Euclidienne on a :

$$d(x, y_i) = \sum_{j=0}^{k-1} (x_j - y_{i,j})^2 \quad (2.13)$$

## 2.2. Techniques de Codage

---

où  $x = [x_0, x_1, \dots, x_{k-1}]^T$  et  $y_i = [y_{i,0}, y_{i,1}, \dots, y_{i,k-1}]^T$ , Les régions de Voronoi sont données par:

$$R_j = \{x \in R^k / Q(x) = y_i, \text{ si } d(x, y_i) \leq d(x, y_j), \forall j \neq i\} \quad (2.14)$$

Avec:

$$R_i \cap R_j = \Phi, \quad i \neq j, \quad \forall i = 0, 1, \dots, N-1, \quad \forall j = 0, 1, \dots, N-1. \quad (2.15)$$

Tout les vecteurs  $x$  de  $R^k$  qui appartiennent à la région  $R_i$  sont représentés par le même vecteur  $y_i$  du dictionnaire  $Y$ .

Soit  $I = \{0, 1, \dots, N-1\}$  l'ensemble des indices correspondant au dictionnaire  $Y$  et  $y_i$  ( $i \in I$ ) l'ensemble des vecteurs de  $Y$ . La compression d'information est réalisée à ce niveau car c'est uniquement l'indice du vecteur-code  $i$  y minimisant le critère de distorsion qui sera transmis ou stocké au lieu du vecteur lui même. Cette opération, qui perd de l'information, fait que la QV est une opération irréversible, le signal original ne pourra plus être restitué.

### 2.2.4 Le décodeur

Le décodeur est considéré comme un récepteur dont la tâche est la reconstruction du vecteur source. Pour cela, il dispose d'une réplique du dictionnaire qu'il consulte afin de restituer le vecteur-code correspondant à l'indice reçu. Le décodeur réalise donc l'opération de décompression. On donne à la Figure 2.2 le schéma général d'un quantificateur vectoriel.

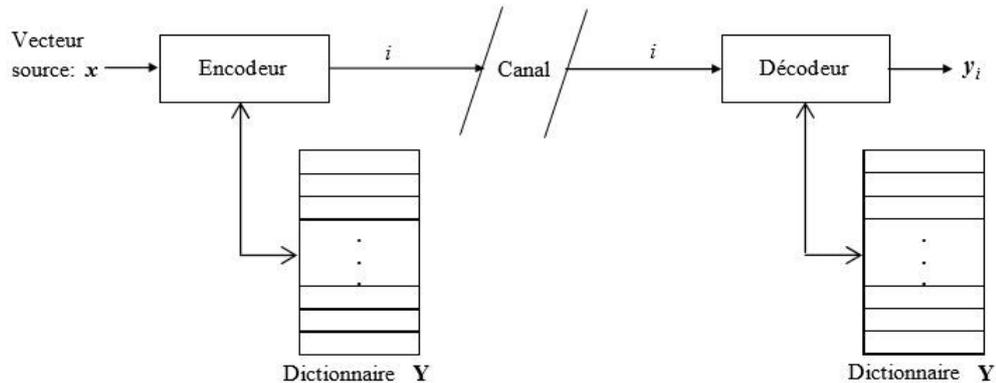


Fig 2.2: Schéma général d'un quantificateur vectoriel

### Quantificateur optimal

Admettant qu'une distance  $d$  a été sélectionnée, un quantificateur est dit optimal, si cette distance minimise au maximum la distorsion moyenne  $D$ .

L'optimalité peut être obtenue en choisissant les codes vecteurs  $y_i$  et les régions de décision  $R_i$  qui minimise la distorsion moyenne  $D$ , pour une densité de probabilité du vecteur aléatoire  $X$  donnée.

**Centroïde** : On définit le centroïde  $\text{cent}(R_0)$ , d'un sous-ensemble non vide  $R_0 \in R^M$ , par le vecteur  $y_0$  (s'il existe), qui minimise la distorsion entre un point  $X \in R_0$  et  $y_0$ , moyenné sur la distribution de probabilité de  $X$  :

$$\text{cent}(R_0) = \{y_0 : E\{d(X, y_0) | X \in R_0\} \leq E\{X, y\} | X \in R_0\}, \quad \forall y \in R^M \quad (2.16)$$

Pour une région de décision donnée, un code vecteur optimal satisfait la relation :

$$y_i = \text{cent}(R_i) \quad (2.17)$$

Dans le codage hybride de la parole qu'on va voir après, et lors de la quantification vectorielle, la densité de probabilité est inconnue, la conception d'un quantificateur vectoriel nécessite alors une séquence d'apprentissage, celle-ci est découpée en  $N$  régions de décision dans  $R^M$ , ou le centroïde de chaque régions forme le dictionnaire, ces régions contenant les centroïdes sont appelées les régions de Voronoi  $R_i$ .

La méthodologie globale pour concevoir un dictionnaire de taille  $N$  est :

1. Commencer avec un dictionnaire initial et calculer la distorsion moyenne.
2. Trouver  $R_i$ .
3. Calculer la distorsion moyenne.
4. Si la distorsion moyenne diminue moins d'un seuil donné, arrêter. Autrement, aller à l'étape 2.

### Quantification par Split

La quantification par split consiste à diviser un vecteur  $x$  en sous vecteurs de dimensions inférieures, cela réduit la complexité de recherche et de stockage dans le dictionnaire. Dans le cas de deux divisions le vecteur d'entrée est :

$$\begin{aligned} x &= [x_1, x_2, \dots, x_M]^T \text{ est divisé en:} \\ x_a &= [x_1, x_2, \dots, x_k]^T, \text{ avec } k < M \text{ et} \\ x_b &= [x_{k+1}, x_{k+1}, \dots, x_M]^T \end{aligned}$$

L'erreur quadratique qui mesure la distorsion de vecteur  $x$  est:

$$\|x - \hat{x}\|^2 = \|x_a - \hat{x}_a\|^2 + \|x_b - \hat{x}_b\|^2 \quad (2.18)$$

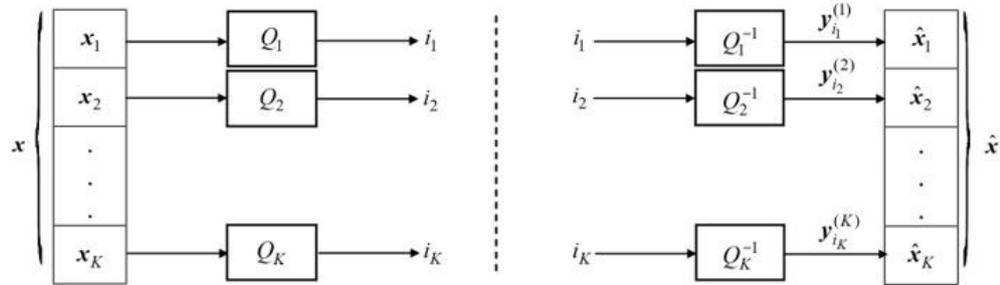


Fig 2.3: Encodeur et décodeur d'une quantification SVQ

Après la division du vecteur  $x$  en sous vecteurs, il est évident qu'on doit créer autant de dictionnaires de dimensions égales à celles des vecteurs, chaque vecteur sera simplement quantifié par le dictionnaire ayant la même dimension que lui, en lui octroyant le code vecteur le plus proche.

### 2.2.5 Codage de forme d'onde

Le codage de forme d'onde consiste à reproduire un signal à partir d'un signal original en tendant à préserver la forme de celui-ci avec une modélisation numérique (quantification), cela sans avoir une information au préalable sur la manière dont le signal original a été généré, c'est un codage qui garde une assez bonne qualité pour des débits supérieurs à 16Kb/s.

Dans la pratique il est très convenable de l'utiliser pour des débits supérieurs à 32Kb/s. Un rapport signal sur bruit mesurerait minutieusement la qualité d'un tel codeur.

### Codage PCM

La PCM (Pulse Code Modulation) est une technique de codage de forme d'onde qui consiste en une quantification scalaire des échantillons instantané du signal discret à coder, ou chaque échantillon est assimilé à un niveau correspondant, à ce dernier est octroyée une séquence binaire unique qui sera transmise au récepteur.

Dans le domaine du codage de la parole, la PCM utilise une quantification pseudo-logarithmique, qui consiste à attribuer plus de niveaux aux échantillons à petites amplitudes, en d'autre termes, le pas de quantification est plus large pour les grandes amplitudes et plus petit pour les échantillons de petites amplitudes. Ce standard PCM non-uniforme est adopté par la norme G711.

Deux lois de compression logarithmique sont utilisées : la loi A adopté par l'Europe ( $A=87.56$ ) et la loi  $\mu$  utilisée par les USA ( $\mu=255$ ), ces deux méthodes sont très populaires à cause de la simplicité de leurs circuits et leur délai de codage. La loi  $\mu$  est basée sur une fonction logarithmique non linéaire qui est donnée par :

$$f(x) = A \frac{\ln\left(1 + \mu \frac{|x|}{A}\right)}{\ln(1 + \mu)} \operatorname{sgn}(x) \quad (2.19)$$

Ou A représente le maximum du signal d'entrée, x le signal à quantifier ( $x \leq A$ ) et  $\mu$  une constante qui contrôle le degrés de compression [12].

La loi A est basée sur une autre fonction logarithmique non linéaire qui est donnée par [5]:

$$f(x) = \begin{cases} \frac{A_0|x|}{1+\ln A_0} \operatorname{sgn}(x) & , |x| \leq \frac{A}{A_0}, \\ \frac{A(1+\ln(A_0|x|/A))}{1+A_0} & , \frac{A}{A_0} \leq |x| \leq A, \end{cases} \quad (2.20)$$

### Codage différentiel DPCM et ADPCM

Le signal vocal est caractérisé par une forte corrélation entre ses échantillons successifs, donc il serait aisé de faire une prédiction d'un échantillon présent en fonction des échantillons antérieurs, ensuite de soustraire une erreur de prédiction qui serait caractérisé par une faible variance [12].

Le codage différentiel DPCM est basé sur cette approche, en effet, dans un tel codeur, c'est l'erreur de prédiction qui est quantifiée et transmise, cela engendre une réduction du débit par rapport un codage PCM classique.

$$\begin{aligned} e(n) &= s(n) - s(n-1) \\ E(z) &= S(z)[1 - z^{-1}] \end{aligned} \quad (2.21)$$

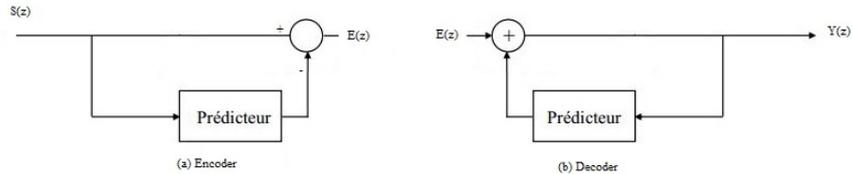


Fig 2.4: Codage PCM

$$E(z) = S(z)(1 - P(z))$$

$$Y(z) = \frac{E(z)}{1 - P(z)} = S(z) \quad (2.22)$$

Le principe du codage différentiel est illustré par la Figure 2.5, l'erreur de prédiction  $e[n]$  est obtenue par la soustraction de la valeur prédite  $s(n)$  depuis la valeur de l'échantillon original  $s(n-1)$ , les indices à la sortie du codeur DPCM représentent le signal codé, ils sont ensuite appliqué au décodeur qui génère l'erreur quantifiée, en combinant celle-ci avec la valeur prédite de l'échantillon, on forme la valeur quantifiée de l'échantillon d'entrée original. le prédicteur utilisera cette dernière pour produire la valeur prédite  $s(n)$  .

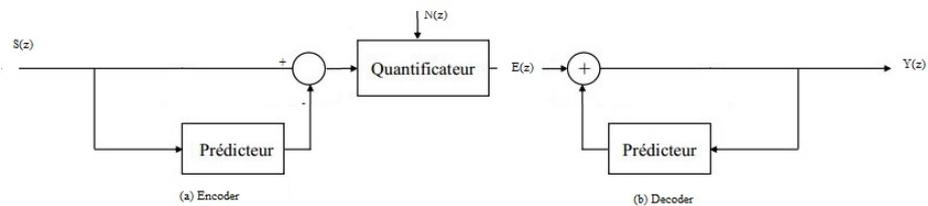


Fig 2.5: Structure première DPCM

D'une façon plus au moins similaire, le décodeur DPCM, à partir des indices reçus, génère l'erreur de prédiction, celle-ci est ajoutée à la valeur prédite de l'échantillon, pour obtenir la valeur quantifiée correspondante à cette dernière. La Figure 2.5 illustre le fonctionnement du décodeur.

Mais le problème qui se pose ici, c'est que l'erreur de quantification à la réception sera intégrée:

$$Y(z) = S(z) + \frac{N_q(z)}{1 - P(z)}$$

de valeur plus grande que le bruit lui-même, donc pour annuler cet effet la:

- La quantification doit être faite à l'intérieur de la boucle.

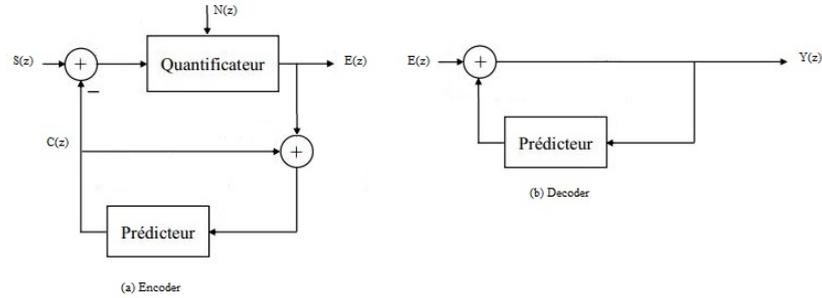


Fig 2.6: DPCM avec Quantificateur à l'intérieur du feedback

$$\begin{aligned}
 E(z) &= S(z) + C(z) + N_q(z) \\
 &= [S(z) + N_q(z)] (1 - P(z)) \\
 C(z) &= \frac{E(z) P(z)}{1 - P(z)} \\
 Y(z) &= S(z) + N_q(z)
 \end{aligned}$$

La non-stationnarité du signal de parole oblige à utiliser un quantificateur scalaire à pas variable pour adapter le codeur aux fluctuations rapides du signal, un codeur PCM adoptant cette technique est appelé codeur ADPCM (Adaptatif PCM), ce dernier fait l'objet de la recommandation G 726 et réduit d'avantage le débit, celui-ci peut atteindre les 16 kb/s.

### 2.2.6 Codage paramétrique

Les codeurs paramétriques modélisent le signal vocal à l'aide d'un modèle paramétrique propre aux signaux de parole. Le codeur extrait les paramètres du modèle et les codes sur un nombre limité de bits. Le décodeur utilise ses paramètres pour alimenter le modèle et synthétiser un signal de parole. Ces codeurs permettent d'obtenir de faibles débits de codage, typiquement inférieur à 4Kb/s, tout en conservant une bonne qualité de parole. Les algorithmes de codage paramétrique sont destinés à des applications de sécurité et ils sont souvent développés et standardisés par des organismes militaires.

Les codeurs de parole utilisant le codage paramétrique diffèrent par le type de modèle utilisé et par la façon de coder les paramètres, le modèle qui est couramment utilisés : le modèle « source-filtre ». Ces codeurs calculent différents types de paramètres, en particulier des paramètres spectraux qui représentent l'enveloppe spectrale du signal.

Les paramètres spectraux sont généralement obtenus par prédiction linéaire (Chapitre1) mais on utilise aussi parfois une transformée de Fourier à court terme [14]. D'autre

paramètre comme le voisement ou le niveau de voisement, la fréquence fondamentale, l'énergie (souvent représentée par un gain), les valeurs de quelques harmoniques sont aussi utilisées. Les codeurs source-filtre utilisent différentes approches pour construire le signal d'excitation. On peut distinguer les codeurs à deux états d'excitation (LPC classique), les codeurs à excitation multi-bande MBE(Multi Band Excited), ainsi de suite [9].

### 2.2.7 Codage hybride

Les codeurs paramétrique décrit ci-dessus permettent une réduction importante du débit, mais au prix d'une qualité moyennement bonne, quand aux codeurs de forme d'onde, il présente une très bonne qualité mais malheureusement pour des débits assez élevés, de l'ordre de 16Kb/s au minimum. Les codeurs hybrides sont une concomitance des deux techniques précédentes, en effet ils codent l'aspect temporel du signal tout en tirant profit d'un modèle de production de la parole et certains aspects de la perception auditive. La plupart des codeurs hybrides sont basés sur l'algorithme CELP (Code Excited Linear Prediction). Cet algorithme et ces dérivés sont à la base de la majorité des standards de téléphonie mobile et les communications vocales sur internet en voix IP. Les codeurs basés sur l'algorithme CELP délivrent un débit moyen allant typiquement de 4 à 16Kb/s en gardant une très bonne qualité du signal.

## 2.3 Mesure de la Distorsion

Pour conclure le chapitre, nous allons énumérer les différents critères couramment utilisés pour juger et classer les méthodes de codage. D'autres critères peuvent être importants, selon le type d'application [15].

### 2.3.1 Débit de Transmission

Le débit de transmission détermine le nombre de bits par seconde alloué au codeur pour la représentation de l'information. L'objectif d'un algorithme de codage est de réduire ce débit en maintenant une bonne qualité du signal. La largeur de la bande disponible dans le système de communication détermine la limite supérieure du débit envisageable du codeur de parole. Lors de la conception d'un système de codage, un choix sera effectué parmi les codeurs à débit de transmission fixe ou variable.

### 2.3.2 Délai de Codage

Le délai de codage est très important dans les transmissions en temps réel. Le délai global est engendré par le temps de traitement du codage et du décodage mais aussi par le délai de transmission qui dépend du canal utilisé et les différents temps d'attente liés à

l'application choisie. De ce temps de restitution du signal dépendra la qualité d'écoute de la communication.

### 2.3.3 Qualité de Parole

Une considération importante dans tout codage de parole est la qualité du signal reconstruit. Les recherches sur les différents types de codage essaient toujours de trouver un bon compromis entre la qualité du signal de parole restitué et le débit de transmission. Pour un débit fixé, le critère de qualité pourra alors être employé pour évaluer un système de codage.

Deux types de mesures, objective et subjective, peuvent permettre l'évaluation de la qualité de parole.

### 2.3.4 Mesures Subjective de La Qualité de Parole

#### Mesures Subjective

La qualité du signal reconstitué peut être déterminée par des tests d'écoute du signal codé et décodé dans des conditions désirées, où des auditeurs jugent subjectivement la qualité globale et l'intelligibilité de la parole. Pour ce type d'étude, un grand nombre de personnes est nécessaire pour effectuer une analyse statistique de leur opinion moyenne (MOS : Mean Opinion Score). Une telle expérimentation est très contraignante à réaliser mais reste incontournable. Un échantillon de résultats de test simple, comme par exemple comparer le signal synthétisé au signal original, peut apporter de premières informations significatives sur les forces et les faiblesses de la procédure de codage.

La perception des caractéristiques de la parole tend à changer considérablement entre les différents auditeurs, mais ces essais restent toutefois utiles pour pointer sur différents aspects déterminés unanimement.

- A : Test MOS (Points Moyens d'Opinion) : un groupe d'auditeurs d'essai évalue la qualité de parole et donne le résultat entre 1 et 5. 1=de mauvaise qualité, 2= faible, 3= qualité juste, 4= de bonne qualité, 5=qualité excellente.

La Table 2.1 donne les résultats de ce test pour les différents types de codeurs de parole.

- B : Test de discrimination : Tester la transparence du quantificateur, demander aux auditeurs de deviner lequel des deux signaux était l'original, et lequel était quantifié.

MOS	Qualité
1	Mauvais
2	Médiocre
3	Passable
4	Bon
5	Excellent

Tableau 2.1: MOS: Points Moyens d'Opinion

- C : Test DRT (Diagnostic Rhyme Test) : Tester l'intelligibilité de la parole codée. Jouer un mot, et donner deux choix aux auditeurs, par exemple sol /seul

Les mesures subjectives ne peuvent pas être calculées automatiquement, donc elles ne peuvent pas être utilisées comme des critères d'optimisation dans un algorithme de codage.

### Mesures Objectives

Les mesures objectives utilisent des fonctions ou des critères mathématiques pour comparer les formes d'onde, les spectres ou les cepstres codées et originales. Certaines mesures donnent des informations utiles selon le type de codage testé. Par exemple, le Rapport Signal sur Bruit (SNR : Signal to Noise Ratio) est représentatif pour les codeurs temporels et certains codeurs hybrides, tels que les codeurs de type CELP, qui incorporent des mécanismes de modélisation de forme d'onde. D'autres évaluent certains éléments des algorithmes de codage, tels que les distorsions cepstrales ou spectrales, employés pour calculer la déformation introduite par la quantification des paramètres LPC. Idéalement, les mesures objectives recourent les résultats obtenus par notre perception subjective de la parole. Toutefois, les tests subjectifs et objectifs peuvent produire des résultats légèrement différents.

### A : Rapport signal à bruit (RSB)

Le rapport signal à bruit représente le rapport de la puissance du signal à la puissance du bruit.

Le RSB s'exprime souvent en décibel suivant la relation

$$RSB = 10 \log_{10} \left( \frac{\sum_n s^2(n)}{\sum_n (s(n) - \hat{s}(n))^2} \right) \quad (2.23)$$

Le RSB n'est pas approprié pour mesurer la distorsion des signaux non-stationnaires. C'est pour cela que l'on a introduit le RSB segmental défini comme la moyenne des RSB calculés sur des trames courtes.

### B : RSB Segmental (RSBS)

C'est une amélioration en ce qui concerne la mesure conventionnelle de RSB, créée pour manipuler la nature dynamique des signaux non stationnaires tels que la parole. La RSBS est définie par :

$$RSB_{seg} = \frac{1}{N} \sum_{m=1}^N SNR_m \quad (2.24)$$

N est le nombre de trames .

### C: Mesure de la distorsion spectrale

Pour déterminer la qualité d'une enveloppe spectrale estimée, la distorsion spectrale (SD) est calculée sur le spectre de puissance dans le plan fréquentiel comme une mesure objective.

$$SD = \sqrt{\frac{1}{N} \sum_{i=0}^{N-1} \left( 10 \log \log_{10} \left( \frac{P_i}{\hat{P}_i} \right) \right)^2} \quad (2.25)$$

Où  $P_i$  et  $\hat{P}_i$  sont les spectres original et estimé respectivement, cette mesure est effectuée chaque trame d'analyse LPC.

## Conclusion

La naissance du traitement du signal numérique a suscité un grand intérêt pour le développement des communications parlées. Dans les années soixante, on codait encore les signaux vocaux avec une simple quantification scalaire puis est venu le codage paramétrique qui sollicitait la conception de modèles caractérisant le système phonatoire humain, ils exploitent sa redondance et utilisent généralement la technique de prédiction linéaire qui s'est avérée imparable pour la compression.

Pour remédier à la détérioration du signal dans le codage paramétrique et éviter les débits élevés du codage de forme d'onde, une technique qui combine en harmonie entre les deux est venue comme issue salvatrice à ce dilemme, c'est la technique du codage hybride qui est basée essentiellement sur le codage CELP qui fera l'objet du prochain chapitre et qui sera détaillé dans le quatrième chapitre.

## Codage CELP

### 3.1 Introduction

L'algorithme du Code-Excited-Linear-Prediction CELP, a été conçu pour développer les codeurs LPC classiques tel que FS1015, afin d'offrir une qualité de parole synthétisée plus naturelle pour des débits moyens, en exploitant la même idée de fonctionnement, mais en remplaçant certains modules par d'autres qui garantissent à la fois un débit et une qualité de parole satisfaisants.

### 3.2 Principe du Codage CELP

La technique CELP constitue la base de beaucoup de codeurs de parole normalisés [7]. Cette technique, permettant des taux de compression significatifs, cherche à exploiter la redondance du signal de parole. Il suffira alors de ne transmettre au décodeur que l'information non prédictible. Le codage CELP est basé sur le codage prédictif et l'analyse par synthèse (LPAS : Linear Prediction Analysis by Synthesis) [7, 16, 17, 18]. Dans le codage LPAS (Figure 3.1), le signal d'entrée est analysé et un signal d'excitation est déterminé. La fonction du codage prédictif consiste à définir les coefficients du filtre de prédiction tandis que le signal d'excitation est modélisé par l'analyse par synthèse.

L'erreur entre le signal d'entrée et celui mis en forme par le filtre de synthèse, reproduisant les résonances (formants) du conduit vocal, est alors minimisée par le critère des moindres carrés MMSE (Minimum Mean Square Error) pour choisir le meilleur signal d'excitation. Les données à transmettre ne sont plus des valeurs du signal échantillonné mais des paramètres résultant de la prédiction linéaire et du codage du signal d'excitation. Le calcul des coefficients de prédiction est effectué par trames de parole en boucle ouverte alors

que la recherche de l'excitation optimale est généralement effectuée par sous-trame (la trame étant subdivisée en deux à quatre sous-frames) en boucle fermée. Les paramètres correspondants sont alors transmis au décodeur qui reconstruira le signal de parole en utilisant la même structure de synthèse que celle utilisée en phase d'analyse.

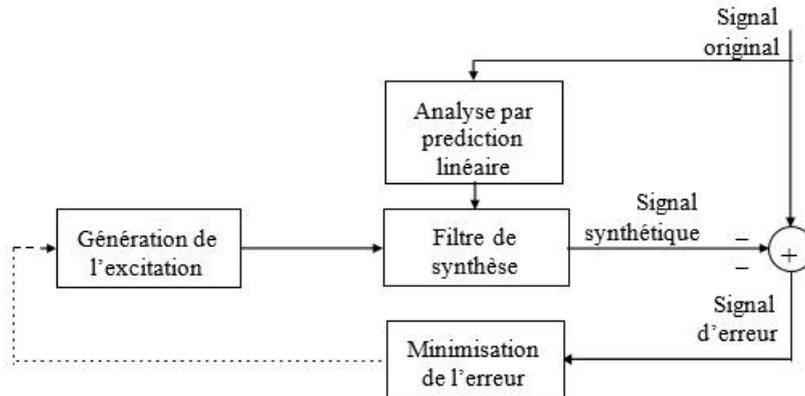


Fig 3.1: Principe du Codeur LPAS

Une des raisons du succès du codage LPAS est la possibilité d'incorporer dans sa structure une fonction qui prend en compte la perception de l'appareil auditif humain. Ce principe a pour but de minimiser un critère d'erreur plus subjectif entre le signal de parole original et le signal de parole synthétique. Ceci est réalisé par pondération des fréquences du signal d'erreur pendant la sélection du signal d'excitation optimal.

#### 3.2.1 L'encodeur

Le schéma de principe du l'encodeur et du décodeur CELP est donné à la Figure 3.2. Une analyse par prédiction linéaire est effectuée pour calculer les coefficients de prédiction du filtre de synthèse  $1/A(z)$ . Le filtre de synthèse de pitch  $1/P(z)$  est remplacé par une recherche dans un dictionnaire adaptatif, dont le contenu est mis à jour à l'aide de l'excitation codée passée. Le signal d'excitation est extrait d'un dictionnaire fixe composé de formes d'ondes prédéterminées. La synthèse du pitch revient à ajouter, à l'excitation extraite du dictionnaire fixe, une excitation extraite du dictionnaire adaptatif composé des séquences d'excitations codées passées. Cette technique de codage sélectionne, en parcourant les dictionnaires adaptatif et fixe, les formes d'ondes pour former le signal d'excitation du filtre de synthèse  $1/A(z)$  qui minimise (en tenant compte d'une pondération perceptive) l'erreur quadratique moyenne entre le signal de parole originale et le signal de parole synthétique [7, 16, 17, 18].

### 3.2. Principe du Codage CELP

Notons que la sélection des formes d'ondes issues des dictionnaires adaptatif et fixe s'effectue de façon séquentielle : la forme d'onde issue du dictionnaire adaptatif est d'abord sélectionnée puis celle issue du dictionnaire fixe. Ce processus, qui consiste à choisir les paramètres d'un dictionnaire (l'indice de la forme d'onde et le gain associé) de manière à optimiser la qualité du signal synthétique, est désigné sous le nom de recherche en boucle fermée et fait suite à l'analyse par prédiction linéaire, dite en boucle ouverte, qui détermine les coefficients de prédiction en analysant uniquement le signal d'entrée.

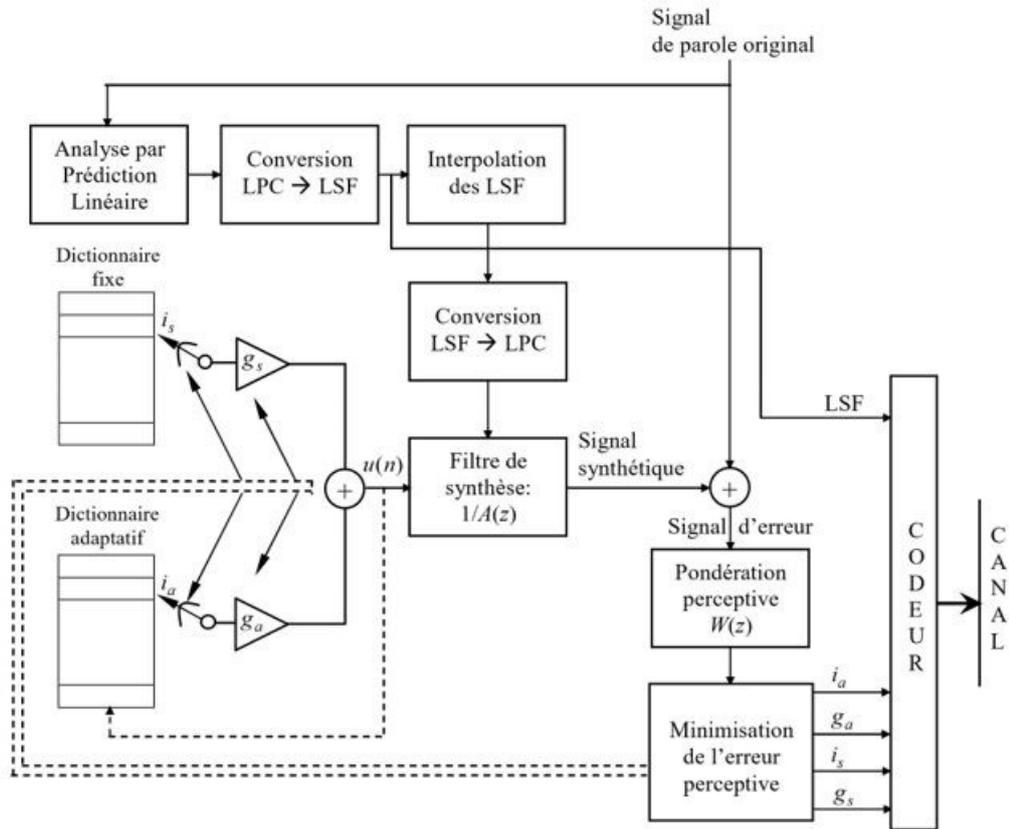


Fig 3.2: (a) Encodeur

#### 3.2.2 Analyse par prédiction linéaire

Une analyse par prédiction linéaire est effectuée sur des trames de 10 à 30 ms pour le calcul des coefficients de prédiction [19]. Les coefficients de prédictions  $\{a_k\}_{k=1,\dots,p}$  sont convertis aux coefficients fréquences de raies spectrales LSF :  $\{\omega_k\}_{k=1,\dots,p}$  [21, 22]. Les coefficients LSF sont donc calculés à chaque trame d'analyse de 10 ms à 30 ms. La trame à coder est généralement divisée en 2 à 4 sous-trames. En effet, la modélisation de l'excitation  $u(n)$  du filtre de synthèse est effectuée pour chaque sous-trame. Dans le traitement par sous-trame, une interpolation linéaire est généralement réalisée entre deux ensembles de p

coefficients LSF chacun (correspondant à deux trames consécutifs) pour former un ensemble intermédiaire de  $p$  coefficients LSF pour chaque sous-trame. L'interpolation linéaire est réalisée pour lisser l'évolution des paramètres LP et réduire ainsi la présence de transitions brutales dues aux changements rapides des paramètres LP entre les trames.

### 3.2.3 Analyse par synthèse

Dans les codeurs LPC tel que le FS1015, la sélection des paramètres représentant le signal de parole se fait par une boucle ouverte, avec cette démarche, on choisit ces paramètres sans avoir à vérifier si ils permettent de reconstituer un signal synthétisé semblable ou pas au signal de parole original.

Le principe de l'analyse par synthèse, nous permet de choisir certains paramètres caractérisant le signal original tout en testant si ils nous fournissent un signal synthétisé analogue à l'original[23], cela est réalisé à l'aide d'un système en boucle fermée (Figure 3.1).

D'après la Figure 3.1, l'analyse par synthèse, nous oblige à synthétiser le signal à l'intérieur du codeur, puis de le comparer avec le signal original dans le but de sélectionner les meilleurs paramètres, ce qui reconstitue une parole proche de l'originale, c'est la raison pour laquelle cette méthode est appelée ainsi.

Théoriquement, les paramètres sont choisis pour présenter le meilleur résultat possible, en pratique, seulement quelques paramètres sont sélectionnés par l'analyse par synthèse. Le codeur CELP est bâti essentiellement sur cette approche, en effet la reconstitution du signal de parole à l'intérieur du codeur nécessite le choix d'une excitation contenu dans un dictionnaire (codebook) (Figure 3.2), cette excitation est à l'entrée du filtre de synthèse du pitch, la sortie de celui-ci vient exciter le filtre de synthèse [23, 24], ce dernier produit le signal de parole synthétisé, qui est comparé au signal original, une erreur alors est induite, elle est minimisée puis exploitée pour sélectionner l'excitation dans le codebook, cela engendre une démarche itérative qui tend à produire une erreur minimale et bien évidemment le signal synthétisé le plus exact, cette itération constitue la boucle fermée de la technique de l'analyse par synthèse.

Comme nous l'avons mentionné précédemment, sauf certain paramètres tel que le gain et l'excitation sont déterminés par l'analyse par synthèse, les coefficients du filtre de synthèse sont déterminés par une boucle ouverte en utilisant la méthode de l'autocorrélation [7][24].

### 3.2.4 Modélisation du signal d'excitation

Le dictionnaire fixe est caractérisé par sa structure. Schroeder et Atal [16] ont proposé un dictionnaire de vecteurs constitués d'échantillons aléatoires gaussiens. Le dictionnaire stochastique obtenu par apprentissage a été proposé en [22].

D'autres types de structures ont été envisagés dans le but de réduire la complexité des algorithmes de sélection de la forme d'onde optimale ou le nombre de bits nécessaires pour son codage. On parle alors de dictionnaires algébriques [25, 26, 27, 28].

Le filtre de synthèse de pitch est généralement considéré comme une recherche dans un dictionnaire adaptatif dont le contenu est mis à jour à l'aide de l'excitation passée. La périodicité à long terme, présente particulièrement pour les sons voisés, est ainsi modélisée en représentant l'excitation courante par une excitation passée pondérée par un gain. Le dictionnaire fixe aura alors pour objectif de modéliser les composants aléatoires restants et les segments non-voisés du signal d'excitation. Pour déterminer les paramètres d'excitation, une recherche exhaustive au sein des dictionnaires adaptatif et fixe est effectuée pour déterminer l'excitation  $u(n)$ , qui est une combinaison linéaire de la contribution des deux dictionnaires (Figure 3.2) :

$$u(n) = g_s c_{i_s}(n) + g_a v_{i_a}(n), \quad 0 \leq n \leq L, \quad (3.1)$$

où les formes d'onde  $\{v_{i_a}(n)\}_{n=1,\dots,L-1}$  et  $\{c_{i_s}(n)\}_{n=0,\dots,L-1}$ , pondérées par les gains  $g_a$  et  $g_s$  respectivement, sont respectivement extraites du dictionnaire adaptatif (à l'indice  $i_a$ ) et du dictionnaire fixe (à l'indice  $i_s$ ). On cherche alors les paramètres  $\{i_a, g_a\}$  et  $\{i_s, g_s\}$  de l'excitation  $u(n)$  qui minimisent l'erreur quadratique moyenne (en tenant compte d'une pondération perceptive) entre les sous-trames de parole originales et celles de parole synthétiques reconstruites par le passage de l'excitation  $u(n)$  à travers le filtre de synthèse  $1/A(z)$ . Pour chaque trame de  $N$  échantillons, l'encodeur transmet  $p$  coefficients LSF et  $N/L$  fois les paramètres du signal d'excitation :  $\{i_a, g_a\}$  et  $\{i_s, g_s\}$ . Le décodeur peut reconstituer le même signal d'excitation qu'à l'encodeur puisque il dispose des mêmes dictionnaires fixe et adaptatif.

### 3.2.5 Pondération perceptive

Le principe de l'analyse par synthèse, utilisée pour déterminer le signal d'excitation, consiste à minimiser un critère d'erreur entre une séquence du signal de parole original  $s(n)$  et le signal de parole synthétisé  $\hat{s}(n)$ . L'excitation optimale pourrait être obtenue par minimisation de l'erreur quadratique moyenne entre le signal original et le signal synthétique:  $\sum_n (s(n) - \hat{s}(n))^2$ . Cependant, ce critère n'étant pas bien adapté à notre système auditif,

une correction est intégrée pour pallier à cet inconvénient. Le critère d'erreur est modifié par une pondération perceptive dont le but est de contrôler la répartition fréquentielle de l'erreur en se basant sur le phénomène de masquage spectral du bruit [29]. Le signal d'erreur est moins perceptible dans les régions où le signal a une grande énergie (zones formantiques). On dit que le bruit est masqué par le signal de parole. Ceci permet donc, de tolérer une erreur plus grande dans les zones formantiques que dans celles antifonnantique du signal.

C'est pourquoi Atal a proposé d'insérer un filtre de pondération de fonction de transfert  $W(z)$  avant le critère à minimisation. Ce filtre est donné par [17][18]:

$$W(z) = \frac{A(z)}{A(a/\gamma)} = \frac{1 + \sum_{k=1}^p a_k z^{-k}}{1 + \sum_{k=1}^p a_k a_k \gamma^k z^{-k}}, \quad 0 < \gamma < 1. \quad (3.2)$$

En général, la valeur de  $\gamma$  est au voisinage de 0.8. La Figure 3.3 donne l'exemple d'un spectre de signal de parole et de la réponse en fréquence du filtre de pondération perceptive associée.

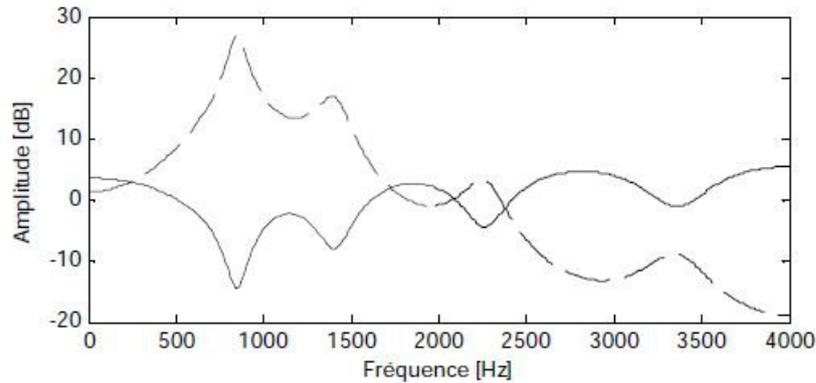


Fig 3.3: Spectre LPC du signal de la parole (en pointillé) et réponse fréquentielle du filtre de pondération perceptive  $A(z)/A(z/\gamma)$  (trait plein) avec  $\gamma=0.8$ .

#### 3.2.6 Le décodeur

Le décodeur dispose des mêmes dictionnaires fixe et adaptatif que l'encodeur. Le dictionnaire adaptatif est actualisé de la même manière que l'encodeur. Le décodeur (Figure 3.4) est relativement plus simple à mettre en œuvre:

- les fréquences de raies spectrales (LSF) sont interpolées et reconverties en coefficients de filtre de Prédiction Linéaire pour chaque sous-trame de  $L$  échantillons.
- l'excitation est construite par combinaison des contributions adaptatif et fixe.

### 3.2. Principe du Codage CELP

- le signal de parole est reconstitué par filtrage de l'excitation à travers le filtre de synthèse  $1/A(z)$ .

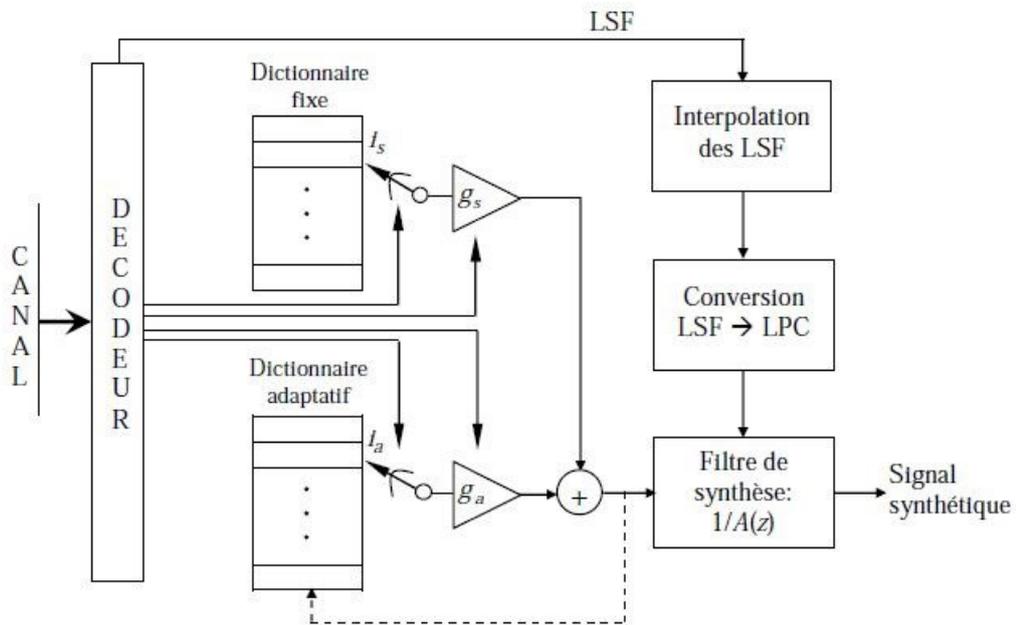


Fig 3.4: (b) Décodeur

## Conclusion

Ce chapitre nous a permis de comprendre le fonctionnement d'un algorithme CELP qui est distingué entre autre par sa boucle d'analyse par synthèse et son dictionnaire d'excitation, ainsi que la méthode de recherche dans ce dernier. Nous nous sommes aussi orientés vers une étude brève du standard FS1016 qui est de loin l'un des standard CELP les plus populaires, ce qui nous a permis d'avoir un aperçu sur son fonctionnement qui est basé sur l'algorithme CELP mais avec le remplacement du filtre de synthèse du pitch par un dictionnaire adaptatif.

# Implémentation sur MATLAB

## 4.1 introduction

Le chapitre précédent nous à permis de comprendre le fonctionnement d'un codeur basé sur l'algorithme CELP, ce mémoire jusque-là consistait en une étude théorique qui transiter au fur et à mesure entre traitement du signal et modélisation mathématique, tout en gardant un coté plus au moins abstrait.

Dans ce quatrième et dernier chapitre, nous allons concrétiser le travail réalisé jusqu'ici par une implémentation de notre algorithme sur MATLAB , nous commencerons par une segmentation d'un signal de parole puis le fenêtrage de celui-ci et voir l'effet de ces deux opérations sur le signal original, puis en tirer les paramètres de prédiction qui seront transformés en LSF et transmis, ensuite on déterminera le pitch par boucle ouverte, et enfin le codebook stochastique.

## 4.2 Calcul des coefficient LPC

L'analyse de Prédiction linéaire (LP) est effectuée sur "chaque trame du speech" [le signal d'entrée cadré et fenêtré] pour calculer les coefficients "du filtre LP" de la prédiction linéaire. Une analyse d'autocorrélation de la parole est utilisée, l'algorithme de wiegner-Levinson-Durbin prend les autocorrélation comme entrée pour calculer les coefficients de prédiction linéaire.

Les coefficients d'autocorrélation sont calculés et la correction de haute fréquence est faite , Les auto-corrélations sont converties en coefficients LP en utilisant la récursion de Levinson-Durbin .après, les coefficients LP sont soumis à une expansion de la bande

## 4.2. Calcul des coefficient LPC

passante pour obtenir des coefficients a bande passante élargie qui seront reconvertis en coefficients de réflexion (CR) en utilisant l'inverse de Levinson-Durbin.

Le schéma de la Figure 4.1 illustre l'analyse d'auto-corrélation linéaire et le module de prédiction . Chaque section à partir de celui-ci, aura un schéma de principe similaire qui illustrera la fonction globale du module décrit dans ce chapitre. Chaque bloc dans le diagramme correspond à un script particulier, avec des programme complémentaire qu'on à écrit et testé dans ce chapitre.

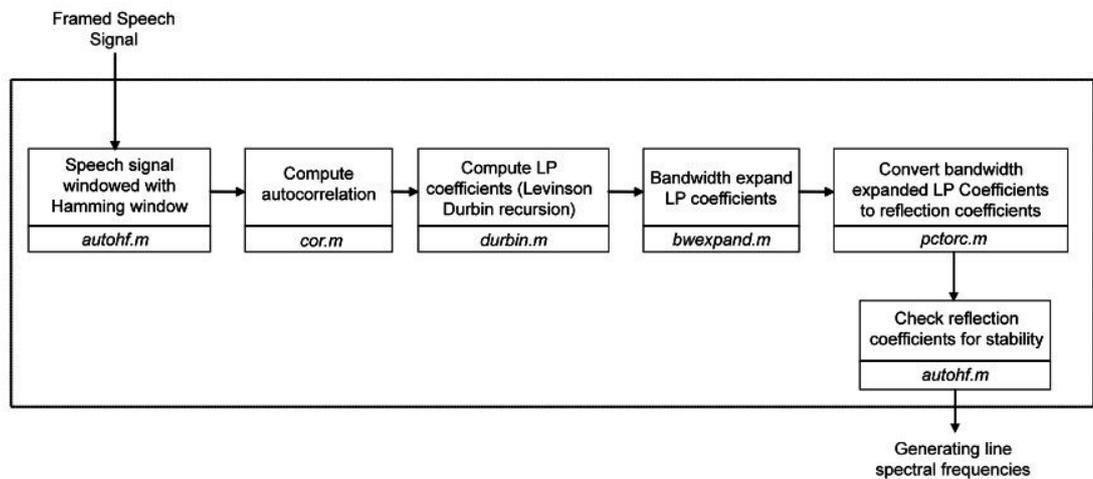


Fig 4.1: Prédiction linéaire et auto-corrélation

### 4.2.1 Fenêtrage et segmentation

La première étape avant de tout commencé, s'agit d'enregistrer le signal à testé et puis le chargé sur MATLAB en utilisant la fonction "wavread" donné par la Figure 4.2, celle ci nous a permet d'obtenir les données du signal chargé qui est en format "wav". Pour adopté l'extension du signal chargé on, a convertit le signal "m4a" vers "wav". L'opération de conversion du son nous à donné la main à précisé la fréquence d'échantillonnage (8 kHz) aussi bien que le nombre de bits associer (16 bits).

## 4.2. Calcul des coefficient LPC

---

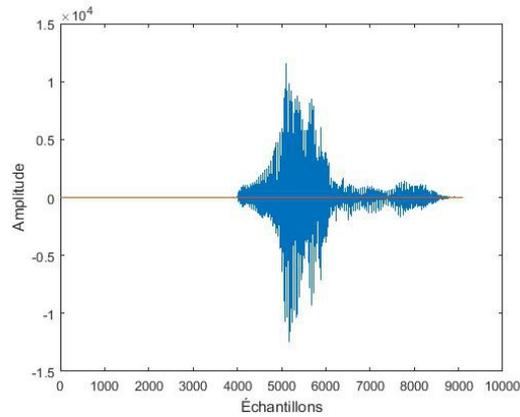


Fig 4.2: Exemple d'un signal engesitré avec le mot "ONE"

L'outil "wavread" nous représente le signal "one.wav" sous forme de vecteur de dimension  $N$ , dont les valeurs sont normalisé par rapport  $Q(x)_{max} = 2^{15} - 1$ .  
N:nombre total d'échantillons.

Le signal de parole est divisé en trames chaque trame contient 240 échantillons qui correspond a 30 ms à une fréquence d'échantillonnage de 8 kHz. pour cela on a utilisé la commande (vec2mat) pour crée une matrice tel que chaque ligne représente une trame, cela pour simplifier les calculs.

chaque trame est filtré par un filtre passe-haut (pour éliminé le bruit basse fréquence) est fenêtré par une fenêtre de Hamming représenté par la Figure 4.3 pour une correction des haute fréquences.

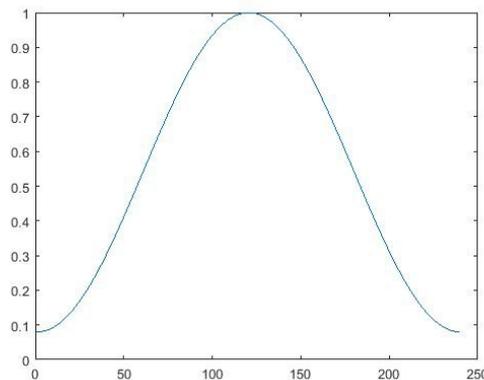
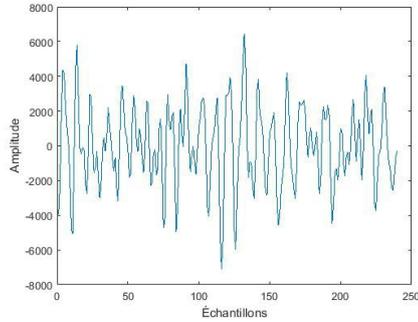


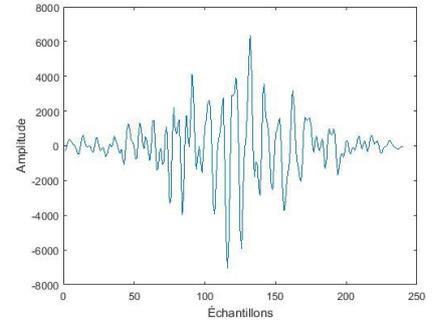
Fig 4.3: Fenêtre de Hamming à 240 points

## 4.2. Calcul des coefficient LPC

Un exemple d'une trame fenêtré avec la fenêtre de Hamming, et non fenêtré est représenté par la Figure 4.4:



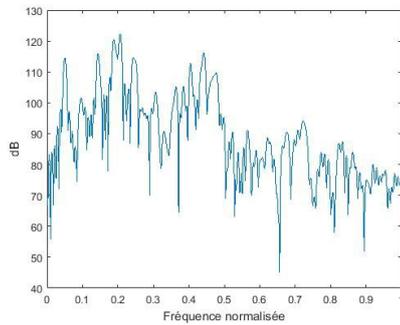
(a) La trame  $s(25,:)$  avant fenêtrage



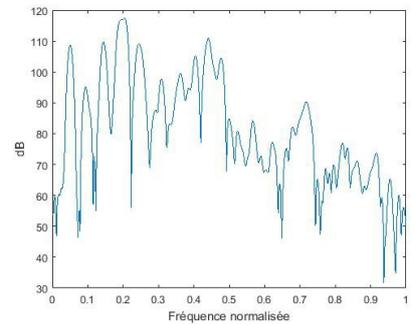
(b) La trame  $s_{new}(25,:)$  après fenêtrage

Fig 4.4: L'effet de la fenêtre de Hamming

La transformé de fourier de la trame fenêtré est plus lisse a cause de la convolution de la réponse fréquentielle de la fenêtre de Hamming 4.5 dans le domaine des fréquences.



(a) Spectre La trame  $s(25,:)$  avant fenêtrage



(b) Spectre La trame  $s_{new}(25,:)$  après fenêtrage

Fig 4.5: L'effet de la fenêtre de Hamming dans le domaine spectrale

### 4.2.2 Calcul des Coefficients d'Auto-corrélation

Les auto-corrélations sont calculées à partir de la formule suivante :

$$r(i) = \sum_{n=i+1}^N u(n)u(n-i), \quad i = 1, \dots, M$$

Où  $r(i)$  sont les entrées de l'algorithme L-D qu'on va voir dans l'étape suivante, voici les auto-corrélations calculées depuis la trames 25 et 30, représenté dans la Table 4.1:

## 4.2. Calcul des coefficient LPC

r(i)	r(1)	r(2)	r(3)	r(4)	r(5)	r(6)	r(7)	r(8)	r(9)	r(10)
( $\times 10^7$ )	47.82	9.046	-17.332	-28.085	-35.417	-39.245	-24.225	10.362	38.871	41.280
	2.1339	1.7523	1.2819	0.8368	0.4686	0.1746	-0.0858	-0.3702	-0.7106	-1.0651

Tableau 4.1: Valeurs d'auto-corrélation  $r(25,:)$  et  $r(30,:)$

### 4.2.3 LEVINSON-DURBIN - LPC

Les coefficients de prédiction linéaire sont les coefficients du polynôme LP  $A(z)$ . Rappelons que l'analyse LP est effectuée en utilisant le filtre tout-zéro  $A(z)$  dont l'entrée est la trame de parole pré-traité et la sortie est le résiduel. Mais, pendant la synthèse, le filtre utilisé a la fonction de transfert  $1 / A(z)$ , qui modélise le conduit vocal et reproduit le signal vocal lorsqu'il est excité par le système LP résiduel. La récursion de Levinson-Durbin est utilisée pour calculer les coefficients du filtre d'analyse LP à partir de l'estimation des autocorrélations. Il exploite la structure de Toeplitz de la matrice d'auto-corrélation pour calculer de manière récursive les coefficients d'un filtre dixième ordre.

Après le calcul des auto-corrélations, les  $r(i)$  entrée sont utilisé pour calculé les coefficients de prédiction linéaire de façon récursive.

Voici les coefficients du prédiction linéaire obtenues par L-D, présenté dans la Table 4.2:

a(i)	a(1)	a(2)	a(3)	a(4)	a(5)	a(6)	a(7)	a(8)	a(9)	a(10)
$data_a(25, :)$	-1.8610	2.1235	-1.7091	1.3676	-1.1694	1.1641	-0.6303	-0.1336	0.3189	-0.1825
$data_a(30, :)$	-2.7045	3.6065	-3.4859	2.9345	-2.2575	1.5995	-0.9372	0.1258	0.4187	-0.2303

Tableau 4.2: Analyse LP des deux trame 25 et 30

la réponse fréquentielle du filtre de synthèse LP ( $1/A(z)$ ), où les coefficients de  $A(z)$  sont obtenues par L-D sont représenté par la Figure 4.6:

## 4.2. Calcul des coefficient LPC

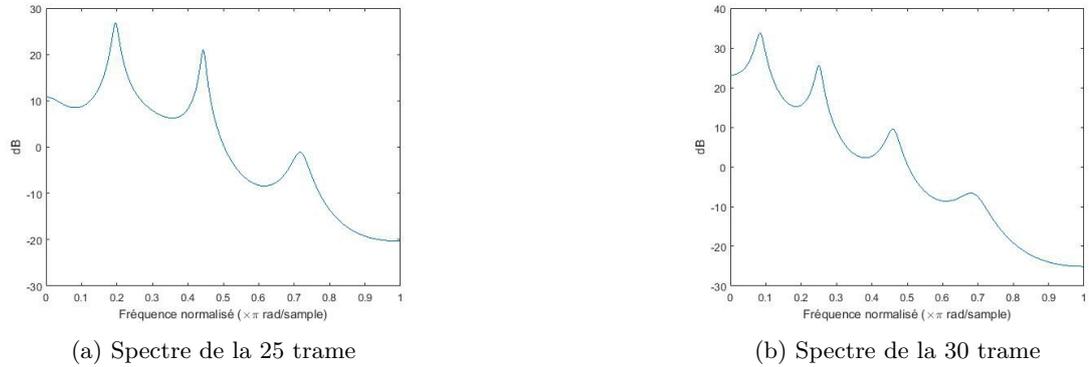


Fig 4.6: Réponse fréquentielle du filtre de synthèse LP

### 4.2.4 Filtre perceptuel

L'élargissement de la bande passante par un facteur  $\gamma$  modifie les coefficients de prédiction linéaire de  $a(i)$  à  $\gamma^i a(i)$ . et déplace les pôles du filtre de synthèse LP vers l'intérieur. L'expansion de la bande passante peut être calculée par:

$$BP = 2\cos^{-1} \left[ 1 - \frac{(1 - \gamma)^2}{2\gamma} \right]$$

A cette fréquence d'échantillonnage (8 kHz), et avec un facteur  $\gamma = 0.994127$ , on aura une expansion de 15 Hz, l'expansion de la bande passante élargie la bande passante des formants et rend la fonction du transfert moins sensible .

Le plan en z des coefficients du filtre LP originales et pondérés sont montré dans la Figure 4.7:

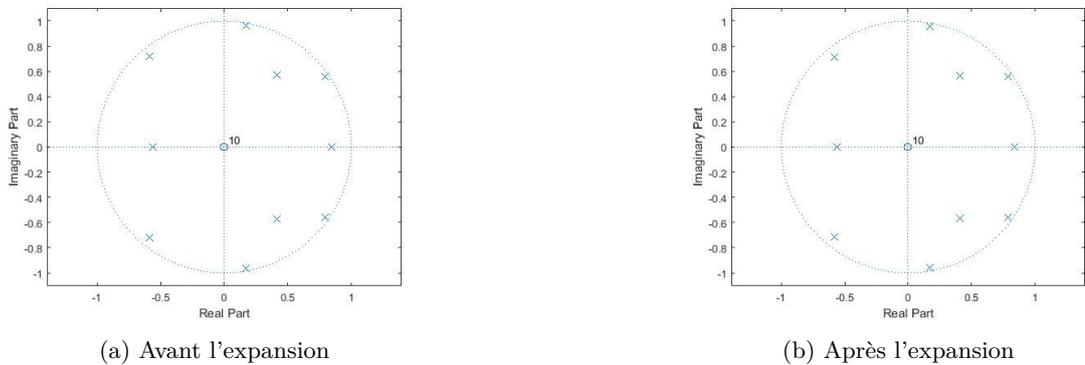


Fig 4.7: Plan en z des coefficients LP de la 25<sup>ème</sup> trame

## 4.2. Calcul des coefficient LPC

---

Le tableau ci contre montre les coefficients avant et après l'expansion montré dans le plan ci-dessus:

Coefficients	Avant l'expansion	Après l'expansion
$a_0$	-1.8610	-1.8501
$a_1$	2.1235	-1.7091
$a_2$	-1.7091	-1.6791
$a_3$	1.3676	1.3357
$a_4$	-1.1694	-1.1355
$a_5$	1.1641	1.1236
$a_6$	-0.6303	-0.6048
$a_7$	-0.1336	-0.1274
$a_8$	0.3189	0.3024
$a_9$	-0.1825	-0.1720

Tableau 4.3: Coefficients du filtre LP avant-après expansion

Pour mieux voir la différence encore voici un graphe qui résume ce qui est dans la Table 4.3:

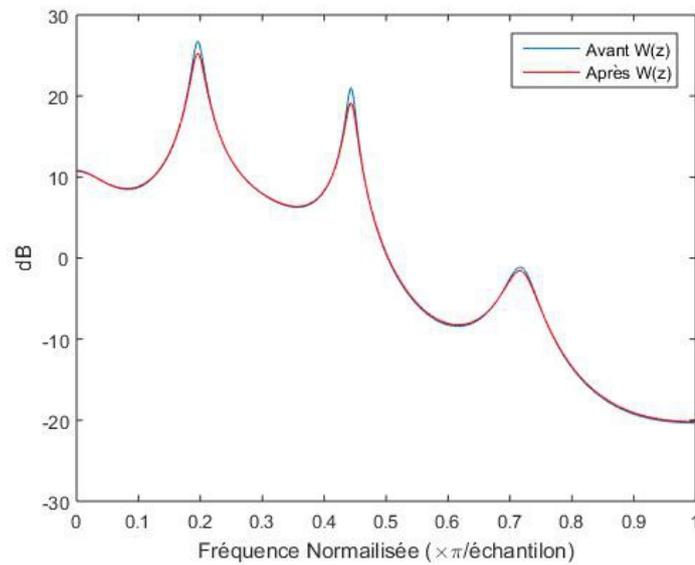


Fig 4.8: Enveloppe Spectrale Avant et Après  $W(z)$

### 4.2.5 L'inverse de Levinson-Durbin

L'inverse de Levinson-Durbin est utilisé pour calculer les RCs à partir des coefficients LP pondérées [30], l'ordre de mise à jour des équations pour le filtre de prédiction avant est arrière et donné sous forme matricielle:

$$\begin{bmatrix} a_i(j) \\ a_i(i-j) \end{bmatrix} = \begin{bmatrix} 1 & k_i \\ k_i & 1 \end{bmatrix} \begin{bmatrix} a_{i-1}(j) \\ a_{i-1}(i-j) \end{bmatrix} \quad j = 0, 1, \dots, i. \quad (4.1)$$

Où  $i$  varie de 1 jusqu'à  $m$ .

Résoudre  $a_{i-1}(j)$  de l'équation ci-dessus, revient à résoudre:

$$a_{i-1}(j) = \frac{a_i(j) - k_i a_i(i-j)}{1 - |k_i|^2}, \quad j = 0, 1, \dots, i. \quad (4.2)$$

Où,  $k_i$  est le coefficient de réflexion d'ordre  $i$ , on commençons par l'ordre le plus grand  $m$ , on calcule les coefficients LP pour un ordre descendant de  $m-1$  jusqu'à 1, on utilisant la formule 4.2, et puis sachant que  $k_i = a_i(i)$ , on détermine les RCs pour l'ordre 1 jusqu'au  $m-1$ .

L'application de la formule 4.2 sur la trame 25 nous à donné les résultats qui fournit dans tableau suivant:

$k_i$	$k_1$	$k_2$	$k_3$	$k_4$	$k_5$	$k_5$	$k_6$	$k_7$	$k_8$	$k_9$
	0.7092	-0.7360	0.3970	-0.7249	-0.1390	-0.0727	0.5409	-0.2229	0.0214	0.1825

Tableau 4.4: RCs de la trame 25

L'analyse LP est appliqué sur chaque trame et les coefficients LP sont obtenus en utilisant l'analyse d'auto-corrélation et l'algorithme de Levinson-Durbin. Ces coefficients sont calculés par une boucle ouverte, qui seront après pondérés par le filtre perceptuel et convertis en RCs en utilisant l'inverse de Levinson-Durbin. Les RCs avec un moins ( $-k_i$ ) sont connues sous le nom des "coefficients PARCOR".

Les RCs on été utilisés dans de nombreux codeurs de première génération, mais la deuxième et la troisième génération utilisent les LSPs, qu'on va utiliser dans la prochaine section.

## 4.3 Calcul et Quantification des LSFs

Les paires de raies spectrales sont une représentation alternative pour les coefficients LP qui sont utilisés dans les systèmes de codage. Cette section va contenir la construction des

### 4.3. Calcul et Quantification des LSFs

deux polynômes LSP, et le calcul de leur zéro en appliquant la règle de Descartes, et puis les quantifier en utilisant un quantificateur non-uniforme, scalaire et indépendant "par split". la quantification scalaire peut engendrer la non-monotonie des LSFs et pour cela une opération d'ajustement de ces derniers est nécessaire, mais nous avons opté à utiliser une autre méthode, qui est la quantification vectorielle qui va être détaillée dans cette section.

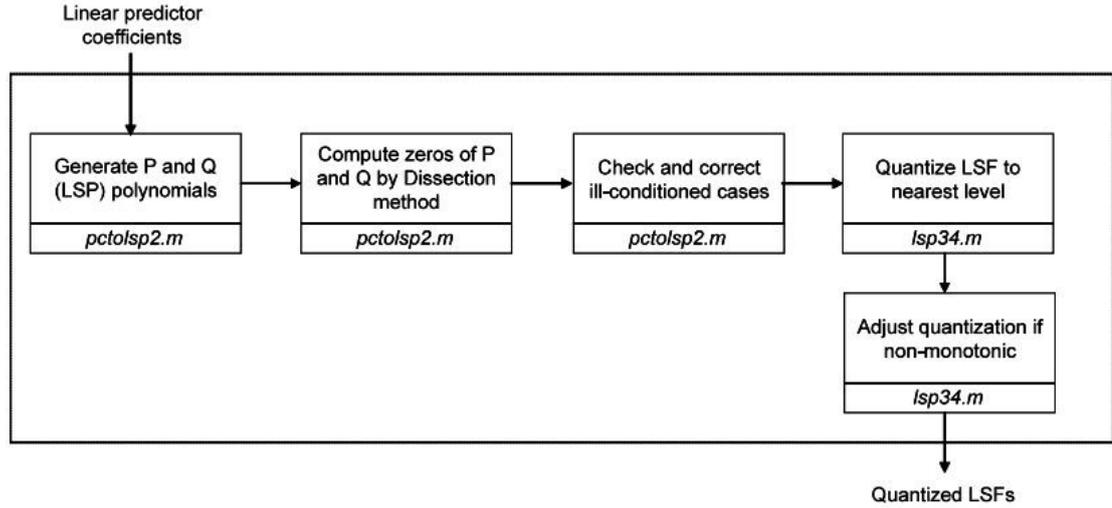


Fig 4.9: Calcul Polynomial et Quantification des LSFs

#### 4.3.1 Construction des Polynômes LSFs

Les Deux polynômes  $P(z)$  et  $Q(z)$  sont construits par le calcul de leur coefficients, utilisant la formule 1.11 et 1.15 les polynômes  $P(z)$  et  $Q(z)$  de la 25<sup>ème</sup> trame sont donnés dans le tableau 4.5:

Coef	$P(z)$
$p_1$	-2.0435
$p_2$	2.424
$p_3$	-1.8426
$p_4$	0.7373
$p_5$	-0.0054

(a) Coef du Polynôme  $P(z)$

Coef	$Q(z)$
$q_1$	-.6786
$q_2$	1.8046
$q_3$	-1.5755
$q_4$	1.9979
$q_5$	-2.3335

(b) Coef du Polynôme  $Q(z)$

Tableau 4.5: Construction des Polynômes  $P(z)$  et  $Q(z)$  pour la trame 25

#### 4.3.2 Zéros du Polynôme Symétrique

Il est impossible de trouver les racines des Polynômes  $P(z)$  et  $Q(z)$  de façon générale. mais en exploitant les propriétés de ces derniers, on peut trouver les racines numériquement de manière traitable.

Étant donné que les zéros se trouvent dans des paires conjuguées complexes, nous calculons uniquement les zéros dans la moitié supérieure du cercle unitaire. En outre, le zéro au point  $(-1, 0)$  dans le cercle unité ne soit pas calculé.

Afin de calculer les racines du polynôme numériquement symétrique en utilisant la règle de Descartes, la procédure suivante est adoptée. La moitié supérieure du cercle d'unité est divisée en 128 intervalles et chaque intervalle est examiné pour des zéros du polynôme. L'hypothèse fondamentale est que l'intervalle est si faible qu'elle contient au plus un seul zéro. Le polynôme  $P'(\omega)$  est évaluée aux fréquences  $f_l$  et  $f_h$ , qui sont les fréquences extrêmes de l'intervalle considéré. Les valeurs du polynôme  $P'(\omega)$  dans les deux fréquences sont enregistrées dans les variables du programme "zero\_sym.mat" (pxr et PXL respectivement).

Initialement, la valeur de  $P'(\omega)$  à la fréquence 0 est la somme de ses coefficients. puis  $P'(\omega)$  est évaluée à la fréquence radian  $\omega$ , où  $\omega$  est représenté comme  $\pi/128$ . D'une manière générale.

Après calcul de pxr et pxl, leurs signes sont comparés. S'il y a un changement de signe, puis par la règle de Descartes, il y a un nombre impair de zéros. Cependant, comme notre intervalle est assez petit, il peut être pas plus d'un seul zéro. S'il n'y a pas de changement de signe, il n'y a pas de zéros et donc le programme passe à l'intervalle suivant. S'il y a un zéro, l'étape suivante est le calcul de la fréquence  $f_m$  milieu en utilisant la méthode de dichotomie. La valeur de pxm est ensuite calculée en utilisant  $f_m$ .

Là encore, les signes de pxm et pxl sont comparés. S'il y a un changement de signe, alors il y a un zéro entre  $f_m$  et  $f_l$ . Par conséquent, l'intervalle est modifié en conséquence. Cette opération est répétée jusqu'à ce que la valeur de pxm se rapproche de zéro, encore l'intervalle de fréquence est traversée pour trouver le zéro. Après avoir trouvé un zéro, nous passons à l'intervalle suivant pour trouver le zéro suivant. Ce processus est répété jusqu'à ce que toute des zéros dans la moitié supérieure du cercle unitaire sont déterminés. Chaque fois qu'un zéro est calculé, le nombre des fréquences calculées (nm) est incrémentée de deux parce que chacun zéro sur la moitié supérieure du cercle unité est un conjugué complexe de l'autre dans la moitié inférieure du cercle unitaire. Par conséquent, trouver un zéro est en fait équivalent à trouver deux LSF.

### 4.3. Calcul et Quantification des LSFs

Les racines du polynôme symétrique pour la trame 25 sont données dans la Table 4.6:

$z_{s_i}$	$z_{s_1}$	$z_{s_2}$	$z_{s_3}$	$z_{s_4}$	$z_{s_5}$
	$0.9158+0.4015i$	$0.6785+0.7346i$	$0.135+0.9908i$	$-0.4063+0.9137i$	$-0.8469+0.5317i$

Tableau 4.6: Zéros de du polynôme P(z) correspond à la trame 25

la représentation de ces racines dans le plan z est montrée dans la Figure 4.10 ,comme il apparait les racines sont sur le cercle unitaire.

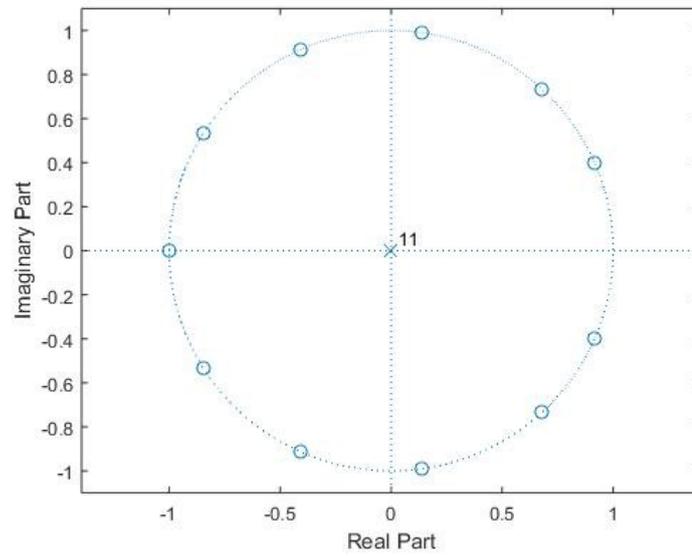


Fig 4.10: Racines du Polynôme LSP Symétrique

#### 4.3.3 Zéros du Polynôme Antisymétrique

La recherche des racines du polynôme antisymétrique est similaire a celle du polynôme symétrique. mais sa seras plus simple on utilisant la propriété qui dit que les racine du P(z) et Q(z) sont entrelacées. et on utilisant l'équation (1.12) et (1.16).

Premièrement, le début et la fin des intervalles sont les zéros successives du polynôme P(z), et puis la procédure précédente et appliquée.

Les racines du polynôme antisymétrique pour la trame 25 sont données dans la Table 4.7:

### 4.3. Calcul et Quantification des LSFs

$z_{as_i}$	$z_{as_1}$	$z_{as_2}$	$z_{as_3}$	$z_{as_4}$	$z_{as_5}$
	0.753+0.658i	0.496+0.8683i	0.2036+0.9791i	-0.6222+0.7829i	-0.9702+0.245i

Tableau 4.7: Zéros du polynôme  $Q(z)$  correspond à la trame 25

la représentation de ces racines dans le plan  $z$  est montrée dans la Figure 4.11 ,comme il apparait les racines sont sur le cercle unitaire.

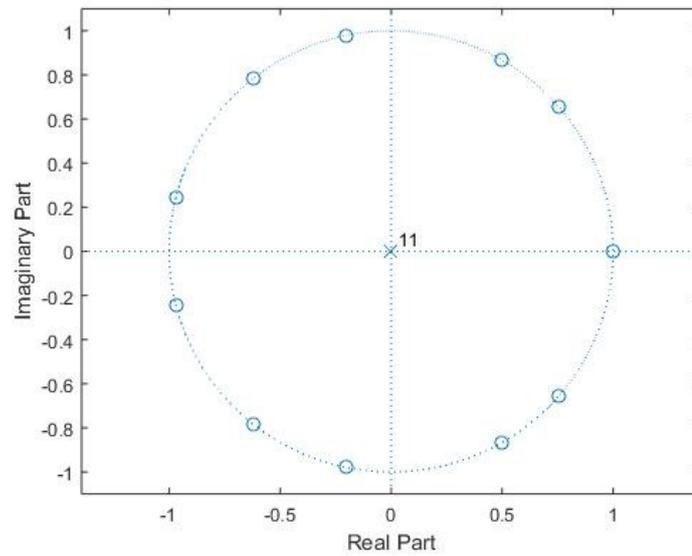


Fig 4.11: Racines du Polynôme LSP Antisymétrique

Les LSFs sont normalisés par rapport à la fréquence d'échantillonnage, c'est pour ça qu'il sont entre  $[0-0.5]$ .les zéros des polynômes  $P(z)$  et  $Q(z)$  de la trame 25 sont représentés par la Figure 4.12:

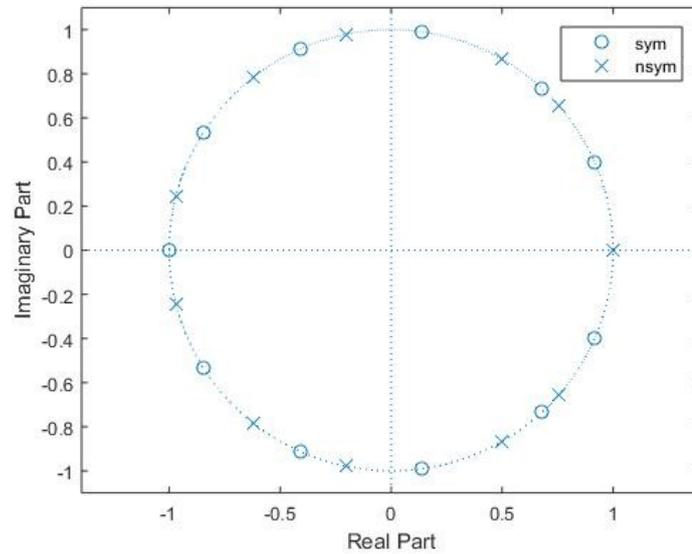


Fig 4.12: Racines da la trame 25 dans le plan z

#### 4.3.4 Test des cas de Mal-conditionnement

Jusqu'à maintenant et d'après les résultat obtenue par la trame 25, On à générés des LSFs à partir d'un filtre LP à phase minimale.

Le cas de mal-conditionnement se produit lorsque le filtre d'analyse utilisé pour généré les LSFs n'est a phase minimale et puis la supposition que les zéros des polynômes LSP qui se trouve à l'intérieur du cercle unité n'est pas valide. Si les zéros des Polynômes LSPs ne sont pas alternés sur le cercle unitaire. Le polynômes LP peut ne pas être à phase minimales aussi.

Un exemple de ce genre est illustré dans la Figure 4.13:

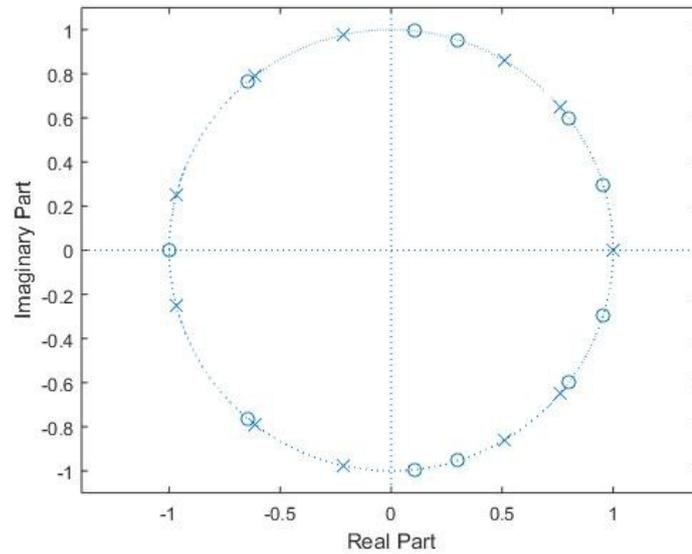


Fig 4.13: Racines des polynômes LSPs pour un filtre à phase non-minimales

En effet le programme teste les cas de mal-conditionnement qui arrive au LSPs et essaye de les corriger, si les LSFs ne sont pas monotones, une tentative de les rendre ainsi est effectuée. si elle échoue il utilise les LSFs précédentes.

### 4.3.5 Quantification des LSFs

#### Méthode 01

Dans la première méthode, les LSF sont quantifiés en utilisant la matrice de quantification présentée par la Table 4.8. Les six fréquences ( $lsf_1$  et  $lsf_6, \dots, lsf_{10}$ ) sont quantifiées en utilisant 3 bits et 4 bits pour le reste ( $lsf_2, \dots, lsf_5$ ), on a donc un total de 34 bits par trame. Les LSFs sont quantifiées de telle façon que la valeur absolue de la distance entre la valeur non quantifiée et le plus proche voisin "niveau de quantification" est minimale.

La première  $lsf_1$  est quantifiée en utilisant les niveaux de quantification présentés sur la première ligne de la matrice donnée dans 4.8,  $lsf_2$  en utilisant la deuxième ligne de cette matrice ... etc. On remarque que sur la même colonne  $lsf_2$ , peut être quantifiée avec une valeur plus petite que  $lsf_1$ , cela est dû au recouvrement de la matrice de quantification.

### 4.3. Calcul et Quantification des LSFs

		Niveaux de Quantification															
Indice de Fréquence	100	170	255	250	280	340	420	500	0	0	0	0	0	0	0	0	
	210	235	265	295	325	30	400	440	480	520	560	610	670	740	810	880	
	420	460	500	540	585	640	705	775	850	950	1050	1150	1250	1350	1450	1550	
	620	660	720	795	880	970	1080	1170	1270	1370	1470	1570	1670	1770	1870	1970	
	100	1050	1130	1210	1285	1350	1430	1510	1590	1670	1750	1850	1950	2050	2150	2250	
	1470	1570	1690	1830	2000	2200	2400	260	0	0	0	0	0	0	0	0	
	1800	1880	160	2100	2300	2480	2700	2900	0	0	0	0	0	0	0	0	
	2225	24400	2525	2650	2800	2950	3150	3350	0	0	0	0	0	0	0	0	
	2760	2880	3000	3100	3200	3310	3430	3550	0	0	0	0	0	0	0	0	
	3190	2370	3350	3420	3490	3590	3710	3830	0	0	0	0	0	0	0	0	

Tableau 4.8: Matrice de Quantification des LSF

Les niveaux de quantification des  $lsf_1, lsf_5, \dots, lsf_{10}$  sont 9 donc 3 bits, pendant que le reste est quantifier sur 16 niveaux (4 bits).

$$bits = [3; 4; 4; 4; 4; 3; 3; 3; 3; 3]$$

Voici un exemple des LSFs quantifier de la trame 25, représenté dans la Table 4.9:

Lsf non-quantifier	Fréquence non-quantifier (Hz)	Indice de la Table Quantification	Fréquence Quantifier (Hz)
0.0476	380.9	7	420
0.0939	751.2	14	740
0.1017	813.9	9	850
0.1388	1110.2	7	1080
0.2012	1609.6	9	1590
0.2213	1770.6	4	1830
0.2233	1866.4	2	1880
0.3325	2660.2	4	2650
0.3612	2889.2	2	2880
0.4026	3220.9	1	3190

Tableau 4.9: LSFs Quantifier (trame 25)

#### 4.3.6 Correction des Quantifications

La quantification des LSFs peut mener à la non-monotonie à cause du recouvrement de la table de quantification 4.8, or qu'il faut garder la monotonie des LSFs pour que les propriétés des deux polynômes restent valable. Donc une vérification de la monotonie des LSFs quantifier est vérifier et s'il se passe que l'lsf suivante et plus petite que l'lsf

### 4.3. Calcul et Quantification des LSFs

---

courante, le problème sera résolu par soit de quantifier lsf suivante par la valeur qu'elle la suite dans la table de quantification ( $lsf_{next} \rightarrow lsf_{next}^+$ ), soit de quantifier l'lsf courante par la valeur précédente ( $lsf_{now} \rightarrow lsf_{now}^-$ ).

Les LSFs quantifier de la trame 25 sont croissants, même pour les autres trames. ce qui nous mène directement à découvrir la deuxième méthode qu'on a proposé.

#### Méthode 02

##### Structure [3;3;4]

Pour la deuxième méthode on a opté à utiliser une quantification vectorielle divisée SVQs [3,3,4]. de tel façon que les trois premiers LSFs [ $lsf_1, lsf_2, lsf_3$ ] sont quantifier avec un quantificateur de dimension 3 et d'une taille de  $2^{10}$  (niveaux) et de même pour [ $lsf_4, lsf_5, lsf_6$ ], pour les quatre LSFs qui restent, dans ce cas un quantificateur de dimension 4 et d'une taille de  $2^9$  et utilisé.

Pour chaque dictionnaire, la séquence d'apprentissage est obtenue en utilisant une distribution gaussienne, à l'aide de la commande "normrnd(E,σ,n point)". Et puis un maillages entre les différentes LSFs dans un dictionnaire est effectué pour initialiser les trois dictionnaires, ou les code-vecteurs non-monotones sont enlevés par une opération de vérification de la monotonie. L'algorithme LBG est utilisé pour construire les dictionnaires optimaux à partir de ces derniers, en spécifiant le nombre de bits pour chaque dictionnaire.

Le Tableau 4.10 présente les LSFs quantifiés en utilisant une SVQ, pour la trame 25, ainsi la distorsion relative associée:

LSFs		0.0476	0.0939	0.1017	0.1388	0.2012	0.2213	0.2333	0.3325	0.3612	0.4026
LSFs quantifiés		0.0460	0.0960	0.1023	0.1327	0.1956	0.2224	0.2313	0.3330	0.3540	0.3955
Distr relative		0.0341	0.0222	0.0057	0.0324	0.0053	0.0019	0.0458	0.0366	0.0041	0.223

Tableau 4.10: LSFs quantifiés par SVQ [3;3;4] (trame 25)

Un autre exemple de la QV est illustré dans le tableau 4.11:

LSFs		0.0362	0.0461	0.0925	0.1243	0.1390	0.2205	0.2367	0.3029	0.343	0.3948
LSFs quantifiés		0.0371	0.0484	0.0916	0.1235	0.1435	0.2213	0.2440	0.3203	0.3597	0.4116
Distr relative		0.0244	0.0501	0.0101	0.0070	0.0330	0.0039	0.0308	0.0575	0.0387	0.0425

Tableau 4.11: LSFs quantifiés par SVQ [3;3;4] (trame 30)

**Structure [3;4;3]**

Pour la structure [3;4;3], la même procédure établie précédemment est adopté en utilisant les même dictionnaires initiaux, les résultats obtenue sur la trame 25 et 30 sont donnés dans la Table 4.12:

LSFs		0.0476	0.0939	0.1017	0.1388	0.2012	0.2213	0.2333	0.3325	0.3612	0.4026
LSFs quantifiers		0.0460	0.0960	0.1023	0.1327	0.1956	0.2224	0.2313	0.3330	0.3540	0.3955
Distr relative		0.0341	0.0222	0.0057	0.0437	0.0276	0.0048	0.0086	0.0014	0.0197	0.0177

Tableau 4.12: LSFs quantifiers par SVQ [3,4;3] (trame 25)

Un autre exemple de quantification SVQ [3;4;3] est montré dans la table 4.13:

LSFs		0.0362	0.0461	0.0925	0.1243	0.1390	0.2205	0.2367	0.3029	0.343	0.3948
LSFs quantifiers		0.0371	0.0484	0.0916	0.1217	0.1342	0.2224	0.2313	0.3069	0.3393	0.3972
Distr relative		0.0244	0.0501	0.0101	0.0210	0.0340	0.0088	0.0228	0.0130	0.0202	0.0061

Tableau 4.13: LSFs quantifiers par SVQ [3;4;3] (trame 30)

Le choix du dictionnaire initial est capital pour une conception efficace de la quantification vectorielle, car il conditionne les résultats finaux de l'algorithme LBG, et comme le temps de calcul est tellement grand "ressources non-disponible" et dans le but de comparer les méthodes de quantifications, une même base de données doit être utilisée.

Pour mieux voir les résultats donnés par les différents quantificateurs, un tableau qui résume tout ça est donné dans 4.14:

### 4.3. Calcul et Quantification des LSFs

Mots	Mesure		Nbits	Débit (Kbits/s)	Dis(abs)	Dis(EQM)	$SNR^+$ (dB)	$SNR^-$ (dB)
	Quantizer							
Test	scalar		34	1.133	0.0019	7.2858e-05	18.2621	
	[3;3;4]		29	0.967	0.0190	1.0582e-04	12.5608	20.5254
	[3;4;3]		29	0.967	0.0086	2.4782e-04	13.4170	
One	scalar		34	1.333	0.0013	5.3032e-05	21.8858	
	[3;3;4]		29	0.967	0.0186	2.2942e-04	15.3442	24.2751
	[3;4;3]		29	0.967	0.0073	9.2082e-05	16.2653	
Télé- phone	scalar		34	1.133	0.0020	4.5159e-05	16.1749	
	[3;3;4]		29	0.967	0.207	2.9325e-04	11.2048	18.0788
	[3;4;3]		29	0.967	0.0092	1.2679e-04	11.9741	
Five	scalar		34	1.133	0.0058	1.0615e-04	17.2735	
	[3;3;4]		29	0.967	0.024	2.8337e-04	11.3150	19.4644
	[3;4;3]		29	0.967	0.0093	1.2776e-04	12.2571	

Tableau 4.14: Comparaison entre QVs [3;3;4] & [3;4;3], & SQ

### Interpretation des Résultats

Les premiers tests ont été fait sur les deux quantificateurs SQ et SVQ, et on a vue une diminution de débit de 0.967 Kbits/s contre 1.333 Kbits/s pour le Quantificateur vectorielle à division (29 bits/trame alloués pour la SVQ). En effet, les poids donnés aux coefficients LSFs ont contribués dans la diminution de la distorsion moyenne ainsi que sur le SNR.

On peut remarquer que la technique de quantification SVQ avec la structure [3;4;3] à contribué d'une manière efficace à la diminution de cette grandeur (bit-rate). Pour [3;4;3] la distorsion moyenne a subit une diminution légèrement supérieur a celle de la structure [3;3;4] ainsi pour le rapport signal sur bruit (SNR), et ca pour les différent mots testé. Quand on passe du quantificateur SQ au quantificateur SVQ une différence de résultat est aussi remarqué du au allocations de bits de 34bits/trame et l'utilisaiton de la base TIMIT.

En conclusion, on peut dire que la technique de Quantification vectorielle a donné une distorsion réduite vue le nombre de bits alloué pour le vecteur LSF. "La comparaison du point de vu allocation de bits n'est pas égale". Et cela est due au calcule fastidieux exigé par LBG qui demande des ressources supérieures a celle que nous disposant.

Les polynômes LSP,  $P(z)$  et  $Q(z)$  sont générés à partir des coefficients LP,  $P(z)$  est un polynôme avec des coefficients symétriques paires tandis que  $Q(z)$  est un polynôme avec des coefficients symétrique impaire, les zéros des polynômes LSP sont calculés en utilisant la méthode de Descartes. Puis les coefficients LSF sont quantifier on utilisant deux méthode la première s'agit d'une quantification scalaire par split où chaque LSF est quantifier toute seule en utilisant une matrice de Quantification LSPQ, la deuxième est de

quantifier les LSFs conjointement pour diminué la distorsion et gagné des bits pour réduire le bit-rate du codeur, la deux structure sont étudier pour choir la meilleur d'entre eux. Pour la première méthode un test de mal-conditionnement est obligatoire pour trillé les LSFs est gardé la propriété du phase-minimal.

## 4.4 Estimation du pitch en boucle ouverte

Le calcule du pitch nécessite une division de la trame de parole en 4 sous-trames, cela peut être accompli par la commande "vec2cell" disponible sur MATLAB, et aussi d'utiliser le codebook adaptative car cette opération représente l'étape la plus délicate dans l'algorithme CELP. Mais comme notre but c'est d'étudier l'algorithme CELP et d'essayer de le modifier "SVQ", pour cela on a calculé le pitch par boucle ouverte.

Cette approche exige de faire une analyse à long terme du signal originale, elle est basée sur une minimisation de l'erreur à la sortie du filtre prédicteur à court terme figure donné dans l'équation 4.3. L'erreur à minimiser est obtenue pour chaque ligne analysée de la matrice  $data_s$ , ce qui équivaut à déterminer un pitch et un gain du filtre LTP pour chaque nouvelle trame du signal originale.

L'erreur quadratique moyenne de la sortie du filtre prédicteur est donné par :

$$E = \sum_{n=0}^{N-1} r(n)^2 = \sum_{n=0}^{N-1} [\epsilon(n) - b_0 \epsilon(n - \alpha)]^2 \quad (4.3)$$

on prend

$$\frac{\partial E}{\partial b_0} = 0 \Rightarrow b_0 = \frac{\sum_{n=0}^{N-1} \epsilon(n) \cdot \epsilon(n - \alpha)}{\sum_{n=0}^{N-1} [\epsilon(n - \alpha)]^2} \quad |b_0| \leq 1 \quad (4.4)$$

et en remplaçant  $b_0$  dans EQM résiduel:

$$E = \sum_{n=0}^{N-1} r(n)^2 - \frac{\left[ \sum_{n=0}^{N-1} \epsilon(n) \cdot \epsilon(n - \alpha) \right]^2}{\sum_{n=0}^{N-1} [\epsilon(n - \alpha)]^2} \quad (4.5)$$

Le filtre prédicteur présente à sa sortie un signal prédit  $\hat{s}$ , ses coefficients sont les coefficients LP obtenus de l'analyse à court terme, ces derniers sont appliqués pour chaque trame du signal (matrice  $data_s$ , la différence entre chaque trame du signal "s(n) et  $\hat{s}(n)$  (ligne du matrice " $data_s$  et  $data_{\hat{s}}$ ") engendre une erreur  $e(n)$ , l'erreur de prédiction est donc représentée par une matrice dont le nombre de lignes est le nombre de trames  $data_e$ , la figure 4.14 présente la 30<sup>ème</sup> trame du signal prédit comparée à la trame du même ordre du signal original ainsi que la trame de l'erreur de prédiction.

## 4.5. Codebook Stochastique

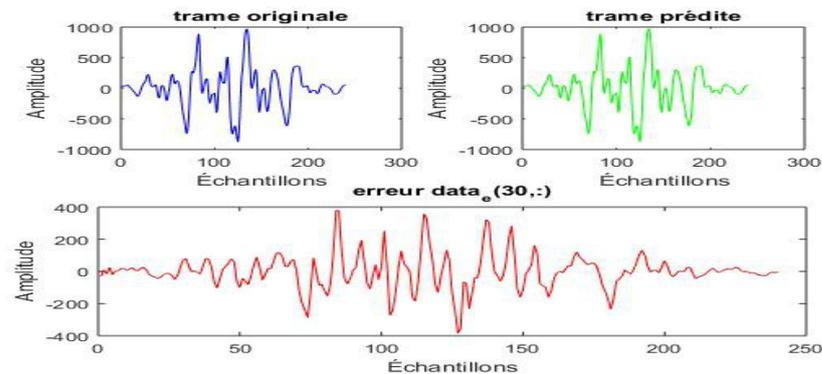


Fig 4.14: Analyse courte et long terme sur la 30<sup>ème</sup> trame

L'erreur de prédiction  $e(n)$  sera ensuite minimisée en minimisant l'équation 4.5, le but de cette minimisation est la détermination du pitch et du gain du filtre LTP, cela en cherchant le pitch qui donne la plus petite erreur de prédiction, pour un intervalle de  $[20;147]$ , puis le remplacer dans l'équation 4.4 pour retrouver le gain correspondant.

Les résultats obtenue par le programme LTP.m qu'on à écrit sur MATLAB et appliqué sur le signal sont donnés dans la Table 4.15 :

Trame	$tr_{10}$	$tr_{17}$	$tr_{23}$	$tr_{25}$	$tr_{30}$	$tr_{34}$	$tr_{37}$
Pitch (Hz)	170.24	333.33	205.12	195.13	150.94	137.01	380.95
Gain	-0.3270	0.3574	0.7821	0.4257	0.5467	0.6976	-0.3595

Tableau 4.15: Pitch & Gain

## 4.5 Codebook Stochastique

Pas mal de types de codebook ont été conçus pour répondre au différents niveaux d'exigences et de performance avec le degré de complexité de calcul et de stockage. Ceux couramment utilisé dans le speech coding incluent: à recherche complète, à recherche binaire, aléatoire (random), adaptatif, cascade et vecteur split.

Dans notre cas on à opté a utilisé un codebook aléatoire. les codebooks aléatoires sont attirants car ils ne nécessitent pas de formation. La conception d'un dictionnaire aléatoire est très simple et ses mots de code peuvent être choisis au hasard parmi une base de données de formation. La performance d'un codebook aléatoire se rapproche de celle d'un optimum codebook lorsque sa taille  $L$  et sa dimension  $N$  augmente.

Pour cela on utilise la commande "rand" sur MATLAB pour créer un codebook de valeurs entre  $[-1,1]$ , qui n'est qu'une matrice  $1024 \times 240$  ou sa taille et 1024 et sa dimension et de 240 "échantillons". le critère du voisin le plus proche est appliqué pour choisir le code vecteur approprié, et ça après le calcul de la cible "excitation" à la sortie du détecteur du pitch et la normalisé, ce dernier supprime les impulsion périodique du résiduel après la prédiction a courte terme. le tableau 4.16 résume les indices des code-valeurs obtenues par le programme cb.mat qui crée le dictionnaire, et target.mat pour le calcul de la cible, ainsi la distorsion moyenne de chaque code-vecteur.

Trame	Indice du code-vecteur	Distorsion moy- enne
<i>tr</i> <sub>14</sub>	78	0.0352
<i>tr</i> <sub>22</sub>	728	0.0839
<i>tr</i> <sub>25</sub>	882	0.0735
<i>tr</i> <sub>27</sub>	78	0.0341
<i>tr</i> <sub>30</sub>	88	0.0413
<i>tr</i> <sub>34</sub>	39	0.0321
<i>tr</i> <sub>36</sub>	531	0.0434

Tableau 4.16: Indice d'excitation et cible calculé pour le mot "One".

## Conclusion

Dans ce chapitre nous avons bien mis en évidence les notions que nous avons nouvellement acquit dans les chapitres précédents, cela grâce l'implémentation des principaux blocs du codeur, de celle-ci nous avons obtenu des résultats et nous avons pu voir concrètement le fonctionnement de l'algorithme ainsi faire une comparaison.

# Conclusion et Perspective

Dans ce projet de fin d'études, en s'inspirant des codeurs déjà existant, nous avons acquis beaucoup de notions de base concernant le codage de la parole, qui nous ont aidés à proposer un codeur (expliqué en détaille dans le chapitre 4).

Au cour de la réalisation du projet, nous avons rencontré beaucoup d'obstacles et de problèmes réels qui ont excité notre curiosité de vouloir savoir, et que nous les considérons comme un premier pas dans la recherche scientifique.

La qualité de la parole peut être améliorée en utilisant un codebook adaptatif pour l'estimation du pitch.

L'utilisation d'autre codebook: algébrique, statistique ... peut réduire le débit au-de-la de 4.8 Kb/s.

# Bibliographie

- [1] L. A. Meacham and E. Peterson. "An experimental multichannel pulse code modulation Bell Sys. Technical Journal, 27: 1-43,1948.
- [2] CCITT: Rec. G. 722. "64 kb/s SB-ADPCM", 1988.
- [3] B. S. Atal and M. I.t. Schroeder. "Stochastic coding of speech signals at verqy low bit rates". In Proc. of IEEE Int. Conf. on Communications, pages 1610-13, Amsterdam, Netherlands, May 1984.
- [4] M. Schroeder and B. S. Atal. "Code Excited Linear Prediction (CELP): High Quality Speech at Low Bit Rates". In Proc. of ICASSP, pages 937-940, Florida, 1985.
- [5] X. Huang. A. Acero, H. Hon. "Spoken language processing a guide to theory, algorithm and system design", Prentice Hall 2001.
- [6] R.Boite et M.Kunt, "Traitement de la parole". Presses polytechniques romandes .1987.
- [7] W.C. Chu. "Speech Coding Algorithms, Foundation and Evolution of Standardized Coders", John Wiley et Sons, Hoboken, 2003.
- [8] J. Makhoul. "Linear prediction: A Tutorial Review", Proceedings of the IEEE, vol.634 pp 561-578, 1975.
- [9] K.K. Paliwal ,W.B. Kleijn. "Quantization of LPC parameters" in Speech Coding and Synthesis, W.B. Kleijn and K.K. Paliwal, Ed. Amsterdam: Elsevier, 1995,pp.443-466.
- [10] S.Furui. " Digital Speech processing, Synthesis, Recognition", Second edition,Marcel Dekker 2001.

- [11] International Telecommunication Union (ITU). "Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear prediction" (CS-ACELP). Recommendation UIT-T G729, March 1996.
- [12] T.Chmayssani, "Modulations sur les canaux vocodés". Thèse de doctorat ès sciences, Université Paris EST. 2010.
- [13] I.Gouicem et S.Soltana, "Implémentation de l'algorithme de codage par prédiction linéaire sur un circuit FPGA". Projet de fin d'études E.N.P. 2008.
- [14] A. Gersho, R.M. Gray, Vector quantization and signal compression, Englewood Cliffs, N.J. Prentice-Hall, 1990.
- [15] A.S. Spanias. "Speech Coding : A Tutorial Review", Proc.IEEE, 82 (1994), pp.1541-1582.
- [16] M. R. Schroeder and B. S. Atal, 'Code-excited linear predictive (CELP) : High quality speech at very low bit rates,' IEEE ICASSP, pp. 937-940, March 1985.
- [17] J. P. Campbell, T. E. Tremain, and V. C. Welch, 'The proposed Federal Standard 1016 4800 bps Voice Coder : CELP', Speech Technology Magazine, pp. 58-64, April 1990.
- [18] N. Moreau, 'Codage prédictif du signal de parole a débit réduit : une présentation unifiée', Annales des Télécommunications, Vol. 46, No. 3-4, pp. 223-239, 1991.
- [19] John Makhoul, 'Linear Prediction: A Tutorial Review,' Proc. of the IEEE, Vol. 63, No. 4, April 1975.
- [20] N. Sugamura and F. Itakura, 'Speech Analysis and Synthesis Methods Developed at ECL in NTT' From LPC to LSP,' Elsevier Speech Comm., pp. 199-215, Jan. 1986.
- [21] P. Kabal, R.P. Ramachandran, 'The computation of line spectral frequencies using Chebyshev polynomials', IEEE Trans. Acoust. Speech, Signal Process. Vol. 34, pp. 1419-1426, Dec. 1986.
- [22] N. Naja, 'Construction de dictionnaires et Quantification vectorielle pour les codeurs de parole LSP-CELP', Doctorat de l'Université de Rennes I, Mention Traitement du Signal et Télécommunications, Juin 1994.
- [23] E.R.Thepie, "Réduction du bruit et annulations de l'écho acoustique dans le domaine des paramètres des codeurs de type CELP, intégrés dans les réseaux mobiles". Thèse de doctorat ès sciences, TELECOM Bretagne. 2009.

- [24] M.De Meuleneire,"Codage imbriqué pour a parole à 8-32 Kb/s combinant thechniques CELP, ondelettes et extension de bande". Thèse de doctorat ès sciences, E.N.S.T Bretagne. 2007.
- [25] ITU-T Recommendation G.729, 'Coding of speech at 8 kbits/s using conjugate-structure algebraic-code-excited linear prediction (CS-ACELP)', ITU-T Publication, Jan. 2007.
- [26] C. Laflamme, J-P Adoul,H. Y. Su, and S. Morissette, 'On Reducing Computational Complexity of Codebook Search in CELP Coder Throught the Use of Algebraic Codes', IEEE ICASSP, pp. 177-180, 1990.
- [27] B. Bessette, R. Salami, R. Lefebvre, M. Jelínek,J. R.-Pukkila, J. Vainio, H. Mikkola, and K. Järvinen 'The Adaptive Multirate Wideband Speech Codec (AMR-WB)', IEEE Trans. on Speech and Audio Process., Vol. 10, No. 8, pp. 620-636, Nov. 2002,
- [28] R. Salami, C. Laflamme, J-P. Adoul, K. Jarvinen, J. Vainio, P. Kapanen, T. Honkanen, and P. Haavisto. 'GSM Enhanced Full Rate Speech Codec", IEEE ICASSP, Vol.2, pp. 771'774, 1997.
- [29] B. S. Atal and M. R. Schroeder, 'Predictive coding of speech signals and subjective error criteria,' IEEE trans. on Acoustics, Speech and Signal Process., Vol. 27, No. 3, pp. 247254, June 1979
- [30] S. Haykin, Adaptive Filter Theory, Prentice-Hall, New Jersey, 1996. 18, 26, 53