

**RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE**

**MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR  
ET DE LA RECHERCHE SCIENTIFIQUE**

**ÉCOLE NATIONALE POLYTECHNIQUE**



**DÉPARTEMENT D'ÉLECTRONIQUE**

**Projet de Fin d'Études**

**En vue de l'obtention du  
Diplôme d'Ingénieur d'État en Électronique**

**Intitulé :**

**Implémentation de l'Algorithme de Codage par  
Prédiction Linéaire sur un Circuit FPGA**

Réalisé par :

M. GOUICEM Imad-Eddine

M. SOLTANA Samir

Proposé et dirigé par :

Pr D. BERKANI

Promotion : Juin 2008

## ملخص

في هذا العمل قمنا بتمثيل خوارزمية التعرف الخطي على دارة قابلة للبرمجة بالحقل , لهذا درسنا إنتاج و بعض مميزات الصوت. ثم درسنا بعض تقنيات التشفير و اخترنا خوارزمية التعرف, الخطي هذا ما جعلنا ندرس الدارة القابلة للبرمجة بالحقل و لغة الوصف المطورة للدارة , وأخيرا قمنا بوصفنا دارة للإخراج عوامل الارتباط الذاتي وبرمجنا خوارزمية ليفنسون داربان

كلمات مفتاحية: التمثيل على دارة FPGA، تشفير تنبئي، Levinson-Durbin، شعاع الارتباط الذاتي , دارات الحساب

## Résumé

Dans ce travail, nous avons implémenté une partie de codage par la prédiction linéaire sur FPGA, pour cela on a étudié la production et quelques propriétés de la parole, puis on a fait une vision sur les techniques de codage et parmi ces technique nous avons choisi à implémenter la prédiction linéaire, ce qui nécessite une étude de l'architecture de la carte VIRTEX II et le langage de description VHDL, et enfin nous avons fait une conception du circuit de calcul pour l'autocorrélation et un programmes pour l'algorithme de Levinson Durbin.

**Mots clés :** implémentation sur circuit FPGA, codage prédictif, Levinson-Durbin, vecteur autocorrélation, circuits de calculs.

## Abstract

In this work we implemented the linear prediction on FPGA. For that, we have studied the production, some properties of the speech, then we have printed a vision on the techniques of coding and among these techniques we chose to implement the linear prediction, which requires to study's the architecture of VIRTEX II and the language of description VHDL. Finally we made a design of channel computation for the autocorrelation and programs for the algorithm of Levinson Durbin.

**Key word:** implementation on FPGA, predictive coding, Levinson-Durbin, autocorrélation vector, calculating circuits.

# 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 tenons à exprimer nos vifs remerciements à notre promoteur Monsieur **Daoued BERKANI** Professeur au département de l'électronique de l'École Nationale Polytechnique pour nous avoir encadré durant notre projet de fin d'études et nous conseillé tout le long de notre travail.

Nous remercions également Monsieur **R. SAADOUN**, directeur du centre de calcul de l'École Nationale Polytechnique.

Nous remercions également Madame **L. HAMMAMI**, Maitre de conférences à l'École Nationale Polytechnique.

Nous remercions également Monsieur **C. LARBESSE** , Maitre de conférences à l'École Nationale Polytechnique.

Nous remercions chaleureusement les membres du jury pour l'honneur qu'ils nous ont fait en acceptant d'évaluer notre travail.

Enfin, nous aimerions adresser nos plus fervents remerciements à nos parents, car nul autres qu'eux se sont plus sacrifiés pour notre bien et l'accomplissement de nos projets. Ils ont fait de nous ce que nous sommes aujourd'hui, et pour cela, nous leurs dédions ce mémoire.

Que tous ceux qui ont contribué de près ou de loin à la réalisation de ce modeste travail « **GEULLAL, DIBE, AMINA, IMANE,...**» trouvent ici l'expression de notre sincère gratitude.



2.5.1. Principes.....	32
2.5.2. Quantification Par Split.....	34
2.6. Principe d'analyse Par Synthèse.....	35
2.7. Les Codeurs à Bas Débit et à Très Bas Débit .....	36
2.7.1. Codage à Excitation Multi-Pulse et Regular-Pulse.....	36
2.7.2. Le Codeur CELP.....	37
2.7.3. Les Codeurs LPC à Excitation Mixte ou MELP.....	38
2.7.4. Codeur à Excitation Multibande (MBE).....	39
2.8. Critères Relatifs au Codage .....	39
2.8.1. Débit de Transmission .....	40
2.8.2. Délai de Codage.....	40
2.8.3. Qualité de Parole.....	40
2.8.4. Mesures Subjective de La Qualité de Parole .....	40
2.8.5. Mesures objectives.....	40
2.9. La décomposition en valeurs singulières SVD .....	42
2.9.1. Principes.....	42
2.9.2. Interprétation Géométrique de la SVD .....	43
Conclusion .....	44
 <i>Introduction aux FPGA</i> .....	 45
Introduction.....	45
3.1. Architecture adoptée par Xilinx.....	45
3.2. Les circuits configurables .....	46
 <i>Le langage VHDL</i> .....	 53
Introduction.....	53
4.1. La structure d'une description VHDL .....	54
4.2. Déclaration des bibliothèques.....	55
4.3. Déclaration de l'entité et des entrées/sorties.....	55
4.4. Déclaration de l'architecture – description du fonctionnement.....	56
4.5. Les instructions de base de la logique combinatoire .....	57
4.6. Les instructions du mode séquentiel .....	59
4.7. Les types .....	60
Conclusion .....	62
 <i>La Carte Virtex-II de Xilinx</i> .....	 63
5.1. La carte système Virtex-II .....	63
5.1.1. Description de la carte système Virtex-II .....	63
5.1.2. La mémoire DDR.....	64
5.1.3. Génération de l'horloge .....	65

5.1.4. Le circuit Reset .....	65
5.1.5. Le port RS 232 .....	66
5.1.6. Le port JTAG .....	67
5.1.7. Le voltage du banc d'entrée/sortie .....	68
5.1.8. ISP PROM .....	68
5.2. Chargement des conceptions.....	69
<i>Programmation et implémentation .....</i>	<i>70</i>
Introduction.....	70
6.1. Les différentes architectures proposées .....	70
6.2. Les blocs élémentaires utilisés pour l'implémentation.....	72
6.2.1. Le registre FIFO.....	72
6.2.2. Le diviseur .....	74
6.2.3. Le multiplieur.....	76
6.2.4. Le multiplieur/accumulateur .....	77
6.2.5. Le multiplexeur .....	78
6.3. Les blocs fonctionnels.....	79
6.3.1. Le bloc d'autocorrélation.....	79
6.3.2. Le bloc de Levinson-Durbin .....	81
CONCLUSION GENERALE.....	88
BIBLIOGRAPHIE.....	90

# INTRODUCTION GENERALE

Les techniques de codage de parole ont connu des développements importants dans les dernières décennies. Plusieurs codeurs ont été proposés pour fournir des débits de plus en plus faibles pour une qualité accrue et pour une large gamme d'applications.

D'une manière générale, le but de codage est de réduire le nombre de bits utilisés pour représenter la forme d'onde discrétisée est de maintenir une qualité acceptable pour des raisons de transmission et de stockage. Habituellement le codage avec perte signifie que les formes d'ondes ne peuvent pas être complètement reproduites par le décodeur.

L'objectif principal dans le codage à bas débit n'est pas reproduire un signal qui est physiquement identique au signal parole original mais de reproduire un son qui semble identique, seules les informations utiles à un auditeur humain sont retenues [1].

La plupart des codeurs de parole sont basés sur la prédiction linéaire, leurs efficacités viennent du modèle simple sur lequel ils se basent. Les méthodes de codage prédictif nécessitent une représentation efficace des coefficients du filtre LPC et de son excitation.

Bien que le progrès important qui a été fait dans le codage des paramètres LPC, ce n'est pas encore possible de coder l'excitation à bas débit et maintenir la haute qualité de la voix dans le signal parole reconstituée.

Les transformées orthogonales offrent des méthodes efficaces de réduction du débit nécessaire pour le codage et la transmission du signal. L'efficacité d'un système de codage par transformation dépend du type de transformation et l'allocation de bits au cours du codage. La plupart des systèmes pratiques sont basés sur les approches sous-optimales de la transformation ainsi que l'allocation de bits.

Dans le Chapitre I nous avons étudié la prédiction linéaire et la modélisation de la parole. Ensuite, dans le chapitre II, nous avons étudié les différentes techniques de codage.

Ensuite, dans les chapitres III et IV, à la présentation des FPGA et du leur langage de programmation le VHDL en l'occurrence. Dans le chapitre V, nous avons étudié la carte Xilinx Virtex-II que nous avons utilisée pour implémenter notre algorithme.

Dans le chapitre IV, on a expliquer les différentes étapes de conception, différentes simulations de blocs réalisés, les différentes propositions pour l'implémentation de notre algorithme et les problèmes rencontrés.

# Prédiction Linéaire

## Introduction

Pour réaliser une analyse efficace du signal parole au niveau acoustique, c'est avantageux d'exploiter la connaissance du processus de la production de parole, cette connaissance est utile pour sélectionner un modèle paramétrique convenable pour la production de parole, une fois le modèle est sélectionné, le rôle des techniques d'analyse de la parole est d'estimer correctement et efficacement les paramètres de ce modèle.

### 1.1 La Production de la Voix

La parole apparaît physiquement comme une variation de la pression de l'air causée est émise par le système articulatoire. Quand une personne parle, sous le contrôle du système nerveux central qui reçoit en permanence des informations par rétroaction auditive et par les sensations kinesthésiques, 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 pharyngienne 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 un instrument de musique produisant un son. En fait, 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 [2].

Pour produire des sons nasaux, la cavité nasale est souvent incluse dans le conduit vocal. La cavité nasale est connectée en parallèle avec la cavité buccale. Le conduit vocal simplifié est montré dans la Figure 1.1.

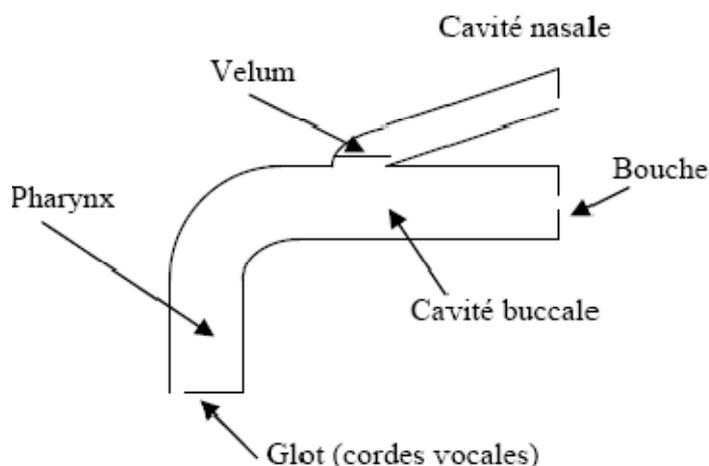


Fig.1.1. Schéma simplifié du conduit vocal

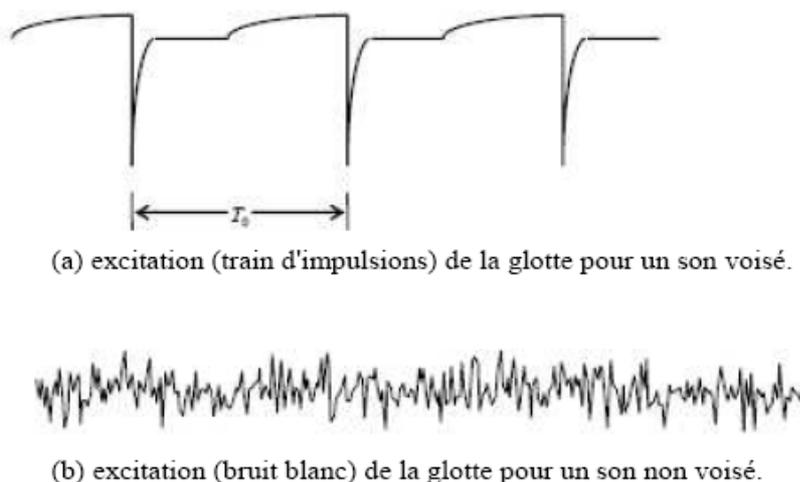
### 1.1.1. Quelques propriétés importantes de la parole

- Les Fricatives • (s, sh, f, th) sont produites quand le conduit vocal est resserré à quelques emplacements et l'air est forcé à travers ces resserrements.
- Les Plosives • (p, k, t) sont produites quand la fin du conduit vocal est resserré ou est fermée momentanément au moment où la pression atmosphérique a développé, ensuite la pression est libérée brusquement.
- Il y'a approximativement 40 phonèmes (éléments de sons) en anglais (16 voyelles, 24 consonnes).
- Dans la parole normale, 10 à 15 phonèmes sont parlés en une seconde [2].

### 1.1.2. Propriétés des Sons Voisés et des Sons Non Voisés

- Les sons voisés (ou *sonores*), tels que les voyelles, sont produits par le passage de l'air des poumons à travers la trachée qui met en vibration les cordes vocales. Ce mode, qui représente 80% du temps de phonation, est caractérisé en général par une quasi-périodicité et une énergie élevée.
- Les sons non-voisés (apériodique), comme des consonnes, sont obtenus par resserrement du conduit vocal, et ont habituellement une énergie inférieure aux sons voisés. Les cordes vocales sont écartées et n'entrent pas en vibration. Ces sons sont

considérés comme ayant les mêmes caractéristiques que le bruit. Ceux-ci sont montrés dans Figure 1.2.



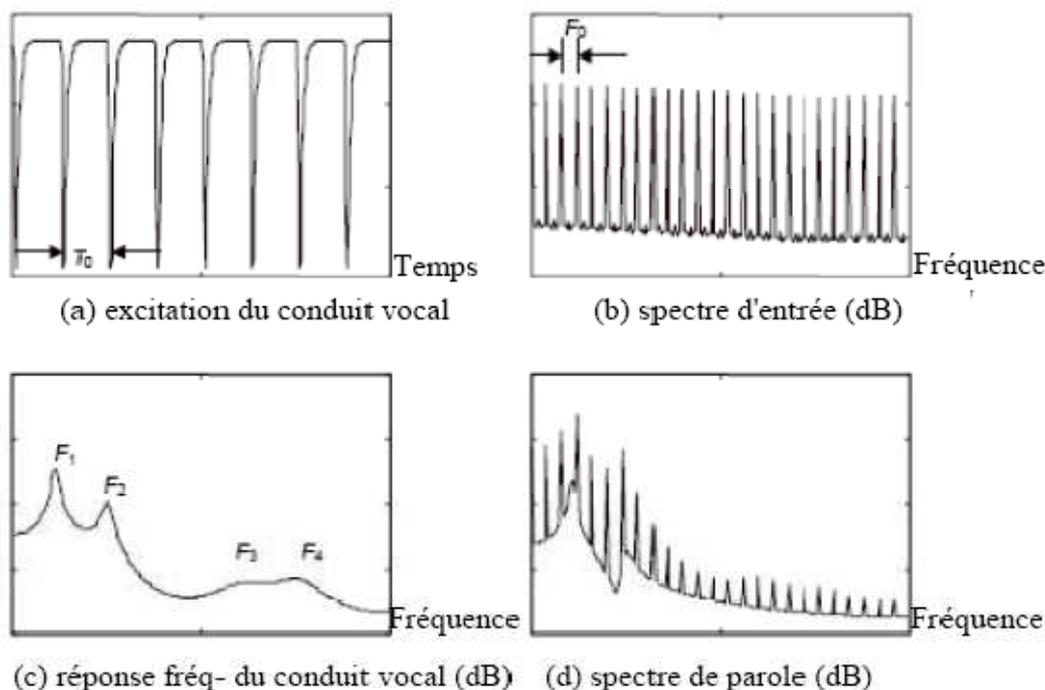
**Fig.1.2.** Deux types d'excitation pour générer le son ( $T_0$ :période du pitch).

La fréquence de pitch  $F_0$  ( $1/T_0$ ) varie d'une personne à l'autre, pour un petit enfant elle peut aller au maximum à 400 Hz, elle s'étend approximativement de 70 à 250 Hz chez les hommes, de 150 à 400 Hz chez les femmes, elle évolue lentement dans le temps.

Cette pulsation de glotte excite la cavité du conduit vocale et produit une voyelle (son voisé). Quelques caractéristiques d'un son vocalique typique sont montrées ci-dessous [3].

Comme montré dans la Figure 1.3 (a), il y'a au moins quatre sommets (pics) résonnants visibles. Les sommets résonnants se produisent à des fréquences dites formants.

Les fréquences formants  $F_1$  et  $F_2$  sont très distinctes par contre les Formants  $F_3$  et  $F_4$  ne sont pas tout à fait distinctes, ces fréquences sont souvent utilisées pour la reconnaissance de la parole. En observant le spectre de la parole, on peut voir qu'il y'a beaucoup d'harmoniques de  $F_0$ , fréquence du pitch, à cause de la richesse en sommets harmonieux il n'est pas facile d'extraire les formants du spectre. D'un autre côté, en entrant un bruit - comme signal d'excitation au conduit vocal, les sons non voisés tels que les Plosives et les Fricatives sont produits [3].



**Fig1.3.** Caractéristiques d'un son vocalique typique

### 1.1.3. Modélisation du Processus de 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 récursif 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* [4].

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 parole est la convolution de l'excitation et la réponse impulsionnelle du système linéaire de génération de 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 linéaires.

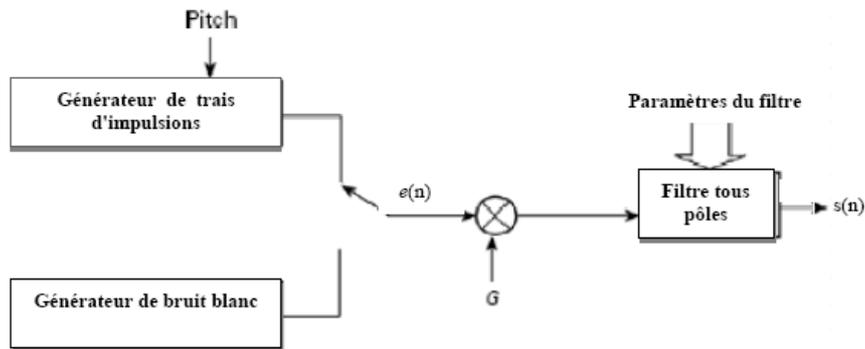


Fig. 1.4. Modèle simplifié de production de la parole:  $G$  est le gain,  $e(n)$  l'excitation,  $s(n)$  le signal parole.

## 1.2. La Prédiction Linéaire

### 1.2.1. Principes

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 donnée par [2] [5]:

$$\hat{s}(n) = a_1 s(n-1) + a_2 s(n-2) + \dots + a_M s(n-M) = \sum_{i=1}^M a_i s(n-i) \quad (1.1)$$

La différence entre l'échantillon actuel et l'échantillon prédit est appelée erreur de prédiction, elle est 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 ( $a_i$ ).

### 1.2.2. 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:

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

Si le signal  $s(n)$  est supposé égal à zéro pour  $n < 0$  et  $n > N$  (par exemple, en le multipliant par une fenêtre de 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 [5]:

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

Où

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

L'équation (4) correspond à l'équation matricielle :

$$\begin{bmatrix} R_0 & R_1 & R_2 & \cdots & R_{M-1} \\ R_1 & R_0 & R_1 & & R_{M-2} \\ R_2 & R_1 & R_0 & & R_{M-3} \\ \vdots & & & \ddots & \vdots \\ R_{M-1} & R_{M-2} & R_{M-3} & \cdots & R_0 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_M \end{bmatrix} = \begin{bmatrix} R_1 \\ R_2 \\ R_3 \\ \vdots \\ R_M \end{bmatrix} \quad (1.5)$$

La matrice d'autocorrélation est une matrice de 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 [2].

### 1.2.3. Algorithme de Levinson-Durbin

Les valeurs initiales

$$E_0 = r(0)$$

$$a_{11} = \kappa_1 = r(1) / E_0$$

$$E_1 = E_0(1 - \kappa_1)^2.$$

avec  $m \geq 2$ , la récursion suivante est exécutée

$$(i) \quad q_m = r(m) - \sum_{i=1}^{M-1} a_{i(m-1)} r(m-i)$$

$$(ii) \quad k_m = q_m / E_{m-1}$$

$$(iii) \quad a_{mm} = k_m$$

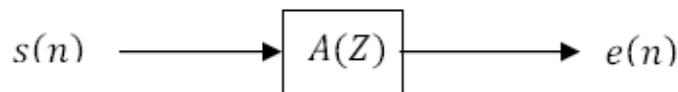
$$(iv) \quad a_{im} = a_{i(m-1)} - k_m a_{(m-i)(m-1)} \quad \text{pour } i = 1, \dots, m-1$$

$$(v) \quad E_m = E_{m-1}(1 - \kappa_m)^2$$

(vi) si  $m < M$ , augmenter  $m$  à  $(m+1)$  et aller à (i).

si  $m = M$ , arrêter.

Une fois les coefficients de prédiction sont calculés, l'équation (1.2) peut être utilisée pour trouver la séquence d'erreurs Figure 1.5.

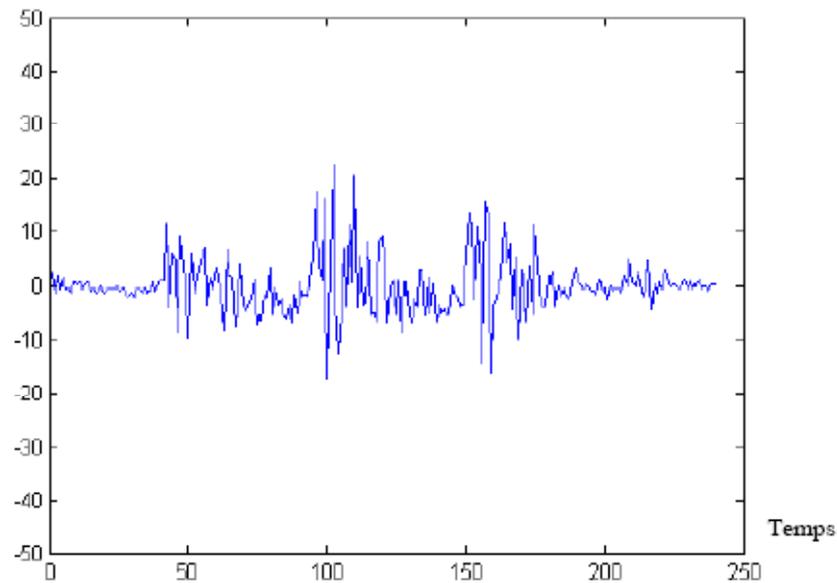


**Fig.1.5.** Filtre d'analyse

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

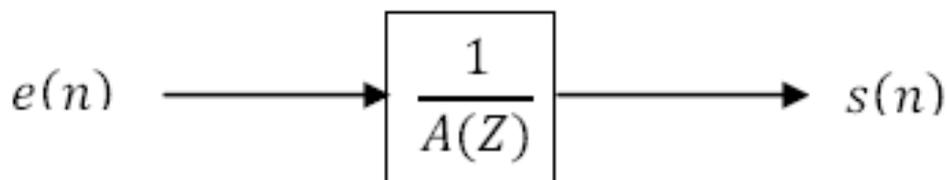
Les échantillons  $s(n)$  ayant une variance plus faible que le signal parole, ils nécessitent moins de bits pour être représentés.

Le signal 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  $s(n)$  et  $e(n)$ .

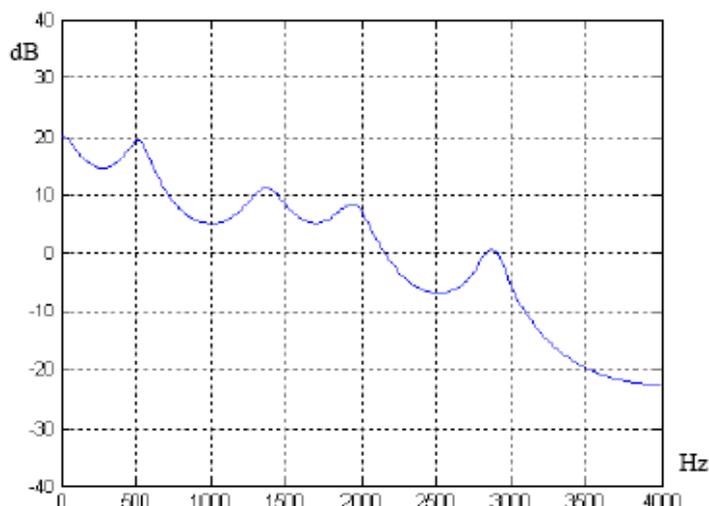


**Fig.1.7.** Filtre de synthèse

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 de la voyelle / uh /, calculé par l'utilisation de la méthode d'autocorrélation avec  $M=10$ , la fenêtre de Hamming est utilisée.



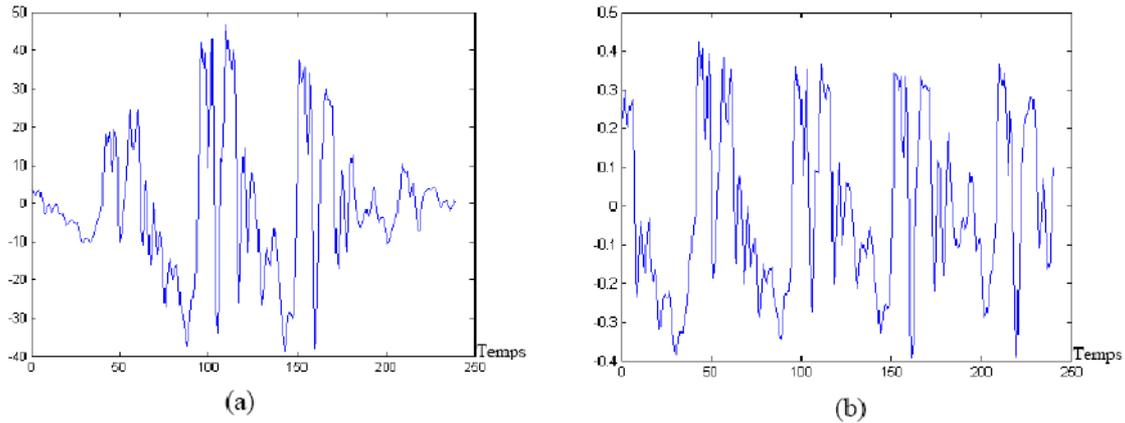
**Fig.1.8.** Spectre d'amplitude du filtre LPC

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.

### 1.3. Analyse de Fourier à Court Terme

L'analyse de Fourier est la technique traditionnelle pour calculer le spectre d'amplitude et de phase du signal parole. La transformée de Fourier standard exige que le signal soit disponible pour tout le temps (c-à-d de moins l'infini à plus l'infini). Puisque le signal parole est de nature non stationnaire, il devient nécessaire d'utiliser une analyse de Fourier à court terme, on pondérant les échantillons de parole par une fenêtre de pondération (souvent une fenêtre de Hamming) Figure 1.9, et en effectuant une transformée de Fourier sur ces échantillons, la DFT (transformée de Fourier discrète) est calculée par l'algorithme (FFT) [2].

$$w(n) = \begin{cases} 0,54 - 0,46 \cos\left(\frac{2\pi n}{N}\right), & 0 \leq n \leq N \\ 0 & \text{ailleurs} \end{cases} \quad (1.9)$$



**Fig.1.9.** 240 échantillons de signal parole "uh ", (a) : fenêtre de Hamming, (b) : fenêtre carrée.

#### 1.4. Prédiction Linéaire rétrograde (Backward) et Filtre en Treillis

La prédiction linéaire peut aussi être appliquée en Backward. Dans ce cas  $s(n-M)$  est prédit à partir de la séquence :  $s(n) s(n-1) \dots s(n-M+1)$  :

$$\hat{s}(n-M) = \sum_{i=1}^M b_i s(n-i+1) \quad (1.10)$$

De la même façon que (1.2), l'erreur de prédiction Backward est définie comme suit :

$$\beta(n) = s(n-M) - \sum_{i=1}^M b_i s(n-i+1) \quad (1.11)$$

La minimisation de cette erreur par rapport aux coefficients de prédiction donne le même ensemble d'équations (1.5).

Le filtre en treillis est une forme de représentation très utile dans le traitement numérique de parole, cette forme d'implémentation (Algorithme équivalent à l'algorithme de Levinson-Durbin) est basée sur les coefficients de réflexion, aussi nommés coefficients PARCOR (corrélation partielle) Figure 1.10, [2].





## 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 tout pôles reste stable après la quantification de ces paramètres. La quantification directe des coefficients LPC n'est pas conseillée parce que les petites erreurs de quantification dans ces coefficients peuvent produire des erreurs spectrales relativement grandes, et peuvent causer aussi une instabilité du filtre  $H(z)$ . Par conséquent, c'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/coefficient (c.-à-d., 60 bits/trame) pour une quantification scalaire des coefficients LPC dans la base de données FM, 25.5% des filtres sont instables, et la distorsion moyenne spectrale est de 1.83 dB [6].

### 1.5.1. Les Coefficients de Réflexions

Les coefficients de réflexions (RCs) peuvent être obtenus des coefficients LPC par l'utilisation de l'algorithme de Levinson -Durbin. Ces coefficients ont deux avantages majeurs sur les coefficients LPC :

- i. ils sont moins sensibles spectralement à la quantification, et
- ii. 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)

Bien que les coefficients de réflexion sont spectralement moins sensibles à la distorsion de quantification que Les coefficients LPC, la distribution statistique de ces coefficients ne ressemble pas à une distribution uniforme. Cependant, cet inconvénient peut être vaincu par l'usage d'une transformation non-linéaire appropriée qui étend la région proche de  $k_i = 1$ , où  $k_i$  est le  $i^{\text{ème}}$  coefficient de réflexion :

- 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 rapport avec les fonctions des surfaces du tube vocale.

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

(1.21)

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

$$J_i = \sin^{-1} k_i \quad (1.22)$$



$$\left\{ \begin{array}{l} P(Z) = (1 + Z^{-1})P'(Z) \\ \quad = (1 + Z^{-1}) \prod_{i=2,4,\dots,M} (1 - 2Z^{-1} \cos \theta_i + Z^{-2}) \\ \\ Q(Z) = (1 - Z^{-1})Q'(Z) \\ \quad = (1 - Z^{-1}) \prod_{i=2,4,\dots,M} (1 - 2Z^{-1} \cos \theta_i + Z^{-2}) \end{array} \right. \quad (1.27)$$

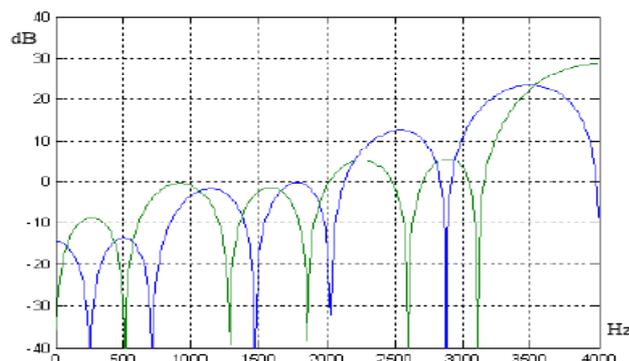
Les paramètres  $\theta_i$  sont rangés en ordre décroissant :

$$0 < \theta_1 < \theta_2 < \dots < \theta_{M-1} < \theta_M < \pi$$

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

$$\begin{aligned} |H(\theta)|^2 &= \frac{1}{|A(\theta)|^2} \\ &= 2^2 |P(\theta) + Q(\theta)|^{-2} \\ &= 2^{-M} \left[ \cos^2 \frac{\theta}{2} \prod_{i=1,3,\dots,M-1} (\cos \theta - \cos \theta_i)^2 + \sin^2 \frac{\theta}{2} \prod_{i=2,4,\dots,M} (\cos \theta - \cos \theta_i)^2 \right]^{-1} \end{aligned} \quad (1.28)$$

Le premier terme à l'intérieur des parenthèses dans l'équation (1.28) se rapproche de 0 quand  $\theta$  ou un des  $\theta_i$  ( $i = 1, 3, \dots, M-1$ ) se rapproche de  $\pi$ . Le deuxième terme se rapproche de 0 quand  $\theta$  ou un des  $\theta_i$  ( $i = 2, 4, \dots, M$ ) se rapproche de 0. Par conséquent, lorsque deux paramètres LSP  $\theta_i$  et  $\theta_j$  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 LSF [3]. Le spectre d'amplitude des filtres  $P(z)$  et  $Q(z)$  est montré sur la Figure 1.11.



**Fig.1.11.** Le spectre d'amplitude de  $P(z)$  et  $Q(z)$  avec  $M=10$ .

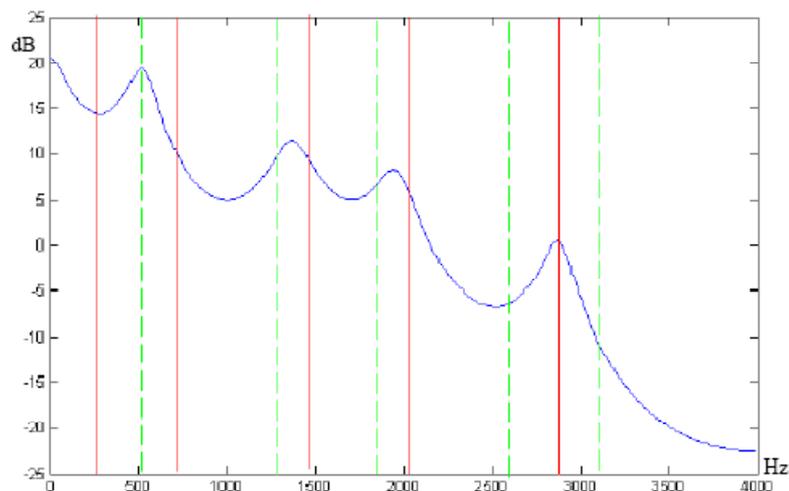
S'il y'a une erreur causée par la quantification d'un paramètres LSF, l'erreur est localisée.

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

### • Algorithme de Calcul

Les paramètres LSF peuvent être trouvés par la procédure suivante [3] :

1. Trouver les coefficients de prédiction linéaires LPC.
2. Former  $P(z)$  et  $Q(z)$ .
3. Estimer la réponse en amplitude de  $P(z)$  et  $Q(z)$ . FFT ou DFT peut être utilisée pour ceci.
4. Les fréquences dans lesquelles les minima locaux se produisent sont les paramètres LSF.



**Fig.1.12.** Spectre du filtre LPC et les positions des paramètres LSF correspondants,

pour une trame de 30 ms de la voyelle /uh/. Les lignes continues représentent

Les zéros de  $P'(z)$ , les lignes en points représentent les zéros de  $Q'(z)$

La propriété  $0 < \theta_1 < \theta_2 < \dots < \theta_{M-1} < \theta_M < \pi$ , permet de réduire considérablement la complexité de recherche des paramètres LSF.

### 1.6. Conversion LSP-LPC



### 1.7.1. Mesure d'Aplatissement du Spectre

Une des méthodes pour décider si la trame de parole est voisée ou non est la mesure d'aplatissement du spectre qui fait usage de la propriété que le spectre de bruit pur est supposé être plat. En d'autres termes, le spectre du segment non voisé est plat et le spectre du segment voisé est moins plat, cette mesure (SFM) est donné par:

$$SFM = \frac{\left( \prod_{k=0}^{N-1} S_j(k) \right)^{\frac{1}{N}}}{\frac{1}{N} \prod_{k=0}^{N-1} S_j(k)} ; S(k) \text{ est TF de la séquence } s(n) \quad (1.35)$$

Cette mesure varie de 0.9 pour un bruit blanc à 0.1 pour un signal sonore. Le seuil est habituellement choisi entre 0.35 à 0.48 [2].

### 1.7.2. Énergie et Taux de Passage par Zéro

L'énergie de la  $j^{\text{ème}}$  trame du signal parole est  $E = \sum_{n=0}^{N-1} s_j^2(n)$  où  $s_j(n)$  est 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.

## 1.8. 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 parole. Il y'a plusieurs algorithmes de détection de pitch, qui ont leurs propres avantages et inconvénients. En général, quatre classes de méthodes sont utilisées communément. Elles sont les méthodes du cepstre, méthodes de corrélation, méthodes temporelles et méthodes fréquentielles [2].

## Conclusion

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

## Techniques de codage de la parole

### Introduction

Dans ce chapitre nous allons présenter brièvement les notions de base de la théorie de l'information relatives au codage source et les techniques du codage de la parole. 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.

### 2.1. Mesure de L'information et Entropie de la Source

Le rôle du codeur source est d'effectuer la compression ou la réduction du débit binaire en enlevant la redondance. Le codage de la source nécessite une description quantitative de la source discrète et en particulier, du contenu et du débit de l'information à transmettre. Ces renseignements sont en général, fournis respectivement par l'entropie de la source et par le débit d'entropie. Considérons une source discrète possédant un alphabet  $Q$  et  $M$  symboles différents  $\alpha_j$  ( $j = 1, 2, \dots, M$ ) de probabilité d'apparition  $p(\alpha_j)$ . Cette probabilité est appelée probabilité a-priori afin de la distinguer de la probabilité a-posteriori qui correspond à la probabilité que le message reçu au récepteur soit correct. On a donc :

$$p(\alpha_j) \geq 0 \quad j = 1, 2, \dots, M \quad \text{et} \quad \sum_{j=1}^M p(\alpha_j) = 1 \quad (2.1)$$

Plus la probabilité d'apparition d'un symbole est faible, moins on s'attend à son apparition. Ce symbole contient donc plus d'informations que lorsqu'il s'agit d'un symbole couramment émis. Cet argument ainsi que d'autres du même type conduisent à la définition usuelle d'auto-information d'une source discrète, valant pour chaque symbole  $\alpha_j$  :

$$I(\alpha_j) = \log\left(\frac{1}{p(\alpha_j)}\right) \quad (2.2)$$

Ce résultat correspond à l'information transmise sous l'hypothèse qu'à la sortie de la source, le symbole  $\alpha_j$  soit fixé dans le temps. L'unité de cette information dépend de la base; on quantifie donc l'information en bits [8].





Le nombre total de bits  $R$  nécessaire pour coder  $N$  scalaires est :

$$R = \sum_{i=1}^N R_i \quad (2.9)$$

À partir de ces équations, on peut calculer le nombre total de niveaux de reconstruction :

$$L = \prod_{i=1}^N L_i = 2^R \quad (2.10)$$

Il est important de noter, à partir des équations, que le nombre total de bits  $R$  est la somme des  $R_i$ , tandis que le nombre de niveaux de reconstruction  $L$  est le produit des  $L_i$ .

Si nous avons un nombre de bits fixe pour coder toutes les valeurs  $N$  en utilisant la quantification scalaire, l'optimisation du nombre de bits alloués à chaque valeur dépend du critère d'erreur utilisé et de la densité de probabilité des scalaires à coder. Une optimisation typique consiste à allouer plus de bits aux scalaires qui ont une grande variance, et un nombre plus petit de bits pour ceux qui ont une petite variance.

### 2.3. Codage de Formes d'ondes

Le codage de la forme d'onde consiste à reproduire le signal vocal par une modélisation numérique de la forme d'onde. Ces techniques peuvent accomplir la haute qualité avec une basse complexité et un bas délai. Cependant, le taux de bits nécessaire dans le codage de forme d'onde est aussi haut.

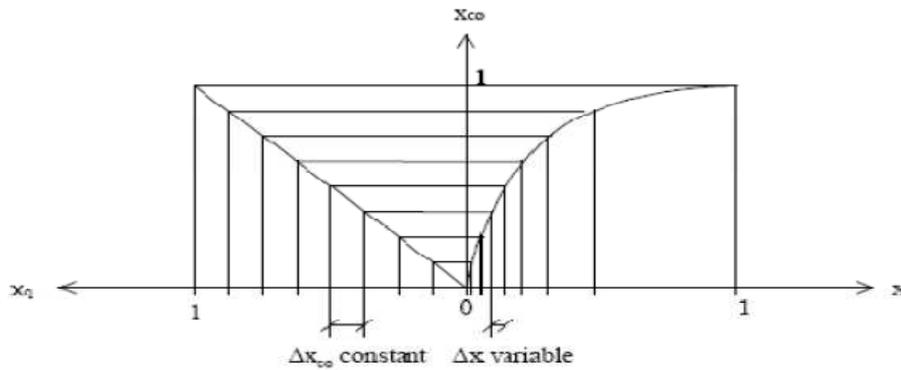
#### 2.3.1. Le Codage PCM

Dans ce type de codage, chaque valeur codée et transmise correspond à la valeur instantanée de l'échantillon de parole, chaque échantillon est quantifié à un nombre fini de niveaux de reconstruction, et à chaque niveau est assigné une séquence unique d'éléments binaires, c'est cette séquence qui est transmise au récepteur [4].

La forme d'onde de parole est prélevée et alors logarithmiquement quantifiée à des niveaux numériques. Il y'a deux méthodes de PCM, qui produisent des débits binaires de 64kb/s (selon la loi  $A$  et selon la loi  $u$ ). Dans ces méthodes les pas de quantification les plus larges sont utilisés pour les échantillons de grandes amplitudes et les pas de quantifications les plus petits sont utilisés pour les échantillons de petites amplitudes. Avec ce type de codage, les échantillons de 13 bits peuvent être comprimés à 8 bits tout en gardant la qualité originale de la parole. La loi  $A$  est le standard d'Europe ( $A=87.56$ ) tandis que les Etats-Unis utilise la loi  $u$  ( $u=255$ ).

Ces méthodes sont très populaires en raison de leur basse complexité et leurs délais de codage, la Figure 2.1 illustre ce principe de compression avec :

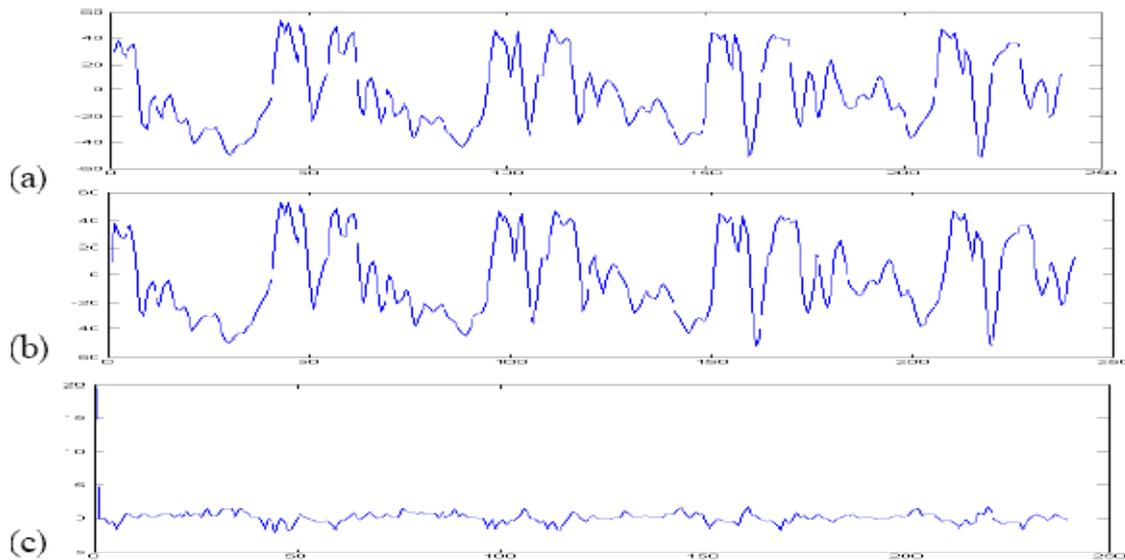
- $x$ : signal original
- $x_{co}$ : signal compressé.
- $x_q$ : signal compressé échantillonné



Loi A (europe)	Loi $\mu$ (USA, Japon)
$x_{co} = \begin{cases} \frac{A}{1 + \ln(A)} \cdot x & \text{si } x \leq \frac{1}{A} \\ \frac{1 + \ln(A \cdot x)}{1 + \ln(A)} & \text{si } x > \frac{1}{A} \end{cases}$	$x_{co} = \frac{\ln(1 + \mu \cdot x)}{\ln(1 + \mu)}$
<p>A = 87.6, donc :</p> $x_{co} = \begin{cases} 16 \cdot x & \text{si } x \leq 1.14 \cdot 10^{-2} \\ \frac{1 + \ln(87.6 \cdot x)}{5.473} & \text{si } x > 1.14 \cdot 10^{-2} \end{cases}$	<p><math>\mu = 255</math>, donc :</p> $x_{co} = \frac{\ln(1 + 255 \cdot x)}{5.545}$

Fig.2.1. Lois de compressions

La Figure 2.2 représente un exemple de quantification avec la loi  $u$  :



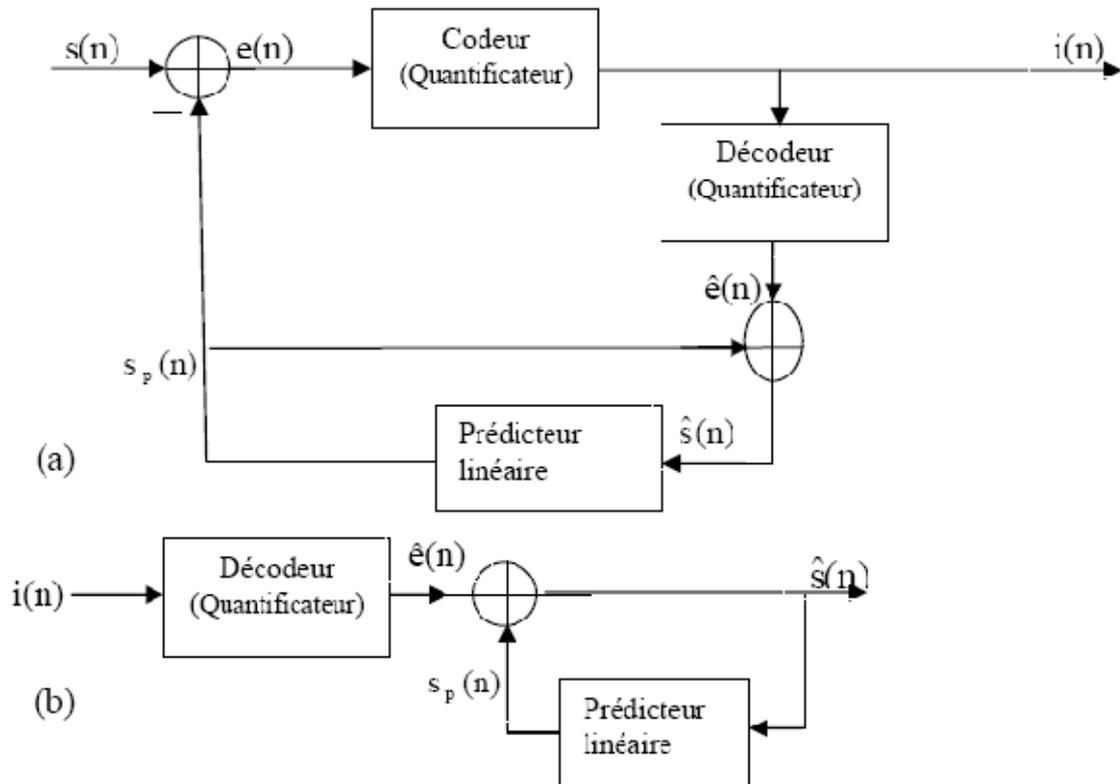
**Fig.2.2** Exemple de quantification (loi  $u$ )

(a) signal original (b) signal quantifié (c) erreur de quantification

### 2.3.2. Le Codage Différentiel DPCM, DM et ADM

Il y'a une forte corrélation entre les échantillons adjacents dans le signal parole. Par conséquent, il est plus rentable de coder non pas les échantillons eux même mais la différence entre des échantillons successifs. Cette technique est connue sous le nom de DPCM. Le signal transmis aura une gamme dynamique beaucoup plus inférieure que le signal original, donc il peut être efficacement quantifié en utilisant un quantificateur avec moins de niveaux de reconstruction. Les échantillons précédents sont employés pour prédire la valeur de l'échantillon actuel [4].

La technique est montrée sur la Figure 2.3, où l'erreur de prédiction  $e(n)$  — obtenue en soustrayant le signal d'entrée  $s(n)$  du signal prédit  $s_p(n)$  — est quantifiée. Les indices à la sortie du quantificateur du codeur représentent le segment de bits de la DPCM. L'entrée quantifiée est obtenue en rajoutant la prédiction de l'entrée  $s_p(n)$  à la différence quantifiée  $\hat{e}(n)$ . La prédiction de l'entrée  $s_p(n)$  est obtenue à partir de  $m$  valeurs quantifiées précédentes de l'entrée. Le décodeur reconstruit l'entrée quantifiée  $\hat{s}(n)$  en rajoutant la prédiction de l'entrée  $s_p(n)$  à la différence quantifiée  $\hat{e}(n)$ . La prédiction  $s_p(n)$  est obtenue à partir de  $m$  valeurs quantifiées ; déjà décodées.



**Fig.2.3** Un système DPCM : (a) Codeur et (b) Décodeur

La modulation delta est une sous-classe de la DPCM dans laquelle l'erreur de prédiction est codée avec un seul bit. La modulation delta opère avec un taux d'échantillonnage plus élevé que celui utilisé de la DPCM. Le pas de quantification peut être adaptatif (ADM). Les codeurs DM et DPCM sont de bas -à- moyenne complexité.

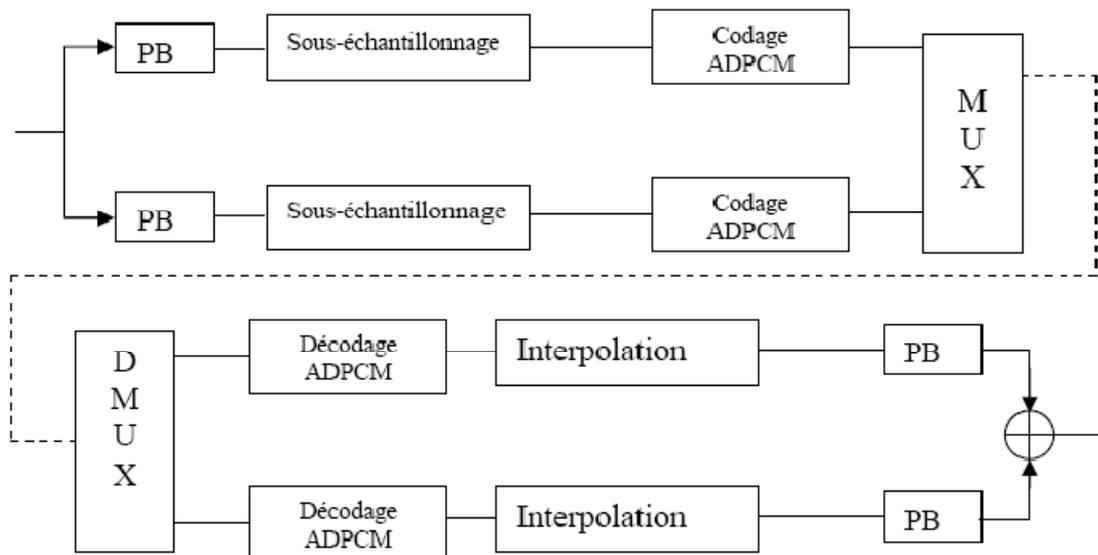
Il est possible de réduire le débit binaire en préservant la qualité du message en combinant le principe de la modulation différentielle avec une adaptation dynamique du pas de quantification. Ainsi, l'UIT-T a défini un procédé de modulation par impulsion et codage différentiel adaptatif ADPCM qui fait l'objet de la recommandation G 721.

## 2.4. Le Codage Fréquentiel

Les codeurs fréquentiels de formes d'ondes divisent le signal en plusieurs composantes séparées de fréquence et les codent indépendamment, le nombre de bits utilisé pour coder chaque composante de fréquence peut varier dynamiquement.

### 2.4.1. Le Codage en Sous-Bande

C'est le plus simple des techniques du codage fréquentiel. Le signal dans un codeur en sous bandes passe à travers un banque de filtres passe bandes, comme montrée sur la Figure 2.4



**Fig.2.4.** Principe de codage en sous bande (PB= filtre passe-bande)

La sortie de chaque filtre est échantillonnée, puis une des techniques du domaine temporel décrites au-dessus est utilisée pour coder chaque sous bande, chaque bande a un nombre de bits assigné, lequel peut être varié d'après l'importance perceptuelle de la bande. Le procédé inverse reproduit le signal original [4].

L'avantage le plus important du codage en sous bande est que le bruit de quantification produit dans une bande est restreint à cette bande. Ceci empêche le bruit de quantification de masquer les composants de fréquence dans d'autres bandes.

## 2.4.2. Codage Par Transformée

### A : Principes

L'idée est de transformer le signal original dans un autre domaine, afin de décorréler ce dernier, la partie majeure d'information (énergie) dans le nouveau domaine est localisée dans un nombre réduit de coefficients transformés à l'aide d'une transformation réversible.

Dans le codage par transformée (TC), des blocs de N échantillons d'entrée sont transformés en N coefficients, qui sont alors quantifiés et transmis, Figure 2.5.











































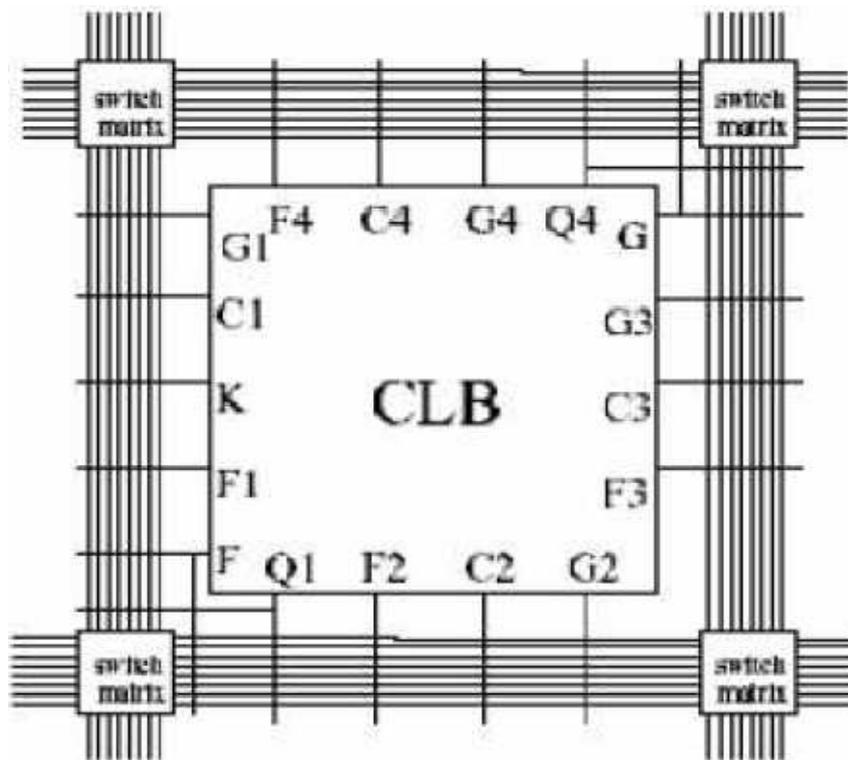


Figure 4.6 Les longues lignes

- Performances des interconnexions

Les performances des interconnexions dépendent du type de connexions utilisées. Pour les interconnexions à usage général, les délais générés dépendent du nombre de segments et de la quantité d'aiguilleurs employés. Le délai de propagation de signaux utilisant les connexions directes est minimum pour une connectique de bloc à bloc. Quant aux segments utilisés pour les longues lignes, ils possèdent une faible résistance mais une capacité importante. De plus, si on utilise un aiguilleur, sa résistance s'ajoute à celle existante.



- Lors de l'étape de simulation, on valide l'application, indépendamment de l'architecture et des temps de propagation du futur circuit cible.
- L'étape de routage génère les informations permettant d'intégrer l'application dans le circuit choisi. Une « métro annotation » est effectuée dans la « netlist » : les temps de propagation du composant cible y sont pris en compte.
- Lors de la simulation temporelle, on peut évaluer les performances de l'application générée.

#### 4.1. La structure d'une description VHDL

Une description VHDL est composée de deux parties indissociables à savoir :

- L'entité (ENTITY), elle définit les entrées et sorties.
- L'architecture (ARCHITECTURE), elle contient les instructions VHDL permettant de réaliser le fonctionnement attendu ; donc on y trouve des descriptions procédurales, fonctionnelles et structurelles. Plusieurs architectures peuvent être associées à une seule entité.

Nous illustrerons cette structure dans un exemple de programmation d'un décodeur (1 parmi 4) comme suit :







































































## BIBLIOGRAPHIE

- [1] B. Atal. "A model of LPC excitation in terms of eigenvectors of the autocorrelation of the impulse response of the LPC filter". ICASSP, Vol. 1. Page(s) 45-48, 1989.
- [2] X. Huang, A. Acero, H. Hon, "Spoken language processing a guide to theory, algorithm and system design", Prentice Hall 2001.
- [3] S.Furui " Digital Speech processing, Synthesis, Recognition", Second edition, Marcel Dekker 2001.
- [4] W.C. Chu, Speech Coding Algorithms, Foundation and Evolution of Standardized Coders, John Wiley & Sons, Hoboken, 2003.
- [5] J. Makhoul, "Linear prediction: A Tutorial Review", Proceedings of the IEEE, vol. 634 pp 561-578.
- [6] 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.
- [7] Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear prediction (CS-ACELP). International Telecommunication Union (ITU). Recommendation UIT-T G729, March 1996.
- [8] B.Chaoub, L.Chontam "Compensation des systèmes standards de compression /décompression d'images".81p. Mémoire de maîtrise ès sciences appliquées. QUBEC : Université de Sherbrooke, Septembre 1995.

- [9] R. M. Gray, "Vector quantization," IEEE ASSP Mag., vol. 1, pp. 4-29, Apr. 1984.
- [10] F. Merazka , D. Berkani, " Vector Quantization Of LPC Parameters By Split", pp IEEE.1998. 434-437.
- [11] Baudoin et al. "Codage de la parole à bas et très bas débit". Annales des Télécommunications, 2000 :p 1-19.
- [12] A.S. Spanias, "Speech Coding : A Tutorial Review", Proc.IEEE, 82 (1994),pp. 1541–1582.
- [13] J.C. Nash "Compact numerical methods for computers: linear algebra and function minimization ", Second Edition Adam Hilger 1990.
- [14] P Corney, JS Mason. "Singular value decomposition and its modelling of speech excitation" IEE Conference, Loughborough, 1991. Page: 305-308
- [15] D.Berkani."Application de la transformée orthogonale SVD en compression de la parole", Séminaire National sur l'Automatique et les signaux SNAS, Annaba 1999.

