REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT D'ELECTRONIQUE
Laboratoire Signal & Communications

PROJET DE FIN D'ETUDES

EN VUE DE L'OBTENTION DU DIPLOME D'INGENIEUR D'ETAT EN ELECTRONIQUE

THEME :

Implémentation d'un codeur de parole CELP sur MATLAB

Proposé et dirigé par :

Pr. D. BERKANI

Réalisé par:

BOUGUERRA Mohamed Samir

CHOUGRANI Houcine

PROMOTION 2011

E.N.P, 10, Avenue Hassan BADI, EL HARRACH, ALGER

ملخص:

في هذا العمل، قمنا ببرمجة المشفر **CELP** على **MATLAB**، بدأنا بدراسة طريقة إنتاج الكلام، و خصائصها مع وصف أهم مشفرا ت الكلام، وهذا بدراسة طريقة التعرف الخطي الذي يعتبر عنصر هام لخوارزمية **CELP**، بعد ذلك اهتممنا بدراسة كل عناصره و بالخصوص طريقة التحليل بالتركيب و طريقة **ZI-ZS**، و أخيرا قمنا بتركيب هاته العناصر و تمثيل المشفر **FS1016** و تقييم خصائصه

كلمات مفتاحية:

التعرف الخطي، **CELP**، التحليل بالتركيب، **PITCH**، **FS1016**، قاموس الرامزات، مصفي الخطأ الملموس.

Résumé :

Dans ce travail, nous avons implémenté le codeur CELP sur MATLAB, nous avons commencé par étudié la modélisation de la production de la parole ainsi que les propriétés de celle-ci en passant par la description des principaux codeurs de la parole, nous avons aussi étudié la prédiction linéaire qui est un élément clé pour l'algorithme CELP ensuit nous nous somme focalisé sur celui-ci et ses différents blocs et méthodes notamment l'analyse par synthèse et la méthode du Zero-State Zero-Input et enfin nous avons réalisé l'implémentation de ces blocs et la simulation du standard FS1016 ainsi que l'évaluation de ses performances.

Mots clés: Prédiction linéaire, CELP, analyse par synthèse, pitch, FS1016, codebook, filtre d'erreur perceptive.

Abstract :

In this work, we implemented the CELP coder on MATLAB, we started by studying the speech production modeling and its properties with describing principal speech coders, we also studied the linear prediction which is a principal element of the CELP algorithm then we study this algorithm and its different blocs and methods and specially the analysis by synthesis and the Zero-State Zero-Input method and finally we realized the implementation of these blocs and the simulation of the FS1016 standard and evaluate its performances.

Key words: Linear prediction, CELP, analysis by synthesis, pitch, FS1016, codebook, error perceptive filter.

Remerciements

Nous remercions Allah le clément le miséricordieux pour la force qu'il nous a donnée pour venir à bout de ce travail.

Nous remercions notre promoteur le professeur D. BERKANI pour les précieux conseils qu'il nous a prodigués, pour sa disponibilité, ses critiques qui ont été cruciales pour l'aboutissement de ce mémoire, pour sa patience et sa gentillesse.

Nous remercions Mr Z. TERRA d'avoir accepté d'évaluer notre modeste travail et d'avoir accepté de présider le jury. Nos remerciements vont aussi à Mr .B. BOUSSEKSSOU qui a accepté d'examiner ce mémoire.

Nos remerciements les plus sincères vont à tout le corps enseignant du département de l'électronique de l'école nationale polytechnique pour le savoir et les connaissances qu'ils nous ont transmis durant ces trois dernières années.

Dédicaces

A la mémoire de mon grand père

A mes chers parents qui sans leurs efforts et leurs sacrifices je ne serai surement pas là où je suis aujourd'hui, leurs conseils ont été, sont et seront toujours d'une importance capitale dans ma vie.

A ma grand-mère.

A mes frères et à ma sœur.

A toute ma famille.

A tous mes amis sans exception.

A tous pour qui je compte et à tous ceux qui comptent pour moi.

HOUCINE

A la mémoire de ma mère.

A mon père qui sans ses préceptes, ses sacrifices et sa présence perpétuelle à mes côtés je ne serai surement pas là où je suis.

A tata Sorya qui sans ses sacrifices, ses encouragements et son dévouement je ne serai surement pas là où je suis.

A ma sœur et à mon frère.

A toute ma famille.

A tous mes amis.

SAMIR

Table des matières

Liste des figures

Chapitre I :

Chapitre II :

Chapitre III :

Chapitre IV :

Liste des figures

Liste des tableaux

Chapitre II :

Chapitre III :

Chapitre IV :

Introduction Générale

Dans le domaine des communications, l'intelligibilité de la parole et sa qualité sont deux facteurs substantiels pour la satisfaction des consommateurs, la réduction du débit en est aussi un pour la satisfaction des concepteurs des systèmes de communications, ces derniers se sont toujours trouvés face à un dilemme qui semble toujours jouer un rôle primordiale lors de l'élaboration d'algorithmes de codage de la parole : Concevoir un algorithme pour un débit réduit tout en gardant une bonne intelligibilité de la parole.

L'avancement des technologies de l'électronique et de la micro-électronique ont permis de concevoir des systèmes autours de microprocesseurs qui possèdent de hautes performances de calcul, cet essor a profité aux spécialistes du traitement du signal et de la parole pour concevoir des algorithmes de calcul puissants donnant accès à la conception de codeurs satisfaisant le compromis qualité de la parole et débit du codeur.

L'émergence du traitement du signal numérique était la genèse des techniques de codage de la parole, la micro-électronique en était le précurseur, en effet, grâce aux progrès dans ces deux domaines, les techniques de codage de la parole ont connu un développement spectaculaire dans les quatre dernières décennies donnant naissance à un algorithme à la fois subtile et efficace, c'est le « Code Exited Linear Prediction » (CELP).

L'algorithme CELP a été introduit par Shroeder et Atal en 1985, considéré par beaucoup de spécialistes du traitement de la parole comme une prouesse dans le domaine, un nombre considérable de codeurs et d'algorithmes ont été conçu autour de cet algorithme, ces derniers ont pris une place importante dans notre vie quotidienne, en allant de la téléphonie mobile à la voix

sur IP en passant par les systèmes de reconnaissances et de sécurité pour des applications militaires…, dans ces dispositifs et dans bien d'autres, l'algorithme CELP et les algorithmes qui en sont dérivés sont de nos jours des pièces primordiales.

Le but de ce mémoire est l'étude de l'algorithme CELP et l'implémentation de ses principaux modules sur MATLAB, dans le premier chapitre nous ferons une introduction au domaine du traitement de la parole par une description physiologique de la génération de la voix puis une étude des caractéristiques d'un signal vocal en vue d'en tirer profit pour la conception de systèmes de communication électronique, ce qui nous permettra de décrire quelque codeurs du signal de parole. Dans le deuxième chapitre, nous étudierons en détail le concept de la prédiction linéaire qui est un fondement pour tous les codeurs de la parole modernes. Le troisième chapitre a été entièrement dédié à l'étude de l'algorithme CELP et ses compartiments, il comprendra aussi une description brève du standard CELP FS1016, l'implémentation des principaux module de l'algorithme a été décrite au chapitre quatre, ainsi que ses résultat et l'évaluation de la qualité d'un signal de parole synthétisé par le standard FS1016, à la fin nous présentons une conclusion générale, dans laquelle, nous donnerons un résumé sur tout le travail effectué dans ce mémoire.

Chapitre I

Généralités sur la parole et son codage.

I.1.Introduction :

La parole est l'aptitude à communiquer la pensée par des sons articulés, c'est le moyen de communication adopté par les humains. La production de ces sons articulés est assurée par les fluctuations de la pression de l'air engendrée par le système phonatoire, elles constituent le signal vocal et elles peuvent être détectées par l'oreille.

Le message vocal se distingue par rapport au message écrit par les différentes intonations qu'il faut prendre en considération lors de l'analyse et du traitement de la parole, le signal vocal est caractérisé par sa redondance, ce qui représente un avantage pour la compression de l'information. [1]

Le fonctionnement du système phonatoire humain est assez intelligible, ce qui a permis aux spécialistes du traitement de la parole de concevoir une modélisation pertinente de ce dernier afin de recréer un système électronique qui aurait les capacités de générer des sons plus au moins analogues à un système phonatoire humain.

Dans ce chapitre nous allons introduire les principes de fonctionnement du système phonatoire ainsi que sa modélisation, et quelques notions sur les systèmes de codage de la parole.

I.2.Caractéristiques de la production de la voix :

I.2.1. La production de la voix :

La parole est engendrée par une variation de pression de l'air causée par le système phonatoire humain, qui est bien entendu contrôlé par le système nerveux central, les poumons sont considérés comme la source d'énergie substantielle pour la production de la parole, c'est

eux qui sont responsable de la génération de l'air, ils le poussent dans la trachée artérielle qui est limitée à son extrémité par le larynx, ce dernier a pour rôle de moduler la pression de l'air, qui sera en suite appliquée à la cavité buccal et nasal qui constituent le conduit vocal.[2]

Les différentes intonations et sons sont émis avec des fréquences fondamentales (pitch) différentes, et c'est la glotte qui en est responsable, la différences entre les sons réside aussi dans les différentes formes que peut prendre le conduit vocal et plus exactement celles de la cavité buccal.[1]

Pour produire quelque sons tel que les (n,m..) la cavité nasal joue un rôle important mais dans les modélisations elle est généralement incluse avec la cavité buccal, la figure 1.1 montre un schéma descriptif du système phonatoire.[1]

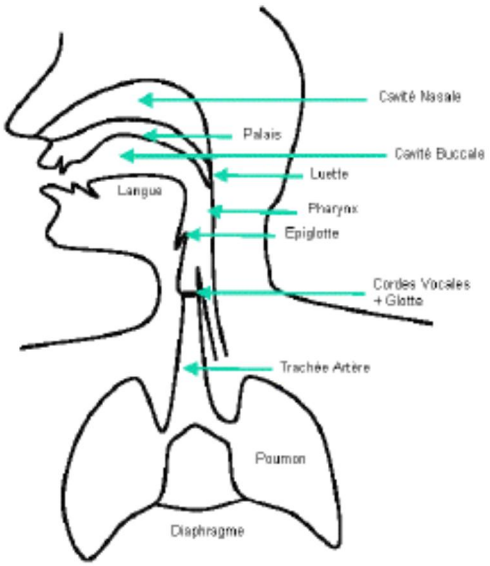


Figure 1.1 : Système phonatoire.

I.2.2. Sons voisés et non voisées :

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 , 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 pour décrire

un spectre vocal, les formants d'ordres supérieurs ont des applications diverses telles que la reconnaissance de voix.[1]

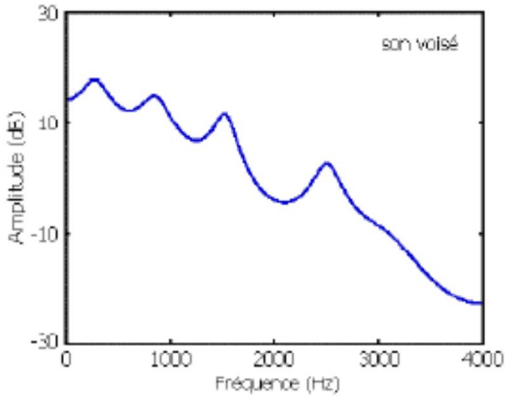


Figure 1.2 : Le spectre d'un son voisé.

Le resserrement du conduit vocal entraine 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 apériodiques, 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.[1]

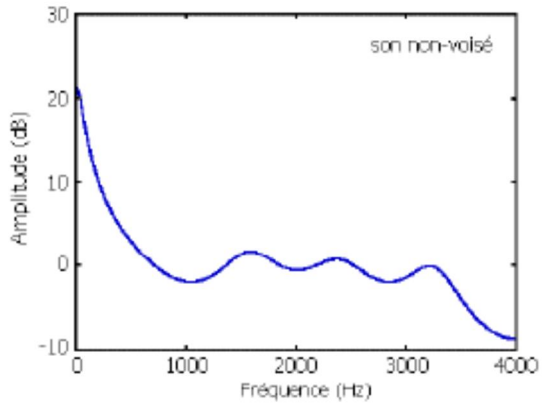


Figure 1.3 : Le spectre d'un son non voisé.

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.

I.2.3. Perception de la parole :

Lors de la transmission d'un signal de parole, la compression de celui-ci entraine 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 $200\ et\ 3700\ Hz$ 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.

I.3.Modélisation du processus de la production de la voix :

Pour définir un algorithme efficient du codage de la parole, il est nécessaire de trouver des paramètres pertinents qui caractérisent un signal de parole ensuite les utiliser pour réduire le débit sans pour autant altérer la qualité du signal. Un modèle simplifié du système phonatoire ne comprenant que ces paramètres significatifs est inéluctable pour la compression. Le système de production de la parole est souvent modélisé par un filtre qui représente le conduit vocal, ce filtre est excité par une source d'énergie qui génère soit des impulsions périodiques pour décrire un son voisé, soit un bruit blanc qui modélise un son non voisé. [1][3]

Une production d'un son compréhensible et naturel, nous astreint à faire une modélisation précise du filtre et des excitations, la figure 1.5 montre un modèle à 2 états qui est évidement le modèle le plus simple.[4]

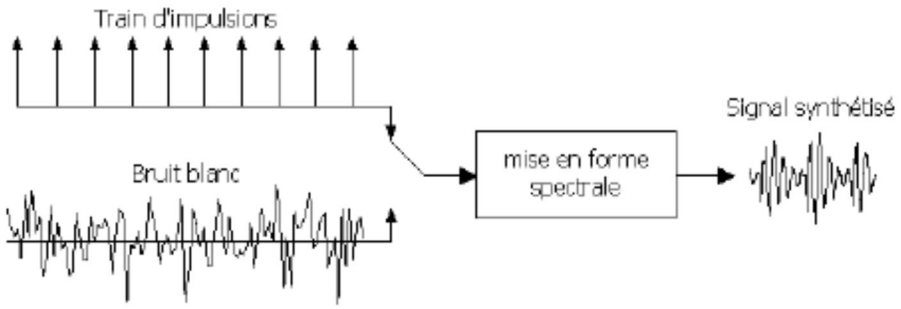


Figure 1.4 : Modèle de synthèse de parole à 2 états.

I.4.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 insatiable 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 effectué afin de maximiser le compromis entre l'efficacité, le cout 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 quelque notion sur les quantifications scalaire et vectorielle, qui jouent un rôle rudimentaire dans les systèmes de communication numérique.

I.4.1.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$Q$, d'un point de vue mathématique, établit une application subjective entre un nombre réel $x \in R$ et un nombre $y_i \in Y$, ou Y est ensemble fini à N éléments :

$$Q: R \to Y$$

$$(y_1, y_2, \dots\dots, y_N) \in Y$$

$$\text{Donc } Q(x) = y_i \; ; x \in R \; ; i = 1, \dots\dots, N$$

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 role d'un codeur et D qui joue celui d'un décodeur[5] :

$$E: R \to I$$

$$D: I \to R$$

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

De plus :

$$\hat{x} = Q(x) = D\big(E(x)\big) = y_i \tag{1.1}$$

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 :

$$x - \hat{x} = x - Q(x) = e \tag{1.2}$$

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[2] :

$$D = \sum_{i=1}^{N} \int_{q_{i-1}}^{q_i} d\big(x, Q(x)\big) p_X(x) dx \tag{1.3}$$

Où :

x est la valeur à quantifier,

$Q(x)$ est la valeur quantifiée,

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

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

N est le nombre de niveaux du quantificateur.

q_i les bornes de quantification.

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

$$r = log_2 N \tag{1.4}$$

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

$$SNR = 10log_{10}\frac{\sigma_x^2}{D} \tag{1.5}$$

Ou $ó_x^2$ est la variance de x.

I.4.2. 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.

I.4.2.1. Principes :

Mathématiquement, un quantificateur vectoriel Q est application qui assigne à chaque vecteur **x** appartenant à un espace euclidien à M dimensions R^M, un vecteur à M dimensions $\mathbf{y_i}$ appartenant à un ensemble de N vecteurs de M dimensions.

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

Avec

$\mathbf{x}=[x_1, x_2, \ldots., x_M]^T$

$(y_1, y_2, \ldots, y_n) \in Y$

$y_i = [y_{i1}, y_{i2}, \ldots, y_{iM}]^T$; $i = 1,2, \ldots N.$

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

$$Q(\mathbf{x}) = \mathbf{y_i}\ ;\ i = 1,2, \ldots, N \tag{1.6}$$

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

$$r = log_2 N \tag{1.7}$$

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 :

$$d\big(\mathbf{x}, Q(\mathbf{x})\big) = \begin{cases} 0 & , \mathrm{Q}(\mathbf{x}) = \mathbf{x} \\ > 0 & , ailleurs \end{cases} \tag{1.8}$$

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

$$d\big(\mathbf{x}, Q(\mathbf{x})\big) = \|\mathbf{x} - Q(\mathbf{x})\|^2 = \sum_{j=1}^{M} (x_\mathbf{j} - \hat{x}_j)^2 \tag{1.9}$$

La distorsion moyenne vaut alors :

$$D = E\{d\big(\mathbf{X}, Q(\mathbf{X})\big)\} = \int_{R^M} d\big(\mathbf{x}, Q(\mathbf{x})\big) f_\mathbf{X}(\mathbf{x}) d\mathbf{x} \tag{1.10}$$

Avec :

X est un vecteur de distribution aléatoire à M dimension

$f_{\mathbf{X}}(\mathbf{x})$ 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 :

$$R_i = \left\{\mathbf{x}: d(\mathbf{x}, \mathbf{y}_i) \leq d\left(\mathbf{x}, \mathbf{y}_j\right); \forall j \in I\right\} ; i \in I$$

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 $\mathbf{y}_i$ par rapport aux autres code vecteurs.

I.4.2.2. 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 $\mathbf{y}_i$ et les régions de décision R_i qui minimise la distorsion moyenne D, pour une densité de probabilité du vecteur aléatoire **X** donnée.

Centroide : On définit le centroide cent(R_0), d'un sous-ensemble non vide $R_0 \in R^M$, par le vecteur $\mathbf{y}_0$(s'il existe), qui minimise la distorsion entre un point $\mathbf{X} \in R_0$ et $\mathbf{y}_0$, moyenné sur la distribution de probabilité de **X** :

$$cent(R_0) = \{\mathbf{y}_0: E\{d(\mathbf{X}, \mathbf{y_0})|\mathbf{X} \in R_0\} \leq E\{\mathbf{X}, \mathbf{y}\}|\mathbf{X} \in R_0\}, \qquad \forall \mathbf{y} \in R^M \qquad (1.11)$$

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

$$\mathbf{y}_i = cent(R_i) \qquad (1.12)$$

Dans le codage hybride de la parole, 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 centroide de chaque régions forme le dictionnaire, ces régions contenant les centroides 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.

Si N est une puissance de 2, un algorithme croissant qui commence avec un dictionnaire de dimension 1 est formé comme suit :

1. Trouver un dictionnaire de dimension 1.
2. Trouver un dictionnaire initial de double dimension en faisant une division binaire de chaque code vecteur. Pour une division binaire, un code vecteur est divis en 2 par des petites perturbations.
3. Appliquer la méthodologie itérative présentée plutôt pour trouver les régions de Voronoi et les codes vecteurs pour obtenir le code optimal.
4. Si le dictionnaire de la dimension désirée est obtenu, arrêter. Autrement, aller à l'étape 2, dans laquelle la dimension du dictionnaire est doublée.

I.4.2.3.Quantification par Split :

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

$\mathbf{x}= [x_1, x_2, \ldots\ldots, x_M]^{\mathbf{T}}$ est divisé en :

$\mathbf{x}_a = [x_1, x_2, \ldots\ldots, x_K]^{\mathbf{T}}$,avec $K < M$ et

$\mathbf{x}_b = [x_{K+1}, x_{K+2}, \ldots\ldots, x_M]^{\mathbf{T}}$

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

$$\|\mathbf{x} - \hat{\mathbf{x}}\|^2 = \|\mathbf{x}_a - \hat{\mathbf{x}}_a\|^2 + \|\mathbf{x}_b - \hat{\mathbf{x}}_b\|^2 \qquad (1.13).$$

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.

I.4.3. 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.

I.4.3.1. 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 μ utilisée par les USA(μ=255), ces deux méthodes sont très populaires à cause de la simplicité de leurs circuits et leur délai de codage.

La loi μ est basée sur une fonction logarithmique non linéaire qui est donnée par :

$$f(x) = A\frac{\ln\left(1+\mu\frac{|x|}{A}\right)}{\ln(1+\mu)}sgn(x) \qquad (1.14)$$

Ou A représente le maximum du signal d'entrée, x le signal à quantifier ($x \leq A$) et μ une constante qui contrôle de degrés de compression [5].

La figure 1.5 montre quelque caractéristiques des la loi μ :

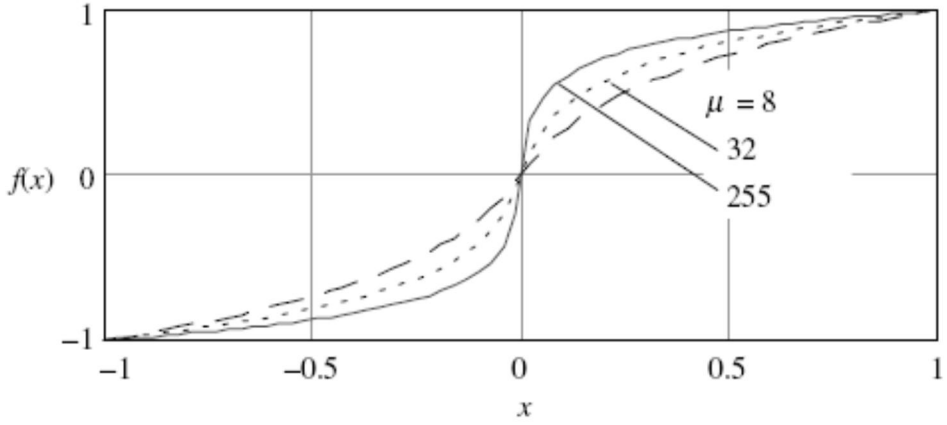


Figure 1.5 : Graphe caractérisant la loi μ avec μ= 255 32 et 8 et A=1.

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

$$f(x) = \begin{cases} \frac{A_0|x|}{1+lnA_0} sgn(x), & |x| \leq \frac{A}{A_0}, \\ \frac{A(1+\ln(A_0|x|/A))}{1+A_0} & \frac{A}{A_0} \leq |x| \leq A, \end{cases} \quad (1.15)$$

Avec A_0 une constante qui contrôle le degrés de compression. La figure 1.6 montre quelque caractéristiques de la loi A

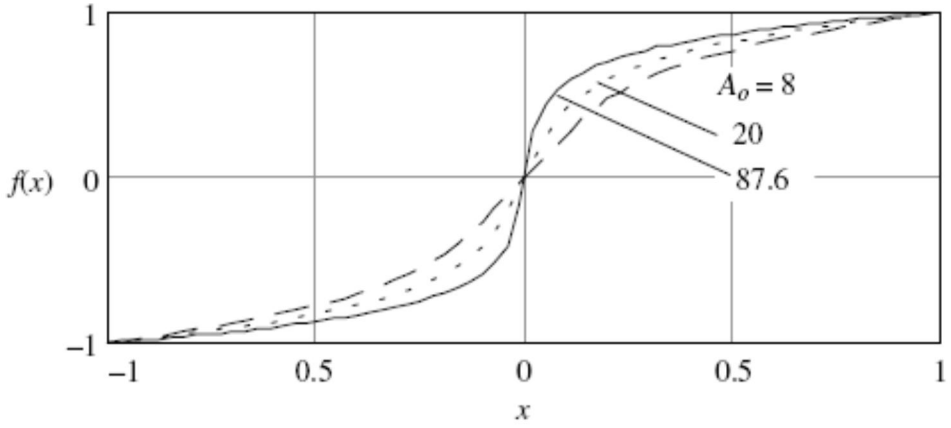


Figure 1.6 : Graphe caractérisant la loi A avec $A_0 = 87{,}6$ 20 et 8 et A=1.

I.4.3.2. 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 soutirer une erreur de prédiction qui serait caractérisé par une faible variance [5].

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. Le principe du codage différentiel est illustré par la figure 1.7, l'erreur de prédiction $e[n]$est obtenue par la soustraction de la valeur prédite $x_p[n]$ depuis la valeur de l'échantillon original $x[n]$, 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 $x_p[n]$.

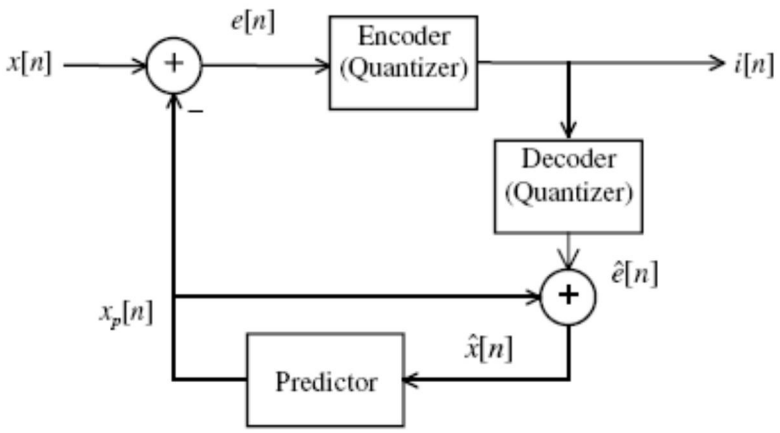


Figure 1.7 : Structure du codeur DPCM.

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 1.8 illustre le fonctionnement du décodeur.

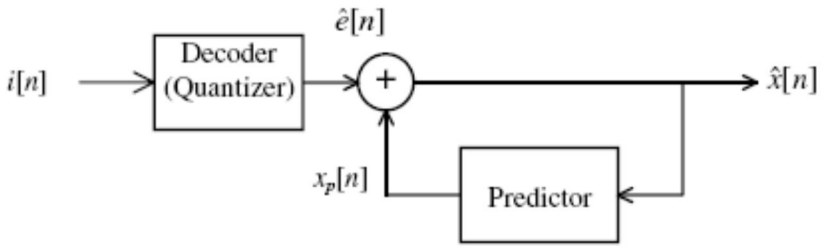


Figure 1.8 : Structure du décodeur DPCM.

La non-stationnarité du signal de parole oblige à utiliser un quantificateur scalaire à pas versatile 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.

I.4.4. 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 comme le DoD ou l'OTAN.

Les codeurs de parole utilisant le codage paramétrique diffèrent par le type de modèle utilisé et par la façon de coder les paramètres, deux types de modèles sont couramment utilisés : le modèle « source-filtre » et les modèles STC (*Sinusoidal Transform Coder*). 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 (Chapitre2) mais on utilise aussi parfois une transformée de Fourier à court terme [2]. 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*), les codeurs à excitation mixte MELP(Mixed Excitation Linear Prediction), les codeurs HSX(*Harmonic Stochastic eXcitation*) et les codeurs à interpolation de forme d'onde prototypes WI*(Waveform Interpolation*) [9].

I.4.5. Codage hybride :

Les codeurs paramétrique décrit ci-dessus permettre 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 malencontreusement 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 délivre un débit moyen allant typiquement de 4 à 16Kb/s en gardant une très bonne qualité du signal.

I.5.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, ces codeurs présentent des débits réduits mais au détriment de la qualité du signal, ils exploitent sa redondance et utilisent généralement la technique de prédiction linéaire qui s'est avérée imparable pour la compression, celle-ci fera l'objet du deuxième chapitre de notre mémoire. 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 est le sujet principal de notre mémoire et qui sera détaillé dans le troisième et quatrième chapitre.

Chapitre II :

Codage par prédiction linéaire

II.1.Introduction :

La prédiction linéaire est devenue une technique fondamentale pour la plupart des algorithmes modernes du codage de la parole, elle exploite la redondance du signal vocal afin de le compresser, cela en tirant profit de la forte corrélation entre ses échantillons voisins, en effet, l'idée de ce type de codage est d'exprimer un échantillon présent par une équation linéaire fonction des échantillons antérieurs.

La modélisation autorégressive du système phonatoire [1] est à l'origine de l'émergence de la prédiction linéaire, en effet, celle-ci exploite cette modélisation qui présente le système phonatoire comme un ensemble de filtres pour tirer des paramètres les représentant, ce sont les coefficients de prédiction.

La prédiction linéaire peut être perçue comme une technique qui élimine l'information redondante qui peut être prédite pour ne laisser que l'information non prédictible qui sera transmise au récepteur, cela requiert donc moins de bits pour la transmission du signal.

Le codage par prédiction linéaire n'est pas une technique à part, il est utilisé dans différents codeurs de parole tels que le LPC (*Linear predictive coding*) qui est bâti essentiellement autour de la prédiction linéaire et le CELP pour lequel cette dernière joue un rôle important pour la compression en plus de la technique d'analyse par synthèse (Chapitre 3).

Dans ce chapitre, nous allons introduire les fondements de la prédiction linéaire ainsi que certains éléments essentiels des codeurs basés sur le codage par prédiction linéaire et évoquer les critères de mesure de ses performances ainsi que les limites d'un codeur LPC, afin d'aborder l'étude du codeur CELP qui est fondé entre autres sur cette technique.

II.2. Modélisation autorégressive du signal vocal :

Le modèle du processus de production de la parole est caractérisé par un filtre décrivant le fonctionnement du système phonatoire de la glotte jusqu'au lèvre dont la transsmitance est en première approximation sous la forme $\frac{1}{A(z)}$, ce système serait soumis à une excitation idéalisée $v(n)$, cette excitation est un train périodique d'impulsions pour les sons voisé ou un bruit blanc de moyenne nulle et de variance unité pour les sons non voisé.

La transmittance $\frac{1}{A(z)}$ est celle d'un filtre tous pôles. La sortie d'un tel filtre représente le signal de parole, sa transformé en z peut s'écrire :

$$X(z) = \frac{V(z)}{A(Z)} \tag{2.1}$$

Le polynôme $A(z)$ est noté :

$$A(z) = \sum_{i=1}^{M} a_i z^{-i} , \qquad a_0 = 1 \tag{2.2}$$

Dans le domaine temporel l'équation (2.1) introduit l'équation aux différences suivante :

$$x(n) + \sum_{i=1}^{M} a_i x(n-i) = v(n) \tag{2.3}$$

La récurrence linéaire (2.3) montre que l'échantillon $x(n)$ est fonction de l'entrée présente et des échantillons des sorties antérieures, un système régi par une telle équation est appelé un système autorégressive (AR)

La figure (2.1) représente la structure du filtre de synthèse AR,

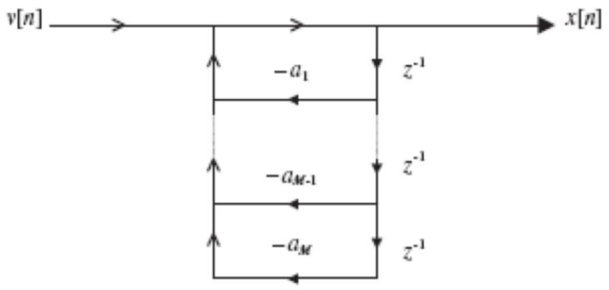


Figure 2.1 : structure du filtre de synthèse AR

Si dans l'équation aux différences (2.3), $v(n)$ est un bruit blanc gaussien stationnaire de moyenne nulle et de variance unité, la sortie $x(n)$ sera appelée signal autorégressive d'ordre M.

II.3. Prédiction linéaire :

La prédiction linéaire est une technique importante pour la compression d'un signal de parole en modélisant le conduit vocal par un filtre dont les coefficients sont déterminés au biais de l'analyse de la redondance du signal, cela est réalisé par la prédiction linéaire qui exploite cette redondance pour prédire un échantillon par une combinaison linéaire d'échantillons antérieurs, c'est l'idée de base du codage par prédiction linéaire LPC.

Lors de l'analyse à court terme, la redondance proche entre les échantillons du signal de parole est supprimée par un filtre d'analyse LP, représentant le conduit vocal. Ce filtre permet d'extraire la structure des formants du signal d'entrée et d'obtenir un signal de sortie de faible énergie qui correspond à l'erreur de prédiction appelée signal résiduel ou d'excitation. Le filtre inverse d'analyse est le filtre de synthèse LP, dont la fonction transfert décrit l'enveloppe spectrale du signal de la parole, il génère le signal de parole synthétisé.

Chaque trame de parole est donc modélisée en sortie du système linéaire par un signal d'excitation. Un meilleur codage de celui-ci pourrait être obtenu en utilisant un prédicteur à long terme qui prendra en compte la corrélation entre les échantillons distants du signal de parole. L'extraction de cette périodicité est obtenue par un estimateur du pitch . Cette analyse n'aura aucun effet sur les sons non voisés.

II.3.1. Principes :

La prédiction linéaire peut être considérés comme une technique d'identification d'un système, où les paramètres d'un modèle AR sont estimés à partir du signal lui-même, cela est illustré par la figure 2.2. Un système autorégressive AR excité par un bruit blanc $x[n]$ fournit à sa sortie un signal autorégressif $s[n]$, les paramètres du système AR sont notés $\widehat{a_i}$.

La prédiction linéaire prédit l'échantillon actuel $s[n]$ par une fonction linéaire de M échantillons passés du signal de parole :

$$s[n] = -\sum_{i=1}^{M} a_s s[n-i] \qquad (2.4)$$

Où les a_i sont les estimés des coefficients du modèle AR, ils sont appelés coefficients de prédiction ou coefficient LP. La constante M représente l'ordre de prédiction. La prédiction linéaire induit une erreur qui est donnée par :

$$e[n] = s[n] - \hat{s}[n] = s[n] + \sum_{i=1}^{M} a_i s[n-i] \quad (2.5)$$

Ou $\hat{s}[n]$ est l'échantillon prédit.

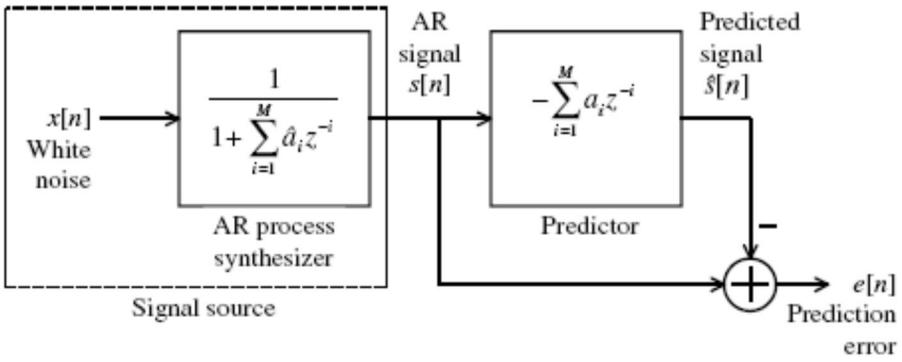


Figure 2.2 : La prédiction linéaire comme système d'identification.

Le problème revient donc à déterminer les coefficients a_i.

II.3.2. Prédiction à court terme :

La prédiction à court terme est basée sur un filtre de synthèse qui génère le signal synthétisé et un filtre d'analyse excité par le signal original qui fournit l'erreur de prédiction qui est l'entrée du filtre de synthèse.

Les coefficients des deux filtres précédents qui ne sont rien d'autre que les coefficients de prédiction, ils sont obtenus à partir d'une analyse du signal de parole original par différentes méthodes, les plus couramment utilisées sont la méthode de l'autocorrélation et la méthode de covariance, la première sera exposée dans la section suivante.

Le filtre de synthèse génère le signal de parole synthétisé à partir de l'erreur de prédiction, il caractérise le conduit vocal, quand ce dernier est modélisé par ce filtre, il est appelé modèle LP, la transmittance de ce filtre est donnée par :

$$H(z) = \frac{1}{1+\sum_{i=1}^{M} a_i z^{-i}} \quad (2.6)$$

Le signal d'excitation (l'erreur de prédiction) doit être multipliée par un gain g qui dépend du signal original, pour le signal résiduel qui excite le filtre de synthèse fournit un signal synthétisé ayant une densité spectral de puissance proche de celle du signal original. La figure 2.3 décrit le filtre de synthèse. Le gain g est donné par la relation :

$$g = \gamma \sqrt{R_s[0] + \sum_{i=1}^{M} a_i R_s[i]} \qquad (2.7)$$

Avec $\boldsymbol{\gamma}$ une constante qui est introduite pour compenser les changements dus au fenêtrage des autocorrélations du signal de parole original, elle varie selon le type de fenêtre appliquée au signal.

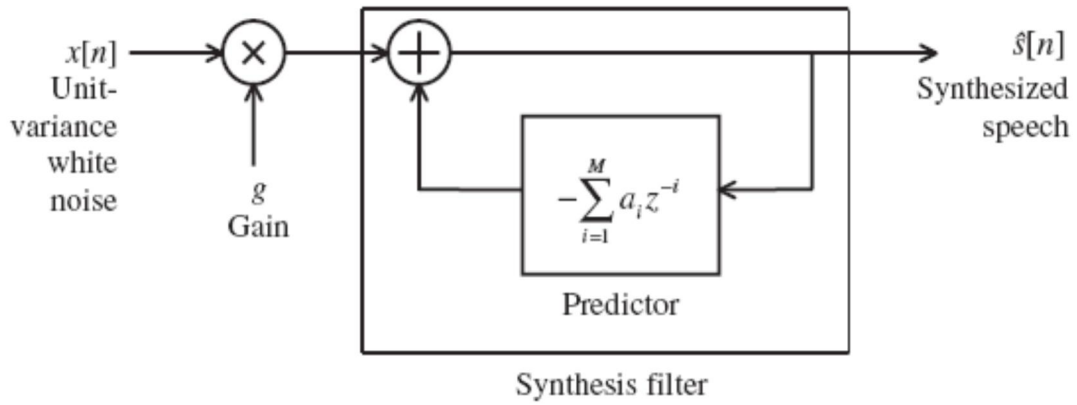


Figure 2.3 : Le filtre de synthèse

La partie analyse contient un filtre d'analyse dont la transmittance $A(z)$ est l'inverse de $H(z)$, ce filtre tout-zéro permet d'extraire l'information prédictible afin de définir le signal résiduel qui excite le filtre de synthèse (Figure 2.4), cela en tirant l'erreur de prédiction entre la trame du signal original et son estimation par prédiction linéaire.

Le signal résiduel est l'excitation idéale pour le filtre de synthèse. Une modélisation précise de ce signal permet d'obtenir un signal reconstruit naturel. Or l'estimation des paramètres de prédiction LP du modèle vocal entraine une approximation du signal d'excitation. L'ajustement du spectre du signal synthétisé est lié à l'ordre de prédiction M, plus celui-ci augmente plus l'enveloppe spectrale est semblable à celle du signal original, l'ordre est le résultat d'un compromis entre une bonne représentation de la structure

formantique et la complexité de calcul. Pour satisfaire ce compromis, l'ordre est choisi généralement de 8 à 16.

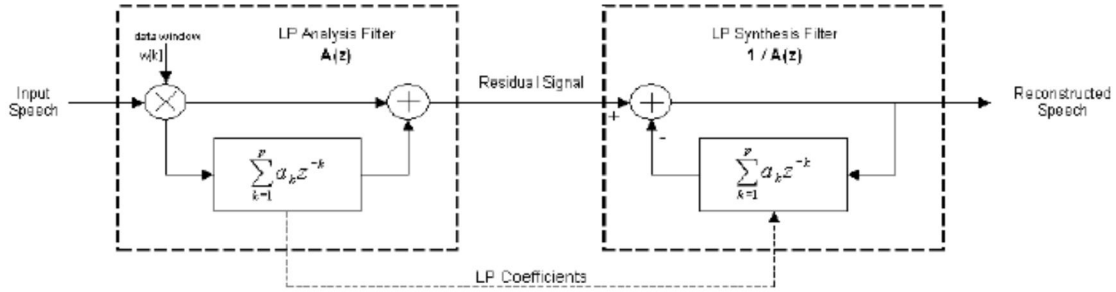


Figure 2.4 : Analyse et synthèse LP

Le filtre d'analyse peut être caractérisé par un gain G, celui-ci est fonction de l'ordre de prédiction comme le montre la figure 2.5.

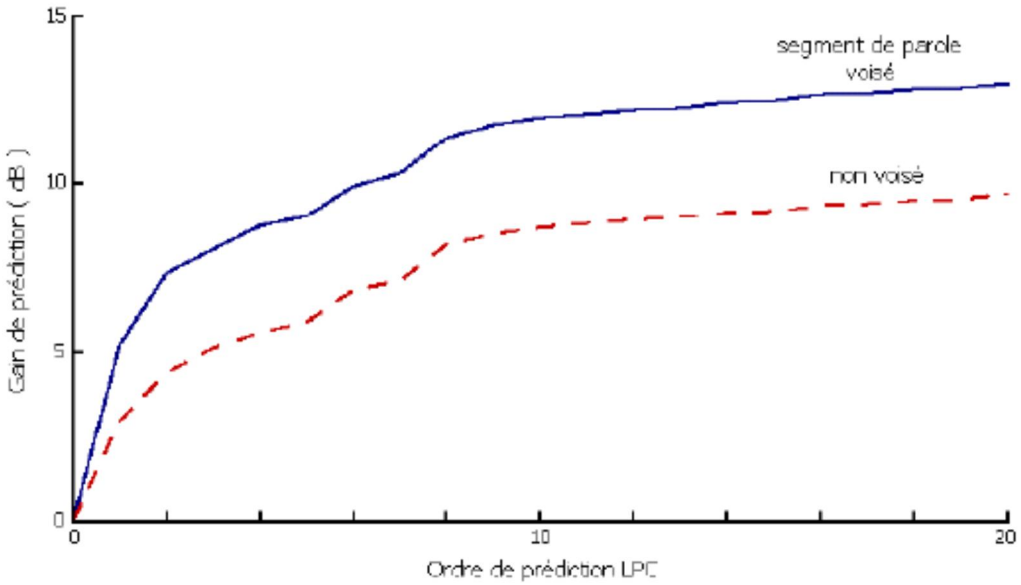


Figure 2.5 : Gain fonction de l'ordre de prédiction

La figure 2.6 montre l'exemple d'un signal vocal traité par un système de prédiction à court terme, ce signal est échantillonné à 8Khz. Après avoir déterminé les coefficients LP du filtre d'analyse, le signal résiduel $e[n]$ est extrait puis appliqué au filtre de synthèse qui fournira le signal synthétisé.

Pour résumer, la prédiction court terme est constituée de deux blocs, le premier est un filtre d'analyse qui extrait le signal résiduel à partir d'une trame du signal de parole original, le deuxième est un filtre de synthèse qui produit le signal de parole synthétisé à partir du signal résiduel. Les deux filtres sont mutuellement inverses et leurs coefficients sont les coefficients de prédiction LP, obtenus par l'analyse du signal original par la méthode de l'autocorrélation.

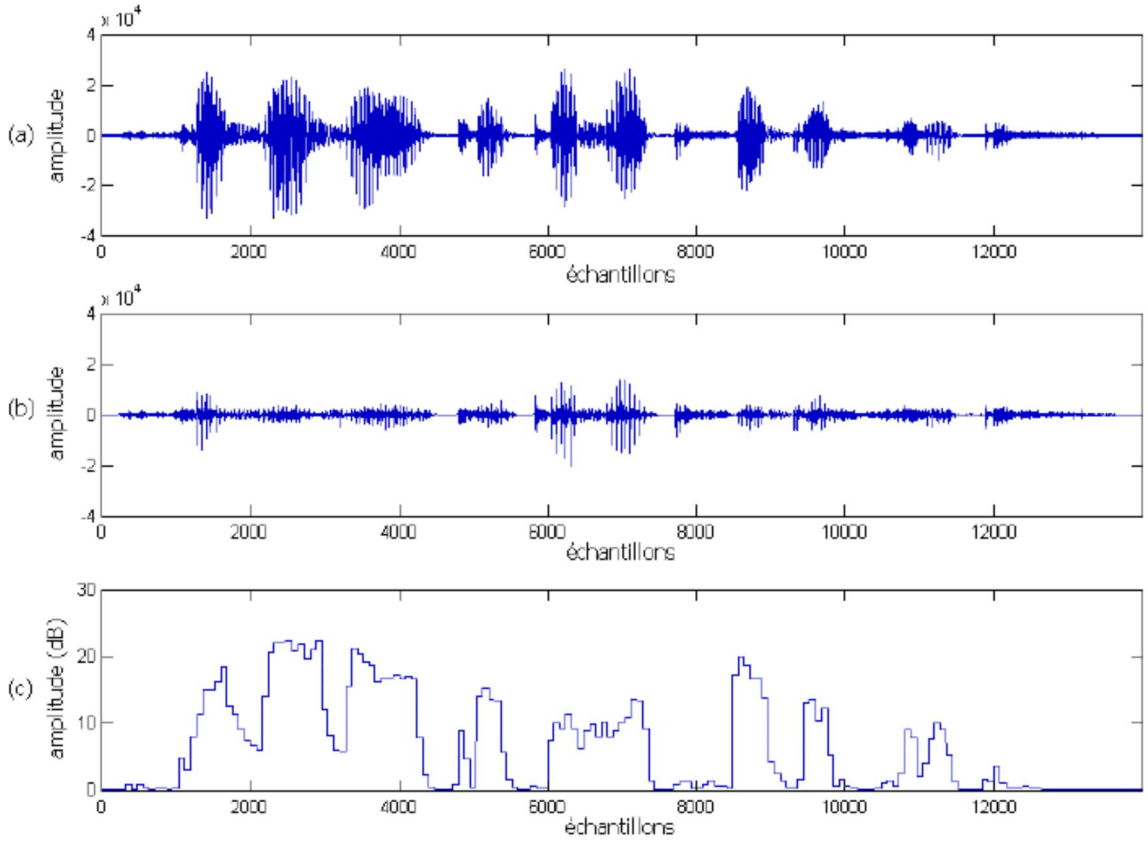


Figure 2.6 : a-Signal source, b-Signal résiduel, c-Gain de prédiction.

II.3.3. Prédiction à long terme :

La prédiction à long terme est un moyen efficace pour représenter la périodicité du signal de parole pour les sons voisés, elle est modélisée par un filtre qui caractérise les vibrations des cordes vocales. Pour se faire on est obligé d'étudier et d'analyser les corrélations entre échantillons éloignés du signal de la parole, ce sont les corrélations à long terme qui décrivent la fréquence fondamentale des sons voisés représentée par un nombre T appelé pitch, celui-ci est égale au nombre d'échantillon durant une période du signal voisé.

La prédiction à court terme décrit l'enveloppe spectrale de la trame du signal de parole, elle comprend deux filtres dont les coefficients sont les coefficients LP qui sont tirés par une analyse sur la trame du signal original, le nombre de coefficients est déterminé par l'ordre de prédiction, ce dernier peut être augmenté pour arriver à représenter au moins une période d'un signal voisé mais le nombre de bits alloués au signal serait prohibitif et cette approche engendrerait une complexité de calcul et d'implémentation lors de l'analyse, c'est la raison pour laquelle l'introduction d'une prédiction à long terme est inévitable. Le concept de l'augmentation de l'ordre de prédiction à court terme ne serait pas concluant car si on dépasse un ordre > 10 , les coefficients supplémentaires ne contribueront nullement à l'accentuation du gain de prédiction, c'est l'idée de base de la prédiction à long terme, en effet, un prédicteur à court terme est connecté en cascade avec un autre à long terme (Figure2.7). Le prédicteur à court terme possède un ordre de prédiction réduit M entre 8 et 12, il élimine la corrélation entre les échantillons voisins, le prédicteur à long terme détecte les corrélations correspondantes à la période du pitch.

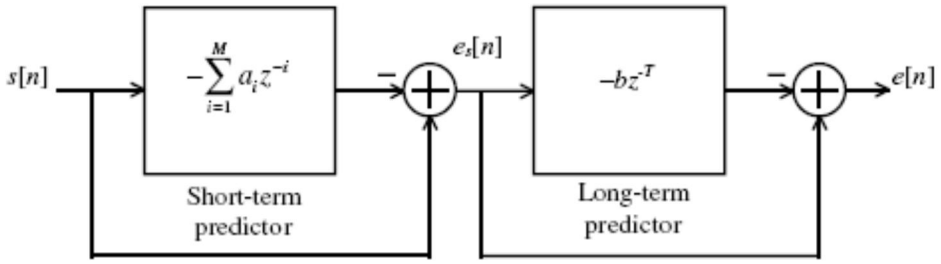


Figure 2.7 : Le filtre de prédiction à court terme connecté en cascade au filtre de prédiction à long terme.

La transmittance du filtre de prédiction à long terme ayant pour entrée l'erreur de prédiction à court terme $e_s[n]$ est :

$$H(z) = 1 + bz^{-T} \tag{2.8}$$

La modélisation de ce filtre nécessite la connaissance de deux paramètres, le pitch T et le gain du filtre de prédiction à long terme b, la détection et l'estimation du pitch est un problème ardu du traitement de la parole, l'étude de ce problème sera évoquée au chapitre 3.

II.3.4. Estimation des paramètres de prédiction :

II.3.4.1. Minimisation de l'erreur de prédiction :

Le problème de la prédiction linéaire réside dans l'estimation des paramètres a_i (équation 2.4), ces coefficients doivent être estimés tel que l'erreur de prédiction $e[n]$ soit minimisée, le critère de minimisation est fondé sur la minimisation de l'erreur quadratique moyenne, celle-ci est donnée par :

$$J = E\{e^2[n]\} = E\{(s[n] + \sum_{i=1}^{M} a_i s[n-i])^2\} \quad (2.9)$$

Le choix des coefficients LP est canalisé par la minimisation de J, les coefficients de prédiction optimaux peuvent être déterminés en annulant les dérivées partielles de l'erreur quadratique moyenne J par rapport au coefficient a_k:

$$\frac{\partial J}{\partial a_k} = 2E\{(s[n] + \sum_{i=1}^{M} s[n-i])s[n-k]\} \quad (2.10)$$

a_k sont les coefficients de prédiction ou coefficients LP, avec $k = 1,2,\ldots,M$, si cette équation est vérifiée, alors les coefficients de prédiction sont égaux aux coefficients du modèle AR décrit dans la section 3.1.

En réarrangeant l'équation (2.10), on peut l'écrire sous la forme :

$$E\{s[n]s[n-k]\} + \sum_{i=1}^{M} a_i E\{s[n-i]s[n-k]\} = R_s[k] + \sum_{i=1}^{M} a_i R_s[i-k] = 0 \quad (2.11)$$

Pour $= 1,2,\ldots,M$.

D'après l'équation (2.11), on a :

$$R_s[k] = -\sum_{i=1}^{M} a_i R_s[i-k] \quad (2.12)$$

Cette équation est équivalente à l'équation matricielle suivante :

$$\mathbf{R_s}\boldsymbol{a} = -\mathbf{r_s} \quad (2.13)$$

Où :

$$\mathbf{R}_s = \begin{pmatrix} R_s[0] & R_s[1] & \cdots & R_s[M-1] \\ R_s[1] & R_s[0] & \cdots & \vdots \\ \vdots & \vdots & \ddots & R_s[1] \\ R_s[M-1] & R_s[M-2] & \cdots & R_s[0] \end{pmatrix} \quad (2.14)$$

$$\mathbf{a}=[a_1, a_2, \dots, a_M]^{\mathbf{T}} \quad (2.15)$$

$$\mathbf{r}_s = [R_s[0], R_s[1], \dots, R_s[M-1]]^{\mathbf{T}} \quad (2.16)$$

Le calcul des coefficients de prédiction a_i se résume donc à inverser la matrice d'autocorrélation $\mathbf{R_s}$, c'est une matrice de Toeplitz symétrique et le système découlant de l'équation matricielle (2.13) est un système de Yule-Walker, l'inversion de la matrice $\mathbf{R_s}$ est une opération qui nécessite un calcul fastidieux et long, un algorithme efficace peut être employé pour déterminer les coefficients LP, c'est l'algorithme de Levinson-Durbin.

II.3.4.2. Algorithme Levinson-Durbin :

L'algorithme de Levinson-Durbin s'est avéré comme une méthode efficace pour résoudre le système (2.13), en effet la résolution de celle-ci exige l'inversion de la matrice d'autocorrélation qu'est une matrice de Toeplitz symétrique. Cet algorithme exploite cette propriété pour fournir une solution aisée afin de calculer les coefficients LP.

L'algorithme de Levinson-Durbin est un algorithme itératif, en effet, le calcul des coefficients de prédiction de l'ordre M nécessite la connaissance des coefficients de prédiction de l'ordre M-1.

Cet algorithme suit les étapes suivantes :

- Initialisation $l = 0$:

$$R_s[0] = J_0$$

- Itération pour $l = 1,2, \dots, M$

  Etape 1 : Calcul des coefficients de réflexion

$$k_l = \frac{1}{J_{l-1}} \left(R_s[l] + \sum_{i=1}^{l-1} a_i^{(l-1)} R_s[l-i]\right) \quad (2.17)$$

  Etape 2 : Calcul des coefficients LP pour l'ordre l

$$a_l^{(l)} = -k_l, \quad (2.18)$$

$$a_i^{(l)} = a_i^{(l-1)} - k_l a_{l-i}^{(l-1)}; i = 1,2, \dots, l-1 \quad (2.19)$$

Arrêter pour $l = M$.

Etape 3 : Calcul de l'erreur quadratique moyenne pour un ordre l

$$J_l = J_{l-1}(1 - k_l^2) \tag{2.20}$$

Pour $l \leftarrow l + 1$; retour à l'étape 1.

Les coefficients de prédiction finaux sont :

$$a_i = a_i^{(M)}; i = 1,2, \dots, M \tag{2.21}$$

Les coefficients k_l sont appelés coefficients de réflexion, ils constituent un formalisme différent pour représenter les coefficients de prédiction LP. Un coefficient $a_i^{(l)}$ représente le i-ème coefficient de l'ordre de prédiction l.

II.4. Représentation des paramètres 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 LP 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. A cause de ce problème les coefficients LP sont représentés sous d'autres formes afin de préserver la stabilité du filtre de synthèse après la quantification.

II.4.1. Les coefficients de réflexion :

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 :

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

II.4.2. Les coefficients cepstraux (LARs) :

Nous avons vu qu'il est préférable de quantifier les coefficients de réflexion que de quantifier les coefficients LP, cependant ceux-ci nécessitent une quantification non uniforme, en effet une fonction appelée fonction de sensibilité spectrale qui mesure la variation de l'amplitude de distorsion spectrale est fonction des coefficient de réflexions k, elle est noté $\Psi(k)$, la figure 2.8 montre que celle-ci prend beaucoup de valeurs pour des coefficients de réflexion proches de +1 à -1 et moins de valeurs pour des coefficient de réflexion proches de zéro, cela conduit à utiliser un quantificateur non uniforme à séquences d'apprentissage de type Lloyd [6], ce dernier utilise un pas de quantification réduit pour les régions ou k de 1 et -1, afin de couvrir la variation rapide de la fonction de sensibilité spectrale.

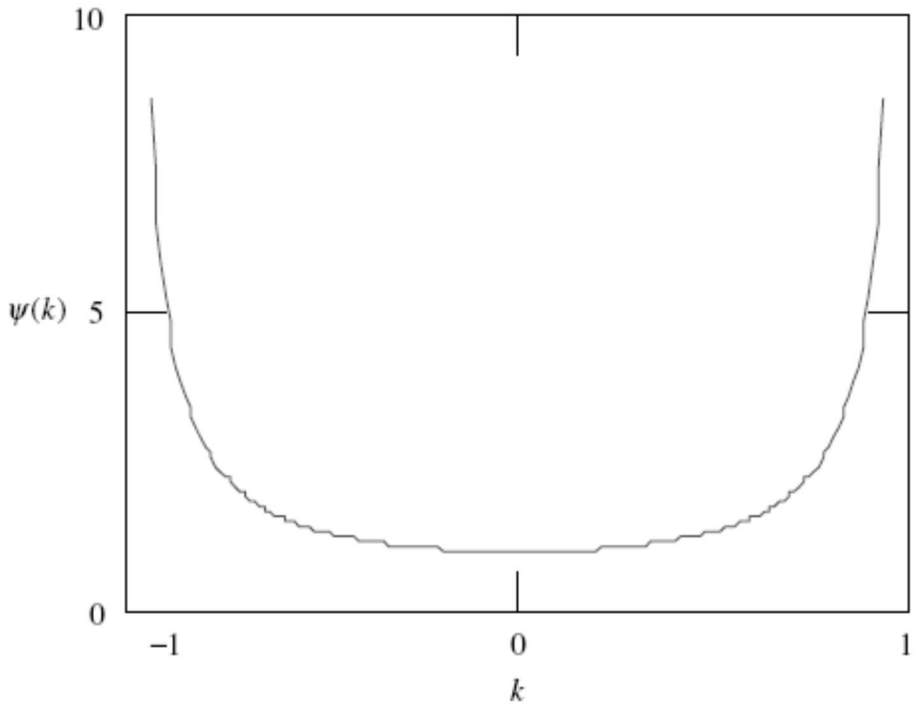


Figure 2.8 : la fonction de sensibilité spectrale

La quantification non uniforme des coefficients de réflexion revient à utiliser un quantificateur uniforme avec une transformation des coefficients de réflexion au biais d'une fonction non-linéaire, les coefficients résultants sont appelés coefficients cepstraux ou coefficients LAR's , cette fonction est donnée par :

$$LAR_i = g = f(k_i) = \log(\frac{1+k_i}{1-k_i}) \quad (2.22)$$

II.4.3. Pairs de fréquences spectrales (LSF) :

La plus répandue des représentations des coefficients de prédiction dans les codeurs de parole standards est la représentation sous la forme de pairs de fréquence spectrale ou coefficients LSF, celle-ci a été introduite par Itakura en 1975 à cause de ses propriétés qui offre plus de robustesse pour la quantification et la transmission, dans cette section nous allons introduire l'idée de la construction des coefficients LSF.

La transmittance du filtre de prédiction s'écrit :

$$A(z) = 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_M z^{-M} = \prod_{i=0}^{M} (1 - z_i z^{-1}) \quad (2.23)$$

Les z_i représentent les zéros du polynôme$A(z)$, on peut se permettre de former deux polynômes $P(z)$ et $Q(z)$ qui seront dérivés du polynôme $A(z)$, ils sont donnés par :

$$P(z) = A(z)\left(1 + z^{-(M+1)}\frac{A(z^{-1})}{A(z)}\right) = \sum_{i=0}^{M+1} p_i z^{-i} \quad (2.24)$$

$$Q(z) = A(z)\left(1 - z^{(M+1)}\frac{A(z^{-1})}{A(z)}\right) = \sum_{i=0}^{M+1} q_i z^{-i} \quad (2.25)$$

Tel que :

$$p_0 = p_{M+1} = 1, \quad (2.26a)$$

$$p_i = p_{M-i+1} = a_i + a_{M-i+1}, \quad (2.26b)$$

$$q_0 = -q_{M+1} = 1, \quad (2.27a)$$

$$q_i = -q_{M-i+1} = a_i - a_{M-i+1}, \quad (2.27b)$$

Les coefficients LSF représentent des valeurs de la fréquence ω tel que :

$P(e^{j\omega}) = 0$ ou $Q(e^{j\omega}) = 0$

On remarque directement que LSF correspondent à des zéros sur le cercle unité, cependant, on ne prendra en considération que les fréquences positives.

Les polynômes $P(z)$ et $Q(z)$ possèdent 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

Les zéros correspondant à 1 et -1 donnent les fréquences $= 0\ ou\ \pi$, ce sont des fréquences qui ne nous intéressent pas, on peut les éliminer en réduisant l'ordre des deux polynomes :

Pour M pair

$$P'(z) = \frac{P(z)}{1+Z^{-1}} = \sum_{i=0}^{M} p'_i z^{-i}$$

$$Q'(z) = \frac{Q(z)}{1+z^{-1}} \sum_{i=0}^{M} q'_i z^{-i}, \qquad (2.28a)$$

Pour M impair

$$P'(z) = P(z) \text{ et } Q'(z) = \frac{Q(z)}{1-z^{-2}} = \sum_{i=0}^{M-1} q'_i z^{-i}\ , \qquad (2.28b)$$

Pour M pair, le polynôme $P'(z)$ est d'ordre M, et le polynôme $Q'(z)$ est aussi d'ordre M, dans le cas ou M est impair les ordes deviennent respectivement M+1 et M-1, dans les deux cas les coefficients des deux polynômes peuvent être calculés à partir des coefficients des polynômes *P(z)* et *Q(z)* [6].

Pour $i = 2,..,M-1$. Les polynomes $Q'(z) et\ P'(z)$ sont symétriques d'ordre pair. Considérons les ordres de $P'(z) et\ Q'(z)$ respectivement$2M_1\ et\ 2M_2$, on a alors :

$M_1 = M_2 = \frac{M}{2}$ pour M pair.

$M_1 = \frac{M+1}{2}$ et $M_2 = \frac{M-1}{2}$ pour M impair.

On construit alors deux fonctions $P_0(z) et\, Q_0(z)$ à partir de $P^{'}(z) et\, Q'(z)$ en remplaçant z par $e^{j\omega}$, les fréquences ω qui annulent ces deux polynômes annulent aussi les polynômes $P(z) et\, Q(z)$, les fonctions $P_0(\omega) et\, Q_0(\omega)$ sont données par :

$$P_0(\omega) = 2\cos(M_1\omega) + 2p'_1\big((M_1-1)\omega\big) + \cdots + 2p'_{M_1-1}cos\omega + p'_{M_1} \quad (2.29)$$

$$Q_0(\omega) = 2\cos(M_2\omega) + 2q'_1\big((M_2-1)\omega\big) + \cdots + 2q'_{M_2-1}cos\omega + q'_{M_2} \quad (2.30)$$

Pour $i = 1,..,M$, ω_i représentent les coefficients LSF et $0 < \omega_1 < \omega_2 < \cdots < \omega_M < \pi$, s'il existe une erreur lors de la quantification des LSF alors celle-ci est localisée, ces paramètres sont directement liés au spectre de la parole, Un spectre d'amplitude des fonctions $P_0(z) et\, Q_0(z)$ est esquissé à la figure 2.9 exposant au passage la localisation des fréquences LSF.

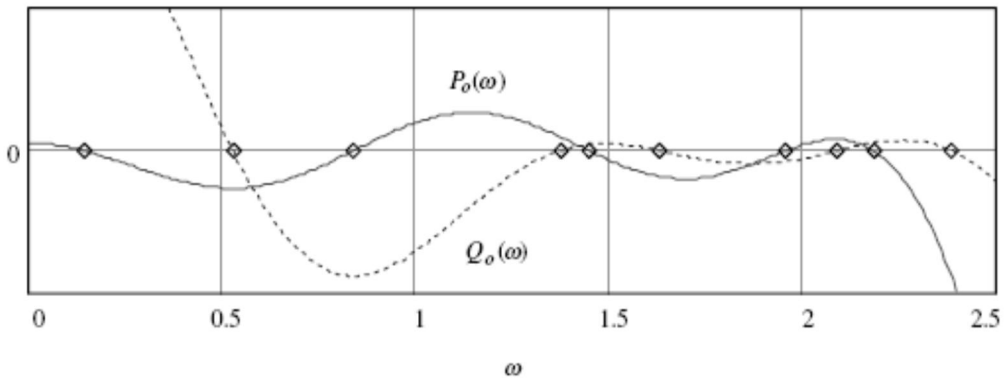


Figure 2.9 : Spectre d'amplitude des fonctions$P_0(z) et\, Q_0(z)$ avec la localisation des LSF

Le passage des polynômes $P^{'}(z) et\, Q^{'}(z)$ aux fonctions $P_0(\omega) et\, Q_0(\omega)$ nous facilitent la tâche pour retrouver les fréquences ω qui représentent les coefficients LSF.

II.4.4. Conversion LSF-LPC :

Une fois quantifiés et transmis, 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 (2.26a) et (2.26b), sur la base des coefficients LSF quantifiés [8].

La méthode adoptée pour retrouver les coefficients LP est la méthode inverse du calcul des LSF, en effet, ces derniers peuvent être transformés en considérant :

$$x_i = cos\omega_i$$

Donc il suffit de transformer les fonctions $P_0(\omega) et\, Q_0(\omega)$ en polynômes $P_0(x)\, etQ_0(x)$, ensuite d'identifier les coefficients p'_i, q'_i des polynômes $P'(z) et\, Q'(z)$ et retrouver les coefficients des polynômes P(z) et Q(z) et enfin de tirer les coefficients LP à partir de ceux-ci :

Une fois les polynômes P(z) et Q(z) retrouvés, on calcule les coefficients de A(z) qui représentent les coefficients LP :

$$a_M = \frac{p_i+q_i}{2} \tag{2.31}$$

$$a_{M-i+1} = \frac{p_i-q_i}{2} \tag{2.32}$$

Pour M pair et $i = 1, \dots, M_1$. Pour M impair, on applique la modification :

$$a_{M_1} = {}^{p_{M_1}}/_{2} \tag{2.33}$$

L'utilisation des coefficients LSF dans les codeurs de parole offre beaucoup plus d'avantage que l'utilisation directe des coefficients LP, parmi ces avantages on évoque :

- La modification des coefficients LSF par des opérations telle que la quantification et la transmission n'affecte pas la stabilité du filtre de synthèse [7].
- La modification de l'un des coefficients LSF n'affecte que localement la densité spectrale de puissance du signale de parole, contrairement aux coefficients LP ou les coefficients RC dont la modification affecte intégralement ce spectre [6].
- Si une erreur se produit lors de la quantification des LSF, celle-ci pourra être localisée [2].

Les coefficients LSF sont généralement quantifiés par une quantification vectorielle par split, dans le cas d'un codeur CELP, on crée trois sous espaces de dimensions 3,3 et 4.

II.5. Mesure de distorsion :

Après avoir effectué la conception d'un algorithme de codage, il est nécessaire de mettre ce dernier sous tests afin de l'évaluer et de voir s'il répond aux normes et aux critères de codage.

Les algorithmes de codage de la parole sont évalués selon plusieurs critères dont les plus importants sont : la qualité du signal, le débit binaire, la complexité de l'algorithme et le retard de communications, nous consacrons cette partie au premier critère qui est la qualité du signal.

Dans les communications numériques, la qualité du signal parole est évaluée selon quatre catégories [9]:

- **Qualité diffusion ou broadcast** :

Qui se réfère aux larges bandes (typique 50-7000 Hz et 20-20000 Hz pour disques compacts) c'est la plus haute qualité qu'on peut atteindre, elle nécessite des débits au moins de 32 à 64 kbps.

- **Qualité réseau ou toll** :

C'est la qualité qui permet d'entendre la parole sur un réseau téléphonique (pour une bande de 200-3200 Hz avec un rapport signal sur bruit de 30 dB et une distorsion moins de 2 à 3 %).

- **Qualité de communications** :

Elle implique une certaine dégradation de la qualité de la parole, néanmoins, elle présente une qualité naturelle et hautement intelligible. Cette qualité peut être atteinte à des débits supérieurs à 4 kbps.

- **Qualité synthétique** :

La parole synthétique est intelligible, néanmoins, elle n'est pas naturelle et permet pas la reconnaissance du locuteur.

Le but actuel dans le codage de la parole est d'atteindre la qualité *toll* pour des débits de 4 kbps. Actuellement, les codeurs opérant en dessous de 4 kbps de débit, fournissent une qualité synthétique. La mesure de qualité est une tâche importante mais très difficile. Il y a deux manières pour mesurer la qualité de la parole, on distingue la mesure subjective et la mesure objective.

II.5.1. Mesure de la distorsion subjective :

La procédure d'évaluation subjective est achevée par des tests d'écoute de l'ensemble des syllabes, mots ou phrases. Le test est souvent concentré sur les consonnes car elles sont plus difficiles à synthétiser que les voyelles. Dans ces tests la qualité est mesurée par l'intelligibilité qui est définie par un pourcentage de mots ou phonèmes correctement écoutés, et avec une sonorité naturelle. Il existe trois types de mesures subjectives de la qualité [12]:

- **Test diagnostique de rime (DRT) :**

Il s'agit d'une mesure d'intelligibilité dont la tâche est de reconnaître un ou deux mots possibles parmi un ensemble de paires de rimes, par exemple (meat - heat).

- **Mesure diagnostique d'acceptabilité (DAM)**

Elle sert pour l'évaluation des systèmes de communications, elle est basée sur l'acceptabilité de la parole par des auditeurs normatifs qualifiés.

- **Le test MOS**

Ce test est largement utilisé pour évaluer la qualité de la parole. Le MOS nécessite 12 à 14 auditeurs (pour le CCITT et TIA les tests nécessitent 32 à 64 auditeurs) qui sont entraînés pour évaluer phonétiquement la qualité selon une échelle de cinq (5) niveaux.

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

Tableau 2.1 : Qualité avec le critère MOS

II.5.2. Mesure de la distorsion objective :

Le système auditif humain est l'évaluateur le plus adéquat de la qualité et des performances d'un codeur de la parole. Il permet de préciser l'intelligibilité et la sonorité

naturelle des sons. Bien que les tests d'écoute subjectifs donnent une bonne évaluation des codeurs de la parole, ils exigent beaucoup de temps et sont inconsistants. Les mesures objectives peuvent donner une évaluation immédiate et efficace de la qualité d'un algorithme de codage.

Les mesures objectives de distorsion peuvent être calculées aussi bien dans le domaine temporel (calcul du rapport signal sur bruit) que fréquentiel (mesure de distorsions).

II.5.2.1. Mesure dans le domaine temporel :

Les mesures objectives les plus importantes dans le domaine temporel sont les suivantes [18]:

- **Le rapport signal sur bruit SNR (Signal to Noise Ratio)**

C'est la mesure objective de la qualité la plus commune pour l'évaluation des performances des algorithmes de compression. Le SNR est défini comme un rapport de l'énergie moyenne du signal parole sur l'énergie moyenne du signal d'erreur, le SNR est généralement exprimé en décibel *dB* et défini par :

$$SNR = 10log_{10}\left(\frac{Energie\ moyenne\ du\ signal\ de\ parole}{Energie\ moyenne\ du\ signal\ d'erreur}\right)\ dB = 10log_{10}\frac{\sum_{n=-\infty}^{\infty} s^2[n]}{\sum_{n=-\infty}^{\infty}(s[n]-\hat{s}[n])^2}\ dB$$

(2.34).

Où ŝ[n] est la version codée du signal parole original s[n]. La mesure SNR n'est par une estimation exacte de la qualité, en effet, le SNR ne donne qu'une seule évaluation pendant toute la durée du signal, on traite le signal parole en tant qu'un seul vecteur, alors qu'en réalité, l'auditeur effectue plusieurs comparaisons pour un signal parole donné. C'est pourquoi on préfère utiliser le SNR segmental.

- **Le SNR Segmental (SNRseg)**

Les variations temporelles de performance peuvent être mieux détectées et évaluées en utilisant un rapport signal sur bruit à court terme (trame par trame), cette mesure s'appelle SNR segmental (SNRseg). Pour chaque trame (typiquement de 15 à 25 msec), on mesure le SNR et la mesure finale sera la moyenne des mesures pour tous les segments du signal. La mesure SNRseg, en dB sur M segments est définie par :

$$SNR_{seg} = \frac{1}{M}\sum_{j=0}^{M-1} 10log\left[\frac{\sum_{n=1}^{N} s^2(n+Nm)}{\sum_{n=1}^{N}(s[n+Nm]-\hat{s}[n+Nm])^2}\right] dB \quad (2.35)$$

Où chaque segment m est de longueur de N échantillons. Pour un signal de parole échantillonné à 8 kHz, la valeur typique de N est entre 100 et 200 échantillons (15-25 msec).

II.5.2.2.Mesure dans le domaine fréquentiel :

La différence entre l'enveloppe spectrale du signal parole original et celle du signal codé, qui peut être traduite par une différence entre les fréquences des formants ou entre leurs largeurs, conduit à des sons phonétiquement différents. C'est pourquoi on fait recours à la distorsion spectrale. Une brève description des différentes mesures de distorsion dans le domaine fréquentiel est présentée dans ce qui suit[9] :

- Distorsion d'Itakura-Saito

La distorsion d'Itakura-Saito, connue sous le nom de mesure de distance du rapport de vraisemblance, mesure le rapport d'énergie entre le signal résiduel obtenu en utilisant le filtre LP avec les coefficients quantifiés et le signal résiduel obtenu en utilisant le filtre LP avec les coefficients non quantifiés.

- Distorsion spectrale Logarithmique

La mesure de distorsion spectrale Logarithmique est la plus fréquemment utilisée, appelée souvent *distorsion spectrale*.

- Distance euclidienne pondérée

Les LSFs possèdent une relation directe avec la forme de l'enveloppe spectrale. Les formants correspondent aux LSFs voisins (étroitement liés) tandis que les LSFs isolés représentent les vallées. Par conséquent, une distance du carré de l'erreur peut être utilisée pour comparer les vecteurs LSFs originaux et les vecteurs LSFs codés.

II.6. Limites d'un codeur LPC :

Un codeur LPC est un codeur qui est fondé sur la prédiction linéaire, il fournit un signal de parole intelligible pour des débits réduits, il est constitué par des filtres de synthèse et d'analyse ainsi que d'autres modules tel que le détecteur de voisement, estimateur de pitch, le filtre de prédiction à long terme….etc., il utilise deux types d'excitations, un train

d'impulsions périodique pour les sons voisés et un bruit blanc pour les sons non voisés, ce qui donne au signal de parole une qualité artificielle, parmi les codeur LPC les plus populaires, on retrouve le standard FS1015 qui a été développé par le DoD en 1982 pour des applications militaires.

Bien que ce codeur produise de la parole à des débits réduit, il présente beaucoup de limites quant à la qualité du signal, en effet, parmi ces limites, on énumère:

- Fréquemment, la trame du signal de parole ne peut pas être classée voisé ou pas par le codeur lui-même, cette inexactitude peut générer des bruits dérangeant comme des bourdonnements ou des bruits tonals.

- Utiliser uniquement un bruit blanc ou un train d'impulsions ne corrobore pas avec les observations pratiques, qui montrent par exemple qu'un signal voisé est une combinaison d'un train d'impulsions et de bruit blanc.

- Les informations sur la phase du signal original ne sont pas préservées et bien que l'oreille humaine ne soit pas sensible à la phase du signal, la préservation de celle-ci donne un aspect plus naturel au signal synthétisé et présente donc une meilleure qualité de la parole.

- L'approche utilisée pour la synthèse des sons voisés qui consiste en l'excitation du filtre de synthèse avec un train d'impulsion est une entorse au fondement du modèle AR.

II.7.Conclusion :

Dans ce chapitre nous nous sommes focalisés sur la technique de codage par prédiction linéaire qui est devenue un rudiment pour la plupart des standards du codage de la parole, nous avons présenté une idée générale sur la modélisation AR de la parole et étudié les compartiments essentiels d'un codeur LPC, en l'occurrence, la prédiction à court terme, à long terme ainsi que l'algorithme qui permet de retrouver les coefficients LP et les différentes représentations de ceux-ci. Ensuite, nous nous sommes intéressés aux techniques de mesure de la qualité de la parole et au codage LPC et à ses limites pour nous permettre de passer au chapitre 3 et à l'étude du codeur CELP qui s'est avéré être vigoureux pour pallier les lacunes d'un simple codeur LPC.

Chapitre III:

L'algorithme CELP et le standard FS1016

III.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.

Comme il a été évoqué dans le chapitre précédent, les codeurs LPC travaillent avec de faibles débits en gardant une moyenne intelligibilité de la parole, cela est plus convenable pour des applications militaires. Pour la radiocommunication civile, il est préférable de fournir une bonne qualité de la parole, Atal et Schroeder [7] en 1985, ont proposé un codeur de parole CELP qui vient remédier aux insuffisances d'un simple codeur LPC.

Un codeur CELP contient des compartiments qui ne sont pas présents dans un LPC classique, il utilise entre autres une boucle d'analyse par synthèse pour choisir une excitation convenable, l'excitation est tirée à partir d'un dictionnaire (Codebook), ces deux techniques manquent dans un codeur LPC classique qui utilisent comme excitation soit un train d'impulsions soit un bruit blanc, ce qui irait à l'encontre de la modélisation AR du signal de parole si le train d'impulsions jouait le rôle d'excitation, en plus du caractère de la phase qui est complètement négligé dans ce type de codeurs, le codeur CELP arrive à contourner ces deux problèmes en plus des autres limites des codeurs le précédant.

Le travail dans ce chapitre consiste à étudier les blocs essentiels d'un algorithme CELP, parmi lesquels deux pouvaient être traités dans le chapitre précédent, mais nous avons préféré les étudier dans celui-ci afin de donner une vision global d'un codeur basé sur cet algorithme, ces deux blocs sont la segmentation et l'estimation du pitch. Après l'étude de ces

deux blocs, nous passons aux compartiments spécifiques aux CELP : l'analyse par synthèse, le filtre d'erreur perceptuelle, le dictionnaire et le choix de l'excitation. Ensuite, nous étudierons d'une manière succincte un standard populaire du CELP qui est le FS1016 ainsi qu'une vision globale sur ses deux dictionnaires stochastique et adaptatif.

III.2.Fenêtrage et segmentation :

Avant tout traitement, le signal de parole doit être échantillonné à 8KHz, chaque échantillon doit être quantifié à un entier de 16bits ensuite filtré par un filtre passe-haut pour éliminer les bruits BF, la fréquence de coupure de ce filtre est généralement de l'ordre de 140Hz pour un codeur G729 [6], sa transmittance est donnée par :

$$F(z) = \frac{0.46363718 - 0.92724705z^{-1} + 0.46363718z^{-2}}{1 - 1.19059465z^{-1} + 0.9114024z^{-2}} \quad (3.1)$$

Les réponses impulsionnelle et fréquentielle de ce filtre sont données par la figure (3.1)

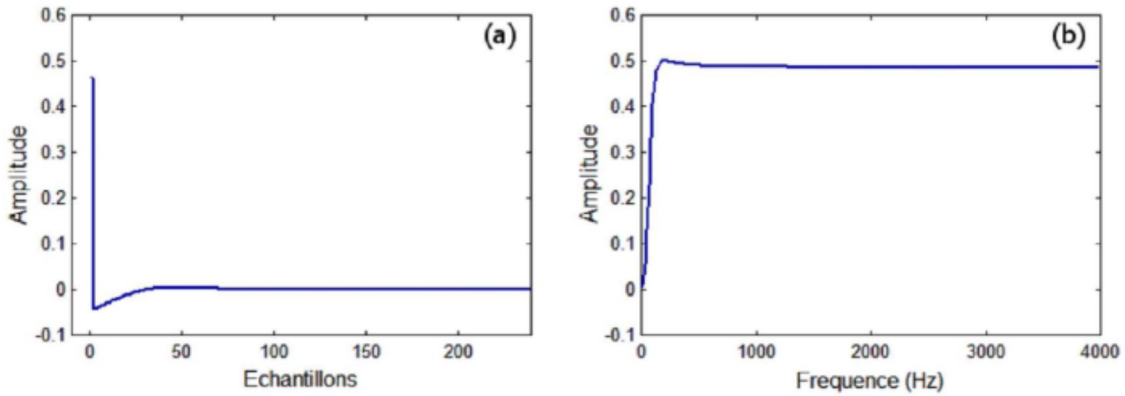


Figure 3.1 : Réponse impulsionnelle(a) et fréquentielle(b) du filtre passe haut

L'estimation des paramètres de prédiction est réalisée trame par trame, cela débute par le fenêtrage de la trame à analyser. La procédure du fenêtrage est décrite à la figure 3.2, la fenêtre appliquée sert à sélectionner la trame ou la sous-trame appropriée à partir du signal de parole original. L'impacte d'une fenêtre sur le signal peut être observé par l'analyse spectrale de celle-ci, la fenêtre la plus simple pouvant être appliquée est la fenêtre rectangulaire :

$$w_{rec} = \begin{cases} 1\,, n = 0,1..,N-1 \\ 0\,,\ Ailleurs \end{cases} \quad (3.2)$$

La fenêtre rectangulaire présente un grand désavantage dans le domaine fréquentiel:

elle possède un lobe principal trop étroit pour un grand nombre de lobes secondaires. A cause de ceux-ci, elle est généralement évitée dans de nombreuses applications.

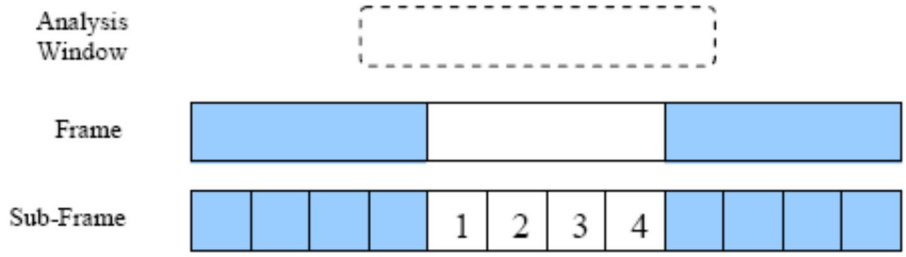


Figure 3.2 : Structure de la fenêtre d'analyse

Dans le but d'éviter les lobes secondaire dans le domaine fréquentiel, un autre type de fenêtre pourrait être appliqué, il s'agit de la fenêtre de Hamming, son expression est donnée par[12] :

$$w_{HAM}(n) = \begin{cases} 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right), & n = 0,..,N-1, \\ 0, & Ailleurs. \end{cases} \qquad (3.3)$$

Le spectre de la fenêtre de Hamming est esquissé à la figure (3.3), elle montre que cette fenêtre a de fortes amplitudes au voisinage des basses fréquences par rapport aux hautes fréquences. La multiplication par une trame va réduire les amplitudes des limites de la trame.

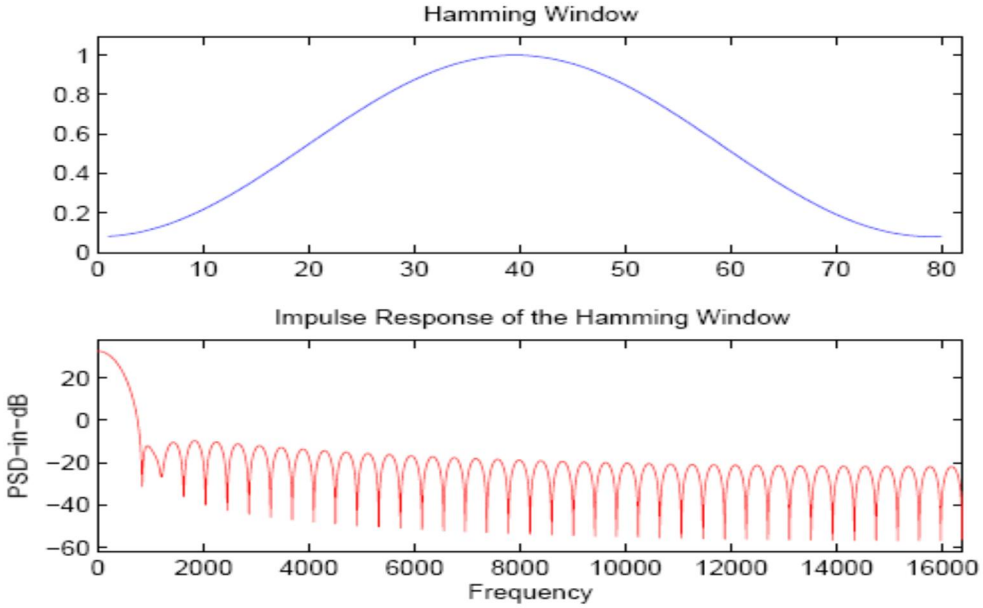


Figure 3.3 : Fenêtre de Hamming

Dans un standard FS1016, à la sortie du filtre, le signal de parole numérisé est segmenté par une fenêtre de Hamming d'une largeur de 240 échantillons, elle est appliquée sur 120 échantillons de l'avant de la trame précédente et 120 échantillons arrières de la trame courante, cela crée trois tampons de parole s_{new} pour les 240 échantillons de la nouvelle trame, s_{old} pour les 240 échantillons de la trame précédente et s_{sub} pour les 240 échantillons issus de la sortie de la fenêtre de Hamming, la durée du signal de parole ainsi segmenté est de 30ms, il est ensuite divisé en quatre sous-trames de largeur 60 échantillons et de durée 7.5ms[6].

L'objectif du fenêtrage et de la segmentation est l'analyse d'un signal de parole stationnaire pour en tirer les coefficients LP et la période du pitch, en effet, l'analyse à court terme par la méthode d'autocorrélation est appliqué sur la trame de parole contenant 240 échantillons, en conséquence, chaque 30ms dix coefficients de prédiction sont tirés et transmis. L'estimation du pitch, par contre, est réalisée avec l'analyse des sous-trames contenant 60 échantillons, cela pour garantir la prise en compte des périodes du pitch inférieures à la longueur de la trame [8] et garantir d'avantage une stationnarité du signal.

III.3.Principe de l'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[14], cela est réalisé è l'aide d'un système en boucle fermée (Figure 3.4).

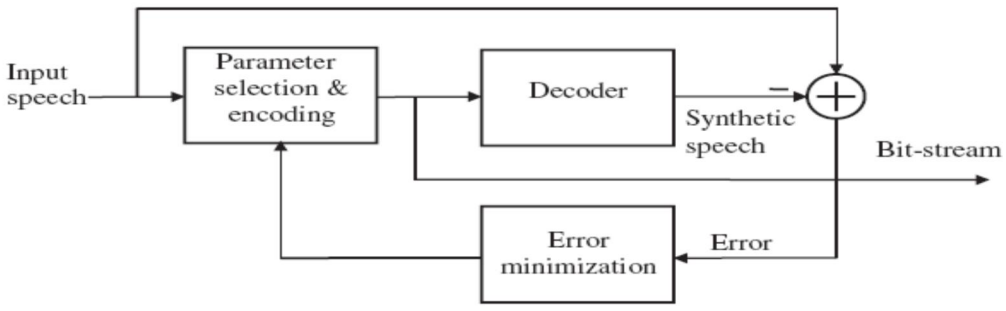


Figure 3.4 : Schéma synoptique d'un codeur de parole utilisant l'analyse par synthèse

D'après la figure 3.4, 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 3.5), cette excitation est à l'entrée du filtre de synthèse du pitch, la sortie de celui-ci vient exciter le filtre de synthèse[7][14], 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.

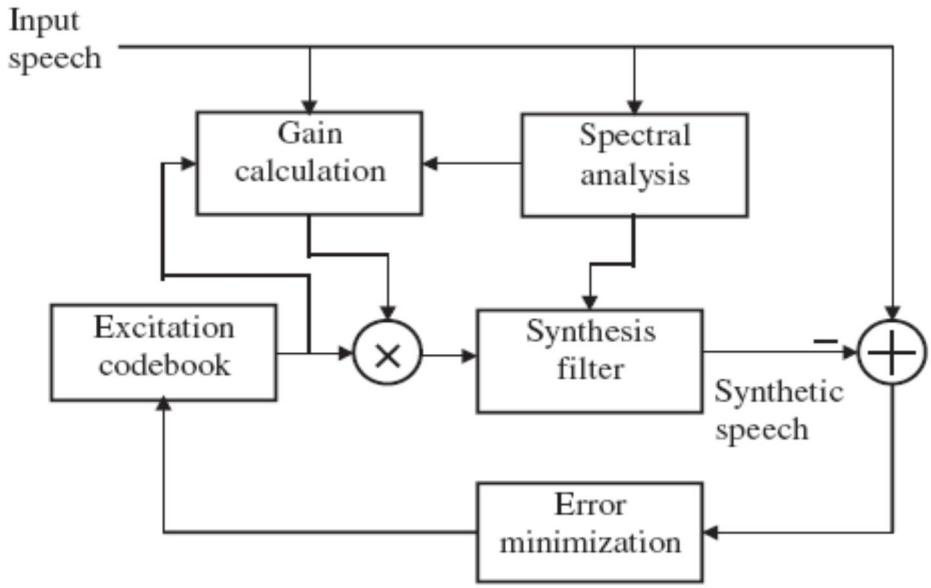


Figure 3.5 : Schéma illustrant la méthode de l'analyse par synthèse dans un codeur CELP

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][6].

III.4.Filtre de pondération perceptive :

L'exploitation de certaine caractéristiques du système auditif s'est avérée nécessaire dans la majorité des codeurs CELP, en effet, celui-ci utilise un filtre de pondération perceptive durant sa boucle d'analyse par synthèse, ce filtre joue un rôle indispensable pour le masquage de fréquences dans la mesure où l'oreille humaine est surtout sensible au signal de parole dans les régions formantiques, par conséquent, elle n'est pas sensible au bruits de quantification dans ces régions ou l'énergie du signal est élevée[7].

Le filtre de pondération perceptive est formé à partir des coefficients de prédiction LP, son rôle consiste à former un bruit afin de rendre son énergie importante dans les régions formantiques, et négligeable dans les vallées (régions entre les pics du spectre), sa transmittance est donnée par :

$$W(z) = \frac{1+\sum_{i=1}^{M} a_i z^{-i}}{1+\sum_{i=1}^{M} \gamma^i a_i z^{-i}} \tag{3.4}$$

Avec γ une constante dont on doit choisir la valeur dans l'intervalle [0,1] afin de contrôler la distribution du bruit.

En faisant varier γ, on contrôle la distribution du bruit, en l'augmentant dans les régions formantiques, il sera masqué par la forte puissance du signal de parole, inversement, il sera plus faible dans les vallées où la puissance du signal de parole n'est pas aussi élevée.

La figure 3.6 montre la boucle d'analyse par synthèse utilisant un filtre de pondération perceptive, ce dernier amplifie les amplitudes du spectre du signal d'erreur dans les régions d'anti-formants et les atténue dans les régions formantiques.

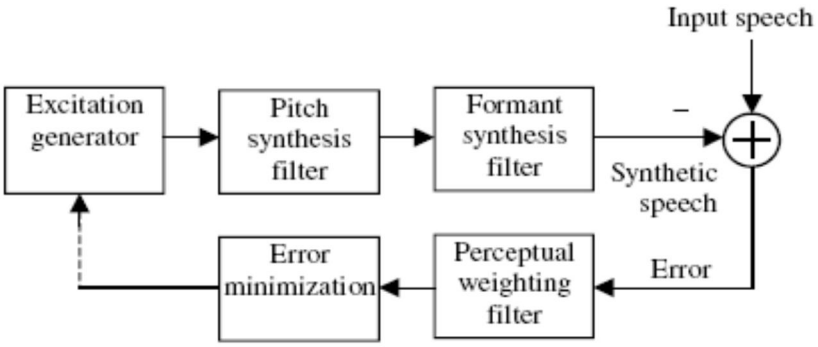


Figure 3.6 : Boucle d'analyse par synthèse d'un codeur CELP utilisant le filtre de pondération perceptive

En d'autre terme, un signal d'erreur dont l'énergie est concentrée dans la région formantique est mieux qu'un signal d'erreur qui est dépourvu de cette propriété, la figure 3.7 montre le spectre d'un filtre de synthèse et le filtre de pondération perceptive correspondant[8].

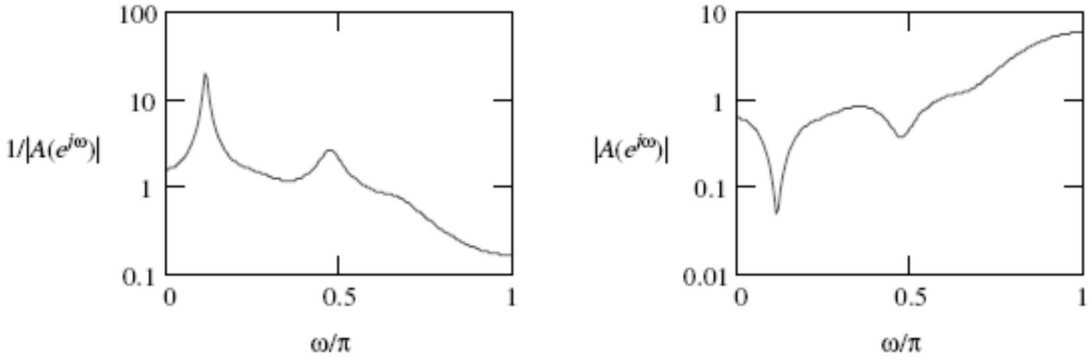


Figure 3.7 : Spectres du filtre de synthèse de formants(gauche) et du filtre de pondération perceptive (droite)

Cependant, pour la réduction de la complexité du calcul itératif de la boucle d'analyse par synthèse et grâce à la linéarité du filtre de pondération perceptive, il est possible de modifier sa position en le mettant en cascade avec le filtre de synthèse ou en le fusionnant avec ce dernier (figure 3.8), avec cette dernière approche le filtre résultant serait[5] :

$$H_f(z) = \frac{1}{A(z/\gamma)} = \frac{1}{1+\sum_{i=1}^{M} \gamma^i a_i z^{-i}} \tag{3.6}$$

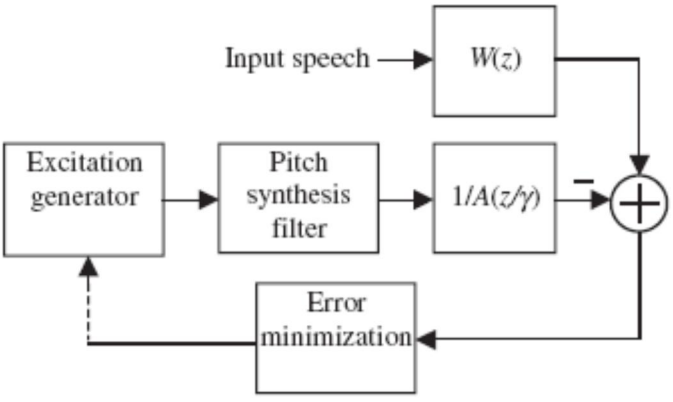


Figure 3.8 : Boucle d'analyse par synthèse avec filtre de pondération perceptive déplacé et fusionné avec le filtre de synthèse de formants.

Avec cette approche et vu le nombre d'excitations dans le codebook qui est élevé, le calcul est réduit par rapport au système de la figure 3.6, en effet, on est pas obligé de filtrer le signal d'erreur.

III.5.La recherche dans le dictionnaire d'excitations :

La recherche dans le dictionnaire d'excitation est la partie la plus délicate et la plus importante dans un codeur CELP, beaucoup d'idées ont été proposées pour accélérer cette procédure sans pour autant dégrader la qualité du signal de sortie, dans cette section nous décrivons le principe de recherche dans un codebook ainsi que la méthode mathématique qui permet la sélection de la meilleure excitation.

III.5.1.Principes :

La procédure de la recherche de l'excitation optimale dans un code-book d'un codeur CELP suit les étapes suivantes[5] :

- Filtrage de la sous-trame du signal de parole original.
- Pour chaque code-vecteur du dictionnaire d'excitations :
    - Calculer le gain optimal et le multiplier par le code-vecteur.
    - Filtrer le code-vecteur après la multiplication par le filtre de synthèse du pitch (Filtre de prédiction à long terme).
    - Filtrer la sortie du filtre de synthèse du pitch par le filtre de synthèse de formants modifié $H_f(z)$.
    - Obtenir le signal d'erreur résiduel par la soustraction de la sortie du filtre de synthèse de formants depuis la sortie du filtre de pondération perceptive dont l'entrée est la sous-trame du signal original.
    - Calculer l'énergie du signal de l'erreur résiduel.
- L'indice de l'excitation présentant l'énergie minimale du signal de l'erreur résiduel est retenu comme information sur la sous-trame.

La procédure décrite ci-dessus est répétée pour chaque sous-trame, il est possible d'améliorer l'efficacité du calcul en décomposant les réponses des filtres en Zero-state et Zéro-input[8], cette méthode est décrite en détail dans le paragraphe suivant, les filtres en question sont les filtres de synthèse du pitch et synthèse de formants montés en cascade comme le décrit la figure 3.9.

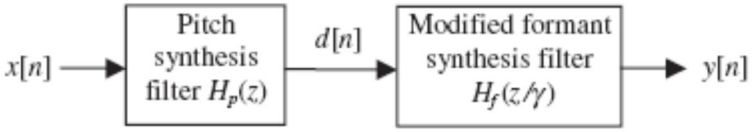


Figure 3.9 : Montage en cascade des filtres de synthèse du pitch et synthèse de formants modifié

Les équations aux différences des soties des deux filtres sont donnée par :

$$y[n] = d[n] - \sum_{i=1}^{M} a_i \gamma^i z^{-i} \quad (3.7)$$

$$d[n] = x[n] - bd[n-T] \quad (3.8)$$

M est l'ordre de prédiction.

a_i sont les coefficients LP

b le gain du filtre de prédiction long terme.

T est la période du pitch.

Ces deux équations aux différences seront utilisées pour le calcul des réponses des filtres pour la méthode du Zero-state Zero-input.

III.5.2.Méthode du Zero-state Zero-input :

III.5.2.1.Principe :

La méthode du Zero-state Zero-input est utilisée dans la boucle de l'analyse par synthèse pour la procédure de recherche du meilleur code-vecteur, elle permet de sauvegarder l'état de la sous-trame présente pour une utilisation ultérieure, cela en séparent les réponses d'un filtre en deux réponses distinctes[12] : l'une pour une excitation nulle mais en étant excité initialement par la sortie de la sous-trame précédente du filtre qui a pour entrée le signal d'excitation (figure3.10).

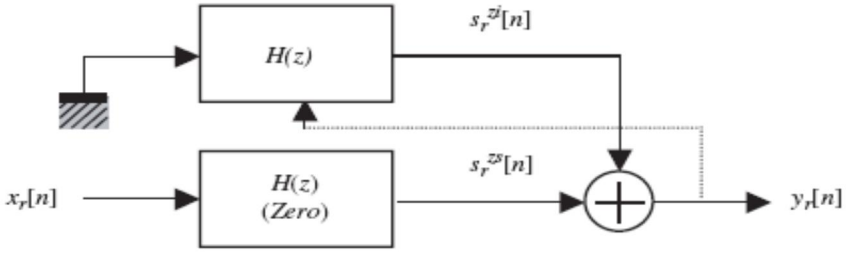


Figure 3.10 : Méthode du Zero-state Zero-input

Dans les équations ci-dessous l'indice r est utilisé pour indiquer la sous-trame en cours de traitement par le filtre par la méthode du Zero-state Zero-input[12] :

- La réponse du filtre Zero-input :

$$s_r^{zi}[n] = y_{r-1}[n+N], \quad -M \le n \le -1 \qquad (3.10)$$

$$s_r^{zi}[n] = -\sum_{i=1}^{M} a_i s_r^{zi}[n-i], \quad 0 \le n \le N-1 \qquad (3.11)$$

L'indice zi dans les deux équations fait référence au filtre de la méthode, dont l'excitation est nulle.

- La réponse du filtre Zero-state :

$$s_r^{zs}[n] = 0, -M \le n \le -1 \qquad (3.12)$$

$$s_r^{zs}[n] = x_r[n] - \sum_{i=1}^{M} a_i s_r^{zs}[n-i], \quad 0 \le n \le N-1 \qquad (3.13)$$

Les a_i représentent les coefficients du filtre et N la longueur de la trame.

III.5.2.2. Application de la méthode du Zero-state Zero-input dans un codeur CELP :

L'application de la méthode du Zero-state Zero-input dans un codeur consiste en son introduction dans la boucle de l'analyse par synthèse de celui-ci en l'appliquant sur les filtres de synthèse du pitch et synthèse de formant modifié (figure 3.11)[8].

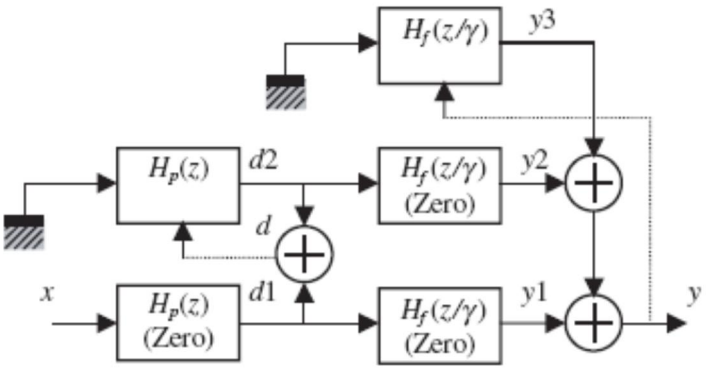


Figure 3.11 : Application de la méthode du Zero-state Zero-input sur le montage en cascade des filtres de synthèse du pitch et synthèse de formant modifié

Les sorties de chaque filtre est donnée par :

- Les réponses du filtre de synthèse du pitch :
  - Zero-state :

$$d1_r[n] = x_r[n], \quad 0 \le n \le T-1 \tag{3.14}$$

$$d1_r[n] = x_r[n] - bx_r[n-T], \quad T \le n \le N-1 \tag{3.15}$$

L'équation (3.14) est ainsi écrite car $d_{1n}[n]$ est nul pour des temps négatifs (pas d'état initial), et de cette propriété découle la terminologie du Zero-state.

  - Zero-input :

$$d2_r[n] = d_{r-1}[n+N], \quad -T \le n \le -1 \tag{3.16}$$

$$d2_r[n] = -bd2_r[n-T], \quad 0 \le n \le N-1 \tag{3.17}$$

Donc la réponse totale du filtre de synthèse du pitch est donnée par :

$$d[n] = d1_r[n] + d2_r[n], \quad 0 \le n \le N-1 \tag{3.18}$$

- Les réponses du filtre de synthèse de formants :
  - Zero-state:

$$y1_r[n] = 0, \quad -M \le n \le -1 \tag{3.19}$$

$$y1_r[n] = d1_r[n] - \sum_{i=1}^{M} a_i \gamma^i z^{-i} y1_r[n-i], \quad 0 \le n \le N-1 \tag{3.20}$$

- Zero-input pour le filtre dont l'entrée est la sortie du Zero-input du filtre de synthèse du pitch :

$$y2_r[n] = 0, \quad -M \leq n \leq -1 \tag{3.21}$$

$$y2_r[n] = d2_r[n] - \sum_{i=1}^{M} a_i \gamma^i z^{-i} d2_r[n-i], \quad 0 \leq n \leq N-1 \tag{3.22}$$

- Zero-input pour le filtre de synthèse de formants dont l'entrée est la sortie du Zero-state du filtre de synthèse du pitch :

$$y3_r[n] = y_{r-1}[n+N], \quad -M \leq n \leq -1 \tag{3.23}$$

$$y3_r[n] = -\sum_{i=1}^{M} a_i \gamma^i z^{-i} y3_r[n-i], \quad 0 \leq n \leq N-1 \tag{3.24}$$

La réponse totale du filtre de synthèse de formants est donnée par :

$$y_r[n] = y1_r[n] + y2_r[n] + y3_r[n], \quad 0 \leq n \leq N-1 \tag{3.25}$$

Dans toutes les équations précédentes, N représente la longueur de la sous-trame, l'indice r la sous-trame en cours de traitement et T la période du pitch.

Considérons un code-book de L excitations, pour la méthode décrite dans la section précédente, le nombre d'opérations pour sélectionner la meilleur excitation est donnée par [11] :

$$Nombre\ de\ sommes = (N(M+2) - T) + (2M+1)N \tag{3.26}$$

$$Nombre\ de\ produits = (N(M+1) - T) + (2M+1)N \tag{3.27}$$

En utilisant cette méthode, d_1 et y_1 doivent étre calculés L fois pour une seule procédure de recherche du meilleur code-vecteur, les signaux d_2, d, y_2 et y_3 sont calculés une seule fois pour cette même procédure. Les équations (3.26) et (3.27) sont valables pour une période du pitch inférieure à la longueur de la sous-trame, pour le cas contraire, elles sont données par [11] :

$$Nombre\ de\ sommes = (M+1)N.L + (2M+1)N \tag{3.28}$$

$$Nombre\ de\ produits = M.N.L + (2M+1)N \tag{3.29}$$

Le tableau 3.1 montre quelques valeur des nombres d'opérations pour N=60, L=512 et un ordre de prédiction M=10, cela pour deux valeurs distinctes de la période du pitch, l'une inférieure à la longueur de la sous-trame et l'autre y est supérieure, en utilisant la méthode du Zero-State Zero-input[17] :

	T=50	T=80
Nombre de sommes	344300	339180
Nombre de produits	313580	308460

Tableau 3.1 : Nombre d'opérations pour la méthode du Zero-state Zero-input dans un codeur CELP pour deux valeurs différentes de la période du pitch.

III.5.2.3. Calcul de l'erreur résiduelle et scalage optimal :

Pour compléter l'utilisation de la méthode du Zero-state Zero-input, il est nécessaire d'introduire dans la figure 3.11 le signal de parole original ainsi que le dictionnaire d'excitations afin de présenter le fonctionnement global de la boucle d'analyse par synthèse (Figure 3.12).

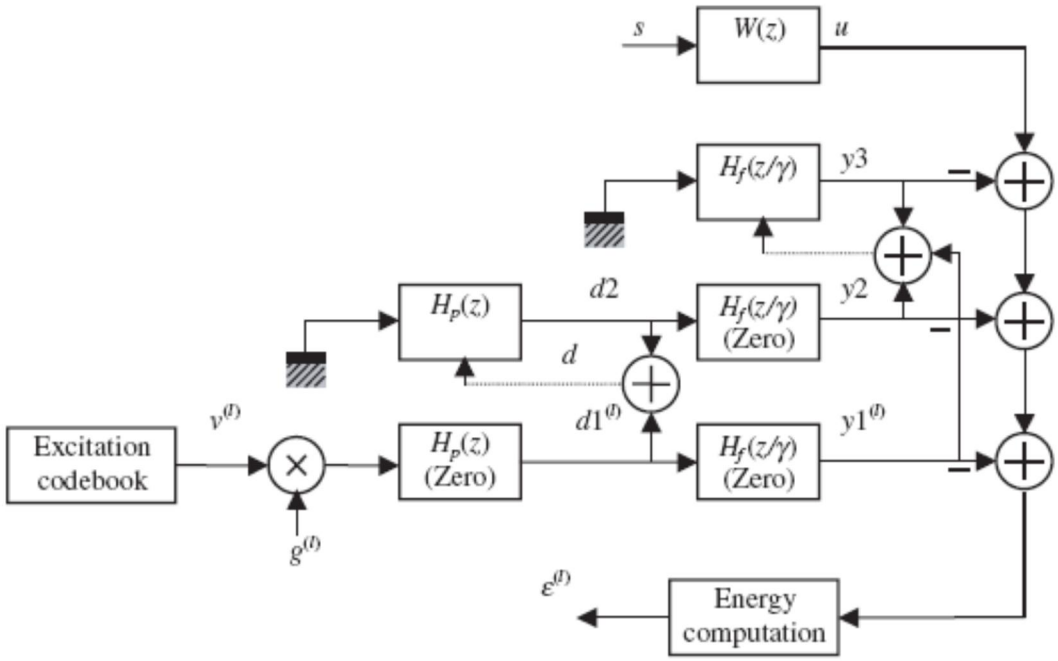


Figure 3.12 : Signaux utilisés pour la recherche de la séquence optimale dans le code-book.

Dans les équations qui suivent, nous supposerons que le traitement se fait sur une sous-trame bien précise et l'indice l indiquera l'essai du $l-ieme$ code-vecteur parmi L.

Le dictionnaire d'excitations contient L code-vecteurs de dimension N qui sont notés :

$v^{(l)}[n],\ \ l = 0, \dots, L-1, \quad n = 0, \dots, N-1$

D'après la figure 3.12, un code-vecteur $v^{(l)}$ est multiplié par un gain $g^{(l)}$, calculé individuellement pour chaque code vecteur, le résultat de la multiplication excitera le filtre Zero-State du LTP (synthèse du pitch), la sous-trame du signal original est filtrée par le filtre de pondération perceptive, elle est noté u.

Avant la sélection du code-vecteur, l'énergie du signal d'erreur résiduel doit être minimisée, elle est donnée par:

$$\varepsilon^{(l)} = \sum_{n=0}^{N-1}(u[n] - y_1^{(l)} - y_2[n] - y_3[n])^2 \quad (3.30)$$

L'indice l présentant la plus petite valeur de ε sera transmis comme information sur la sous-trame, pour peaufiner la procédure de recherche, on doit trouver le gain $g^{(l)}$:

La réponse du Zero-state du filtre de synthèse des formants pondéré excité par le $l - ieme$ code-vecteur normalisé filtré par le Zero-state du filtre LTP est donnée par :

$$y1_0^{(l)}[n] = \frac{y1^{(l)}[n]}{g^{(l)}}, \quad n = 0, .., N-1 \quad (3.31)$$

On a aussi :

$$u_0[n] = u[n] - y2[n] - y3[n] \quad (3.32)$$

Donc l'équation (3.30) pourrait être donnée par :

$$\varepsilon^{(l)} = \sum_{n=0}^{N-1}(u_0[n] - g^{(l)} y1_0^{(l)}[n])^2 \quad (3.33)$$

Le gain est calculé de telle façon à ce que l'énergie du signal de l'erreur résiduelle soit minimisée, il est calculé par l'annulation de la dérivée de $\varepsilon^{(l)}$, il est donné par [7] :

$$g^{(l)} = \frac{\sum_{n=0}^{N-1} u_0[n] y1_0^{(l)}[n]}{\sum_{n=0}^{N-1}(y1_0^{(l)}[n])^2} \quad (3.34)$$

L'équation (3.34) est équivalente à :

$$\varepsilon^{(l)} = \left(\sum_{n=0}^{N-1}(u_0[n])^2\right) - P^{(l)} \quad (3.35)$$

Ou :

$$P^{(l)} = \frac{\left(\sum_{n=0}^{N-1} u_0[n] y1_0^{(l)}[n]\right)^2}{\sum_{n=0}^{N-1} (y1_0^{(l)}[n])^2} \quad (3.36)$$

La minimisation de (3.35) revient donc à maximiser (3.36), cette approche est utilisée dans la plupart des codeurs CELP [11].

- **Calcul de la réponse du Zero-state par la convolution circulaire[11] :**

Notons la réponse impulsionnelle du filtre résultant par la mise en cascade du filtre LTP et le filtre de synthèse des formants par $h[n]$, donc la sortie est donnée par :

$$y1^{(l)}[n] = g^{(l)} \sum_{n=0}^{N-1} h[n-k] v^{(l)}[k] \quad (3.37)$$

En considérons l'entrée et la sortie comme vecteurs, l'équation (3.37) conduit à l'équation matricielle suivante :

$$\mathbf{y1}^{(l)} = g^{(l)} \mathbf{H} \mathbf{v}^{(l)} \quad (3.38)$$

La matrice **H** est la matrice de la réponse impulsionnelle du filtre, c'est une matrice NxN elle est donnée par :

$$\mathbf{H} = \begin{pmatrix} h[0] & 0 & \dots & 0 \\ h[1] & h[0] & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ h[N-1] & h[N-2] & \cdots & h[0] \end{pmatrix}$$

L'approche par le calcul matriciel nous permet de trouver le gain optimal :

$$g^{(l)} = \frac{\mathbf{u}_0^T \mathbf{H} \mathbf{v}^{(l)}}{\left\|\mathbf{H} \mathbf{v}^{(l)}\right\|^2} \quad (3.39)$$

$\left\|\mathbf{H} \mathbf{v}^{(l)}\right\|^2$ représente la norme euclidienne du vecteur $\mathbf{H} \mathbf{v}^{(l)}$.

L'énergie du signal d'erreur résiduel minimisé est alors donnée par :

$$\varepsilon^{(l)} = \|\mathbf{u}_0\|^2 \left[1 - \frac{(\mathbf{u}_0^T \mathbf{H} \mathbf{v}^{(l)})^2}{\|\mathbf{u}_0\|^2 \left\|\mathbf{H} \mathbf{v}^{(l)}\right\|^2}\right] \quad (3.40)$$

L'approche de la convolution circulaire peut paraitre facultative mais la plupart des codeurs sont basés dessus, notamment le FS1016 qui l'utilise durant la procédure de recherche dans son dictionnaire adaptatif[15].

III.6.Estimation de la période du pitch :

La modélisation des vibrations des cordes vocales se fait par l'estimation d'un paramètre représentant leur fréquence de vibration, il est appelé pitch, celui-ci est égal au nombre d'échantillons sur une période du signal de parole voisé, l'estimation de ce paramètre constitue l'une des taches les plus délicate du traitement de la parole, dans ce paragraphe nous décrirons brièvement les deux méthodes les plus utilisée, à savoir, la méthode par boucle ouverte et la méthode par boucle fermée.

III.6.1.Méthode par boucle ouverte :

Le filtre de synthèse du pitch LTP est monté en cascade après le filtre de prédiction à court terme, il est excité par l'erreur de prédiction à court terme, sa transmittance est donnée par :

$$H_p(z) = \frac{1}{1-gz^{-T}} \tag{3.41}$$

T est la période du pitch, g est le gain du filtre LTP, ce filtre prédit la valeur de l'échantillon de l'erreur de prédiction à court terme par un échantillon éloigné d'une durée égale à la période du pitch[9] :

$$\widehat{e_s}[n] = -be_s[n-T] \tag{3.42}$$

La procédure revient à trouver le gain et le pitch qui minimisent la valeur quadratique moyenne MSE de la sortie du filtre LTP :

$$J = \sum_n (e_s[n] - \widehat{e_s}[n])^2 = \sum_n (e_s[n] + be_s[n-T])^2 \tag{3.43}$$

En annulant la dérivée de J par rapport au gain, on trouve ce dernier :

$$g = \frac{\sum_n e_s[n]e_s[n-T]}{\sum_n e^2{}_s[n-T]} \tag{3.44}$$

L'erreur quadratique moyenne peut être alors écrite :

$$J = \sum_n e^2{}_s[n] - \frac{[\sum_n e_s[n] e_s[n-T]]^2}{\sum_n e^2{}_s[n-T]} \quad (3.45)$$

La minimisation de J équivaut à la maximisation du terme à gauche de l'équation (3.45), en testant différentes périodes du pitch dans un intervalle $[T_{min}, T_{max}]$, le pitch maximisant cette équation est retenu comme information sur la trame. Il existe une autre méthode en boucle ouverte, la maximisation de l'autocorrélation, mais elle est généralement évitée car elle n'est guerre efficace pour des petites valeurs du pitch, en particulier si le pitch est inférieur à la longueur de la trame[5].

III.6.2.Méthode par boucle fermée :

L'estimation du pitch par une boucle fermée est utilisée lors de la procédure de l'analyse par synthèse dans un codeur CELP, son but principal est l'augmentation de l'exactitude, cette méthode inclut obligatoirement un deuxième dictionnaire dit adaptatif qui comme son nom l'indique, garde des informations sur les sous-trames précédentes. Cette méthode nous permet d'estimer des pitch dont la valeur est inférieure à la longueur de la trame, le principe par contre reste le même, il est toujours basé sur la minimisation d'une erreur résiduelle qui est combinée avec le dictionnaire stochastique pour estimer le pitch[6].

III.7.Description du standard CELP FS1016 :

III.7.1.Description générale :

Le standard FS1016 est un codeur basé sur l'algorithme CELP, il est le fruit de recherches menées à la fin des années 80's dans le but d'améliorer la qualité du signal de parole synthétisé. En plus des blocs décrits précédemment, il englobe un dictionnaire adaptatif dont les excitations sont régulièrement mises à jour et un dictionnaire stochastique dont les excitations sont fixes (Figure 3.14)[15].

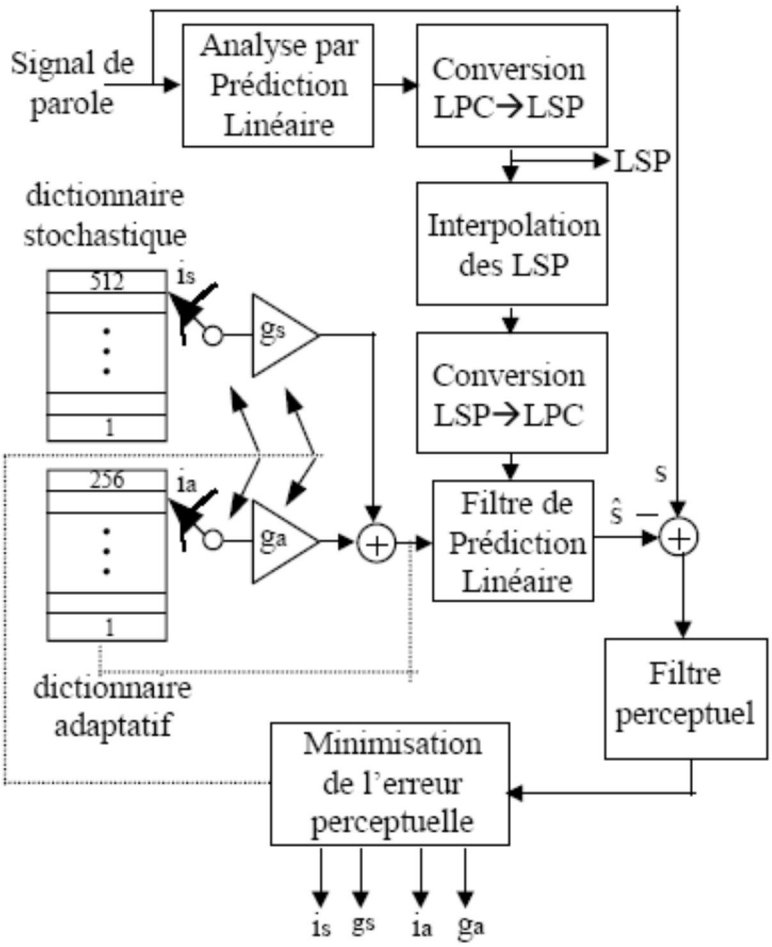


Figure 3.13 : Principe du codeur FS1016.

Le but de l'incorporation du dictionnaire adaptatif est l'amélioration de l'estimation et de la représentation du pitch.

Le signal de parole original dans ce standard est échantillonné à 8KHz et représenté sur 16Bits, la trame contient 240 échantillons et la sous-trame 60 échantillons, les autres compartiments fonctionnent de la même manière que ce qui a été décrit jusqu'à lors, exceptée l'incorporation du dictionnaire adaptatif. Le dictionnaire stochastique est le même que celui décrit dans la section 5. Le tableau 3.2 montre l'allocation des bits dans un standard FS1016.

Paramètres/fenêtre.	Bits/fenêtre	Débits
10 LSP	34 [3,4,4,4,4,3,3,3,3,3]	1,13333
4 Indices ia 4 gains ga	8 + 6 + 8 + 6 4 x 5	1,600
4 Indices is 4 gains gs	4 x 9 4 x 5	1,86667
Bit de synchronisation	1	0,2
Bits de correction d'erreur	4	
Bit non utilisé	1	
Débit total	144	4, 8 Kbits/s

Tableau 3.2 : Allocation des bits pour les paramètres à transmettre dans un codeur FS1016.

III.7.2.Amélioration de la prédiction à long-terme :

L'amélioration de la prédiction à long terme est directement liée à l'exactitude des paramètres du filtre LTP (synthèse du pitch), l'idée globale dans le standard FS1016 est de déterminer ces paramètres à savoir : la période du pitch et le gain du filtre LTP, de la même façon que l'excitation et le gain sont déterminés, c'est-à-dire avec la minimisation de l'énergie de l'erreur perceptive, cela en introduisant une seule modification[6] :

L'énergie de l'erreur perceptive de la $r-ième$ sous-trame s'écrit :

$$\varepsilon^{(l)} = \sum_{n=0}^{N-1}(u[n] - y1_r^{(l)} - y2_r[n] - y3_r[n])^2 \qquad (3.46)$$

Pour minimiser cette énergie, il faut trouver l'ensemble des paramètres : le gain du dictionnaire, l'excitation optimale, le gain du filtre LTP et la période du pitch mais cette démarche engendrerait un calcul fastidieux.

L'issue pour contourner ce calcul fastidieux, consiste en la division de la recherche en deux étapes, on suppose que le gain de l'excitation du dictionnaire est nul et on procède de la même façon sur l'équation (3.46) en annulant le terme $y1_r^{(l)}$, cela conduit à la recherche des paramètres du filtre LTP :

$$J = \sum_{n=0}^{N-1}(u[n] - y2_r[n] - y3_r[n])^2 \qquad (3.47)$$

Donc on retrouve son gain et la période du pitch. Dans la seconde étape, on conserve le filtre LTP tel qu'il est, avec ses paramètres retrouvés puis on minimise l'équation (3.46) pour retrouver le gain du dictionnaire et l'excitation optimale.

Pour la première étape, la minimisation de J se résume à la minimisation de $y2_r$ par rapport à b pour retrouver ce dernier :

$$\frac{\partial y2_r}{\partial b} = \sum_{k=0}^{n} h[k] \frac{\partial d2_r[n-k]}{\partial b}; \quad 0 \leq n \leq N-1 \tag{3.48}$$

$h[n]$ est la réponse imulsionnelle du filtre de synthèse des formants pondéré.

La réponse Zero-input du filtre LTP est donnée par :

$$d2_r[n] = -b.d2_r[n-T]; \quad 0 \leq n \leq N-1, \tag{3.49}$$

Il est évident que pour exprimer la sous-trame actuelle de $d2_r[n]$ en fonction des échantillons passés, il faut prendre en considération la valeur de T, en effet, si $T \geq N$, les indices de la partie droite de l'équation (3.49) sont négatifs, ils caractérisent donc des échantillons passés, ce qui n'est pas le cas si $N/2 \leq T \leq N$, dans ce cas l'équation devient :

$$d2_r[n] = \begin{cases} -bd2_r[n-T]; & 0 \leq n \leq T-1 \\ b^2 d2_r[n-2T]; & T \leq n \leq N-1 \end{cases} \tag{3.50}$$

On remarque que la complexité de calcul augmente au fur et à mesure que la valeur de T diminue. La méthode décrite dans ce paragraphe est peu commode pour des valeurs réduite du pitch, c'est pour cette raison qu'un autre type de dictionnaire a été introduit dans le standard FS1016.

III.7.3.Dictionnaires du codeur FS1016 :

III.7.3.1.Dictionnaire stochastique :

Le dictionnaire stochastique du codeur CELP est similaire à une matrice L×N pour son format non imbriqué, c'est-à-dire que les code-vecteurs sont organisés de tel manière à ce qu'ils forment cette matrice. Le problème majeur de ce dictionnaire est la taille de mémoire requise pour stocker tous les code-vecteurs.

La solution pour ce problème de mémoire est l'utilisation d'un dictionnaire imbriqué ou entre deux vecteurs successifs, il existe des échantillons communs. Un dictionnaire

imbriqué avec un décalage S est montré à la figure 3.15, avec cette approche la taille totale du dictionnaire est L(S-1)+N pour une taille NL pour un dictionnaire non-imbriqué, donc la réduction de mémoire par l'utilisation du dictionnaire imbriqué est remarquable et importante.

Le dictionnaire non-imbriqué n'est rien d'autre qu'un dictionnaire imbriqué avec un décalage égal à la dimension des vecteurs, S=N[5].

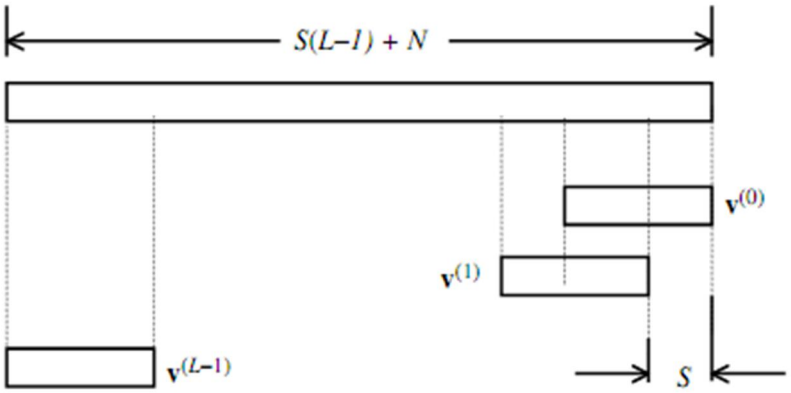


Figure 3.14 :Structure du dictionnaire stochastique.

Dans un dictionnaire stochastique, on peut décrire tous le contenu avec un seul vecteur :

$v[n];\ \ n = 0,1, \dots, S(L-1) + N - 1$

Chaque code vecteur peut être décrit par :

$$v^{(0)}[n] = v[n + (L-1)S],$$

$$v^{(1)}[n] = v[n + (L-2)S], \qquad (3.51)$$

$$\vdots$$

$$v^{(L-1)}[n] = v[n],$$

De cette équation, on peut montrer que pour $n = 0$ à $n = N - 1$:

$$v^{(l+1)}[n] = v^{(l)}[n - S]; \quad S \le n \le N - 1 \qquad (3.52)$$

Le dictionnaire stochastique contient des bruits blancs décorrelés et des séquences décalées de bruits blancs aussi décorrelés.

Un autre avantage du dictionnaire imbriqué est la sauvegarde des résultats avec l'application de la convolution circulaire, en effet, cette dernière permet de sauvegarder le résultat puisque la réponse du Zero-State pour une excitation donnée peut être retrouvée depuis la réponse correspondante à la version décalée de celle-ci.

Le FS1016 utilise des excitation gaussiennes à moyenne nulle et à variance unité quantifiées à +1,0 et -1.

III.7.2. Le dictionnaire adaptatif :

Le concept du dictionnaire adaptatif a été développé pour remédier à la complexité de calcul et la détermination du pitch pour des valeurs inférieures à la longueur de la sous-trame pour celui-ci, cependant, il utilise toujours une procédure de minimisation d'erreur perceptive pendant la boucle de l'analyse par synthèse. Avec cette approche, la réponse du filtre Zero-input est redéfinie de la manière ci-dessous[5] :

$$d2_r[n] = \begin{cases} d2_r[n-T]; & 0 \le n < T \\ d2_r[n-2T]; & T \le n < 2T \\ d2_r[n-3T]; & 2T \le n < 3T \end{cases} \qquad (3.53)$$

Le résultat de cette réponse pour $T < N$ est montré à la figure 3.16 :

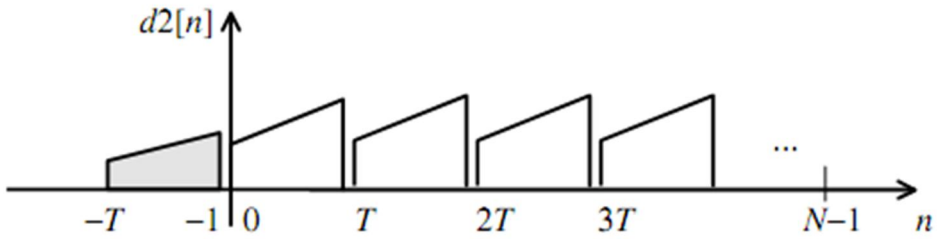


Figure 3.15 :Extraction d'un code-vecteur depuis le dictionnaire adaptatif : les échantillons entre –T et -1 sont régénérés pour former le code-vecteur

Pour T>N, la procédure de recherche définie précédemment reste valable, pour T<N, la sortie du dictionnaire adaptatif est créée par une duplication d'échantillons passés qui seront multipliés par le gain du filtre de synthèse du pitch. Avec l'utilisation du dictionnaire adaptatif, la réponse du filtre Zero-input pour la sous-trame présente est déterminée à partir de l'historique (échantillons passés), la valeur optimale du gain du filtre de synthèse du pitch

peut être ainsi obtenue par la recherche de la valeur du pitch T en utilisant cette procédure de recherche avec le dictionnaire adaptatif.

Avec l'utilisation du dictionnaire adaptatif, on peut se passer de l'utilisation du filtre LTP, en effet, ce dictionnaire est similaire au dictionnaire stochastique à la seule différence qu'il n'est pas fixe, on extrait donc une séquence périodique à partir du dictionnaire adaptatif puis celle-ci est multipliée par un gain optimal approprié qui est équivalent au gain du filtre LTP(Figure 3.14), cette procédure est équivalente à l'utilisation d'un filtre de synthèse du pitch. Les code-vecteur de ce dictionnaire sont imbriqués comme ceux du stochastique mais ils sont mis à jours à chaque sous-trame [15].

III.8.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.

Chapitre IV:

Implémentation et résultats

IV.1.Introduction :

Les deux chapitres précédents nous ont 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 des principaux modules d'un algorithme CELP, 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.

L'implémentation de la boucle d'analyse par synthèse nécessite l'utilisation d'un dictionnaire fixe, l'absence de celui-ci nous a obligé à réaliser l'algorithme sans pouvoir le tester, nous allons expliquer les démarches suivies pour sa réalisation sans pouvoir obtenir les deux paramètres restant en l'occurrence: le gain et l'indice du dictionnaire.

Enfin, nous présenterons une application sur MATLAB qui a été conçu pour la simulation du standard FS1016, et comparer son signal synthétisé au signal original, par une évaluation objective de celui-ci.

IV.2.Fenêtrage et segmentation :

Pour pouvoir implémenter le codeur sur MATALB, il faut y charger les données d'un signal de parole, pour se faire, nous avons utilisé une commande appelée 'wavread', celle-ci

nous permet non seulement de charger ces données mais nous évite la tache de concevoir un programme qui échantillonne le signal à 8KHz et le quantifie à des entiers de 16Bits. Cette commande présente le signal sur MATLAB comme vecteur dont la dimension est le nombre total d'échantillons.

Le résultat de cette commande sur un échantillon de parole de format « wav » est donné par la figure 4.1 :

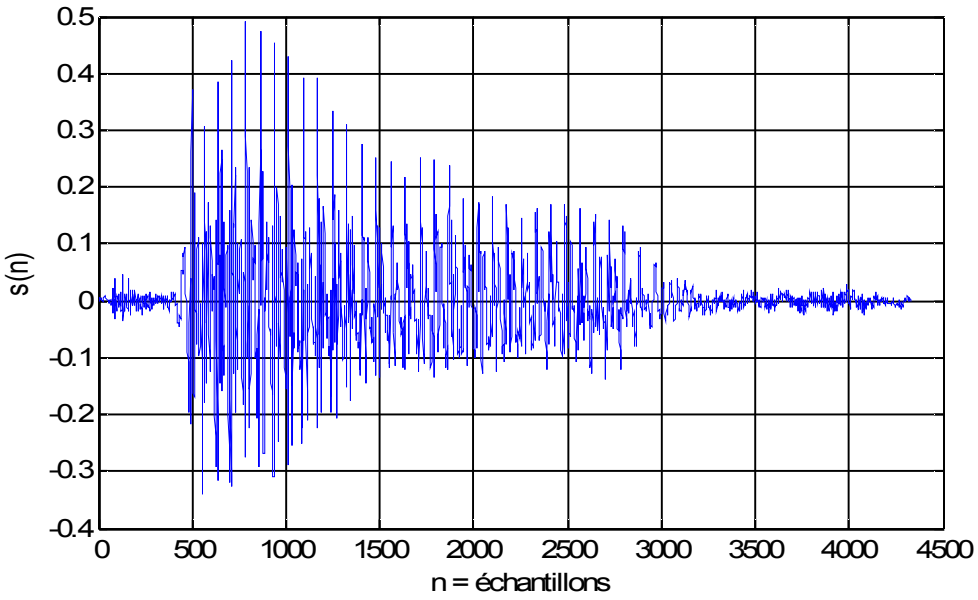


Figure 1.4 : Le signal de parole 'FIVE' en format « wav » chargé sur MATLAB par la commande 'wavread'

Le signal de parole chargé est 'FIVE', il est composé de k=4329 échantillons.

Le signal obtenu sur MATLAB est un vecteur, la segmentation de celui-ci exige qu'on le décompose en trames, la longueur de la trame que nous avons choisie est 240 échantillons par analogie au standard FS1016, cette décomposition est réalisée en transformant le vecteur obtenu par la commande 'wavread' en une matrice 240xL qui sera appelée $snew$.

L est le nombre de trames du signal de parole, il est calculé en arrondissant la valeur de $\frac{k}{240}$ à l'entier qui y est supérieur. Le nombre de trames pour le signal 'out' est 57 trames pour k=13440 échantillons.

Donc, la ligne i de la matrice ainsi obtenue représente la $i-$ème trame du signal de parole. La figure 4.2 représente la 33ème trame du signal 'one'.

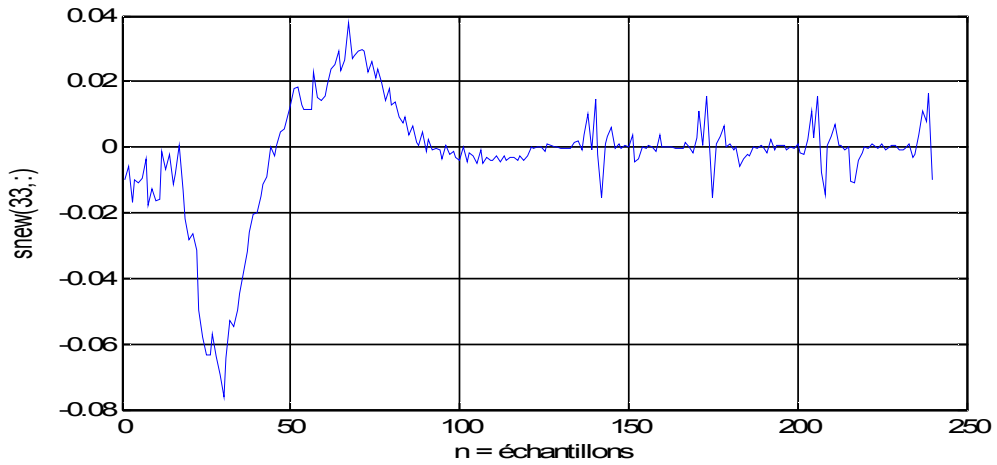


Figure 4.2 : 33ème trame du signal 'one'

La matrice *snew* obtenue est filtrée au préalable par un filtre passe-haut qui élimine les bruits basse-fréquences, puis par une fénétre de Hamming, qui est utilisée pour les corrections haute-fréquences.

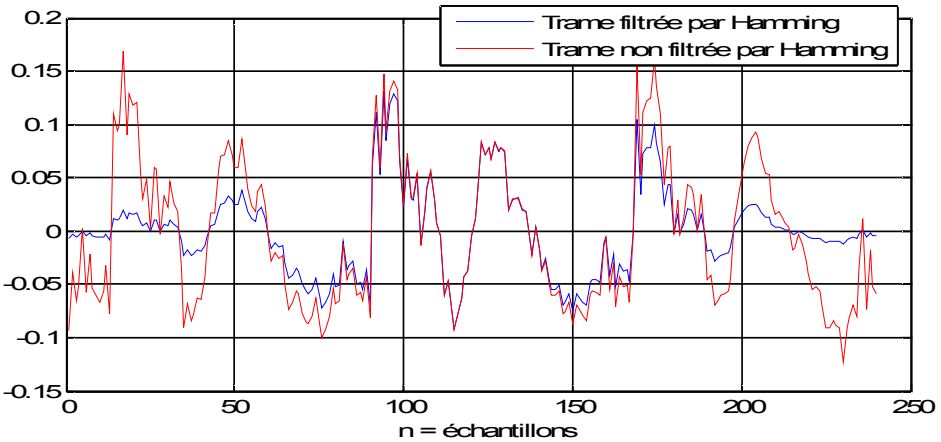


Figure 4.3 : Comparaison entre la 10ème trame filtrée et non filtrée par la fenetre de Hamming du signal 'FIVE'

D'après la figure 4.3, Les échantillons des limites de la trame ont des amplitudes réduites, cela est dû à l'application de la fenêtre de Hamming, avec l'application d'une fenêtre rectangulaire, ce léger problème n'apparaitrait pas mais le signal serait trop altéré dans le domaine fréquentiel à cause du lobe principal étroit et des lobes secondaire que présente celle-ci, ce qui justifie le choix de la fenêtre de Hamming.

Les caractéristiques fréquentielles sont données par la figure 4.4, on remarque plus clairement la correction fréquentielle, en effet le signal fenêtré est plus lisse et moins bruité après l'application de la fenêtre de Hamming.

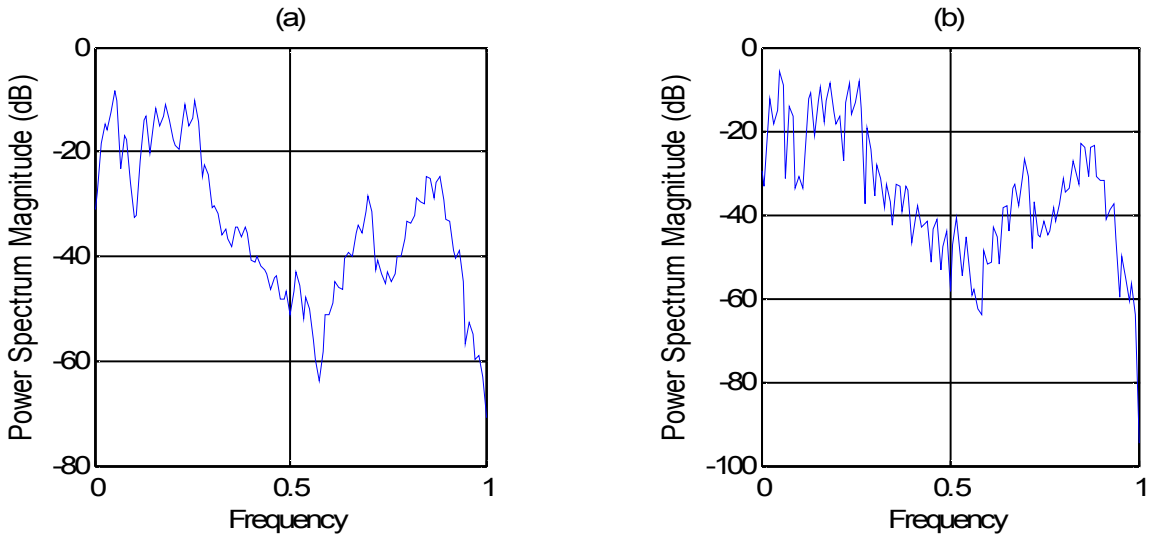


Figure 4.4 : Spectre de la 5ème trame du signal 'FIVE' : (a) :Filtrée par la fenêtre de Hamming, (b): Non filtrée par la fenêtre de Hamming

La matrice obtenue pour la segmentation comprend les trames à analyser, c'est-à-dire les trames utilisées pour tirer les coefficients de prédiction, il nous faut donc créer une autre matrice qui sera constituée de lignes représentant les trames du signal original utilisées dans la boucle d'analyse par synthèse dans la comparaison avec le signal synthétisé. Cette matrice sera appelée ssub, la ligne i de cette matrice contiendra les 120 derniers échantillons de la trame $i-1$ et les 120 premiers échantillon de la ligne du même rang de la matrice $snew$.

La matrice caractérisant le signal à tester par la boucle d'analyse par synthèse est constituée de lignes dont le nombre est le nombre de trames, ces lignes doivent être divisées en

quatre sous-trames chacune pour pouvoir tirer le gain et l'indice du dictionnaire, en effet, cette opération s'applique sur des sous-trames. La démarche que nous avons suivie est basée sur la représentation de chaque ligne de la matrice par quatre vecteur de longueur 60 échantillons, ils sont désignés par :

$subj(i,:)$: pour chaque valeur de i, $subj$ représente la $j-$ème sous-trame de la $i-$ème trame.

La figure 4.5 représente la 4ème sous-trame de la 23ème trame du signal 'one' :

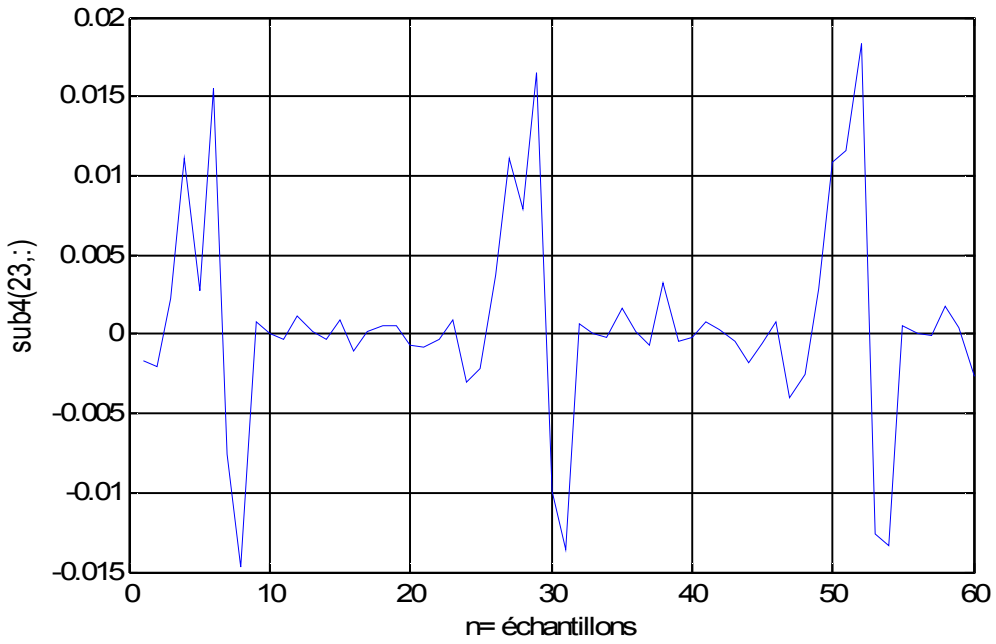


Figure 4.5 : La 4ème sous-trame de la 23ème trame pour le signal 'one'

La figure 4.6 représente le schéma synoptique des programmes MATLAB utilisés pour le fenêtrage et la segmentation.

Le programme *segmt_ham.m* permet de filtrer le signal obtenu par *'wavread'* par un filtre passe-bas, créer les trames et leur appliquer une fenêtre de Hamming. Le programme *sub_fram.m* crée les sous-trames pour l'étape de l'analyse par synthèse.

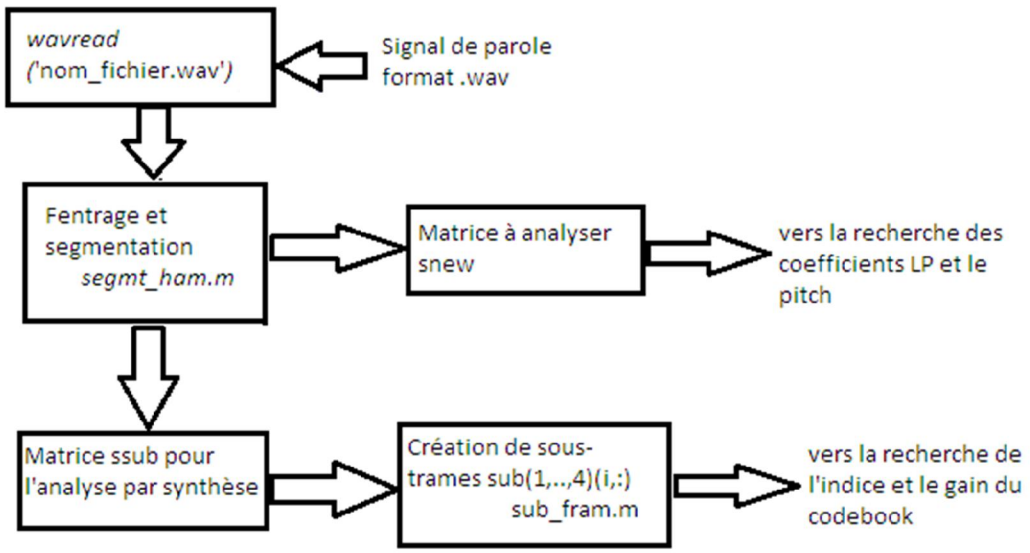


Figure 4.6 : Schéma synoptique des codes MATLAB du fenêtrage et la segmentation et la création des sous-trames

IV.3.Estimation des coefficients LP :

Après la segmentation et le fenêtrage du signal de parole, nous passons à l'analyse du signal issu de l'étape précédente qui est désignée dans notre implémentation par la matrice $snew$.

Le but de l'analyse est d'extraire les coefficients de prédiction LP pour chaque trame du signal de parole, c-à-d pour chaque ligne de notre matrice $snew$.Comme il a été évoqué dans le chapitre II, ces coefficients peuvent être obtenus par une analyse à court terme sur la trame en question, l'équation découlant de cette analyse est donnée par :

$$\begin{pmatrix} R_s[0] & R_s[1] & \cdots & R_s[M-1] \\ R_s[1] & R_s[0] & \cdots & \vdots \\ \vdots & \vdots & \ddots & R_s[1] \\ R_s[M-1] & R_s[M-2] & \cdots & R_s[0] \end{pmatrix} [a_1, a_2, \dots, a_M]^{\mathbf{T}} = -[R_s[0], R_s[1], \dots, R_s[M-1]]^{\mathbf{T}}$$

L'objectif de cette partie de l'implémentation est divisé en deux principales parties : la première consiste en le calcul des M autocorrélations correspondantes aux M retards de

l'équation précédente, la deuxième en la résolution de celle-ci. La figure 4.7 illustre le schéma synoptique des programmes MATLAB qui réalisent ces deux opérations.

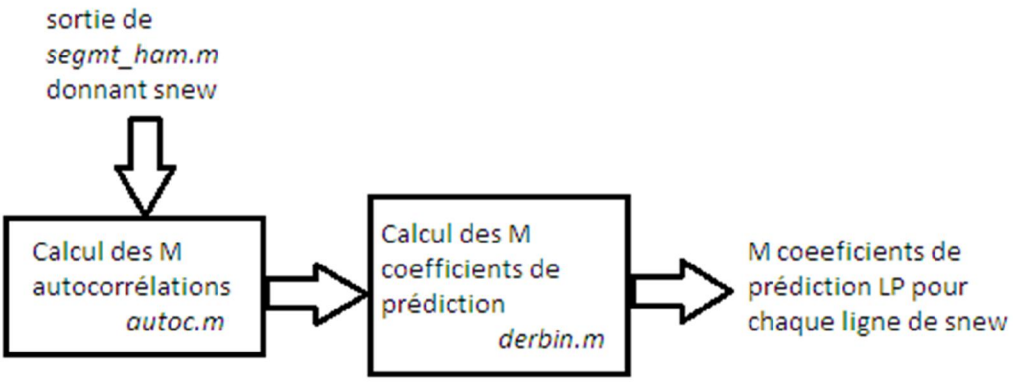


Figure 4.7 : Schéma synoptique de la détermination des coefficients LP.

Le programme *autoc.m* calcule les *M* autocorrelations qui représentent les entrées du programme *derbin.m* qui est une implémentation sur MATLAB de l'algorithme de Levinson-Durbin celui-ci présente comme sorties les *M* coefficients de prédiction LP. Le tableau 4.1 présente les coefficients LP de 3 trames du signal ''FIVE'', les premiers coefficients pour chaque trame ou ligne de *snew* sont toujours égaux à un, ils ne figurent cependant pas sur ce tableau.

Les coefficients LSF peuvent être retrouvés sur MATLAB en utilisant la commande '*poly2lsf*'qui a pour entrées les coefficients de prédiction, pour notre implémentation, le résultat de cette commande nous fournira une matrice dont les lignes sont les coefficients LSF pour chaque trame du signal, le tableau 4.2 montre les coefficients LSF correspondant aux LPs du tableau 4.1

trame	Les coefficients LP's
Snew(1, :)	$a(1) = -0.5096, a(2) = 0.5542, a(3) = -0.1812, a(4) = 0.3334$ $a(5) = -0.1368, a(6) = 0.1530, a(7) = -0.0091, a(8) = 0.4177$ $a(9) = -0.3853, a(10) = 0.3274$
Snew(9, :)	$a(1) = -1.1605, a(2) = -0.0216, a(3) = 0.2281, a(4) = 0.5623$ $a(5) = -0.7798, a(6) = 0.1177, a(7) = 0.0071, a(8) = 0.6225$ $a(9) = -0.6350, a(10) = 0.1644$
Snew(18, :)	$a(1) = -0.6421, a(2) = 0.2335, a(3) = -0.4727, a(4) = 0.1943$ $a(5) = -0.4899, a(6) = 0.2695, a(7) = -0.1367, a(8) = 0.5644$ $a(9) = -0.5194, a(10) = 0.2079$

Tableau 4.1: Coefficients de prédiction pour des trames du signal ''FIVE''

trame	Les coefficients LSF
Snew(1, :)	$LSF(1) = 0.3792, LSF(2) = 0.5676, LSF(3) = 0.8699,$ $LSF(4) = 1.1213$ $LSF(5) = 1.3249, LSF(6) = 1.5460, LSF(7) = 1.9912,$ $LSF(8) = 2.0617 \;\; LSF(9) = 2.5288, LSF(10) = 2.7203$
Snew(9, :)	$LSF(1) = 0.1679, LSF(2) = 0.3563, LSF(3) = 0.5872,$ $LSF(4) = 0.9264$ $LSF(5) = 1.0722, LSF(6) = 1.4751, LSF(7) = 2.0242,$ $LSF(8) = 2.1827, \;\; LSF(9) = 2.6072, LSF(10) = 2.6827$
Snew(18, :)	$LSF(1) = 0.1557, LSF(2) = 0.3626, LSF(3) = 0.7706,$ $LSF(4) = 1.1842$ $LSF(5) = 1.3019, LSF(6) = 1.5720, LSF(7) = 1.9953,$ $LSF(8) = 2.0899 \;\; LSF(9) = 2.5720, LSF(10) = 2.6676$

Tableau 4.2: Paramètres LSF pour des trames du signal 'FIVE'

IV.4. Estimation du pitch par boucle ouverte :

La détermination du pitch représente l'une des tâches les plus délicate de l'algorithme CELP, une représentation précise des paramètres du filtre LTP nous impose utilisation d'un dictionnaire adaptative, cependant le but de notre mémoire est l'implémentation d'un codeur CELP général, pour ces deux principales raison nous avant opté pour une détermination du 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(4.8). L'erreur à minimiser est obtenue pour chaque ligne analysée de la matrice snew, ce qui équivaut à déterminer un pitch et un gain du filtre LTP pour chaque nouvelle trame du signal originale

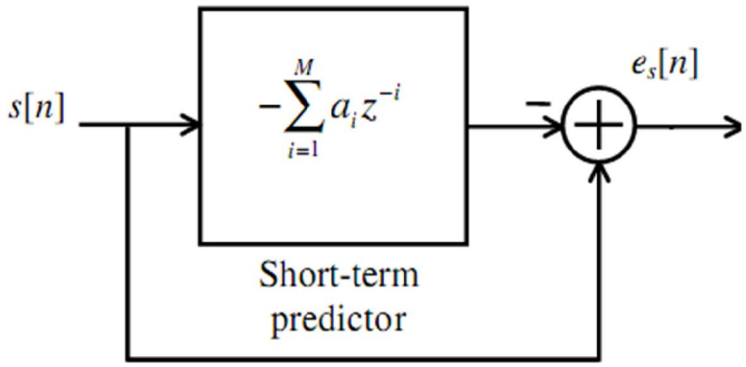


Figure 4.8 : L'erreur à minimiser pour la détection du pitch issue du filtre prédicteur à court terme

Le filtre prédicteur présente à sa sortie un signal prédit, ses coefficients sont les coefficients LP obtenus de l'analyse à court terme, ces coefficients sont appliqués pour chaque trame du signal, c'est-à-dire pour chaque ligne de la matrice $snew$, le système de la figure 4.8 engendre une erreur qui est la différence entre la matrice $snew$ et la matrice du signal prédit f, l'erreur de prédiction est donc représentée par une matrice dont le nombre de lignes est le nombre de trames, la figure 4.9 présente la 50ème trame du signal prédit comparée à la trame du même ordre du signal original ainsi que la trame de l'erreur de prédiction.

L'erreur de prédiction sera ensuite minimisée en minimisant l'équation (3.45) du chapitre 3, le but de cette minimisation est la détermination du pitch et du gain du filtre LTP, cela en cherchant le pitch qui donnent la plus petite erreur de prédiction, il appartient à un intervalle $[Tmin, Tmax]$. puis le remplacer dans l'équation (3.44) pour retrouver le gain correspondant.

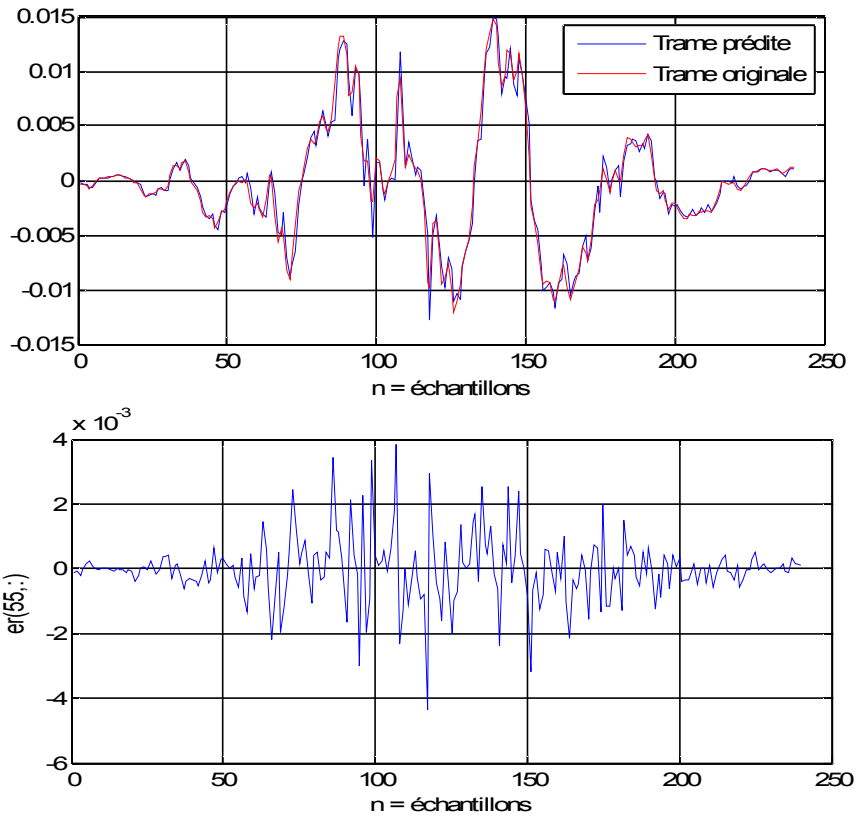


Figure 4.9 : (a) Comparaison entre la 55ème trame du sinal original 'one' avec le signal prédit (b) Erreur de prédiction de la 50ème trame du signal ''one''

Pour implémenter cette partie, nous avons considérer notre erreur comme une matrice ou chaque ligne représente une trame, la figure 4.10 présente l'organigramme pour la recherche des paramètres LTP pour une trame bien précise, l'intervalle des pitch que nous avons choisi est [20,150], cela est en relation avec les caractéristiques temporelles et fréquentielles de la parole ou le pitch ne dépasse pas les bornes de cet intervalle.

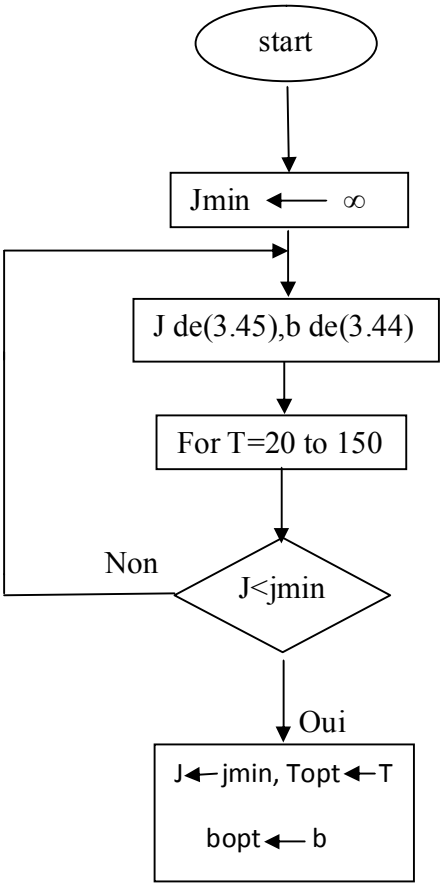


Figure 4.10 : Organigramme de l'estimation du pitch et le gain du filtre LTP pour une trame

Pour les programmes MATLAB, nous avons crée deux fonctions : *err_pitch.m* qui calcul l'erreur de prédiction pour chaque trame du signal original et la présente à sa sortie comme une matrice, celle-ci est l'entrée d'une seconde fonction *pitch.m* qui calcul le pitch et le gain du filtre LTP en minimisant chaque ligne de la matrice de l'erreur et les présente sous forme de vecteurs de dimension égale au nombre de trame, donc le pitch T[i] et le gain b[i] sont les paramètre du filtre LTP pour la $i-$ ème trame pour le signal original.

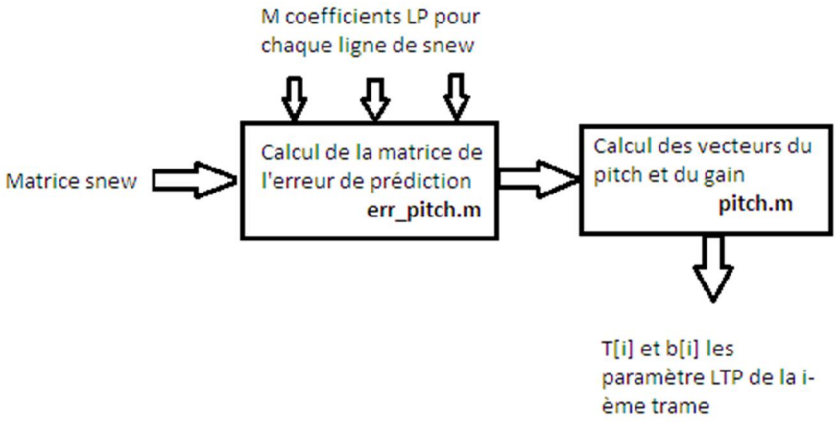


Figure 4.11 : Schéma synoptique des codes MATLAB pour la détection du pitch

Le tableau 4.2 donne les pitch et les gains du filtre LTP pour différenets trames de deux signaux différents :

signal	trame	Pitch Tf	Gain bf
‘’FIVE’’	Snew(4, :) Snew(13, :) Snew(18, :)	148 78 32	0.3752 0.1230 0.1693
‘’one’’	Snew(4, :) Snew(13, :) Snew(18, :)	89 128 68	0.1181 -0.0381 0.4981

Tableau 4.3 : Différents pitch’s et gains pour les signaux ‘’FIVE’’ et ‘’one.

IV.5.Analyse par synthèse :

Nous avons réalisé le programme de l’implémentation de la boucle de l’analyse par synthèse en réalisant différents codes MATLAB mais malheureusement nous n’avons pas pu le tester à cause l’indisponibilité d’un codebook, dans cette section nous allons expliquer la conception des filtres, la minimisation de l’erreur perceptive et l’intégralité de la boucle.

Pour l’implémentation des filtres de cette boucle, nous avons réalisé deux fonctions MATLAB, la première est *LTP.m* elle définit le filtre de synthèse du pitch Zero-Input, elle a pour entrée la période du pitch, le gain du filtre, l’état de la séquence précédente, celle-ci est un

vecteur dont la longueur est la période du pitch. La deuxième fonction est *synth_formant.m* , elle représente le filtre de synthèse de formant modifié en Zero-Input, elle a pour entrées les coefficients de prédiction de chaque trame, l'ordre de prédiction et la séquence de l'état précédent qui est un vecteur dont la longueur est l'ordre de prédiction. La figure 4.12 représente ces deux filtres.

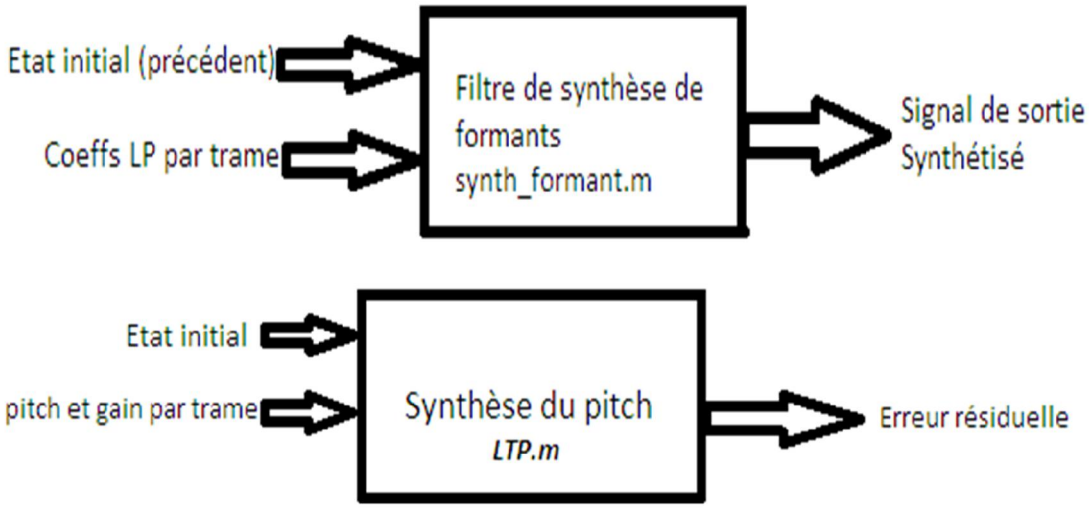


Figure 4.12 : Les fonctions LTP.m et synth_formant.m utilisée dans l'analyse par synthèse

Nous avons créé une autre fonction MATLAB permettant de minimiser l'erreur perceptive pour tirer les indices du dictionnaire d'excitation est la fonction *minimis.m* , l'algorithme est basé sur les équations (3.34) et (3.36) du chapitre 3. Cette fonction a pour entrées les sous-trames du signal original et les sous-trames du signal synthétisé et pour sorties l'indices du meilleur code-vecteur et son gain approprié pour chaque sous-trame, elle contient les deux fonctions *LTP.m* et *synth_formant.m* , la figure 4.13 montre l'organigramme de la fonction *minimis.m*

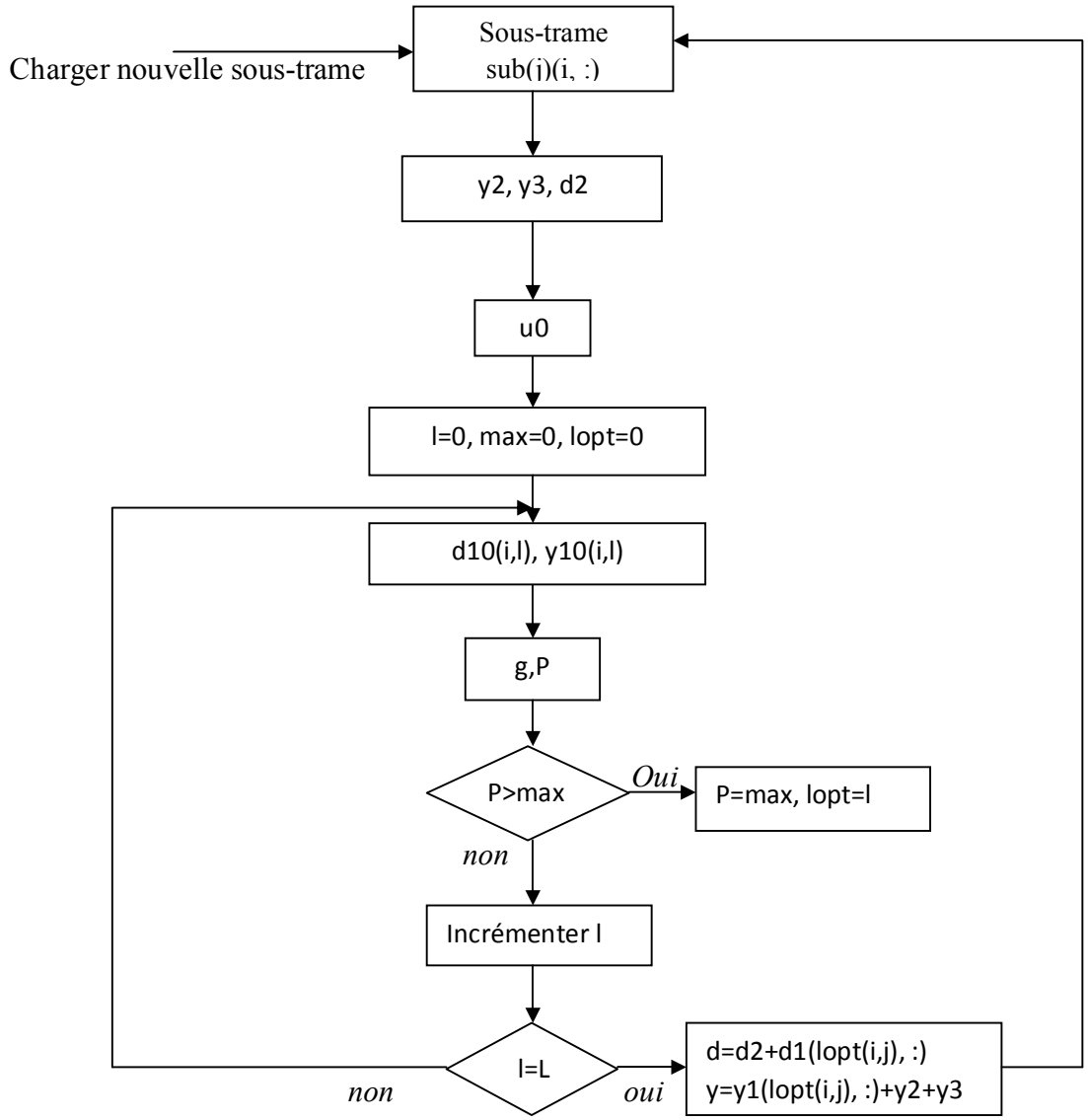


Figure 4.13 : Organigramme du programme de recherche du meilleur code-vecteur minimis.m

Le programme *minimis.m* prend en compte la méthode du Zero-State Zero-Input et l'applique sur les filtres de synthèse du pitch et de formants modélisés par les fonctions *LTP.m* et *synth_formant.m* en manipulant leurs état précédents (entrées précédentes), il présente pour chaque sous-trame sub(j)(i, :) un gain et indice optimaux tel que :

$lopt(i,j)$: est l'indice optimal de la $j-$ème sous-trame de la $i-$ème trame.

$gopt(i,j)$: est le gain optimal de la $j-$ ème sous-trame de la $i-$ ème trame.

Pour prendre en considération tous le signal, un programme globale *analyse_synthese.m* englobant toutes les fonctions décrites précédemment a été élaboré, il présente à sa sortie tous les gain et indices présentés sous formes de deux matrices de dimensions m×4, ou m est le nombre total de trames pour le signal de parole, la figure 4.14 montre le schéma synoptique de ce programme.

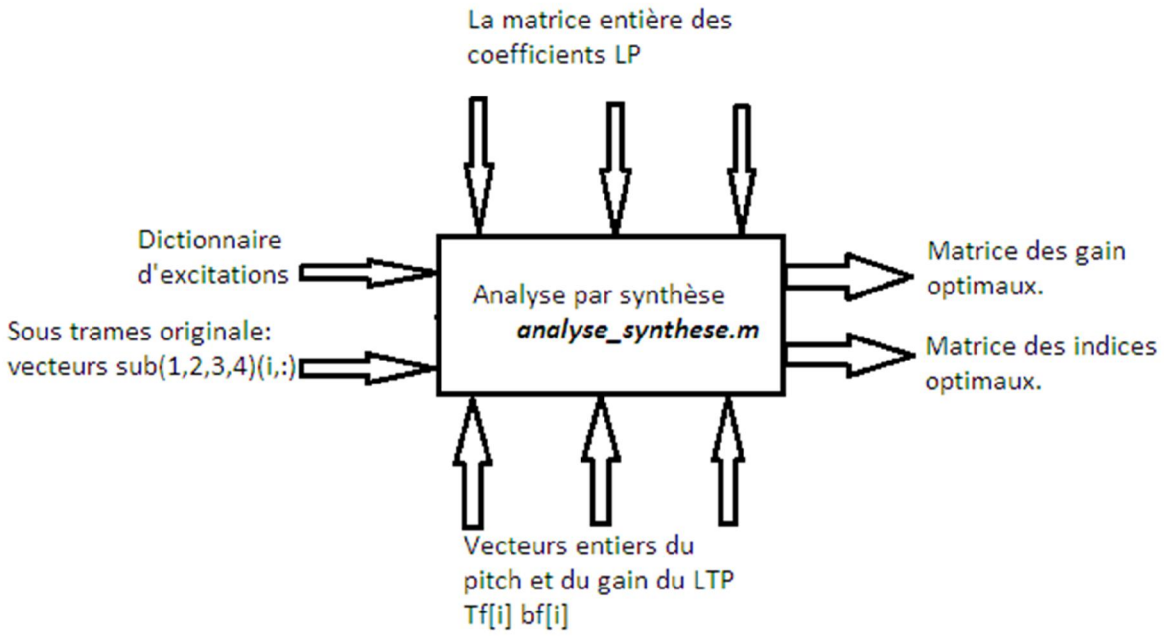


Figure 4.14 : Schéma synoptique du code MATLAB d'analyse par synthèse

IV.6.Simulation par l'application FS1016 sur MATLAB :

Dans cette partie de l'implémentation, nous ferons une simulation d'un standard FS1016 et mesurer les performances de celui-ci par une évaluation objective qui est la SNR segmental, cela avec l'utilisation d'une application sur MATLAB qui n'est rien d'autre que l'implémentation du FS1016 réalisée par A.Spanias et T.Painter en 1999[6].

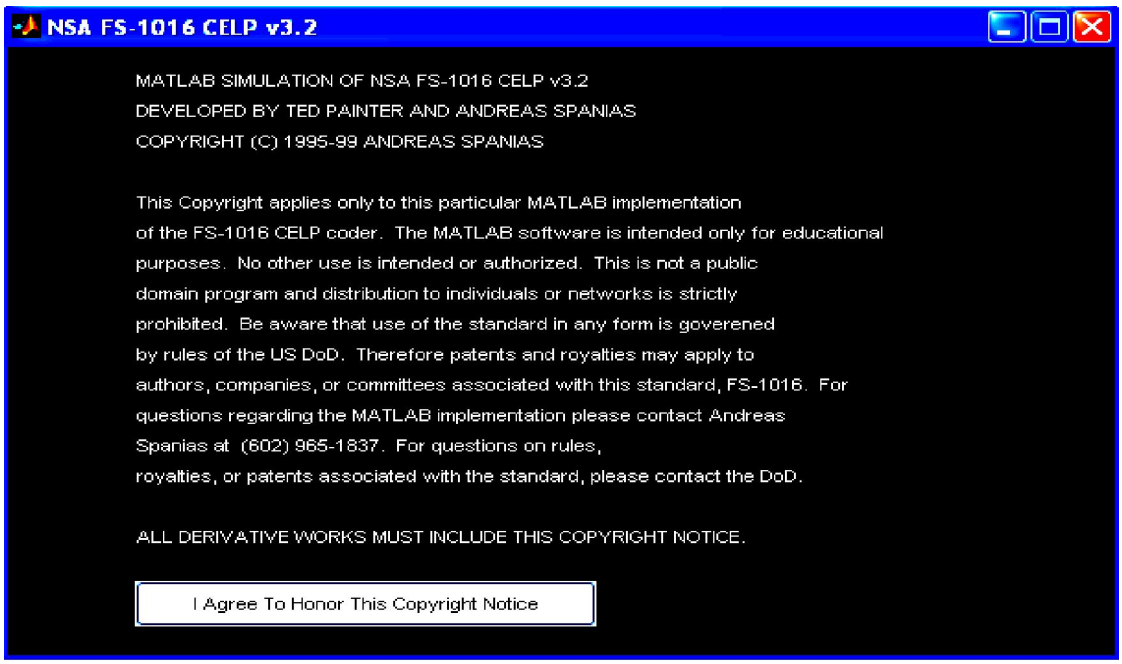
IV.6.1.Présentation :

L'application FS1016 est une interface développée par A.Spanias et T.Painter de 1994 à1999 pour des objectifs éducatifs, elle consiste en une implémentation du codeur standard FS1016 sur MATLAB, le répertoire de l'interface FS1016 est constitué de :

CPYRIGHT.TXT : Copyright notice
.M : CELP 3.2 simulation .m files. fs1016.m.
.DAT: Table des valeurs

Le standard FS1016 pour MATLAB emploie des fichiers de paroles de type « wav ». Les signaux de parole d'entrée doivent être des signaux de 16 bit, et prélevés à 8khz. De même pour les signaux de parole de sortie produits par l'algorithme sont des signaux de 16 bit, échantillonnés à 8khz.

Nous commençons par changer le répertoire de travail MATLAB en allant vers le répertoire FS1016, puis nous lançons l'application sur MATLAB. Nous acceptons les conditions d'utilisation, nous choisissons un signal à traiter, il apparait alors une interface qui montre le programme en exécution trame par trame pour fournir à la fin un signal synthétisé, la figure 4.15 montre les étapes suivies pour la synthèse de la parole avec cette application très ergonomique.



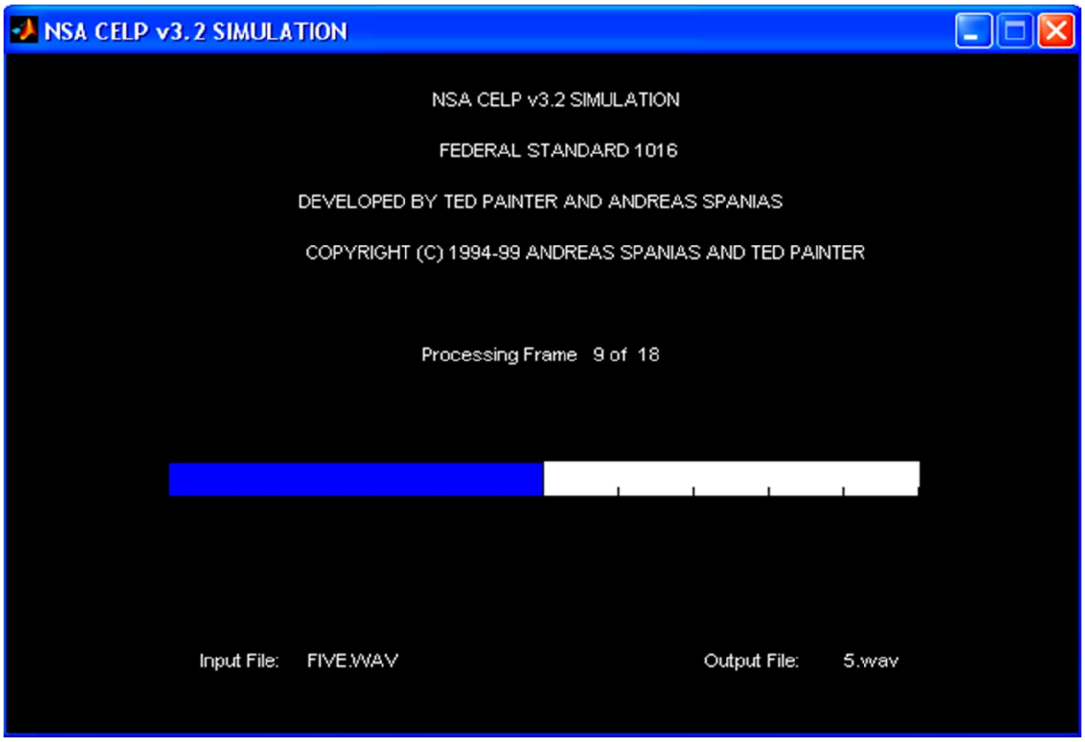


Figure 4.15 : Synthèse du signal 'FIVE' par l'application FS1016

Le signal synthétisé est nommé 5.wav, une fénétre apparait pour nous demander de nommer le signal de sortie.

IV.Comparaison entre le signal original et le signal synthétisé :

Après avoir synthétisé le signal avec l'application, on doit mesurer la qualité du signal synthétisé, la figure 4.16 montre le signal original et synthétisé, elle montre qu'il n'est pas entièrement similaire, le but de cette partie est de mesurer la qualité de la parole et les performance du codeur en mesurant le degrés de similitude entre les deux signaux par des mesures objectives comme il est illustré dans le tableau 4.4.

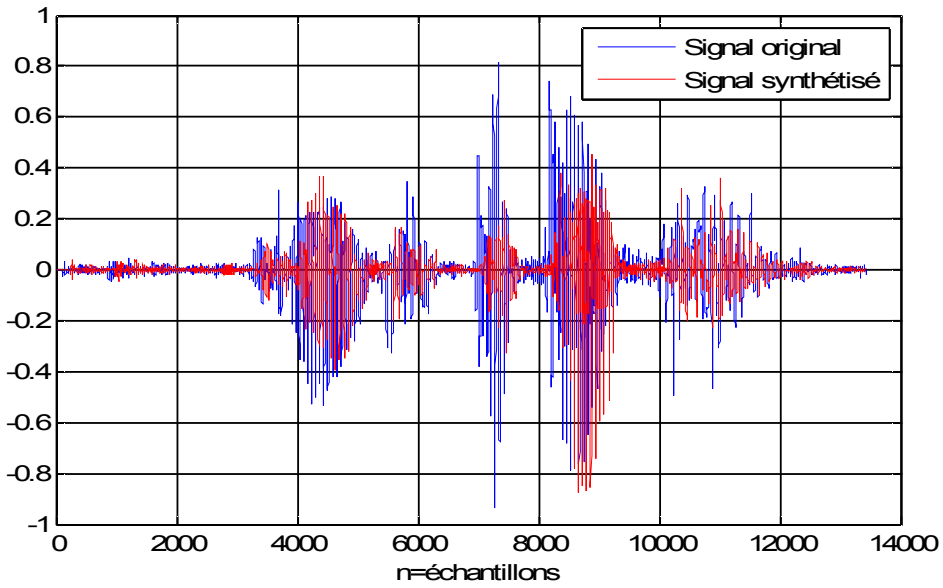


Figure 4.16 : Comparaison entre le signal original et synthétisé par le FS1016 du signal 'one'

Pour évaluer la qualité de la parole synthétisée, on a utilisé le rapport signal-sur-bruit segmental SNRseg, et le PESQ.

Le modèle PESQ a accès aux signaux de référence et dégradés, qu'il transforme et compare pour obtenir une note de qualité d'écoute[6][11]. La note PESQ est appliquée à une échelle de type MOS, sous forme d'un scalaire compris entre **-0.5** et **4.5**. Pour l'évaluation par le SNR segmental, nous avons réalisé un programme *SNRseg.m,* pour le modèle PSEQ, nous avons utilisé une interface d'évaluation sur MATLAB réalisée par J.L.H.Hansen et B.L.Pellom[11].

	SNRseg	PESQ
« one.wav »	-1.7906	3.6370
« five.wav »	-1.8457	2.9639

Tableau 4.4 : Evaluation des signaux 'one' et 'FIVE' par le SNR segmental et le PESQ

D'après les résultats obtenus, présentés dans le tableau ci-dessus et d'après les valeurs de PESQ, on peut constater que le codeur FS1016 est performant pour des milieux non bruités, en effet, les tests sont réalisés sur des signaux issus de milieux propres.

Nous ne pouvons pas juger à 100% les performances du codeur, d'une part il faut le tester pour différents environnements (différentes conditions) et d'autre part, la mesure subjective reste le moyen le plus adéquat pour juger ces performances.

IV.7.Conclusion :

Ce chapitre nous a permis de renforcer les notions que nous avons nouvellement acquise dans les chapitres précédents, cela grâce l'implémentation des principaux blocs du codeur CELP, de celle-ci nous avons obtenu des résultats et nous avons pu comprendre concrètement le fonctionnement de l'algorithme, seuls le gain et l'indice du dictionnaire n'ont pas pu être retrouvés, ce qui nous a poussés à utiliser l'application FS1016 pour évaluer la qualité de la parole synthétisé.

Conclusion Générale

Grâce à ce projet de fin d'étude, nous nous sommes familiarisé avec un domaine très important en électronique qu'est le traitement de la parole, bien que les connaissances et les notions acquises jusqu'ici ne soient que superficielles, nous avons pu comprendre le fonctionnement de différents codeurs de parole et avoir une idée générale sur les différentes démarches que les spécialistes ont empruntées depuis plusieurs décennies pour satisfaire le compromis débit et qualité de la parole.

Le travail réalisé dans ce mémoire consistait à étudier différentes techniques de codages et en particulier le Code Excited Linear Prediction (CELP), cette tache a pu aboutir grâce à une étude brève des différents codeurs le précédant, c'était l'objectif du premier chapitre. Le deuxième chapitre reposait sur l'étude de la technique de prédiction linéaire qui est présente dans pratiquement tous les codeurs de parole moderne et notamment le CELP, 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 presque exhaustive de l'algorithme CELP général et une description concise du standard FS1016 dont les performances ont été évalué dans le chapitre quatre, dans celui-ci nous avons réalisé l'implémentation de l'intégralité des blocs du codeur CELP, cette implémentation visait l'extraction de tous les paramètres caractérisant le signal au biais de cet algorithme.

La partie analyse par synthèse n'a pas pu être testée à cause de l'indisponibilité du dictionnaire du CELP, néanmoins, nous avons réalisé l'algorithme de recherche de l'indice et du gain du codebook. Ce léger conte-temps nous a conduits à évaluer différents signaux synthétisés par l'application FS1016 sur MATLAB.

Nous nous fixons comme perspectives en tant qu'étudiants pour la suite de nos études en post-graduation l'implémentation de l'algorithme CELP sur microprocesseurs ainsi que celle d'un codeur CELP ayant un dictionnaire adaptatif, en l'occurrence, le FS1016 sur MATLAB.

Références Bibliographiques

[1]. R.Boite et M.Kunt,, '' Traitement de la parole''. *Presses polytechniques romandes* .1987.

[2]. 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.

[3]. G.Madre,''Application de la transformée en nombre entier à l'étude et au développement d'un codeur de parole pour transmission sur réseaux IP''. *Thèse de doctorat ès sciences, Université de Bretagne occidentale*. Octobre 2004.

[4]. A.Bouafia et N.Belgroune,''Amélioration du codec G.729 par entrelacement de trames''. *Projet de fin d'études E.N.P*. 2006.

[5]. T.Chmayssani,''Modulations sur les canaux vocodés''. Thèse de doctorat ès sciences, Université Paris EST. 2010.

[6]. W.C.Chu,''Speech coding algorithms, foundation and evolution of standardized coders''. *John Wiley & sons, Inc., Hooken, New Jersey*. 2003.

[7]. 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.

[8]. M.Djamah, M.Boudraa, B.Boudraa et M.Bouzid,''Qauntification adaptative des coefficients LSF pour le codage de la parole à bas débit''. *Laboratoire de communication parlée et traitement du signal, Faculté du génie électrique et d'informatique USTHB.*

[9]. M.Fedila et F.Z.Merabet,''Mise au point d'un codeur CELP FS1016 fonctionnant 4.8Kb/s''.*Mini projet, post graduation : Communication parlée , département d'électronique USTHB*.2007.

[10]. J.H.Hansen and B.L.Pellom, ‘’An effective quality evaluation protocol for speech enhancement algorithm’’ *Robust speech processing laboratory, Duke University.*

[11]. M.R.Shroeder and B.S.Atal,’’Code excited linear prediction(CELP): High quality speech at very low bit rates’’, *Proc. ICASSP-85*, Apr. 1985

[12]. A.V.Oppenheim and R.W.Schafer,’’Digital Signal Processing’’,*2nd Edition Prentice Hall, Upper Saddle River, New Jersey 07558*,1999

[13]. K.N.Ramamurthy and A.S.Spanias,’’MATLAB software of the Code Exited Linear Prediction algorithms, the federal FS1016’’. *Synthesis lecture on algorithms and software engineering, Morgan & Claypool publisher series*. 2009.

[14]. 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.

[15]. P.Kroon and E.F.Deprettere,’’A class of analysis-by-synthesis predictive coders for high quality speech coding at rates between 4.8 and 16Kb/s’’, *IEEE Journal on selected area in communication. Vol. 6, No. 2*, February 1988.

[16]. T.V.Ramabadran and C.D.Lueck,’’Complexity reduction of CELP speech coders Through the use of phase information’’*, IEEE transaction on Communications vol.42 NO.2/3/4*.1994.

[17]. M.Mauc et G.Baudoin,’’Codeur CELP à complexité réduite’’. *Journal de la physique VI, Colloque C1, supplément au journal de la physique III , volume 2, ESIEE département de signaux et télécommunications*.1992.

[18]. M.Djamah, M.Boudraa, B.Boudraa et M.Bouzid,’’Un logiciel de codage de la parole basé sur le FS1016’’. *Laboratoire de communication parlée et traitement du signal, Faculté du génie électrique USTHB*.

[19]. G.Baudoin,’’Codage de la parole à bas et très bas débit, transformation de la voix’’.*Mémoire d’habilitation à diriger des recherches, Université de Marne La Vallée*.2000.