REPUPLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique


## Ecole Nationale Polytechnique

D. E. R. Génie Electrique & Informatique

Département d'Electronique


**Thèse**

**EN VUE DE L'OBTENTION**

**DU GRADE DE MAGISTER**

**EN ELECTRONIQUE APPLIQUEE**

Option : Télécommunications



Thème

# Conception et réalisation d'un codeur/décodeur de la parole à bande étroite (300 - 3400 Hz), à 16 kbits/s et à faible retard (< 5 ms)


Etudié par : **DJEDDOU MUSTAPHA**

Ingénieur d'Etat en Electronique


Soutenue publiquement le : 11/02/1997. Devant le jury composé de :

| | | |
|---|---|---|
| Président : | | |
| Mr BERKANI D. | Maître de conférences | ENP |
| Rapporteurs : | | |
| Mlle GUERTI M. | Maître de conférences | ENP |
| Mr HALIMI M. | Chargé de recherche | CDTA |
| Examinateurs : | | |
| Mr GUESSOUM A. | Maître de conférences | Univ. de Blida |
| Mr BELOUCHRANI A. | Docteur au département d'électronique | ENP |
| Mr BOUDRAA B. | Chargé de recherche | USTHB |

# Résumé

Un codeur de parole "LD-CELP" de débit 16 Kb/s et possédant un retard de codage inférieur à 2 ms a été réalisé. Cette réalisation a nécessité la prise en compte d'une taille réduite du vecteur d'analyse (5 échantillons) à traiter ainsi qu'une adaptation régressive du prédicteur LPC et du gain d'excitation. Le prédicteur pitch dans le CELP conventionnel est supprimé à cause de sa sensibilité aux erreurs du canal. Pour compenser la perte de performance (surtout pour un signal de parole prononcé par un locuteur féminin), l'ordre de prédiction LPC est augmenté de 10 à 50. Cette modification permet au codeur d'être moins spécifique pour les signaux de parole.

Deux types de dictionnaire sont utilisés pour la QV (Quantification Vectorielle) de l'excitation :

- stochastique ;
- algébrique.

Le premier a été conçu par apprentissage (similaire à l'algorithme LBG) optimisé en boucle fermée en utilisant une base de donnée de signaux de parole.

le deuxième a été conçu par l'utilisation d'un générateur de code ternaire. Une réduction significative de complexité a été obtenue en utilisant les caractéristiques du vecteur d'excitation ternaire.

Les mesures objectives et subjectives montrent que la qualité de la parole synthétisée est de haute qualité pour les deux types d'excitations.

Le travail a été extrapolé pour avoir un codeur de parole large bande (50 - 7000 Hz) à un faible retard (< 1 ms) à un débit de 32 Kb/s. Le signal de parole décodé possède une qualité transparente, mais la complexité de calcul se trouve pratiquement doublée.

Le codeur LD-CELP à 16 Kb/s est utilisé dans les réseaux téléphoniques et peut être utilisé comme programme de compression de signaux de parole dans les supports de stockage.

# Abstract

A LD-CELP 16 kb/s speech coder with a delay less than 2 ms was designed. It's achieved by taking a small vector frame (5 samples) and by making the LPC predictor and the gain excitation backward adaptive. The pitch predictor in conventional CELP coder is not used due to its sensitivity to channel errors, and the resulting performance loss (especially for female speech signals) is compensated for by increasing the LPC predictor order from 10 to 50. This modification makes the encoder less specific to speech signals.

Two kinds of codebooks are used for VQ (vector quantization) of excitation :

- the first one is stochastic, designed by using training speech data base and codebook design algorithm optimized in closed loop;
- the second one is algebraic, generated by ternary code generator. Significant reduction in complexity is achieved by using characteristics of ternary vector excitation.

Objective and subjective measures show that the coder achieves high quality with the two kinds of excitation.

We extrapolate our work to obtain a wide band (50 - 7000 Hz) speech coder with a coding delay less than 1 ms at 32 kb/s bit rate. The decoded speech has a transparent quality, but the complexity is doubled.

The 16 kb/s LD-CELP coder is used in telephonic networks and may be used as compression program for speech signals on storage supports.

# ملخص

كان اهتمامنا منصبا فى هذه الاطروحة فى انشاء مرمز للكلام (الحزمة الهاتفية) LD-CELP بسرعة تدفق 16 كيلوبايت/الثانية، مع مهلة تأخر لاتتجاوز 2 ميلي ثانية. للتمكن من ذلك، كان علينا اتخاذ : طول الشعاع التحليلي ذو 5 عناصر، مع جعل مرشح تنبؤ LPC و طويلة شعاع التحريض يعملان بطريقة ملائمة الى الوراء (backward). منبىء مقام الصوت (pitch) فى نظام الترميز (CELP) العادى تم حذفه نظرا لشدة حساسيته لاخطاء قنوات الاتصال، لتعويض نقص فعالية المرمز (خاصة فى الكلام النسوى)، تم رفع درجة التنبؤ مرشح التركيب من 10 الى 50 . هذا التعديل جعل من المرمز اقل خاصية للكلام.

نوعان من القواميس قد تم استعمالهما لعملية التكميم الشعاعى للتحريض، الاول : إحصائى منشأ باستعمال خوارزمية انشاء القواميس مشابه لخوارزمية (LBG) المعروفة. الثانى : منشأ بمولد للرموز ثلاثى. خاصيات المولد الثلاثى سمحت بتخفيض معتبر لدرجة التعقيد. القياسات اكدت النوعية العالية للكلام المعالج المحصل عليه لكلا نوعي التحريض المستعملين .

وقد تم توسيع العمل بانشاء مرمز للكلام ذو حزمة موسعة (50-7000 هرتز) بسرعة تدفق 32 كيلوبايت في الثانية، مع مهلة تأخر لاتتجاوز 1 ميلي ثانية. الكلام المحصل عليها ذو نوعية عالية لكن مع درجة تعقيد مظاعفة.

المرمز المحصل عليه يستعمل في [illegible] مع امكانية استعماله كبرنامج لتشفير الاشارات الكلام في وسائل التخزين الالكترونية.

# REMERCIEMENTS

Je tiens à exprimer tous mes remerciements :
Au Docteur M. Halimi, mon Directeur de thèse au Centre de Développement des Technologies Avancées pour son aide inestimable. J'ai ainsi pu apprécier ses qualités humaines, sa bienveillance et la pertinence de ses remarques scientifiques. Je n'omettrai certainement pas de mentionner l'excellente ambiance qui a rendu mon séjour très agréable et fructueux.
Au Docteur M. Guerti, mon Directeur de thèse à l'ENP, pour ses conseils, son soutien et son sens professionnel. Je lui suis très reconnaissant pour la confiance et l'intérêt qu'elle ma constamment accordée.
Au Docteur D. Berkani d'avoir accepté de présider le jury de cette thèse et aussi à sa manière originale d'enseigner (durant l'année théorique) qui a laissé certainement une marque indélébile sur mon orientation vers le codage de la parole.
Aux membres de jury pour avoir accepté de juger mon modeste travail.

Je tiens à exprimer ma profonde gratitude à Monsieur Ahmed Sid le chef de centre de calcul de l'EMP et Monsieur Benaissa A. ainsi que Monsieur Aissa N. pour leurs précieuses aides.
Que Monsieur Kessal M. et Monsieur Zaknoune R. trouvent ici ma profonde reconnaissance pour leurs aides et conseils.
Finalement, je remercie toute ma famille pour son soutien morale et pour m'avoir toujours préparé les meilleures conditions de vie

*A Yacine*

# Tables des Matières

# Liste des Figures

# Liste des tableaux

# Liste des Acronymes

| | | Première apparition dans les pages |
|---|---|---|
| PCM | Pulse Coded Modulation | 1 |
| ADPCM | Adaptive Delta Pulse Coded Modulation | 1 |
| CELP | Code Excited Linear Prediction | 1 |
| MPLPC | Multipulse Linear Predictive Coding | 1 |
| APC | Adaptive Predictive Coding | 1 |
| ATC | Adaptive Transform Coding | 1 |
| SBC | Sub Band Coding | 1 |
| LD CELP | Low Delay Code Excited Linear Prediction | 2 |
| MOS | Mean Opinion Score | 14 |
| LPC | Linear Prediction Coding | 17 |
| RSB | Rapport Signal à Bruit | 17 |
| RSB seg. | Rapport Signal à Bruit segmental | 18 |
| RMS | Root Mean Square | 18 |
| DRT | Diagnostic Rhyme Test | 19 |
| DAM | Diagnostic Acceptability Measure | 19 |
| QV | Quantification Vectorielle | 19 |
| VQ | Vector Quantization | 19 |
| GLA | Generalized Lloyd Algorithm | 24 |
| ARMA | Auto Régressif à Moyenne Ajustée | 28 |
| AR | Auto Régressif | 29 |
| MA | Moyenne ajustée | 29 |
| LP | Linear Prediction | 34 |
| ZIR | Zero Input Response | 65 |
| MSE | Mean Squared Error | 67 |
| LBG | LINDE BUZO GRAY (Algorithme) | 74 |

# Introduction Générale

Avec l'avènement de nouvelles applications multimédia et l'utilisation accrue d'environnement à bande limitée telle que le canal téléphonique, la radio, les liaisons satellites et celle des supports de stockage tels les CD ROM, la réduction du débit d'information, lorsqu'on transmet ou on stocke un signal de parole, présente un intérêt stratégique. Le signal restitué ne doit pas être distordu par l'opération de transmission ou de stockage suite à la compression et le coût de l'opération de traitement doit rester raisonnable. Pour cela, la recherche sur les techniques de codage de la parole a été une préoccupation majeure des chercheurs depuis des décennies.

Avant, pour la transmission de la parole en bande téléphonique, la haute qualité était accessible seulement avec des codeurs de débits élevés comme le 64 Kbits/s log **PCM** (Pulse Coded Modulation) ou le 32 Kbits/s **ADPCM** (Adaptive Delta Pulse Coded Modulation). Actuellement, plusieurs techniques de codage peuvent produire la qualité requise à 16 Kbits/s. Parmi ces techniques, nous pouvons citer :

- Code Excited Linear Prediction (**C.E.L.P**) [1] ;
- Multi Pulse Linear Predictive Coding (**M.P.L.P.C**) [2] ;
- Adaptive Predictive Coding (**A.P.C**) [3] ;
- Adaptive Transform Coding (**A.T.C**) [4] ;
- Sub Band Coding (**S.B.C**) combiné avec le **ADPCM** [5], etc...

Toutes ces techniques exigent un grand délai de codage qui varie entre 30 et 60 ms. Ce délai, nécessaire pour ce type de codeurs afin d'exploiter les redondances du signal de parole, est indésirable ou inacceptable dans plusieurs applications tels les réseaux téléphoniques commutés, les liaisons longue distances, etc.

Dans ce mémoire, le but de notre travail est la réalisation d'un codeur/décodeur nommé "Low Delay Code Excited Linear Prediction (L.D C.E.L.P)" ayant les caractéristiques suivantes :

- débit égal à 16 Kbits/s ;
- bande téléphonique ( 300 Hz - 3400 Hz ) ;
- retard de codage inférieur à 5 ms.

Les codeurs de cette classe, appelés codeurs hybrides, essaient d'imiter le processus de production du son humain. Ainsi, le couple source-conduit vocal est substitué dans le codeur par un couple séquence d'excitation-filtre. Le filtre, défini en utilisant des techniques adaptatives prédictives qui retirent la corrélation du signal de parole, est actualisé à des intervalles réguliers avec un cycle de 4 vecteurs. Le codeur LD-CELP utilise la quantification vectorielle pour le codage. Nous choisissons une excitation, issue d'un dictionnaire connu aussi bien de l'émetteur que du récepteur, qui est la plus proche (selon un critère de minimisation de l'erreur pondérée) du vecteur de parole à coder. On obtient un codage efficace car seule l'adresse (ou l'indice) de la séquence choisie est transmise. La procédure de recherche de la meilleure séquence d'excitation dans le dictionnaire est optimisée afin de réduire au minimum le coût de calcul.

Ce mémoire comporte quatre chapitres :

le premier comporte des généralités. Il décrit brièvement les systèmes de phonation et d'audition humaine, et est suivi d'une brève description de système de communication numérique. Nous décrivons ensuite les dimensions de performances d'un codeur de la parole pour lesquelles nous sommes amenés à faire des compromis durant la conception. Un aperçu sur la quantification vectorielle sera donné en fin de chapitre ;

le deuxième chapitre est consacré à l'analyse par prédiction linéaire et à la description de l'algorithme de codage CELP ;

le troisième chapitre décrit l'algorithme de codage adopté LD-CELP, la procédure de recherche de la meilleure séquence ainsi que l'algorithme de conception des dictionnaires "forme" et "gain" ;

Le quatrième chapitre expose d'une façon détaillée la mise en oeuvre de l'algorithme LD-CELP. Les formes d'ondes de phrases codées sont illustrées par des figures et les mesures objectives de la qualité de la parole synthétisée sont illustrées par des tableaux de valeurs. Notre travail est ensuite extrapolé à un codeur utilisant un générateur de code ternaire. Une comparaison de performance entre un codeur utilisant une excitation stochastique et une excitation ternaire sera décrite. Le codeur LD-CELP réalisé a été ensuite adapté pour le codage de signaux de parole large bande à un débit de 32 Kbits/s.

# Chapitre 1

## Généralités

Ce chapitre comporte des généralités qui regroupent des notions de production, d'audition et les propriétés statistiques d'un signal de parole. Nous donnons un aperçu sur les systèmes de communication, les dimensions de performance des codeurs et les mesures objectives et subjectives utilisées pour l'évaluation de la qualité du signal de parole synthétisé. Ensuite, nous abordons un aspect très important dans toute opération de codage qui est la quantification vectorielle. Des définitions générales et des notions de base de la quantification vectorielle seront données. Nous décrivons ensuite l'algorithme de Lloyd généralisé pour la conception de dictionnaire stochastique et une technique de conception de dictionnaire initial.

## 1.1 Aspects physiologiques de la phonation

Les principaux organes composant l'appareil phonatoire sont : les poumons, la trachée artère, le pharynx, les cavités buccales et nasales figure (1.1).

La parole est le résultat de l'action volontaire et coordonnée des appareils respiratoires et masticatoire. Cette action se déroule sous le contrôle du système nerveux central.

Les sons voisés résultent d'une vibration périodique des cordes vocales; des impulsions périodiques de pression sont ainsi appliquées au conduit vocal. Ce dernier est un ensemble de cavités situées entre la glotte et les lèvres ; on peut distinguer la cavité pharyngienne, la cavité buccale et en dérivation la cavité nasale.

Le conduit vocal peut être considéré comme une succession de tubes ou cavités acoustiques de sections différentes.

Les sons non voisés, fricatifs ou plosifs, sont générés en laissant volontairement les cordes vocales ouvertes. Pour les fricatives ( /f/, /s/, /ch/), le débit d'air, forcé à travers un resserrement du conduit vocal, donnera naissance à un bruit. Un resserrement lié aux vibrations des cordes vocales générera des fricatives voisées (e. g. /v/). Un son plosif (ou occlusif) est produit par une occlusion momentanée du conduit vocal en un point donné suivi par une ouverture brusque; Il peut être voisé (/b/, /d/, /g/) ou non voisé (/p/, /t/, /k/ ) [6] et [7].

Un son est donc dit voisé lorsque les cordes vocales sont excitées. Elles s'ouvrent progressivement, sous l'action de la pression de l'air, et laissent apparaître une ouverture de forme triangulaire et d'étendue variable : la glotte.

Elles se mettent alors à vibrer donnant naissance à l'onde glottique caractérisée par des variations impulsionnelles périodiques de la pression et du débit de l'air. Les sons voisés sont générés par une onde glottique excitant les cavités supra glottiques (pharynx, cavités buccale et nasale). L'ouverture de la glotte changera en fonction des longueurs, épaisseurs et tensions des cordes vocales. Plus la tension des cordes vocales est élevée, plus la fréquence fondamentale sera élevée.

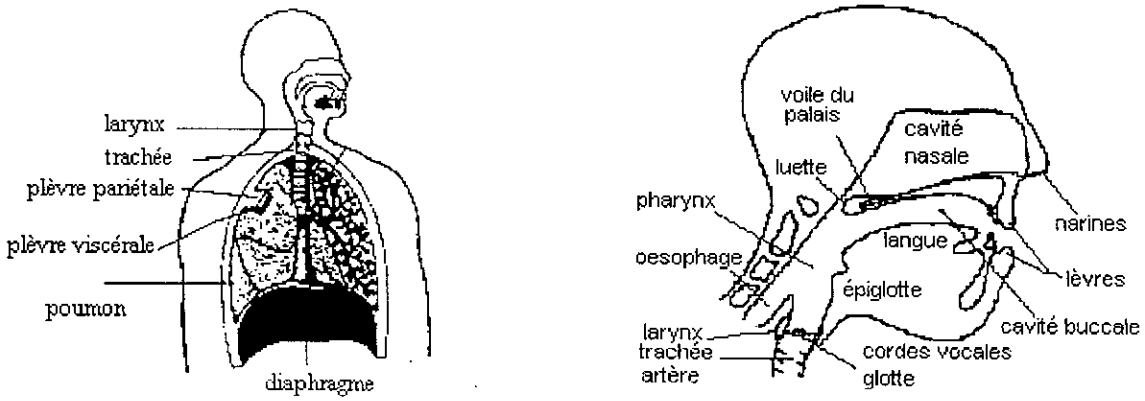



**Figure 1.1** Le système phonatoire d'après [6] et [7]

La fréquence du pitch peut varier [6] :

- de 80 à 200 Hz pour une voix masculine ;
- de 150 à 450 Hz pour une voix féminine ;
- de 200 à 600 Hz pour une voix d'enfant.

Un exemple de spectre d'un son voisé est montré en figure (1.2.c). On y observe les raies qui correspondent aux harmoniques du fondamental Fo (structure pitch); l'enveloppe de ces raies présente des maximums appelés formants et qui correspondent aux fréquences propres $F_i$ (i = 1, 2, 3,...) du conduit vocal (structure formantique).

Les trois premiers formants sont essentiels pour caractériser le spectre vocal ; les formants d'ordre supérieurs ont une influence plus limitée.

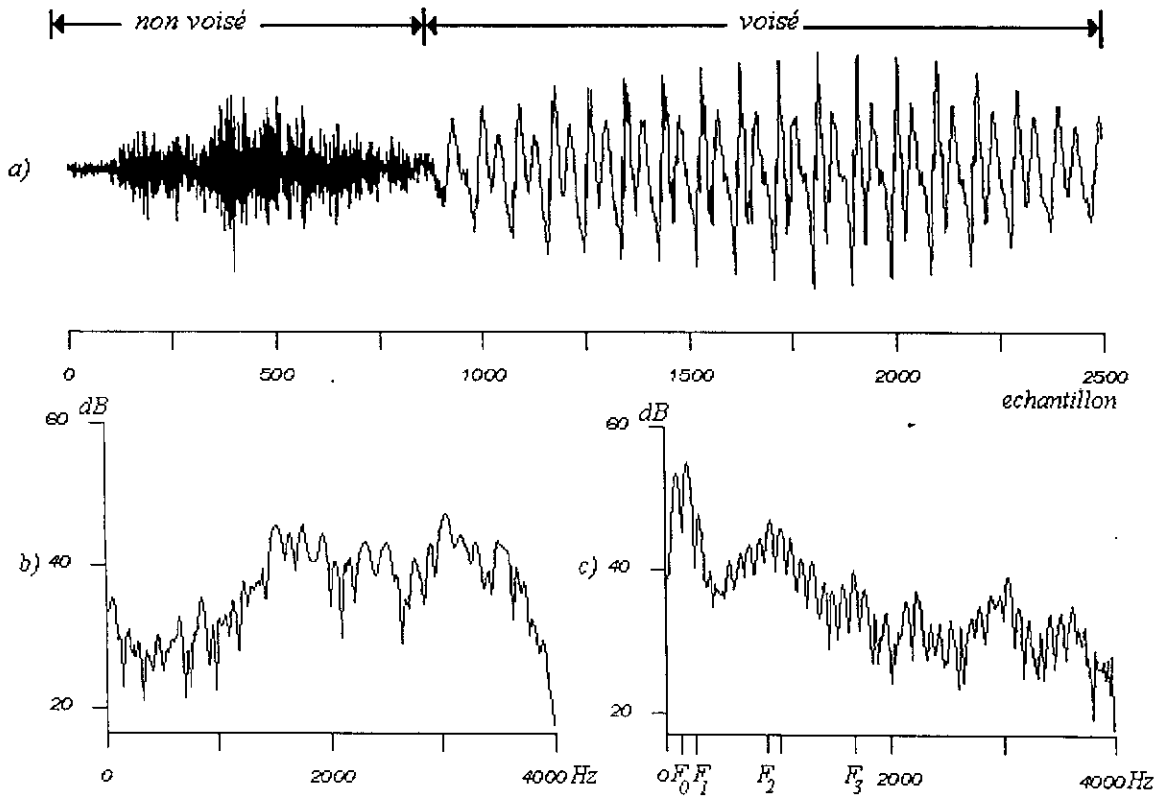



**Figure 1.2.** a) Représentation temporelle de segment de parole voisé et non voisé. b) et c) Puissances spectrales calculées dans les régions non voisées et voisées respectivement. La puissance spectrale est calculée sur des segments de longueur de 30 ms et pondérés par une fenêtre de Hamming.

Un son non voisé ne présente pas de structure périodique. Il peut être considéré comme un bruit blanc filtré par la transmittance de la partie du conduit vocal et les lèvres, son spectre ne présente donc pas de structure pitch (figure 1.2.b). La figure (1.3) nous montre un exemple de modèle de production de la parole. Le rapport relatif des composantes voisées et non voisées est contrôlé dans le modèle par ajustement des gains correspondants. Pour un signal "purement voisé" le gain de la source de bruit est nul, et pour un signal "purement non voisé" le gain de la source de train d'impulsions est nul.

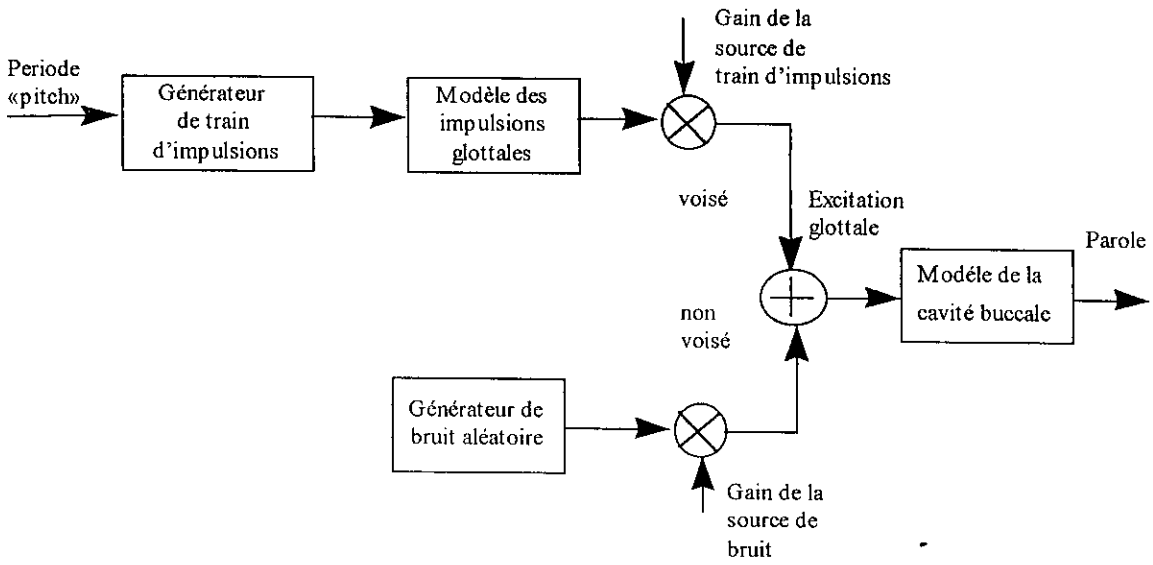



**Figure 1.3.** Exemple de modèle de production de la parole d'après [8].

## 1.2 Caractéristiques psycho-acoustiques

L'oreille est un récepteur complexe, nous nous contenterons d'énumérer quelques grandeurs caractéristiques.

Le seuil d'audition de l'oreille est non linéaire par rapport aux fréquences. L'oreille atteint sa sensibilité maximale entre 3 et 4 kHz.

L'oreille est extrêmement sensible à la période du signal de parole. Elle est capable de déceler des variations de la fréquence du fondamental de 0.5 à 1% d'où le problème

de la mesure précise de cette fréquence. La perception est peu sensible à la phase du signal.

***Seuil d'audition masqué :*** Il est bien connu que les sons faibles cessent d'être entendus en présence de sons forts. Tel est l'effet de masque. On distingue en fait le masquage temporel du masquage fréquentiel.

***Le masquage fréquentiel*** peut apparaître lorsqu'on entend en même temps deux sons purs de fréquences différentes. Il arrive que l'un d'entre eux, le son masqué, devienne inaudible. Cet effet de masque, qui peut être partiel ou total, dépend des intensités et des fréquences relatives des deux sons.

***Le masquage temporel*** apparaît lorsque deux sons ne sont pas présentés en même temps mais sont séparés par un bref silence. Ce masquage est la conséquence du fait que la forme temporelle des excitations auditives produites est supérieure à la durée physique des signaux [7].

## 1.3 Les redondances dans le signal parole

On va essayer de déterminer la cadence maximale à laquelle un auditeur peut assimiler un message. Pour cela, définissons d'abord l'information associée à un message constitué par des éléments discrets $x_i$ appartenant à un ensemble donné X.

Soit $P_k$ la probabilité a priori d'occurrence du symbole $x_k$ : $P_k = Pr\{ X(n) = x_k \}$.

Alors une appréciation numérique de l'information reçue est donnée par :

$$I(x_k) = -\log_2 P_k \text{ [en bits]}. \tag{1.1}$$

La quantité d'information moyenne reçue ou entropie est alors [en bits/échantillon] :

$$H(X) = E(I(X)) = -\sum_{k=1}^{K} P_k . \log_2 P_k \tag{1.2}$$

On a :

$$0 \leq H(X) \leq \log_2 K$$

- si H(x) = 0 la source est totalement prédictible,
- si H(x) = $\log_2 K$ la source est non prédictible, les symboles sont équiprobables.

Par exemple, si X est l'ensemble des 42 phonèmes de la langue anglaise dont les probabilités d'occurrence sont connues, on trouve la valeur moyenne H = 4,9 bits [38]; si tous les phonèmes étaient équiprobables, on aurait trouvé 5,39 bits (car $2^{5,39} = 42$), en d'autres termes, chaque phonème doit être codé avec 5 bits si l'on tient compte des probabilités a priori, sinon avec 6 bits.

Exemple :

Soit deux codages binaires d'une source de quatre messages tableau 1.1;

| message | probabilité | code C1 | code C2 |
|---|---|---|---|
| $m_1$ | 0.5 | 00 | 0 |
| $m_2$ | 0.25 | 01 | 10 |
| $m_3$ | 0.125 | 10 | 110 |
| $m_4$ | 0.125 | 11 | 111 |

Tableau 1.1. Exemple d'un codage binaire.

L'entropie de la source est :

$$H(x) = 0.5 \log 2 + 0.25 \log 4 + 0.125 \log 8 + 0.125 \log 8 = 1.75 \text{ bits}$$

Dans le code C1 ou $n_i = n = 2$

l'efficacité vaut : $\eta = \frac{H}{2} = 0.875$

la redondance vaut : $\rho = 1 - \eta = 0.125$

L'efficacité de 87.5 % (ou la redondance de 12.5 %) provient du fait qu'on avantage de la même manière des messages de fréquences différentes.

Le code C2, au contraire, affecte les mots les plus courts aux messages les plus fréquents (ici : $p_i = 2^{-n_i}$). On a :

$$n = 1x0.5 + 2 x 0.25 + 3 x 0.125 + 3 x 0.125 = 1.75$$

$$\eta = 1, \quad \rho = 0.$$

Le code est efficace à 100 %. Sa redondance est nulle.

Dans la conversation courante, environ dix phonèmes sont prononcés par seconde; l'information moyenne est donc inférieure à 50 bits/s [6].

De l'autre coté, pour garder une haute qualité de la parole avec une représentation numérique du signal de parole, l'utilisation d'un système de conversion A/D réclame plus de 100 000 bits par seconde.

Il y a donc apparemment une redondance énorme dans le signal de parole [6].

La suppression partielle des redondances permet une représentation plus efficace des données. La compression des données peut se faire sans pertes d'information (e.g. code Huffman) ou avec pertes en exploitant dans ce cas la tolérance de l'organe récepteur (e.g. l'oreille). La compression du signal consistera à réduire les redondances du signal de parole. Ces dernières sont essentiellement dues [9]:

- au manque de platitude du spectre court-terme ;
- à la quasi-périodicité des signaux voisés ;
- à la limitation des formes et des vitesses de mouvement possibles du conduit vocal ;
- aux distributions non uniformes des valeurs des paramètres de transmission.

Les trois premières sont dues à des propriétés physiques du mécanisme de production de la parole. La dernière est fonction du codage utilisé.

Le manque de platitude du spectre court-terme est lié au fait que les échantillons de parole adjacents sont corrélés entre eux. On peut décorréler ces échantillons de parole par un filtrage spectrale adapté. La quasi-périodicité des signaux parole voisé peut être supprimée en utilisant un prédicteur long-terme. La lenteur du conduit vocal permet d'envoyer les paramètres des filtres toutes les 10-30 ms. La dernière des redondances citées peut être exploitée par un codage approprié.

## 1.4 Propriétés statistiques du signal de parole.

On peut s'intéresser au signal de parole non plus comme une manifestation d'un phénomène physique mais comme une fonction du temps au sens de la théorie du signal. L'exemple de la figure (1.4) indique plusieurs caractéristiques significatives de formes d'ondes. Ces caractéristiques sont :

- la densité de probabilité des amplitudes ;
- le degré de corrélation entre échantillons ;
- l'existence d'une structure macroscopique comme une quasi-périodicité ;
- la non-stationnarité. Cette dernière résulte des changements au cours du temps aussi bien de la source que de la forme et des dimensions du conduit vocal.

La figure (1.5) montre que la forme d'onde voisée de /u/ à basse fréquence varie plus lentement que la haute fréquence de la forme d'onde non voisée de /s/. On notera que le segment voisé est quasi périodique, avec environ "7" périodes sur 50 ms, ce qui correspond dans ce cas, à une fréquence pitch de 140 Hz.

La forme d'onde non voisée, d'autre part, est ressemblante à un bruit. Le niveau d'énergie est plus bas que celui du segment voisé. La forme d'onde de la parole est extrêmement non stationnaire avec des différences inter-segment très significatives en termes de niveaux d'amplitudes et du contenu spectral. La figuré (1.5) montre la dépendance temporelle de la variance de segment $\sigma^2_x(k)$ mesuré à travers des segments de 32 ms de parole. On constate que la dynamique de la variance court-terme peut excéder 40 dB sur une durée de moins de 1 seconde.

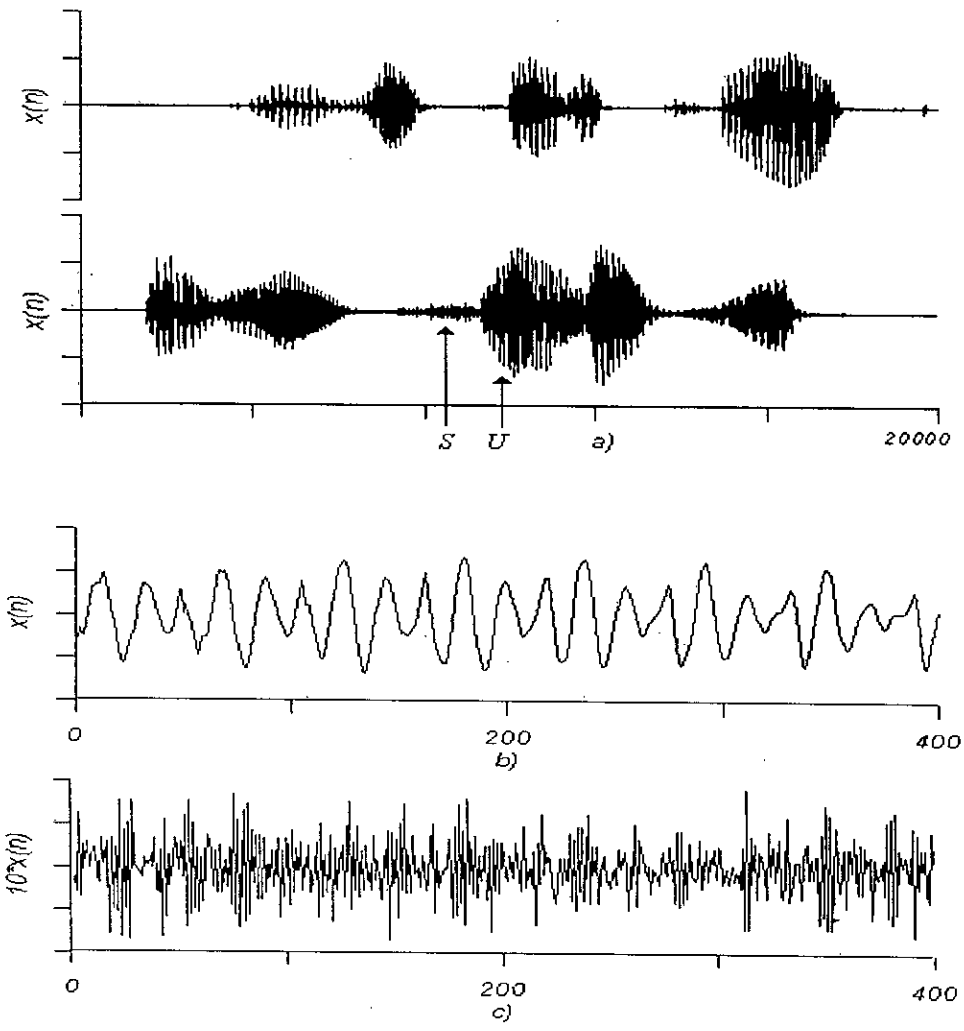



**Figure 1.4.** Formes d'ondes d'amplitudes de parole pour a) phrase de longueur 2.50 secondes (locuteur masculin) "**La vaisselle propre est mise sur l'évier**". b) segment de parole voisé court terme "u" du mot "sur" (50 ms); et c) segment de parole non voisé court-terme "s" du mot "sur" (50 ms).

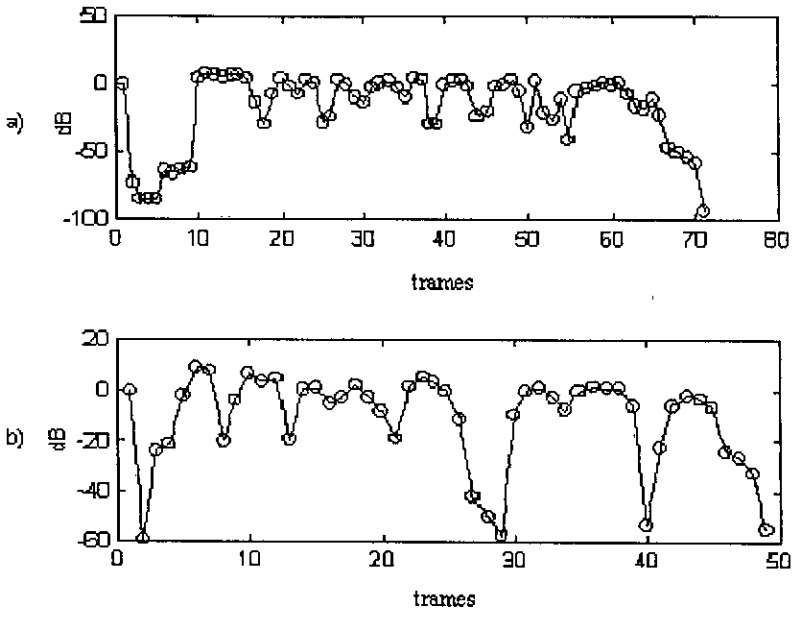



**Figure 1.5.** Evolution de la variance de la parole court terme **a)** signal de 3.5 s et **b)** un signal parole différent de 2.39 s.

## 1.5 Système de communication

Le codage d'un signal est la procédure de représenter un signal information de façon qu'il réalise l'objectif de communication telle une conversion analogique numérique, transmission à bas débit ou cryptage de message. Dans la littérature les termes codage de source, codage numérique, compression de données et compression des signaux sont tous utilisés pour désigner les techniques utilisées pour achever une représentation numérique compacte du signal [10]. La figure (1.6) représente un schéma bloc de codage numérique. On notera que le but final est le récepteur humain.

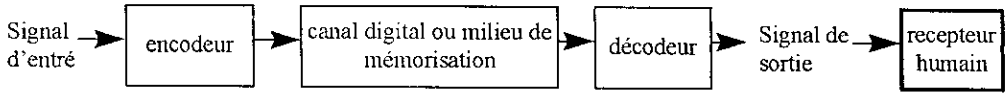



**Figure 1.6.** Système de codage pour compression de signaux d'après [10].

### 1.5.1 Chaîne de communication numérique

La figure (1.7) décrit les différents blocs d'une chaîne de communication numérique :

- le codeur source essaie de minimiser le débit binaire nécessaire pour représenter fidèlement le signal d'entrée ;
- le MOdulateur DEModulateur (MODEM) cherche à maximiser le débit binaire que peut supporter un canal donné sans causer un niveau inacceptable de probabilité d'erreur binaire ;
- Le bloc de codage canal ajoute des redondances au flux binaire pour la protection contre les erreurs.

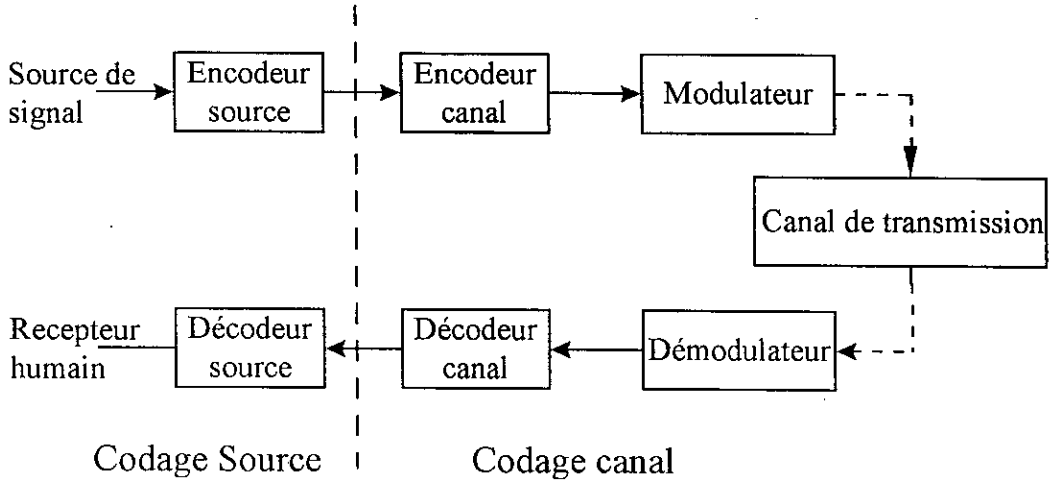



**Figure 1.7.** Schéma bloc d'un système de communication numérique d'après [10].

La capacité de la compression des signaux a été une majeure préoccupation de la technologie de la communication longue-distance, stockage de haute qualité et le cryptage de message.

## 1.5.2 Critères de performance dans le codage de la parole

Le problème essentiel dans la compression du signal est de minimiser le débit binaire dans la représentation numérique du signal toute en maintenant des niveaux adéquats de qualité du signal, de complexité d'implantation et de retard de communication.

### 1.5.2.1 Qualité du signal

La qualité du signal perçu est souvent évaluée sur une échelle de 5 points qui est connue comme étant l'échelle MOS (Mean Opinion Score) dans les tests de la qualité de la parole : une moyenne à travers un grand nombre d'entrée parole, locuteurs et testeurs

d'écoute évaluant la qualité du signal. Les cinq points de la qualité sont associés à un ensemble d'adjectifs de description : mauvais, médiocre, inacceptable, bon, excellent. On attribue ainsi un seul niveau à chaque signal parole à évaluer durant la procédure d'évaluation subjective.

### 1.5.2.2 Débit binaire

On mesure le débit binaire d'une représentation digitale en bits par échantillon, ou bit par seconde (b/s) selon le contexte. Le débit en bits par seconde n'est que le produit de la fréquence d'échantillonnage et le nombre de bits par échantillon. La fréquence d'échantillonnage doit être au moins deux fois plus grande que la largeur de bande du signal correspondant. Dans le cas de la téléphonie, pour une bande de 3.2 kHz (200-3400 Hz), la fréquence d'échantillonnage de 8 kHz est utilisée.

### 1.5.2.3 Complexité

La complexité d'un algorithme de codage est l'effort de calcul exigé pour implanter les processus de l'encodage et du décodage dans les cartes de traitement du signal (hardware), mesuré en terme de la capacité arithmétique (évalué en MIPS) et l'espace mémoire utilisé. D'autres mesures de complexité peuvent être signalées telles que la taille physique de l'encodeur ou du décodeur ou codec, le prix et la consommation de puissance (en Watt ou en milliwatt, mW) ce dernier étant un important critère dans un système portable.

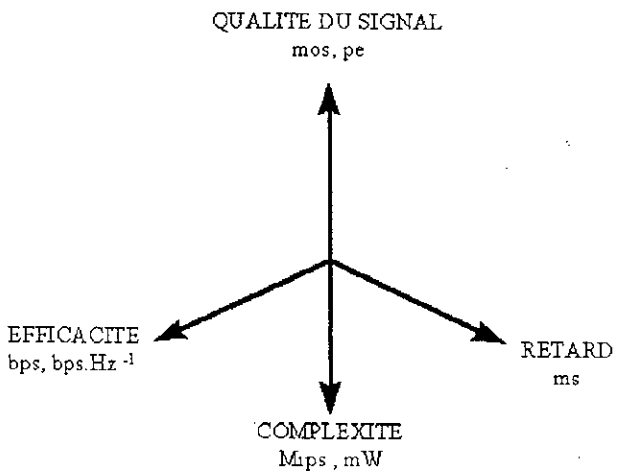



**Figure 1.8.** Critères de performance d'un codeur

### 1.5.2.4 Retard de communication

La complexité dans un algorithme de codage est souvent accompagnée d'une augmentation de la durée de traitement dans l'encodeur et le décodeur. Bien que l'évolution des capacités des processeurs de traitement du signal est un facteur en faveur d'utilisation d'algorithme plus sophistiqué, le besoin de limiter le retard de communication ne doit pas être d'une importance moindre. Ce besoin impose des restrictions pratiques importantes dans l'utilisation des algorithmes. Selon l'environnement de communication, le retard total permis à un sens peut être aussi bas qu'une milliseconde (comme en réseaux téléphoniques sans annulateur d'écho).

Le retard de codage à un seul sens est défini comme étant le temps écoulé entre l'instant où l'échantillon du signal de parole arrive à l'entrée de l'encodeur et l'instant où le même échantillon apparaît à la sortie du décodeur, moins tout retard introduit par les autres équipements de communication (comme les MODEM ) entre la paire encodeur-décodeur et le retard de propagation du signal qui dépend de la distance. En d'autres termes, c'est comme si l'encodeur et le décodeur sont directement connectés par fils sans aucun équipement entre eux. Cette définition fait que le retard de codage dépend seulement de l'algorithme de codage.

Avec cette définition, le retard de codage des codeurs CELP peut être grossièrement déterminé en fonction de la taille de la trame du signal de parole utilisée.
Le retard de codage consiste en trois catégories de retard [11] :

- retard algorithmique de bufferisation ;
- retard de traitement ;
- retard de transmission binaire.

- Le premier type de retard est dû à l'analyse LPC (Linear Prediction Coding) adaptative progressive. Le codeur CELP doit bufferiser une trame d'échantillon avant de commencer le codage du premier échantillon de cette trame. Cette opération introduit au moins une trame de retard de bufferisation.

- En supposant que le hardware utilisé est assez rapide pour exécuter le codage en temps réel (ce qui est le cas en général -cartes TMS320C30- par exemple.), alors, cela peut prendre presque une valeur d'une trame de retard de traitement pour achever l'encodage et décodage de la trame de parole bufferisée.

- Il est supposé que l'encodeur ne commence l'envoi de bits correspondant à une trame donnée que si l'encodage de la trame entière soit terminé. Le décodeur ne commence le décodage que si tous les bits de cette trame sont reçus, d'où, une trame additionnelle de retard de transmission binaire doit être introduite. Ceci est dû au fait que le temps mis entre l'envoi du 1$^{er}$ bit de la trame et la réception du dernier bit de la trame est égal à une trame, étant donné que le débit binaire du canal de communication est le même que le débit binaire du codeur de la parole. Alors, le retard de codage total (à un sens) du codeur CELP vaut environ trois trames.

En pratique, on peut réduire le retard de traitement en utilisant des processeurs plus rapides.

Si on prend un retard de 2.5 à 3 trames comme moyenne, alors le codeur CELP, avec des trames de 20 ms chacune, aura un retard de codage global de 50 à 60 ms. Celui ci est dû principalement à la bufferisation de 20 ms de la trame qui est exigé pour la détermination du modèle autorégressif (coefficients de prédiction LPC).

## 1.6 Mesure de la qualité

Les mesures objectives de la qualité de la parole sont purement des mesures mathématiques évaluées en utilisant des distances euclidiennes et les mesures subjectives de qualité évaluent la qualité de codage par des tests d'écoute.

La mesure objective de la qualité la plus couramment utilisée, pour les codeurs qui essaient de préserver la forme du signal, reste le rapport signal à bruit (RSB).

Si $s$ est le signal de parole original.

$\overline{s}$ est le signal de parole synthétisé.

Alors le signal d'erreur est donné par:

$$e(n) = s(n) - \overline{s}(n) \tag{1.3}$$

pour un signal de N échantillons, on définit l'énergie du signal

$$E_s = \sum_{n=0}^{N-1} s^2(n) \tag{1.4}$$

et l'énergie de l'erreur :

$$E_e = \sum_{n=0}^{N-1} e^2(n) \tag{1.5}$$

le RSB est alors donné par :

$$RSB(N) = 10\log\left(\frac{E_s}{E_e}\right) \quad \text{en dB} \tag{1.6}$$

Le signal de parole est par nature non-constant. Certains segments du signal peuvent avoir une énergie plus ou moins grande. En supposant que l'énergie de l'erreur soit à peu près constante, le RSB pourra être très important comme très faible.
On utilise plutôt le RSB segmental. Le signal est découpé en "M" segments de 15 à 30 ms puis on calcule une moyenne des RSB.

$$RSB_{seg} = \frac{1}{M}\sum_{i=0}^{M} 10\log\left(\frac{\sum_{n=0}^{N-1} s^2(n)}{\sum_{n=0}^{N-1} e^2(n)}\right) \tag{1.7}$$

Cette mesure présente l'avantage de tenir compte de l'évolution du RSB au cours du temps et de bien prendre en compte les segments de faible énergie. Nous essayons en outre de limiter les trop grands écarts. Si RSB(m) d'un segment de signal est supérieur à 35 dB, on le remplace par 35 dB. De même, dans les zones de silence, le RSB peut atteindre des valeurs très négatives: dans ce cas, on peut ou bien retirer du calcul les zones ou bien fixer un seuil inférieur "T" tel que $0 \le T \le (-10)$ dB [9].

L'exemple suivant nous donne une idée sur le fait que le RSB segmental reflète mieux la réalité de distorsion du signal [12]. Soit le tableau de valeurs suivant :

| Numéro du segment | M = 1 | M = 2 |
|---|---|---|
| RMS du signal d'entrée | 10 | 1 |
| RMS du bruit d'entrée | 1 | 1 |
| RSB(m) | 100 | 1 |
| RSB(m) dB | 20 | 0 |

Rapport signal à bruit

| | |
|---|---|
| RSB conventionnel | (100 + 1) / 2 = 50.5 → 17 dB |
| RSB segmental | (20 dB + 0 dB) / 2 → 10 dB |

**Tableau 1.2.** RSB segmental et RSB conventionnel.

La valeur RSB conventionnel = 17 dB est très influencée par RSB(1) = 20 dB, correspondant au segment de haute énergie. Alors que, RSB seg = 10 dB, montre une pondération égale des valeurs des composantes RSB(1) = 20 dB et RSB(2) = 0 dB.

Les essais d'écoute sont nécessaires car le récepteur humain représente le dernier bloc d'un système de codage de la parole. De plus, le RSB n'est pas nécessairement corrélé avec la qualité d'écoute.

Les méthodes les plus utilisées sont les suivantes :

- Diagnostic Rhyme Test (DRT) qui mesure l'intelligibilité sur un grand nombre de mots
- Diagnostic Acceptability Mesure (DAM) qui mesure le naturel perçu de la parole ;
- Mean Opinion Score (MOS) ou l'auditeur évalue un codeur sur une échelle absolue allant de 1 à 5 avec :
  1. Mauvais.
  2. Médiocre
  3. Passable
  4. Bon
  5. Excellent

## 1.7 Aspect de la Quantification Vectorielle

### 1.7.1 Principes de la Quantification Vectorielle

La quantification est une partie intégrante dans tout codeur de la parole. La plupart des paramètres utilisés pour représenter le signal de parole doivent être quantifiés. Si, à chaque instant, une valeur ou un vecteur de dimension inférieur ou égale à trois est quantifié, on parlera de quantification scalaire. La valeur quantifiée est représentée par l'une des valeurs discrètes fixes dites niveaux de quantification. Dans la quantification scalaire uniforme, ces niveaux sont régulièrement espacés. Dans la quantification logarithmique, l'espacement est uniforme dans le domaine logarithmique. Dans le cas ou la grandeur à quantifier est composée de plusieurs variables (supérieur à trois variables): on parlera de Quantification Vectorielle (QV) [13, 14].

La collection des représentations possibles d'un vecteur est dite dictionnaire. On utilise en générale plus d'un dictionnaire pour représenter le vecteur. Plusieurs procédures ont été proposées pour créer, organiser et scruter les dictionnaires. Ces méthodes englobent : tree-structured VQ (Vector Quantization), transform VQ, product code VQ,

split VQ, gain-shape VQ, multistage VQ, hierarchical VQ.

Nous appellerons quantificateur vectoriel de dimension m à k niveaux une application Q qui, à un vecteur d'entrée $x = (x_1, x_2,.., x_m)$, fait correspondre une valeur approchée $y_i$ choisie dans un ensemble fini de k éléments $y = \{ y_i; i = 0,1,.., k-1\}$.
L'ensemble y est un dictionnaire de k représentants. En posant $R = \log_2 k$, nous dirons que les vecteurs d'entrée sont quantifiés sur k niveaux et codés sur R bits.
Il n'y a rien de mystérieux à considérer des espaces de grandes dimensions, il suffit de savoir que tout s'organise autour des coordonnées des vecteurs et qu'il n'y a pas lieu de s'imposer une représentation mentale géométrique. A titre d'illustration, nous précisons qu'un vecteur de l'espace $R^m$ est simplement une matrice colonne constituée de k nombres réels $x_i$ : $x = (x_1, x_2,...,x_m)^T$, et que par exemple, une sphère entièrement caractérisée par son centre $u = (u_1, u_2,...,u_m)^T$ et son rayon $\rho$ est constitué de points dont les coordonnées satisfont la relation: $\sum_{i=1}^{m} (x_i - u_i)^2 = \rho^2$

Nous appellerons distance entre x et $y_i = Q(x)$, généralement notée par $d(x , y_i)$ le degré de distorsion dû à l'approximation du vecteur d'entré x par le vecteur «arrondi» $y_i$. Un quantificateur vectoriel est alors complètement défini par le dictionnaire y et la distance d. En général, la fonction "d" nécessaire à la définition d'une distance entre deux éléments x et y, d(x , y) est défini par l'application :

$$R^k \xrightarrow{\ d\ } D$$
$$D = \{y_i \in R^k / i = 1, 2, ..., k\} \qquad (1.8)$$

Doit avoir les propriétés suivantes :

- $d(x , y) \geq 0$
- $d(x , y) = 0$ si $x = y$
- $d(x , y) = d(y , x)$ (symétrie)
- $d(x , z) \leq d(x , y) + d(y , z)$ (inégalité triangulaire)

Dans le cas de la parole, la distance doit avoir deux propriétés supplémentaires :

- d(x,y) doit avoir une interprétation physique;
- d(x,y) doit être simple à calculer.

La mesure de la distorsion doit avoir une certaine signification dans le domaine spectrale selon les propriétés spectrale de la parole. Les différences entre l'enveloppe spectrale du signal original et l'enveloppe spectrale du signal codé qui peuvent conduire à des sons phonétiquement différents sont les suivantes :

- les formants de l'enveloppe spectrale du signal original et ceux de l'enveloppe spectrale du signal codé se produisent à des fréquences différentes ;
- les bandes de ses formants différent significativement.

Exemples de distances :

$$x = (\alpha_1, \alpha_2, \ldots\ldots \alpha_n)$$
$$y = (\alpha'_1, \alpha'_2, \ldots\ldots \alpha'_n)$$

$$d(x,y) = \sum_{i=1}^{n} \left|\alpha_i - \alpha'_i\right| \qquad \text{distance de Minkowsky}$$

$$d(x,y) = \left( \sum_{i=1}^{n} \left|\alpha_i - \alpha'_i\right|^2 \right)^{1/2} \qquad \text{distance Euclidienne} \qquad (1.9)$$

$$d(x,y) = \underset{i}{\text{Max}} \left|\alpha_i - \alpha'_i\right| \qquad \text{distance de Chebychev}$$

D'autres mesures de distorsion spectrale peuvent être utilisées selon le contexte telles que : la mesure de distorsion spectrale logarithmique, la mesure d'ITAKURA SAITO, la mesure cepstral, ect

Par exemple, la distorsion d'ITAKURA SAITO équation (1.10) 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.

$$d_{IS} = \frac{1}{2\pi} \int_{-\pi}^{\pi} \left[ e^{V(\omega)} - V(\omega) - 1 \right] d\omega \qquad (1.10)$$

avec $V(\omega) = \log(S(\omega) - \log(\hat{S}(\omega)))$

$$S(\omega) = \frac{G}{\left|A(e^{i\omega})\right|^2} \qquad G : \text{facteur gain du filtre LP} \qquad (1.11)$$

En supposant que la grandeur d'entrée est un vecteur aléatoire distribué selon une loi p(x), les performances du quantificateur peuvent être mesuré par la distorsion moyenne $D_Q$ introduite, c'est à dire par l'espérance mathématique de la distance d :

$$D_Q = E\left[d(x,Q(x))\right] = \int d(x,Q(x)).p(x).dx \qquad (1.12)$$

Dans la pratique, la distribution des points d'entrée étant généralement inconnue, on approximera $D_Q$ par une distorsion moyenne calculée sur un large nombre d'échantillons $\{x_1, x_2, .., x_N\}$de vecteurs d'entrée. L'ergodicité et la stationnarité nous permettent d'écrire:

$$D_Q \cong \frac{1}{N}\sum_{j=1}^{N} d(x_j,Q(x_j)) \qquad (1.13)$$

La distance introduit implicitement une partition de l'ensemble des vecteurs d'entrée en k classes $\{S^i, i = 0, 1, ...., k - 1\}$, la classe $S^i$ étant l'ensemble des vecteurs associés à $y_i$ par le quantificateur :

$$S^i = Q^{-1}(y_i) = \{x; Q(x) = y_i\} \qquad (1.14)$$

Nous appellerons centroïde de la classe $S^i$ le vecteur $c^i$ tel que sa distance moyenne à tous les éléments de la classe soit minimale (en géométrie euclidienne, le centroïde est le centre de gravité) :

$$E\left[d(x,c^i); x \in S^i\right] = \inf_{x^i}\left\{E\left[d(x,x^i); x \in S^i\right]\right\} \qquad (1.15)$$

Etant donné une distance et une taille de dictionnaire, on cherche un quantificateur optimal qui minimise la distorsion moyenne ou qui se rapproche de l'optimalité.

## 1.7.2 Conditions d'optimalité

Pour une distribution statistique donnée de la source et un débit fixé :

- le quantificateur globalement optimal est celui qui minimise la distorsion moyenne ;
- un quantificateur localement optimal a un dictionnaire qui peut être légèrement perturbé sans que la distorsion moyenne augmente.

Il n'existe pas de méthode qui décrit la façon de concevoir de dictionnaire globalement optimal pour les quantificateurs vectoriels. Seuls des propriétés suffisantes sont connues qui permettent de construire des dictionnaires localement optimaux.

Un quantificateur se décompose en deux applications : un codeur et un décodeur (figure 1.9). Le quantificateur (localement) optimal est alors celui réunissant les points suivants [14, 15].

- un codage optimal (pour un dictionnaire fixé), celui-ci respecte "la règle du plus proche voisin" que nous allons décrire ;
- le décodage optimal (pour une partition $S^i$ donnée), le vecteur représentant $y^i$ doit minimiser la distorsion associée au voronoï $S^i$, $y^i$ est donc le centroïde de cette cellule : $y^i = cent\ (S^i)$ ;

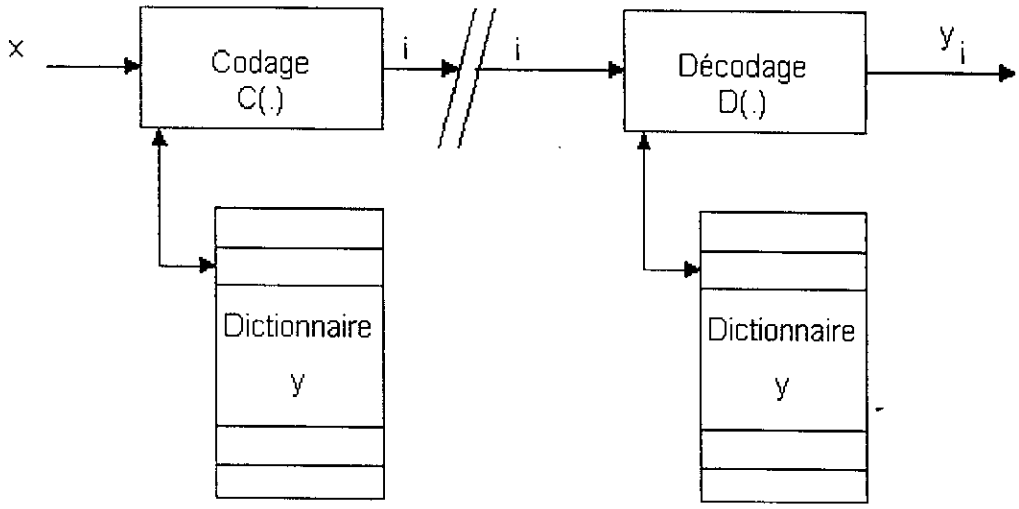



**Figure 1.9.** Schéma général d'un quantificateur vectoriel

**Condition du plus proche voisin**

Etant donné un décodeur et son ensemble fini de mots codes de sortie C, les classes de partitions $S^i$ de l'encodeur sont optimales si, elles satisfassent :

$$S^i \subset \{x \mid d(x, y_i) \leq d(x, y_j)\ ;\ \forall j\} \tag{1.16}$$

Les régions de partition sont définis par les mots codes $\{y_i\}$ dans C :

$$Q(x) = y_i \text{ seulement si } d(x,y_i) \leq d(x,y_j)\ \forall j \tag{1.17}$$

**Condition du centroïde**

Etant donné une partition d'encodeur $P = \{ S^i \mid i=1,...N\}$,

les mots codes optimaux $y_i$ dans C sont les centroïdes dans chaque partition $S^i$ :

$$y_i = Cent\ (S^i)$$

$$y_i = \min E(d(x,y) \mid x \in S^i)$$

## 1.7.3 Construction de quantificateurs statistiques

Supposons que nous disposons d'une certaine distance d. Construire un quantificateur revient donc à établir une stratégie de choix du dictionnaire associé. Cette stratégie est intimement liée à la nature de la distribution des vecteurs à quantifier.

Dans le cas ou les points d'entrée sont distribués d'une façon non uniforme, on adoptera une approche statistique visant à tirer parti de cette non-uniformité (bien qu'il existe des techniques de transformation de la structure non-uniforme en une structure uniforme et appliqué une approche algébrique). Le dictionnaire sera construit par apprentissage : à partir d'une large base de vecteurs d'entrée où sera sélectionné un nombre réduit de points susceptible d'en refléter les propriétés statistiques.

En revanche, si la distribution des vecteurs d'entrée est plutôt uniforme, on aura intérêt à conférer à l'espace de représentation une structure mathématique forte, indépendamment de la "réalité" des données à traiter. Cette approche algébrique utilise généralement les propriétés des réseaux réguliers de points.
On s'intéressera exclusivement à la première approche.

**Algorithme de Lloyd Généralisé (GLA : Generalized Lloyd Algorithm)**

Les conditions d'optimalité citées précédemment conduisent à la conception d'un algorithme qui réalise, à partir d'une séquence d'apprentissage représentative de la statistique de la source à coder, la construction d'un dictionnaire (localement) optimal.
Cet algorithme de classification, appelé aussi algorithme des K-moyens (K-means [15]) est l'extension au cas vectoriel de l'algorithme de Lloyd-Max (cas scalaire).
Il s'agit d'un algorithme d'optimisation itératif opérant à partir d'un dictionnaire initial. A chaque itération (dite "itération de Lloyd"), deux opérations distinctes sont appliquées :

- une classification suivant la règle du plus proche voisin ;
- une optimisation suivant la condition du centroïde.

Chaque itération de Lloyd, en modifiant localement le dictionnaire, réduit ou laisse inchanger la distorsion moyenne. L'algorithme converge en un nombre fini d'itérations vers un minimum local. Ce minimum local varie en fonction du choix du dictionnaire initial. Le choix de ce dernier est donc capital.

**étape 1** : on commence par un dictionnaire initial $C_1$ , mettre m = 1;

**étape 2a** : ayant un dictionnaire $C_m = \{y_i\}$, partitionner la séquence d'entraînement en un ensemble de classe (cluster) $S^i$ en utilisant la condition du plus proche voisin,

$$\text{où} \qquad S^i = \{x \in T \mid d(x, y_i) \leq d(x, y_j) \text{ ; pour tout } j \neq i\} \tag{1.18}$$

**étape 2b** : en utilisant la condition de centroïde, calculer les centroïdes pour l'ensemble des classes trouvés en étape 1 pour obtenir le nouveau dictionnaire :

$$C_{m+1} = \{Cent(S_i) \mid i = 1,...N\} \tag{1.19}$$

**étape 3** : calculer la distorsion moyenne pour $C_{m+1}$ , si elle a changé d'une petite quantité par rapport à l'itération précédente stop, sinon, mettre m = m+1 et répéter étape 2 et 3.

La figure 1. 10. illustre la procédure de fonctionnement de l'algorithme de Lloyd.

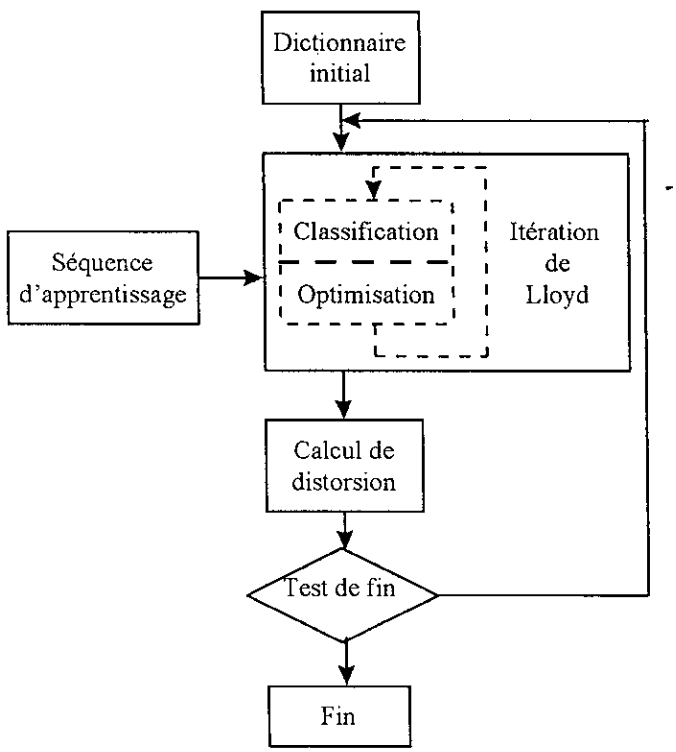



**Figure 1.10.** Schéma de fonctionnement de l'algorithme de Lloyd.

### 1.7.4 Technique de conception du dictionnaire initial

L'algorithme ainsi décrit est une procédure d'optimisation itérative basée sur la méthode de projections successives et donc conduit vers un minimum local.
La vitesse de convergence des itérations de l'algorithme de Lloyd généralisé et les performances du dictionnaire obtenu après convergence dépendent du dictionnaire initial $C_1$. Par conséquent, il est important de trouver un bon dictionnaire initial.
L'une des technique d'initialisation pour l'algorithme de Lloyd généralisé, est décrite ci-dessous [16].
Le principe de la technique, consiste à donner une attention particulière aux vecteurs d'apprentissage qui sont les plus éloignés l'un de l'autre, car ils sont susceptibles d'appartenir à des classes différentes.

Soit $v_i$, i = 1, ..., M, la séquence d'apprentissage des vecteurs. La procédure peut être exprimée comme suit :

- calculer les normes de tous les vecteurs de l'ensemble d'apprentissage. Choisir le vecteur ayant la norme maximum comme mot code ;
- calculer la distance de tous les vecteurs d'apprentissage par rapport au premier mot de code, et choisir le vecteur ayant la plus grande distance comme second mot de code. On a alors un dictionnaire de taille 2 ;
- généralement, avec un dictionnaire de taille i, i = 2, 3,...L, nous calculons la distance entre les vecteurs d'apprentissage restants $v_k$ et tous les mots codes existants. La plus petite valeur trouvée est appelée distance entre $v_k$ et le dictionnaire. Alors, le vecteur d'apprentissage ayant la plus grande distance par rapport au dictionnaire est choisie pour être le $(i+1)^{ieme}$ mot code. La procédure s'arrête quand on obtient un dictionnaire de taille L.

L'idée de base de cette procédure est d'utiliser le vecteur le plus "différent" des codes vecteurs existants comme un nouveau mot code.
La procédure est applicable pour n'importe quelle taille de dictionnaire. Dans ce cas nous n'avons pas besoin de définir un seuil quelconque.

A titre d'exemple, appliquons cette procédure pour concevoir un dictionnaire initial de 16 mots de codes en dimension 2, la séquence d'apprentissage est une source gaussienne

de moyenne nulle et de variance 1. On obtient, comme on le remarque dans la figure 1.11, une partition de mots de codes très proche de la structure 1-6-9 qui est la structure optimale pour une source gaussienne à 2 dimensions.

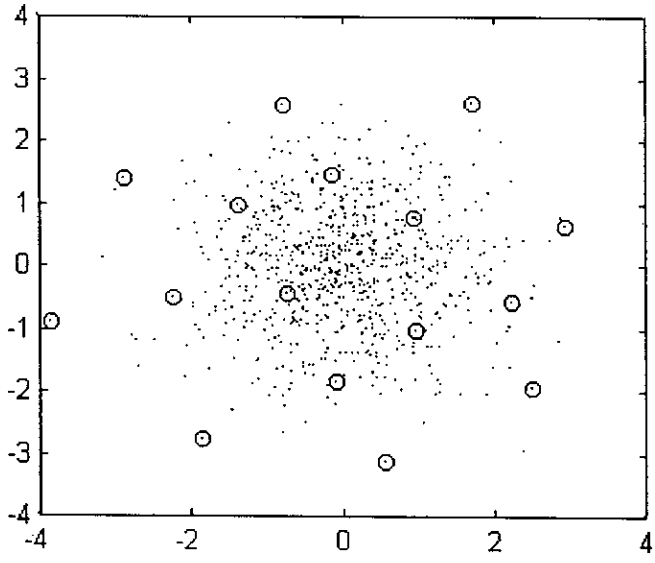



**Figure 1.11.** Structure obtenue pour une source gaussienne après une seule itération (proche de la structure 1-6-9 qui est la structure optimale pour cette source).
(. : échantillon, o : mot de code obtenu)

## 1.8 Conclusion

Le but de l'opération de codage est de réduire le taux d'informations à envoyer à chaque seconde sans dégrader la qualité de la parole reconstruite. La connaissance du mode de production, d'audition et des caractéristiques du signal de parole permet d'utiliser certaines techniques pour la modélisation du couple "source conduit-vocal", la réduction du taux de redondance dans le signal parole et l'utilisation de filtre de mise en forme du bruit pour améliorer l'agrément d'écoute. L'utilisation de la quantification vectorielle permet d'avoir une représentation des signaux plus compacte.

# Chapitre 2

## Codeur hybride utilisant la prédiction linéaire

Dans ce chapitre, nous exposons le principe de codage par prédiction linéaire qui modélise le signal de parole comme une combinaison linéaire des échantillons de parole passés. Nous présentons la méthode d'obtention des paramètres spectraux d'un signal parole (court-terme). Ensuite, le principe des Codeurs Prédictifs Linéaires Excités par Codes (CELP) est présenté.

### 2.1 Analyse par Prédiction Linéaire

Le codage des paramètres spectraux de la parole est une composante intégrale du codage de la parole. Le modèle de production de la parole source-filtre nous permet d'utiliser la prédiction linéaire pour analyser l'aspect court-terme du signal sur une trame de parole. Le signal s(n) peut être modélisé comme la sortie d'un système Auto Régressif à Moyenne Ajustée (ARMA) avec une entrée u(n) [17] et [18]:

$$s(n) = \sum_{k=1}^{p} a_k s(n-k) + G\sum_{l=0}^{q} b_l u(n-l), \qquad b_0 = 1, \tag{2.1}$$

Où $\{a_k\}$, $\{b_l\}$ et le gain G sont les paramètres du système. L'équation (2.1) prédit la sortie courante en utilisant une combinaison linéaire des sorties antérieures et les entrées courantes et antérieures.

Dans le domaine fréquentiel, la fonction de transfert du modèle de prédiction linéaire de la parole est de la forme :

$$H(z) = \frac{B(z)}{A(z)} = \frac{G\left[1 + \sum_{l=1}^{q} b_l z^{-l}\right]}{1 - \sum_{k=1}^{p} a_k z^{-k}} \tag{2.2}$$

Les racines du dénominateur et le numérateur sont, respectivement, les pôles et les zéros du système ou modèle pôle-zéro H(z).

Si $a_k = 0$ pour $1 \leq k \leq p$, H(z) devient un modèle tout zéro ou modèle à moyenne ajustée (MA). Si $b_l = 0$ pour $1 \leq l \leq q$, H(z) devient un modèle tout pôle ou modèle Auto Régressive (AR):

$$H(z) = \frac{1}{A(z)} \tag{2.3}$$

Dans l'analyse de la parole, les classes de phonèmes comme les fricatives et les nasales contiennent des vallées spectrales qui correspondent aux zéros dans H(z). Par contre les voyelles contiennent des résonances qui peuvent être modélisées par le modèle tout-pôle [65]. Pour des raisons de simplicité, ce modèle est préféré pour l'analyse par prédiction linéaire de la parole.

Ainsi, le signal prédit est égal à :

$$s(n) = \sum_{k=1}^{p} a_k s(n-k) \tag{2.4}$$

et l'erreur de prédiction ou résiduel du signal est la sortie e(n) :

$$e(n) = s(n) - \sum_{k=1}^{p} a_k s(n-k) \tag{2.5}$$

L'ordre p du système est choisi de façon que l'estimation de l'enveloppe spectrale soit adéquate. Une façon de procéder est d'allouer une paire de pôles pour chaque formant présent dans le spectre. On ajoute 2 ou 3 pôles pour approximer les zéros dus aux sons non voisés [17].

Quand la prédiction linéaire est basée sur les échantillons de parole passés s(n), celle-ci, est dite Prédiction Linéaire Adaptative Progressive (Forward) et dans ce cas les coefficients de prédiction doivent être transmis au décodeur. Si la prédiction linéaire est basée sur les échantillons de parole reconstruits antérieurs $\bar{s}(n)$, celle-ci, est dite Prédiction Linéaire Adaptative Régressive (Backward). Pour avoir les coefficients du filtre court terme $\{a_i\}$ du processus AR, la méthode classique des moindres carrés peut être utilisée. La variance ou l'énergie, du signal erreur e(n) est minimisée sur une trame de parole. Deux grandes approches sont utilisées pour l'analyse LPC (Linear Prediction Coding) court-terme : la méthode d'autocorrélation et la méthode de covariance.

## 2.2 Méthode d'Autocorrélation

La méthode d'autocorrélation garantit la stabilité du filtre LP.

Les suppositions de cette méthode sont les suivantes :

- Le signal est défini pour toutes les valeurs du temps ; il est identiquement nul en dehors d'une séquence de "N" échantillons, où "N" est un entier ; ceci équivaut à multiplier le signal de parole par une fenêtre de longueur finie correspondant à N échantillons.

$$s_f(n) = w(n).s(n) \qquad 0 \le n \le N-1$$
$$s_f(n) = 0, \qquad \text{ailleurs}$$

la fonction de pondération la plus courante est la fenêtre de Hamming :

$$w(n) = 0.54 - 0.46 \cos(2\pi n/(N-1)); \qquad 0 \le n \le N-1$$
$$w(n) = 0; \qquad \text{ailleurs.}$$

- Chaque échantillon peut être prédit approximativement à partir de p échantillons précédents. Ceci est valable pour toutes les valeurs du temps : $-\infty < n < +\infty$.

L'erreur quadratique totale entre le signal fenêtré et le modèle (signal prédit) est minimisée sur l'ensemble des échantillons.

Après la multiplication du signal parole avec la fenêtre d'analyse, les coefficients d'autocorrélations du segment parole fenêtré sont calculés. La fonction d'autocorrélation du signal fenêtré $s_f(n)$ est :

$$R(i) = \sum_{n=i}^{N-1} s_f(n) s_f(n-i) \qquad 0 \le i \le p \qquad (2.6)$$

La fonction d'autocorrélation est une fonction paire R(i) = R(-i).

Pour trouver les coefficients du filtre LPC, l'énergie du résiduel de prédiction sur l'intervalle fini $0 \le n \le N-1$ doit être minimisée :

$$E = \sum_{n=-\infty}^{\infty} e^2(n) = \sum_{n=-\infty}^{\infty} \left(s_f(n) - \sum_{k=1}^{p} a_k s_f(n-k)\right)^2 \qquad (2.7)$$

En annulant les dérivations partielles par rapport aux coefficients du filtre :

$$\frac{\partial E}{\partial a_k} = 0, \qquad 1 \le k \le p \tag{2.8}$$

on obtient p équations linéaires avec "p" coefficients inconnus $a_k$ :

$$\sum_{k=1}^{p} a_k \sum_{n=-\infty}^{\infty} s_f(n-i)s_f(n-k) = \sum_{n=-\infty}^{\infty} s_f(n-i)s_f(n) \qquad 1 \le i \le p \tag{2.9}$$

Alors, les équations linéaires peuvent être écrites sous la forme :

$$\sum_{k=1}^{p} R(|i-k|)a_k = R(i) \qquad 1 \le i \le p \tag{2.10}$$

Sous la forme matricielle, l'ensemble des équations linéaires est représenté par **R a = v** qui peut être réécrit en :

$$\begin{bmatrix} R(0) & R(1) & . & . & R(p-1) \\ R(1) & R(2) & . & . & R(p-2) \\ . & R(0) & . & . & . \\ . & . & . & . & . \\ R(p-1) & R(p-2) & . & . & R(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ . \\ . \\ a_p \end{bmatrix} = \begin{bmatrix} R(1) \\ R(2) \\ . \\ . \\ R(p) \end{bmatrix} \tag{2.11}$$

la matrice d'autocorrélation p x p obtenue est une matrice Toeplitz. L'algorithme de Levinson-Durbin est utilisé pour trouver les coefficients de prédiction minimisant la moyenne quadratique de l'erreur de prédiction.

## 2.3 Méthode de Covariance

les méthodes d'autocorrélation et de covariance diffèrent dans l'emplacement de la fenêtre d'analyse. Dans la méthode de covariance, le signal erreur est fenêtré au lieu du signal parole de façon que l'énergie à minimiser soit :

$$E = \sum_{n=-\infty}^{\infty} e_f^2(n) = \sum_{n=-\infty}^{\infty} e^2(n)w^2(n) \qquad (2.12)$$

En annulant les dérivations partielles par rapport aux coefficients du filtre $\delta E / \delta a_k = 0$ pour $1 \leq k \leq p$, on a "p" équations linéaires.

$$\sum_{k=1}^{p} \phi(i,k)a_k = \phi(i,0) \qquad 1 \leq i \leq p \qquad (2.12)$$

où la fonction de covariance $\phi(i , k)$ est définie par :

$$\phi(i,k) = \sum_{n=-\infty}^{\infty} w^2(n)s(n-i)s(n-k) \qquad (2.13)$$

Sous forme matricielle, les p équations deviennent $\Phi \mathbf{a} = \Psi$ , ou :

$$\begin{bmatrix} \phi(1,1) & \phi(1,2) & . & . & \phi(1,p) \\ \phi(2,1) & .\phi(2,2) & . & . & \phi(2,p) \\ . & . & . & . & . \\ . & . & . & . & . \\ \phi(p,1) & \phi(p,2) & . & . & \phi(p,p) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ . \\ . \\ a_p \end{bmatrix} = \begin{bmatrix} \psi(1) \\ \psi(2) \\ . \\ . \\ \psi(p) \end{bmatrix} \qquad (2.14)$$

Où $\Psi(i) = \phi(i,0)$ pour $1 \leq i \leq p$

La matrice $\Phi$ n'est pas une matrice Toeplitz, elle est symétrique et définie positive. La matrice de covariance peut être décomposée en matrices triangulaires supérieures et inférieures :

$$\Phi = \mathbf{L}\,\mathbf{U} \qquad (2.15)$$

la décomposition de Cholesky est utilisée pour convertir la matrice de covariance en :

$$\Phi = \mathbf{C}\,\mathbf{C}^T \qquad (2.16)$$

ou $\mathbf{C} = \mathbf{L}$ et $\mathbf{C}^T = \mathbf{U}$. Le vecteur a est trouvé en résolvant d'abord l'équation :

$$\mathbf{L}\,y = \Psi \qquad (2.17)$$

puis :

$$\mathbf{U}\,a = y \qquad (2.18)$$

## 2.4 Algorithme de résolution

Nous admettons que la fonction d'autocorrélation R(k) est connue pour k = 0, 1,..., p. La moyenne du signal est supposée nulle; dans le cas contraire, elle est estimée et soustraite.

Il s'agit donc de résoudre le système linéaire (2.11), c'est à dire en fait d'inverser une matrice d'ordre "p". Les méthodes algébriques classiques exigent pour cela un nombre d'opérations (multiplication + addition) de l'ordre de $p^3$, ce que l'on note $O(p^3)$. L'algorithme qui va être décrit profite de la structure particulière (Toeplitz symétrique) de la matrice d'autocorrélation pour résoudre (2.11) par une récurssion sur l'ordre de prédiction : autrement dit, ils fournissent toutes les solutions d'ordre m = 1,2,..,p, Le nombre d'opérations est seulement $O(p^2)$.

La variance de l'erreur de prédiction $\alpha_p$ sera obtenue également par une récurrence sur l'ordre m.

***Algorithme de Levinson-Durbin.***

Rappelons que la fonction d'autocorrélation est supposée connue et que pour un signal stationnaire, on a:

$$R(i,j) = R(|i-j|) = R(k) \tag{2.19}$$

Initialisation :

$$a_m(0) = 1, \quad (m = 1,2,\ldots,p) \quad E_0 = R(0) = \sigma_x^2 \tag{2.20}$$

Récursion :

pour m = 1, 2, ..., p.

$$k_m = -\frac{1}{E_{m-1}}\left[R(m) - \sum_{k=1}^{m-1}\alpha_{m-1}(k)\, R(m-k)\right] \tag{2.21}$$

pour $k = 1, 2, \ldots, m-1$

$$\alpha_k(m) = \alpha_k(m-1) - k_m\alpha_{m-k}(m-1) \tag{2.22}$$

$$E_m = E_{m-1}(1-k_m^2) \tag{2.23}$$

les coefficients $\alpha_k(m)$ résultants, quand m = p, représentent les coefficients de prédiction d'un prédicteur linéaire d'ordre p :

$$a_k = \alpha_k(p) \qquad 1 \le k \le p$$

## 2.5 Analyse par synthèse : Codeurs prédictifs linéaires excités par codes (CELP)

Les codeurs de la parole sont des algorithmes qui compressent les représentations numériques des signaux de parole pour minimiser le nombre de bits nécessaires pour les représenter.

Ceci est atteint en tirant profit, des degrés de variations, des redondances dans le signal parole et de certaine propriétés de l'audition humaine.

Les applications des codeurs peuvent être classées dans deux catégories : la transmission numérique des signaux de parole et leur stockage. Dans la première catégorie, on peut citer les systèmes de communication, la radio mobile, la téléphonie cellulaire et les systèmes vocales sécurisés. Dans cette catégorie, les conditions du canal, le retard et le débit binaire sont des facteurs importants. Dans la seconde catégorie, des applications comme les machines à réponse vocale, la qualité de la parole et la capacité de stockage sont en général les facteurs les plus critiques.

Pour la sélection d'un codeur de la parole, il faut prendre en considération certains facteurs et faire certains compromis. Les plus importants facteurs (comme on l'a déjà vu) sont la qualité du signal synthétisé, la complexité du codeur, le débit binaire; et le retard de codage. En général, l'obtention de la très haute qualité de la parole à des bas débits s'accompagne d'une haute complexité. La diminution de la complexité du codeur s'accompagne généralement d'un accroissement du débit binaire ou d'une perte de la qualité.

Pour minimiser la distorsion de codage perçue, les systèmes de codage modernes tiennent compte des caractéristiques du système auditif ; du "conduit vocal" et du langage.

En général, trois caractéristiques distinctes du système auditif peuvent être utilisées.

Celles ci sont l'effet de masquage du bruit, la sensibilité variable de la perception auditive vis-à-vis des fréquences et l'insensibilité relative de l'oreille par rapport à la phase.

La plus simple technique de codage de la parole est la Modulation d'Impulsions Codées, (PCM), Elle ne tient compte d'aucunes hypothèses sur les caractéristiques du signal à coder(sauf la dynamique du signal). Les codeurs utilisant cette technique doivent avoir un haut débit binaire pour générer la bonne qualité de la parole. Par contre, ils ont l'avantage d'être utilisés pour coder des signaux autres que ceux de la parole vu qu'ils ne font aucune préférence sur les classes de signaux.

Les systèmes comme la modulation d'impulsions codées différentielle (DPCM) et la modulation delta utilise un modèle statistique stationnaire long terme pour la production de la parole. La Modulation d'Impulsions Codées Différentielle Adaptative (ADPCM) et la modulation delta adaptative utilise aussi la nature de la variation lente de l'énergie court terme, provoquant la corrélation restreinte du bruit avec le signal.

Plusieurs formes de codeurs par transformée adaptative et de codeurs prédictifs adaptatifs (souvent dit vocodeur excité par le résidu) utilisent les caractéristiques auditives et du "conduit vocal". Ces systèmes génèrent une très bonne qualité de la parole à des moyens et bas débits. Il en est de même pour les nouveaux codeurs paramétriques, comme les Codeurs Prédictifs Linéaires Excités par des Impulsions Multiples (MPLPC) et les codeurs prédictifs linéaires excités par codes (CELP), qui sont les plus utilisés et les plus étudiés.

### 2.5.1 Principe des codeurs CELP

Comme nous l'avons déjà signalé, le signal de la parole n'est pas stationnaire. Ses propriétés statistiques varient au cours du temps. Alors, pour le traiter, on doit faire une approximation qui consiste à le considérer comme localement stationnaire sur des intervalles de temps de l'ordre de 10 à 30 ms.

Des fenêtres d'analyse sont alors introduites de longueur 80 à 240 échantillons si l'on choisit une fréquence d'échantillonnage de 8 kHz. On dispose ainsi de "N" échantillons qu'on utilisera pour extraire les caractéristiques du modèle. Ces dernières, codées à l'émetteur et transmises dans le canal de transmission, sont décodées au récepteur pour reproduire les "N échantillons".

Ces caractéristiques représentent des valeurs moyennes sur cet intervalle de temps. On répète indéfiniment ce traitement pour les fenêtres qui suivent [19].
Le principe de la modélisation est donné en figure (2.1).

On cherche à construire un signal synthétique $\bar{s}_0 \ldots . \bar{s}_{N-1}$ le plus ressemblant possible au signal de départ $s_0 \ldots . s_{N-1}$ sur l'ensemble de la fenêtre d'analyse.

Il faut donc définir le mode de construction du signal synthétique et préciser le critère de ressemblance. Une règle de construction possible est le passage d'un signal $\bar{r}_0 \ldots . \bar{r}_{N-1}$ au travers d'un système.

Le choix se rapporte en général sur des filtres linéaires, invariants (dans la fenêtre d'analyse) et causaux. Ensuite, on détermine les valeurs numériques des différents paramètres qui interviennent dans ce modèle. On minimise un critère de ressemblance entre le signal réel et le signal synthétique. Pour des raisons de simplicité, on choisit presque toujours un critère quadratique de la forme :

$$E = \sum_{n=0}^{N-1} (s_n - \bar{s}_n)^2 = \sum_{n=0}^{N-1} \varepsilon_n^2 \qquad (2.24)$$

Ici, il faut déterminer $\bar{r}_0 \ldots . \bar{r}_{N-1}$ et les coefficients du filtre : on note que c'est un problème de déconvolution. Vu qu'on ne peut déterminer simultanément l'entrée et les coefficients du filtre, une solution sous optimale consiste à déterminer d'abord les coefficients du filtre en faisant des hypothèses très simplificatrices au niveau de l'entrée puis on raffine le modèle de l'entrée.

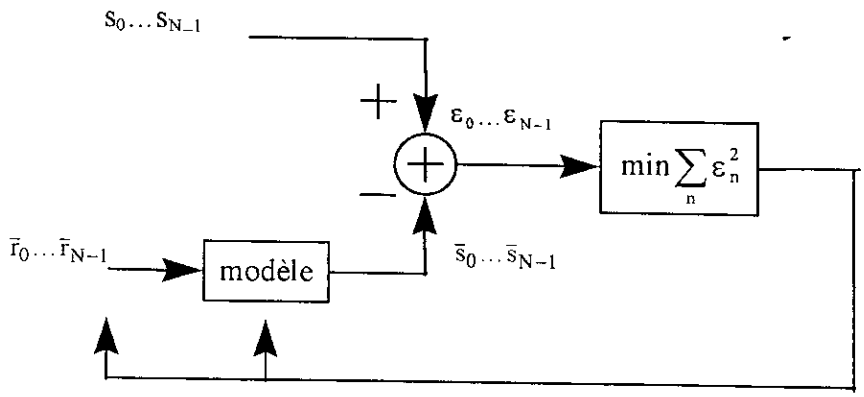



**Figure 2.1.** Principe de modélisation paramétrique [19]

## 2.5.2 Modèle de synthèse de la parole dans un codeur CELP

La synthèse de la parole dans un codeur CELP [1], [2] et [20] est identique à celle utilisée dans les codeurs prédictifs adaptatifs [2]. Elle consiste en deux filtres récursifs

linéaires à coefficients variables dans le temps, ayant chacun un prédicteur dans la boucle de contre réaction figure (2.2). La première boucle contient un prédicteur long-terme qui génère les périodicités du pitch d'un son voisé, la seconde contient un prédicteur court-terme pour restaurer l'enveloppe spectrale.

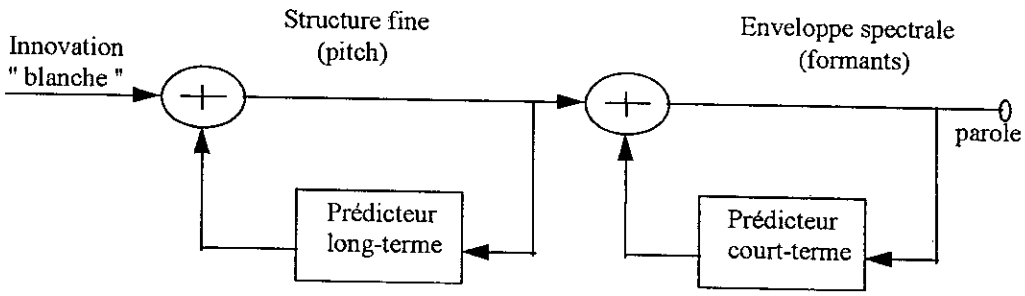



**Figure 2.2.** Modèle de synthèse de la parole avec prédicteur long-terme et court-terme d'après [1]

Les deux prédicteurs sont déterminés en utilisant les méthodes décrites dans les références [21] et [22]. Le prédicteur court-terme a, par exemple, 16 coefficients et est déterminé en utilisant en analyse LPC la méthode de covariance stabilisée pondérée toutes les 10 ms [2] et [22]. Dans cette méthode, l'erreur de prédiction instantanée est pondérée par une fenêtre de Hamming de 20 ms et les coefficients sont obtenus en minimisant l'énergie de l'erreur pondérée.

Le prédicteur long-terme est souvent appelé prédicteur pitch, yu que son rôle principale est d'exploiter les périodicités du pitch dans la parole voisée.
La forme générale de ce prédicteur est :

$$\frac{1}{P(z)} = \frac{1}{1 - \sum_{j=-1}^{1} \beta_j z^{-(M+j)}} \qquad (2.25)$$

Où $\beta_j$ : les coefficients gain.

M : la période pitch ( $20 \leq M \leq 147$ )

J : ( prend les valeurs -1, 0, et 1).

Le décalage M approxime la périodicité du signal. Le gain $\beta$ peut être interprété comme un indicateur du "niveau de périodicité" avec $\beta$ approchant la valeur 1 pour des signaux "très périodique". Ces paramètres sont déterminés par analyse-synthèse (boucle

fermée). L'estimation initiale des paramètres est souvent déterminée avec des méthodes en boucle ouverte.

Pour les sons voisés, la fréquence du pitch varie de 54 Hz à 400 Hz. Pour une fréquence d'échantillonnage de 8 kHz, l'échelle de décalage est comprise entre 20 et 147 échantillons (128 délais possibles) ce qui nécessite 7 bits pour le codage.

L'utilisation du prédicteur pitch est importante dans le codeur CELP. Sans lui, le dictionnaire aléatoire ne peut générer efficacement les composantes périodiques dans le signal d'excitation [23], [24] et [25]. En effet, cette périodicité correspond, physiologiquement, à la période de vibrations des cordes vocales.

### 2.5.3 Filtrage perceptuel et critère de minimisation :

Les fonctions de coût quadratique se prêtent bien aux calculs : elles possèdent la bonne propriété de fournir un système linéaire lorsque l'on dérive ce critère par rapport aux paramètres inconnus. Par contre, ce critère n'est pas forcément bien adapté au système auditif humain.

Pour pallier à cet inconvénient, un critère expérimental a été mis au point par Atal et al [26]. Il tend à donner au spectre du signal d'erreur, ou différence entre le signal reconstruit et le signal original, une forme voisine du spectre de l'original un peu étalé, en exploitant le phénomène psychoacoustique de masquage du bruit. Ce phénomène est la traduction dans le domaine fréquentiel de la remarque suivante : la différence entre le signal original et celui reconstruit est moins perceptible dans les zones ou le signal a beaucoup d'énergie. On cherche donc une fonction de pondération qui attribue moins d'importance aux zones fréquentielles énergétiques (c'est à dire zones formantiques).

La fonction de transfert $W(z) = \dfrac{A(z)}{A(z/\gamma)}$ avec $0 < \gamma < 1$ joue ce rôle.

En effet, si on note :

$$A(z) = 1 + a_1 z^{-1} + \ldots\ldots\ldots\ldots a_p z^{-p} = \prod_{i=1}^{p} (1 - p_i z^{-1}) \qquad (2.25)$$

on remarque que :

$$A(\frac{z}{\gamma}) = 1 + a_1 \gamma\, z^{-1} + \ldots\ldots\ldots\ldots a_p \gamma^p\, z^{-p} = \prod_{i=1}^{p} (1 - \gamma\, p_i z^{-1}) \qquad (2.26)$$

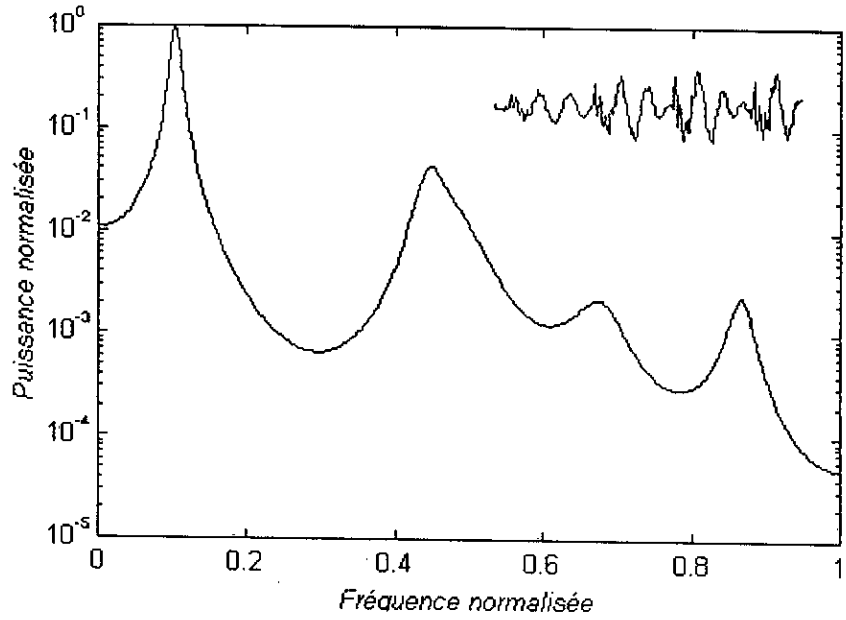



**Figure 2.3.** a) Spectre du signal et b) réponse en fréquence de la fonction de pondération $W(z) = A(z)/A(z/\gamma)$ avec $\gamma = 0.8$.

Le module de la réponse en fréquence du filtre $1/A(z/\gamma)$ présente des pics moins accentués que celui du filtre $1/A(z)$ puisque les pôles du filtre $1/A(z/\gamma)$ sont ramenés vers le centre du cercle unité par rapport à ceux du filtre $1/A(z)$. Le module de la fréquence du filtre $W(z) = \frac{A(z)}{A(z/\gamma)}$ a donc la forme souhaitée.

### 2.5.4 Sélection de la séquence optimale dans le dictionnaire

Considérons par exemple, le codage d'un bloc de signal parole échantillonné à 8 kHz et de durée 5 ms. Chaque bloc consiste en 40 échantillons. Un débit binaire de 1/4 bits par échantillon correspond à 1024 séquences possibles de longueur 40 pour chaque bloc.

La procédure de sélection de la séquence optimale est illustrée à la figure (2.4). Chaque élément du dictionnaire fournit 40 échantillons du signal innovation. Celui-ci est mis à l'échelle par un facteur d'échelle. Ce facteur est réactualisé à une nouvelle valeur chaque cycle d'adaptation. Les échantillons ainsi mis à l'échelle sont filtrés séquentiellement à travers les deux filtres récursifs, pour introduire la périodicité des sons voisés et l'enveloppe spectrale. Les échantillons du signal de parole régénérés à la sortie du deuxième filtre sont comparés aux échantillons correspondant du signal de parole original pour former le signal différence.

Ce dernier, représentant l'erreur objective, est alors traité à travers le filtre perceptuel de pondération. Pour chaque innovation, on détermine le vecteur erreur puis on calcule sa valeur quadratique moyenne correspondante. L'innovation fournissant la valeur minimale de l'erreur est sélectionnée, son indice est alors transmis avec les paramètres des filtres court-terme et long-terme et l'indice représentant la quantification du gain. La procédure est répétée pour chaque nouveau vecteur de parole.

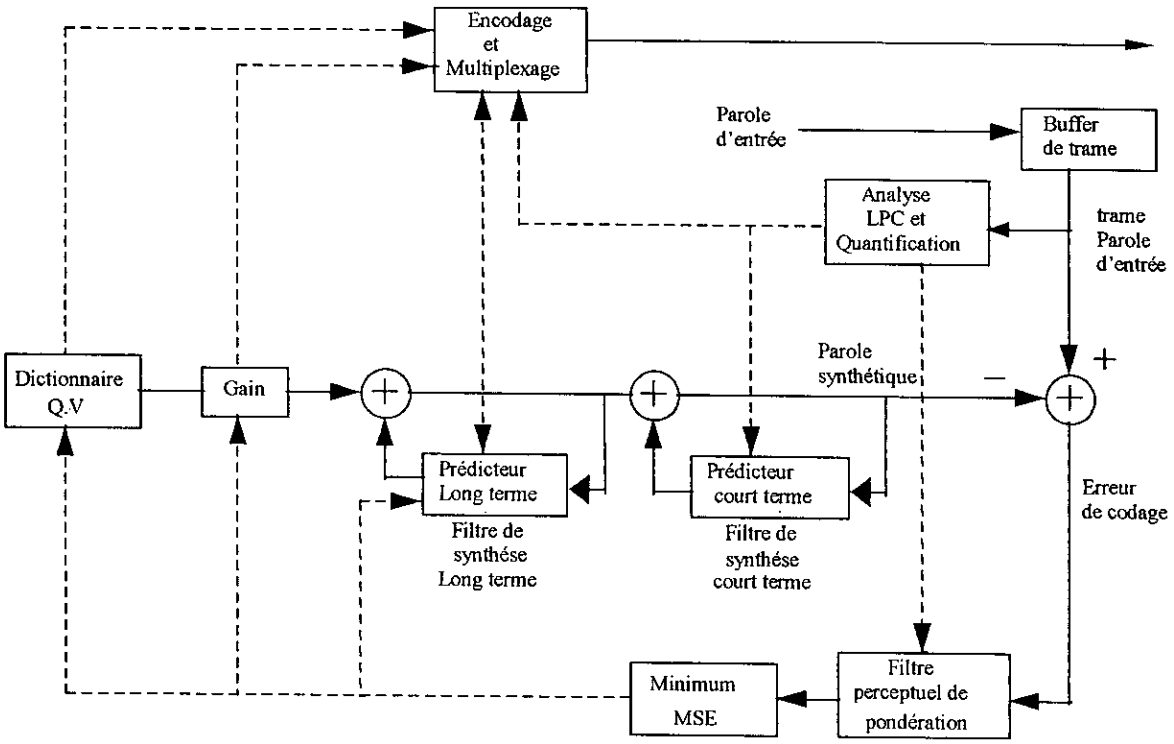



**Figure 2.4.** Procédure pour trouver la séquence optimale.

## 2.6 Conclusion

Dans les codeurs CELP, la synthèse de la parole consiste en deux filtres récursifs linéaires à coefficients variables dans le temps, ayant chacun un prédicteur dans la boucle de contre réaction. Un prédicteur long-terme génère les périodicités du pitch d'un son voisé, le second prédicteur (court-terme) restaure l'enveloppe spectrale. Les coefficients de ce dernier sont déterminés par la méthode d'autocorrélation ou la méthode de covariance. Le principe de recherche dans le dictionnaire consiste en une minimisation de l'erreur quadratique moyenne. L'erreur est la différence entre les signaux pondérés (original et synthétique).

procédure que l'encodeur. Ainsi, il n'est plus nécessaire de transmettre des bits d'information externes (side information) pour spécifier ces coefficients ;

- il n'est pas nécessaire de bufferiser 20 ms du signal de parole d'entrée pour l'analyse LPC. Par conséquent le vecteur d'excitation devient l'unité de base de bufferisation. Le retard de codage est ainsi hautement réduit [11].

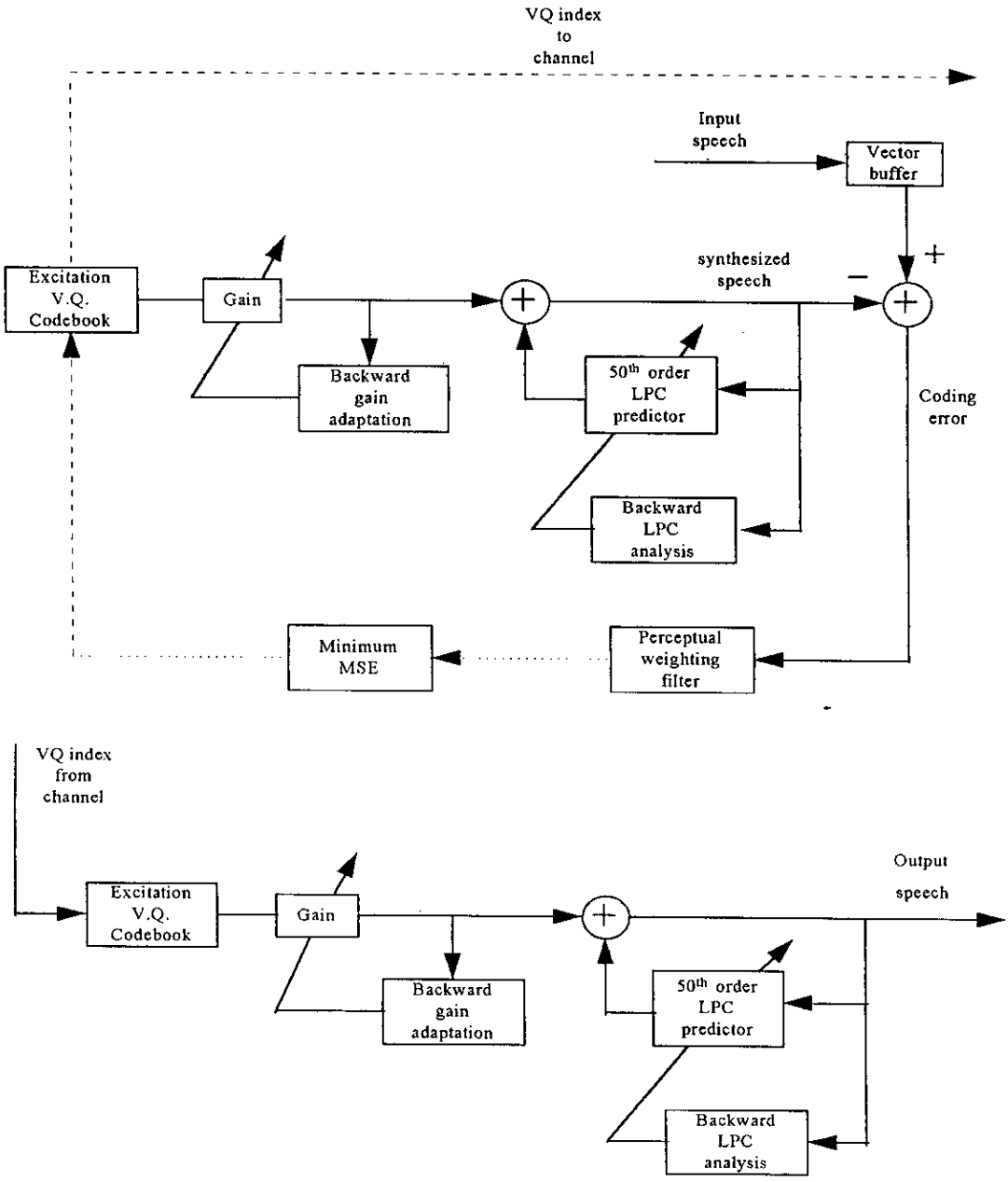



**Figure 3.1.** Codeur et Décodeur LD-CELP (Code Excited Linear Prediction) d'après [27].

Avec une fréquence d'échantillonnage standard de 8 kHz, un retard de 2 ms correspond à 16 échantillons. Sachant que le retard de codage est environ trois fois la durée de la dimension du vecteur, la dimension du vecteur qu'on peut utiliser est de 5 échantillons (0.625 ms). Avec un vecteur de dimension 5 et un débit binaire de 2 bits/échantillon, on a seulement 10 bits pour coder l'excitation.

## 3.2 Prédicteur LPC d'ordre supérieur

Les codeurs CELP conventionnels utilisent un prédicteur long-terme adaptatif progressif pour exploiter les redondances du pitch dans le signal de parole. Dans le LD-CELP, vue la contrainte de taille limitée du bloc, il est naturel, si nous songeons à utiliser un prédicteur pitch, de le faire d'une façon adaptative backward [28] et [29]. Cependant, l'adaptation backward du prédicteur pitch est très sensible aux erreurs de canal. Cette mauvaise adaptation (divergence) due à la présence des bits erreurs nous a conduit à supprimer le prédicteur pitch [27].

Cependant, l'élimination de ce prédicteur affecte beaucoup plus la qualité de la parole prononcée par un locuteur féminin que celle prononcée par un locuteur masculin [27]. Pour compenser cette perte de qualité, nous avons augmenté l'ordre du prédicteur LPC de 10 à 50. La motivation de ce choix est d'exploiter les redondances du pitch dans la parole prononcée par un locuteur féminin. Pour un locuteur masculin le gain de prédiction se sature à l'ordre 20 alors que pour un locuteur féminin cet ordre n'atteint la saturation qu'à l'ordre 50 (figure 3.2).

Le choix de l'ordre 50 du filtre LPC est dû aux justifications suivantes :

- l'analyse LPC adaptative régressive est apparue plus robuste aux erreurs de canal même après l'augmentation de l'ordre LPC à l'ordre 50 [27];
- on n'est plus obligé d'allouer des bits pour les 50 coefficients du modèle LPC vu qu'ils sont adaptés en backward ;
- le gain de prédiction est généralement saturé pour des ordres supérieurs à l'ordre 50
- un autre avantage possible est apparu, en éliminant le prédicteur pitch et en utilisant à la place un prédicteur LPC d'ordre 50, le codeur devient moins spécifique pour le signal parole parce qu'il ne tient compte d'aucune quasi-périodicité du pitch dans le signal d'entrée et peut de ce fait fonctionner assez bien avec des signaux autres que celui de la parole (comme les données MODEM).

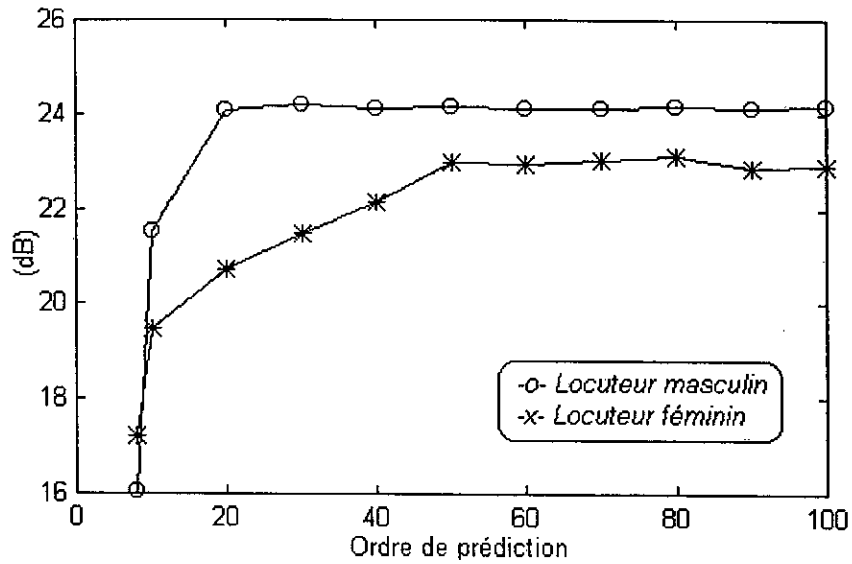



**Figure 3.2.** Evolution du gain de prédiction en fonction de l'ordre de prédiction du filtre de synthèse.

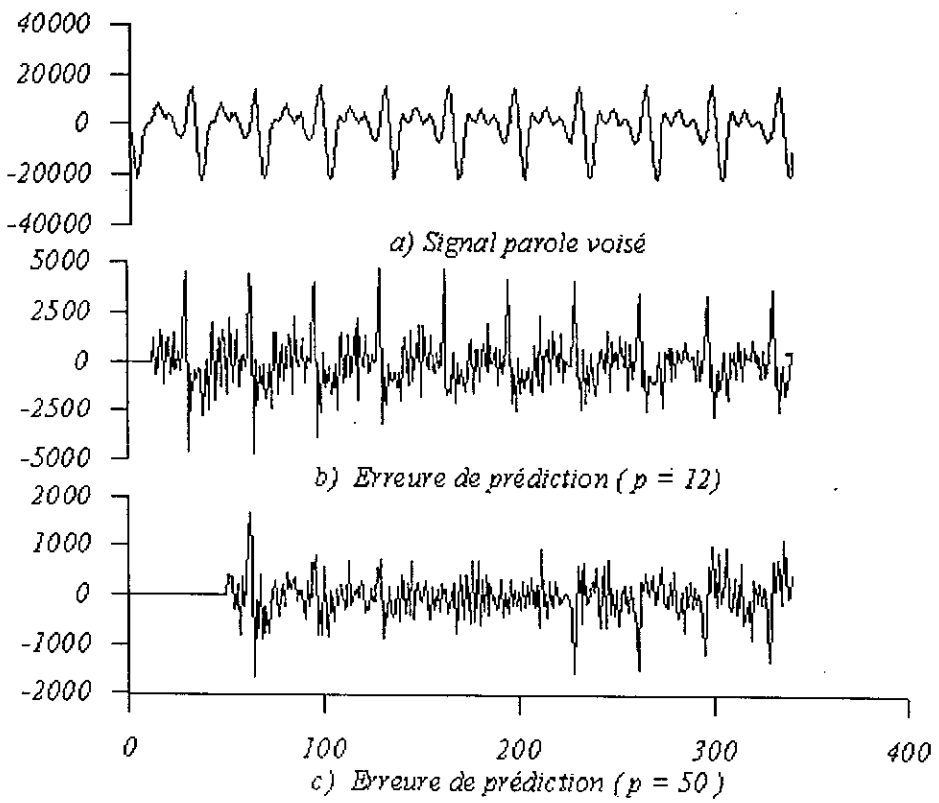



**Figure 3.3.** Structure Pitch dans l'erreur de prédiction pour des ordres de prédiction du filtre de synthèse p = 12 et p = 50.

## 3.3 Opération de fenêtrage : Fenêtre de Barnwell

En général, pour l'analyse LPC, une fenêtre de Hamming est utilisée. Cette méthode nécessite une opération de bufferisation, et un calcul intensif (multiplication et addition) est nécessaire pour l'estimation de la fonction d'autocorrélation. La fenêtre de Hamming a été remplacée par la version modifiée de la fenêtre de Barnwell qui est la réponse impulsionnelle d'un filtre numérique récursif.

Cette méthode est une alternative pour le calcul de la fonction d'autocorrélation d'une façon récursive. Vue que la matrice d'autocorrélation est de type Toeplitz, la résolution du système d'équations peut être accomplie par récursion (sans inversion de matrice) en utilisant la méthode de Levinson-Durbin.

La forme de la fenêtre w(n) doit être de telle sorte que le signal dans le passé immédiat soit plus pondéré que le signal dans le passé lointain. Ceci nous assure que le prédicteur agit selon les modes de changement de la stationnarité du signal de parole d'entrée.

Les paramètres importants dans le choix d'une fenêtre sont la forme et la longueur effective de cette fenêtre. Un autre facteur à prendre en considération est la complexité de calcul de l'algorithme de réactualisation. Une méthode pour réduire cette complexité dans la procédure de réactualisation est d'utiliser une fenêtre w(n) qu'on peut considérer comme la réponse impulsionnelle d'un filtre causal d'ordre fini. L'utilisation de ces fenêtres nous permet d'avoir la possibilité d'obtenir des équations de réactualisation récursives [30].

Soit s(n) la séquence d'entrée, divisée en trames, à un intervalle fixe. Soit m l'indice du dernier échantillon dans une certaine trame, et soit w(n), le n $^{ieme}$ échantillon de la fonction fenêtre, de façon que w(n) = 0, pour n < 0 et w(n) est indicé en backward dans le temps.

Le signal de parole fenêtré S(n,m), où "n" est l'indice temps et "m" le numéro de la fenêtre, est donné par:

$$S(n,m) = s(n).w(m-n) \qquad (3.1)$$

la fonction d'autocorrélation exacte du signal fenêtre est déterminée à partir de :

$$R(k,m) = \sum_{n=-\infty}^{+\infty} S(n,m)\, S(n+k,m) \qquad (3.2)$$

Où R(k , m) est le $k^{ieme}$ coefficient d'autocorrélation pour la fenêtre m. Si la longueur de la fenêtre est finie, alors les limites de la sommation dans (3.2) sont aussi finies. Ces coefficients d'autocorrélation sont alors utilisés comme entrée pour l'algorithme d'inversion de la matrice de Toeplitz pour trouver les paramètres du filtre LPC.

En général pour une bonne qualité de la parole les fenêtres doivent être recouvrantes (overlapping). Ainsi, plusieurs échantillons de la parole sont utilisés dans différentes trames successives pour avoir les fonctions d'autocorrélation. De plus, l'opération de recouvrement conduit à une augmentation considérable de bufferisation et de calcul. Ceci peut être évité si le besoin d'une longueur de fenêtre finie est écarté. D'où l'intérêt de la classe des fenêtres de longueur infinie. L'une des fenêtres de cette classe peut être formée à partir de la réponse impulsionnelle d'un filtre numérique de second ordre ayant deux pôles réels.

La transformée en z correspondant à ce filtre est :

$$H(z) = \frac{1}{(1-\alpha z^{-1})(1-\beta z^{-1})} \qquad (3.3)$$

Où $\alpha$ et $\beta$ sont les lieux des pôles.

En appliquant (3.1) à (3.2) les fonctions d'autocorrélations pour les séquences fenêtrées peuvent être écrites comme suit :

$$R(k,m) = \sum_{n=-\infty}^{n=+\infty} S(n)\, S(n+k)\, w(m-n)\, w(m-n-k) \qquad (3.4)$$

$$\begin{aligned} &\text{soit :} && W(n,k) = w(n).w(n-k) \\ &\text{et} && S(n,k) = s(n).s(n+k) \end{aligned} \qquad (3.5)$$

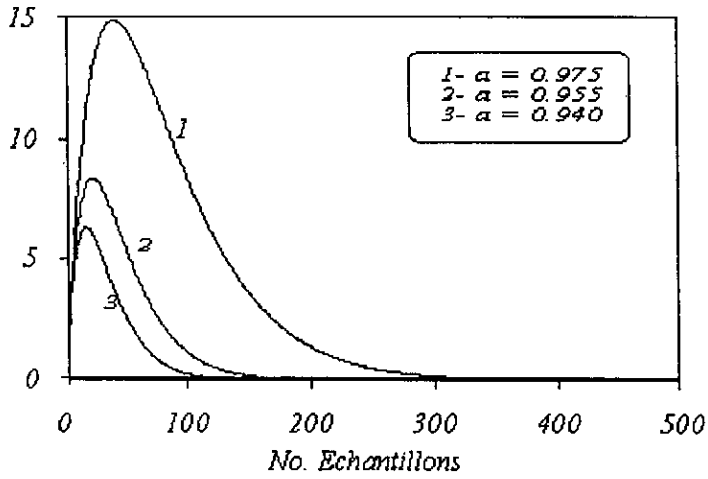



**Figure 3.4.** Exemples de réponses impulsionnelles pour des filtres IIR avec un pôle réel double à $z = -\alpha$

l'expression 3.4 devient :

$$R(k,m) = \sum_{n=-\infty}^{n=+\infty} S(n,k)\, W(m-n,k) \qquad (3.6)$$

A partir de cette équation, on peut exprimer le $k^{ieme}$ coefficient d'autocorrélation comme la convolution de la séquence {S(n,k)} et la fonction W(n,k). Sachant que W(n,k) est le produit de deux fonctions fenêtres, alors, $W_k(z)$ ( transformée en z de W(n,k) ) est donnée par la convolution des transformées en z des deux fonctions fenêtres w(n) et w(n-k).

Si la fenêtre est supposée de longueur infinie et est permise d'être la réponse impulsionnelle d'un filtre de second ordre dont la transformée en z est donnée par H( z ), alors $W_k(z)$ peut être écrit comme :

$$W_k(z) = \frac{1}{2\pi j} \oint H(v)\, H\left(\frac{z}{v}\right) v^{-k-1}\, dv \qquad (3.7)$$

Si H(z) est pris comme la fonction de transfert du filtre de second ordre donnée par (3.3 ), $W_k(z)$ devient :

$$W_k(z) = \frac{1}{2\pi j} \oint \frac{\nu^{-k-1}}{\left(1-\alpha\nu^{-1}\right)\left(1-\beta\nu^{-1}\right)\left(1-\alpha\frac{\nu}{z}\right)\left(1-\beta\frac{\nu}{z}\right)} d\nu \qquad (3.8)$$

l'évaluation de cette expression donne :

$$W_k(z) = \frac{b(0,k) + b(1,k).z^{-1}}{1 - a(1,k).z^{-1} - a(2,k).z^{-2} - a(3,k).z^{-3}} \qquad (3.9)$$

Avec :

$$b(0,k) = \frac{\alpha^{k+1} - \beta^{k+1}}{\alpha - \beta} \qquad (3.10)$$

$$b(1,k) = \frac{\alpha^2\beta^{k+1} - \beta^2\alpha^{k+1}}{\alpha - \beta} \qquad (3.11)$$

$$a(1,k) = \alpha^2 + \beta^2 + \alpha\beta$$

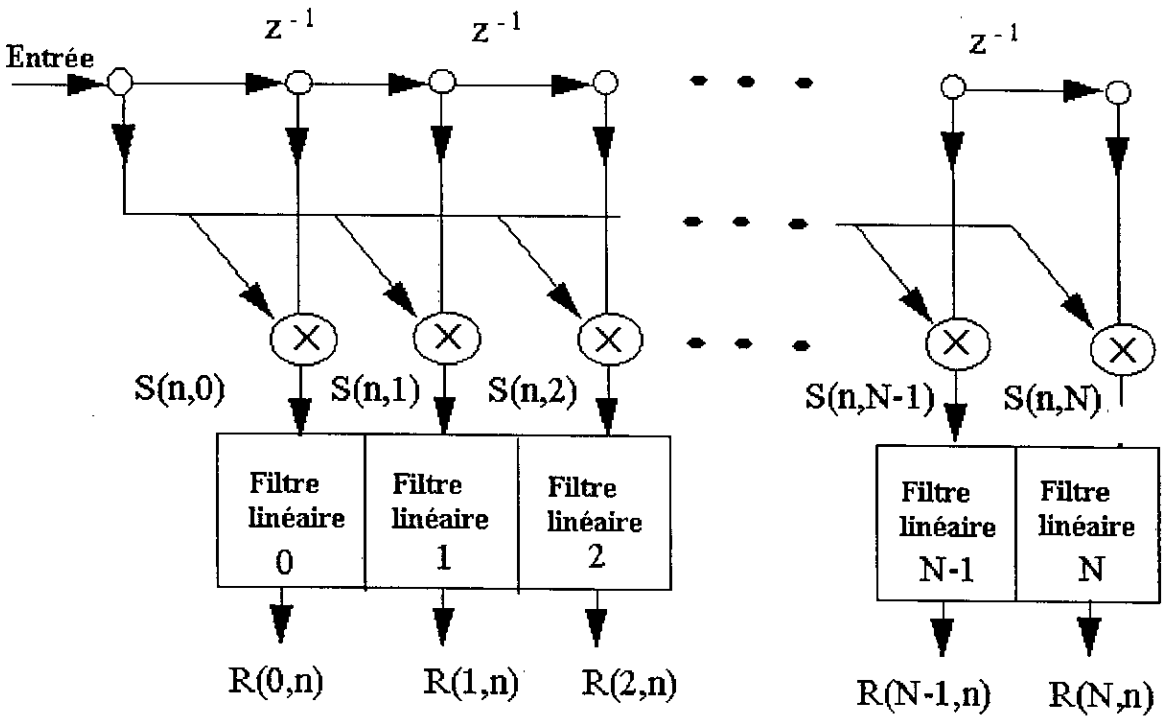
$$a(2,k) = -(\alpha^2\beta^2 + \alpha^3\beta + \alpha\beta^3) \qquad (3.12)$$

$$a(3,k) = \alpha^3\beta^3$$

Dans le cas où $\alpha = \beta$ , (on fait tendre $\alpha$ vers $\beta$) on a :

$$\begin{aligned} b(0,k) &= (k+1).\alpha^k \\ b(1,k) &= -(k-1).\alpha^{k+2} \end{aligned} \qquad (3.13)$$

$$\begin{aligned} a(1,k) &= 3.\alpha^2 \\ a(2,k) &= -3.\alpha^4 \\ a(3,k) &= \alpha^6 \end{aligned} \qquad (3.14)$$




**Filtre lineaire**

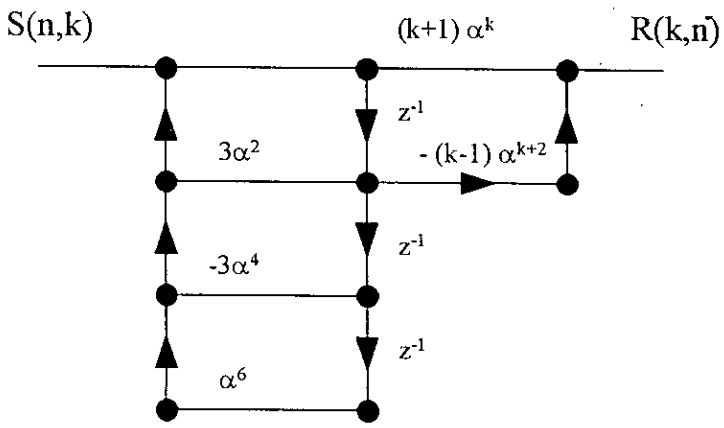



**Figure 3.5.** Structure pour le calcul récursif de la fonction d'autocorrélation estimée pour une analyse d'ordre N.

**Remarques sur la structure récursive :**

- c'est un système point à point qui opère d'une façon identique sur chaque échantillon, il n'y a pas de bufferisation additionnelle autre que celle mentionnée à la figure 3.5 ;
- le paramètre $\alpha$ contrôle entièrement la longueur de la fenêtre ;
- les deux multiplications dans la partie non récursive du filtre [$(k+1)\alpha^k$ et $(k-1)\alpha^{k+2}$] sont calculées une seule fois par trame et non sur chaque échantillon ;
- la logique de contrôle pour la structure entière est très simple. Vu que toutes les informations concernant la fenêtre sont contenues dans les coefficients du filtre linéaire, alors un stockage extensif de la fonction fenêtre n'est pas nécessaire.

D'après cette méthode, on peut calculer les coefficients d'autocorrélation récursivement en passant le signal de parole à travers la structure spéciale du banc de filtres (fig. 3.5).

Pour une analyse LPC d'ordre 50, il y a 51 filtres de forme directe d'ordre 3 dans le banc de filtres, un pour chaque coefficient d'autocorrélation. Cependant, l'obtention de ces coefficients par la méthode de Barnwell telle qu'elle est exposée, provoque des problèmes de précision numérique dans la séquence de l'analyse LPC d'ordre 50 dans le cas où l'on travaille en virgule fixe [11].

Ce problème est contourné en remplaçant le filtre de forme directe d'ordre 3 par des filtres du 1$^{er}$ ordre en cascade [27].
Les fonctions de transfert de ces filtres du 1$^{er}$ ordre sont respectivement :

$$\frac{1}{1-\alpha^2 z^{-1}},\ \frac{1}{1-\alpha^2 z^{-1}}\ \text{et}\ \frac{(k+1)\alpha^k-(k-1)\alpha^{k+2}z^{-1}}{1-\alpha^2 z^{-1}} \qquad (3.15)$$

où $\alpha$ est le paramètre de contrôle de la forme de la fenêtre qui sera fixé ultérieurement.

## 3.4 Adaptation logarithmique du gain

Pour obtenir une échelle dynamique adéquate du signal synthétisé avec une taille réduite du dictionnaire d'excitation, nous avons utilisé une technique de prédiction du

gain adaptatif en backward [31]. Cette procédure permet de suivre assez fidèlement la puissance court-terme du signal parole.

$$e(n) = \sigma(n).y(n)$$
$$\text{et} \quad \log[\sigma_e(n)] = \log[\sigma(n)] + \log[\sigma_y(n)] \tag{3.16}$$

Où y(n) : mot code excitation (appartenant au dictionnaire de 10 bits) à l'instant n.

e(n) : version gain mise à l'échelle.

$\sigma_y(n)$ : valeur **RMS** (**R**oot **M**ean **S**quare) de y(n).

$\sigma_e(n)$ : valeur RMS de e(n).

$\sigma$ (n) : gain d'excitation adaptatif en backward, utilisé pour mettre en échelle y(n).

Le but de ce prédicteur est d'obtenir $\sigma(n)$ aussi proche que possible de $\sigma_e(n)$ en exploitant les informations disponibles à l'instant n.

Soit $\log[\sigma(n)]$ la prédiction de $\log[\sigma_e(n)]$ en se basant sur $\log[\sigma_e(n-1)]$, $\log[\sigma_e(n-2)]$,...

$$\log[\sigma(n)] = \sum_{i=1}^{10} p_i.\log[\sigma_e(n-i)] \tag{3.17.a}$$

$$\log[\sigma(n)] = \sum_{i=1}^{10} p_i.\log[\sigma(n-i)] + \sum_{i=1}^{10} p_i.\log[\sigma_y(n-i)] \tag{3.17.b}$$

Notons que $\log[\sigma(n)]$ peut être vu comme la sortie d'un filtre pôle-zéro d'ordre 10 avec $\log[\sigma_y(n-1)]$ comme entrée. En réactualisant les coefficients $p_i$ du prédicteur log-gain à travers une analyse LPC régressive adaptative sur la séquence antérieure $\log[\sigma_e(n)]$} (avec soustraction d'un gain offset de 32 dB), la méthode d'autocorrélation garantit la stabilité du filtre pôle-zéro définie par l'équation (3.17.b).

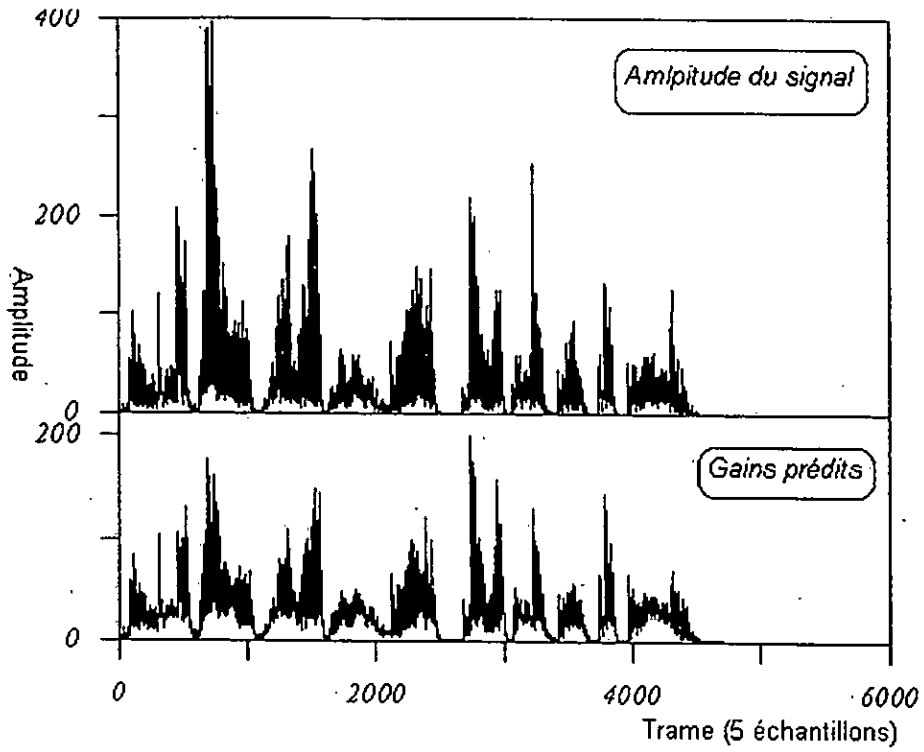



**Figure 3.6.** Variation du Gain de mise à l'échelle par rapport à l'amplitude du signal de parole à coder.

## 3.5 Filtre perceptuel de pondération

Comme dans le CELP conventionnel, un filtre perceptuel pour le masquage du bruit par les formants est utilisé.

Soit le prédicteur d'ordre 10 représenté par la fonction de transfert :

$$Q(z) = \sum_{i=1}^{10} q_i z^{-i} \tag{3.18}$$

Où $q_i$ sont les coefficients du prédicteur.

Les coefficients du filtre perceptuel sont calculés selon les équations suivantes :

$$W(z) = \frac{1 - Q(\frac{z}{\gamma_1})}{1 - Q(\frac{z}{\gamma_2})} \quad 0 < \gamma_2 < \gamma_1 < 1 \tag{3.19}$$

$$Q\left(z/\gamma_1\right) = \sum_{i=1}^{10} \left(q_i \gamma_1^i\right) z^{-i} \qquad (3.20)$$

$$Q\left(z/\gamma_2\right) = \sum_{i=1}^{10} \left(q_i \gamma_2^i\right) z^{-i} \qquad (3.21)$$

Le filtre perceptuel de pondération est un filtre pôle-zéro d'ordre 10 défini par la fonction de transfert w(z) dans l'équation (3.19).

Les valeurs optimales de $\gamma_1$ et $\gamma_2$ sont déterminées par des tests d'écoute.

En général, le filtre perceptuel est déplacé vers les deux branches avant l'opération de soustraction entre le signal original et le signal synthétique. Ceci peut être fait, vu que le filtre est linéaire. La minimisation est équivalente alors à la minimisation de l'erreur quadratique moyenne de l'erreur entre le signal original pondéré et le signal synthétique pondéré comme montré en figure 3.7.

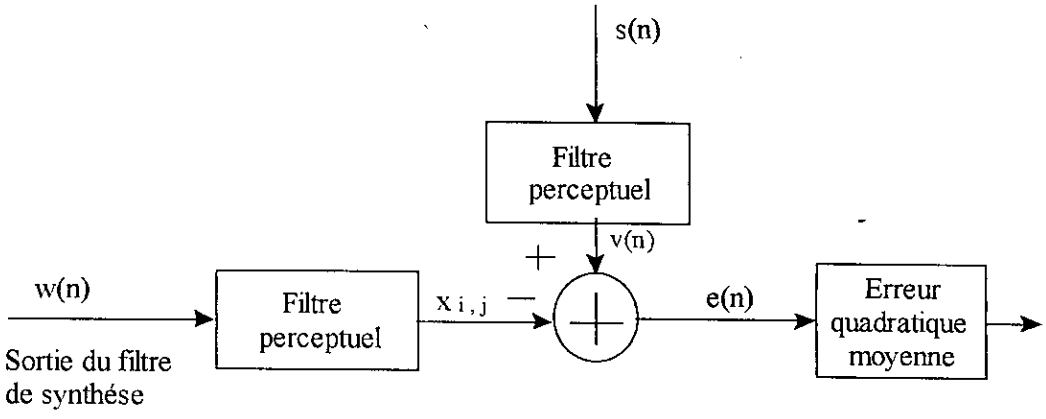
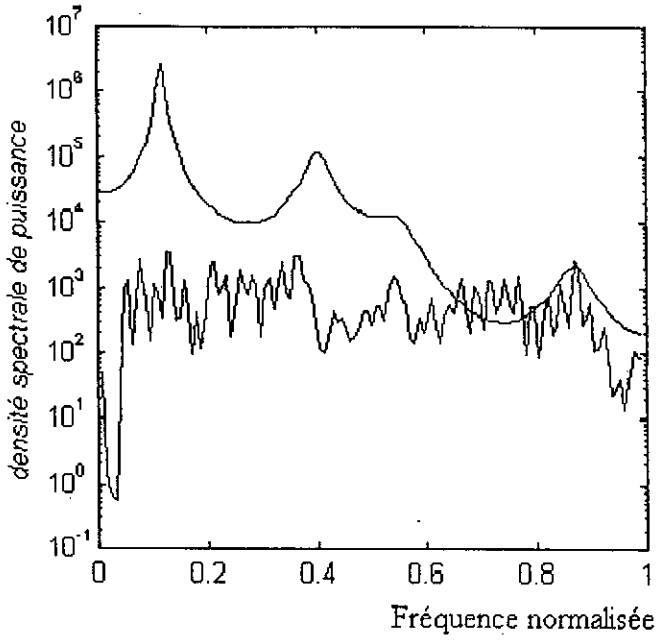


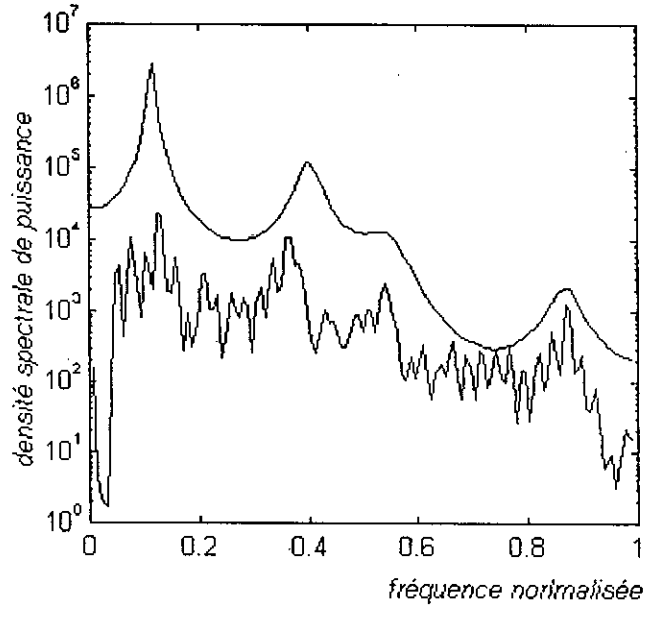

**Figure 3.7 :** Déplacement du filtre perceptuel vers les deux branches d'entrées

L'effet du filtre de perception :

- diminue les pics des formants (augmente leur largeur de bande) ;
- l'erreur est redistribuée.




a)




b)

**Figure 3.8.** Densité spectrale de l'erreur **a)** sans pondération **b)** avec pondération.

Le tableau 3.1 montre les types d'adaptation (forward ou backward) pour les différents paramètres dans le CELP conventionnel et le LD-CELP. Se distinguant des codeurs CELP conventionnels qui transmettent cinq types d'informations différentes comme listés dans le tableau 1, le LD-CELP transmet seulement l'indice de 10 bits (ou "l'adresse") de la QV du mot code (meilleure excitation sélectionnée pour chaque vecteur de parole).

| Paramètre | CELP conventionnel | LD-CELP |
|---|---|---|
| Coefficients LPC | Forward | Backward |
| Gain du Prédicteur "Pitch" | Forward | Non utilisé |
| Période du prédicteur pitch | Forward | Non utilisé |
| Gain d'excitation | Forward | Backward |
| Vecteur d'excitation | Forward | Forward |

**Tableau 3.1.** Types d'adaptations (forward et backward) pour différents paramètres dans le CELP conventionnel et le LD-CELP d'après [11].

## 3.6 Conclusion

Dans ce chapitre, nous avons décrit le codeur LD-CELP. Afin d'avoir un retard de codage assez faible et une haute qualité de la parole, la plupart des blocs du codeur CELP conventionnel ont subi des modifications. Les principales caractéristiques du codeur sont :

- un prédicteur LPC d'ordre supérieur et une analyse LPC adaptative en backward ;
- adaptation régressive du gain d'excitation à l'aide d'un prédicteur log-gain adaptatif;
- l'opération de fenêtrage pour les analyses LPC est assurée par la fenêtre de BARNWELL modifiée qui fournit les coefficients d'autocorrélation d'une manière récursive; cette fenêtre est en fait la réponse impulsionnelle d'un filtre IIR d'ordre 2;
- un filtre perceptuel de pondération est utilisé pour le masquage du bruit de codage.

# Chapitre 4

## Mise en œuvre du codeur et décodeur LD-CELP

Dans ce chapitre, une description détaillée de la mise en œuvre de l'algorithme de codage LD-CELP est décrite. Chaque bloc du codeur/décodeur est présenté. Une attention particulière est donnée pour le module de recherche dans le dictionnaire d'excitation. Une méthode de recherche permettant une réduction de complexité est appliquée. La procédure de conception du dictionnaire "forme" et "gain" est décrite, la conception est optimisée en boucle fermée.

Comme extrapolation de notre travail, nous explorons la possibilité de réduire la complexité de calcul dans le LD-CELP réalisé. Les mot codes sont remplacés par une structure de nombres ternaires qui nous permet de calculer les sommes partielles contenues dans l'expression de la distorsion sans effectuer d'opérations de multiplication, celles-ci étant remplacées par des additions. De plus, la structure ternaire nous permet de trouver des relations de récurrence pour calculer la distorsion. La complexité se trouve ainsi réduite. Les performances des deux dictionnaires sont ensuite comparées. Une autre extension consiste en l'application au codeur réalisé des signaux de parole large bande et nous procédons à l'évaluation de ses performances.

### 4.1. Encodeur LD-CELP

La figure (4.1) représente le schéma bloc de l'encodeur LD-CELP:
Pour chaque variable à décrire, k est l'indice de l'échantillon et les échantillons sont pris à des intervalles de 125 μs.
Un bloc de 5 échantillons consécutifs dans un signal donné est dit vecteur de ce signal, par exemple 5 échantillons consécutifs de parole forment un vecteur parole, 5 échantillons d'excitation forment le vecteur d'excitation.

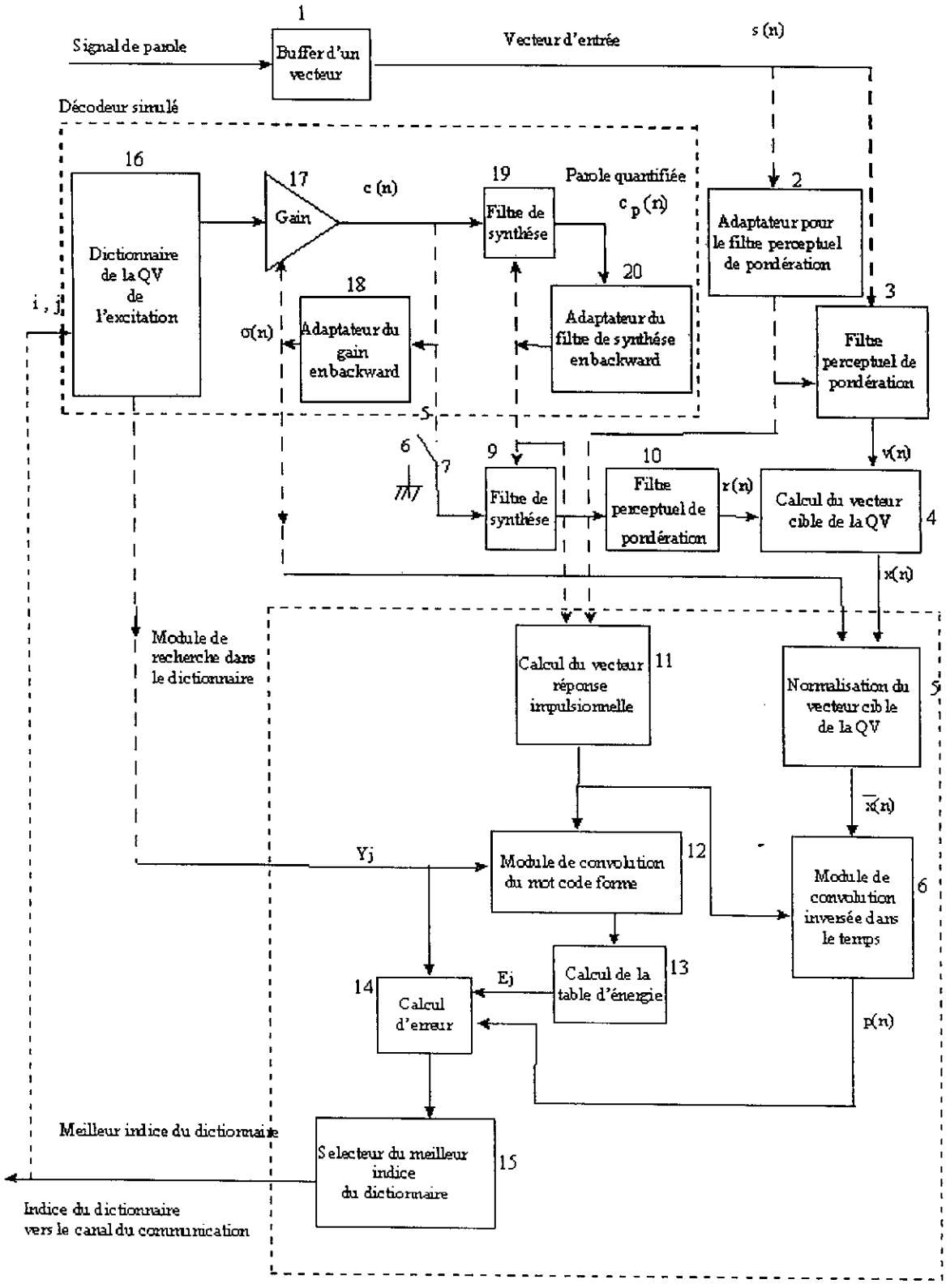



**Figure 4.1.** Schéma bloc détaillé de l'encodeur LD-CELP d'après [32].

Nous utilisons "n" pour désigner l'indice vecteur, qui est différent de l'indice de l'échantillon "k".

L'indice du dictionnaire qui est la quantification du vecteur d'excitation (QV) est la seule information qui est explicitement transmise de l'encodeur au décodeur. Trois autres types de paramètres sont réactualisés périodiquement :

- le gain d'excitation ;
- les coefficients du filtre de synthèse ;
- les coefficients du filtre perceptuel de pondération.

Ces paramètres sont dérivés d'une manière adaptative en backward des signaux déjà reçus avant l'encodage du vecteur du signal de parole courant. L'actualisation est effectuée de la manière suivante : le gain d'excitation est actualisé à chaque nouveau vecteur, alors que les coefficients des filtres de synthèse et perceptuel sont réactualisés une fois tous les 4 vecteurs (c'est à dire, tous les 20 échantillons ou toutes les 2,5 ms), ceci pour alléger les charges de calcul.

Il est à noter que bien que la séquence de traitement dans l'algorithme a un cycle d'adaptation de 4 vecteurs (20 échantillons), la taille du vecteur de bufferisation de base est seulement de 1 vecteur (5 échantillons).
Une description de chaque bloc de l'encodeur est donné ci-dessous.

### 4.1.1. Mémoire tampon du vecteur (Bloc 1)

Ce bloc bufferise 5 échantillons consécutifs de la parole s(5n), s(5n+1), s(5n+2), s(5n+3), s(5n+4) pour former un vecteur parole de dimension 5. S(n) = [s(5n), s(5n+1), ...,s(5n+4)];

### 4.1.2. Adaptateur pour le filtre perceptuel de pondération (Bloc 2)

La figure (4.2) montre l'opération détaillée de ce bloc. Cet adaptateur calcule les coefficients du filtre de pondération perceptuel une fois toutes les 4 vecteurs en se basant sur l'analyse par prédiction linéaire et en utilisant la parole non quantifiée. Les coefficients sont maintenus constants entre les réactualisations.

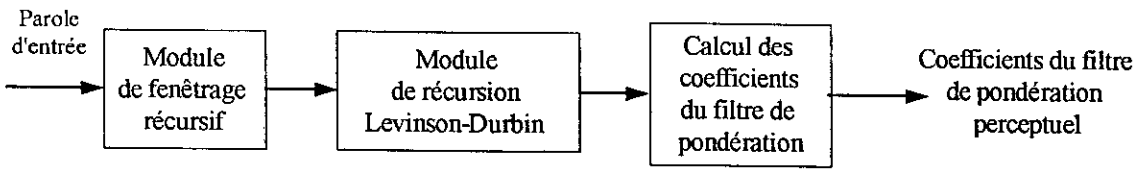



**Figure 4.2.** Adaptateur pour le filtre perceptuel

En se reportant à la figure (4.2), le calcul se fait comme suit : le vecteur parole d'entrée (non quantifié) est passé au travers du module de fenêtrage récursive qui applique une fenêtre sur les vecteurs antérieurs de parole et calcule récursivement les 11 premiers coefficients d'autocorrélation. Le module de récursion Levinson-Durbin convertit ces coefficients d'autocorrélation en coefficients du prédicteur.

L'adaptateur du filtre perceptuel de pondération réactualise périodiquement les coefficients de w(z) selon les équations (3.20) et (3.21) et transmet les coefficients au module de calcul du vecteur réponse impulsionnelle et au filtre perceptuel de pondération.

### 4.1.3. Filtre perceptuel de pondération (Bloc 3)

Dans la figure (4.1) le vecteur parole d'entrée courant s(n) est passé au travers du filtre perceptuel de pondération, qui donnera un vecteur parole pondéré v(n). Il est a noter que, excepté durant l'initialisation, la mémoire du filtre (c'est-à-dire valeurs restantes dans les unités de décalage du filtre) ne doit pas être initialisée à zéro à n'importe quel instant.

Autrement dit, la mémoire du filtre perceptuel doit avoir un traitement spécial comme il sera décrit plus loin.

### 4.1.4. Filtre de synthèse (Blocs 9 & 19)

Dans la figure (4.1), il y a deux filtres de synthèse ayant des coefficients identiques.

Les deux filtres sont réactualisés par l'adaptateur du filtre de synthèse backward (bloc 20). Chaque filtre de synthèse est un filtre tout-pôle d'ordre 50. La fonction de transfert du filtre de synthèse est :

$$f(z) = \frac{1}{[1 - p(z)]} \tag{4.5}$$

Où p (z) est la fonction de transfert du prédicteur LPC d'ordre 50.

Après l'obtention du vecteur parole pondéré v(n), le vecteur r(n) qui est une réponse à entrée nulle, doit être généré en utilisant le filtre de synthèse (bloc 9) et le filtre de pondération perceptuel (bloc 10). Pour accomplir ceci, on ouvre d'abord l'interrupteur 5, (le pointé vers le noeud 6). Ceci implique que le signal allant du noeud 7 vers le filtre de synthèse 9 doit être nul. On laisse alors ce dernier et le filtre de perceptuel (10) "tourner" durant 5 échantillons (1 vecteur). Ceci veut dire qu'on continue l'opération du filtrage pour 5 échantillons avec un signal zéro appliqué au noeud 7.
La sortie r(n) du filtre perceptuel 10 est la réponse à une entrée nulle.
Il est à noter que sauf après initialisation, la mémoire des filtres 9 et 10 n'est en général pas nulle. Alors le vecteur sortie r(n) est aussi diffèrent de zéro en général, bien que l'entrée du filtre venant du noeud 7 vers le filtre de synthèse 9 soit nulle. En effet, ce vecteur r(n) est la réponse des deux filtres aux vecteurs d'excitation à gain mis à l'échelle c(n-1), c(n-2),...antérieur à l'instant n-1. Ce vecteur, actuellement, représente l'effet de la mémoire du filtre.

### 4.1.5. Calcul du vecteur cible de QV (Bloc 4)

Ce bloc retranche le vecteur r(n) (réponse à l'entrée nulle) du vecteur parole pondéré v(n) pour obtenir le vecteur cible x(n) qui sera utilisé dans le module de recherche dans le dictionnaire (ceci est dû à l'utilisation de la technique du vecteur ZIR, (chapitre 4.2).

### 4.1.6. Adaptateur du filtre de synthèse (Bloc 20)

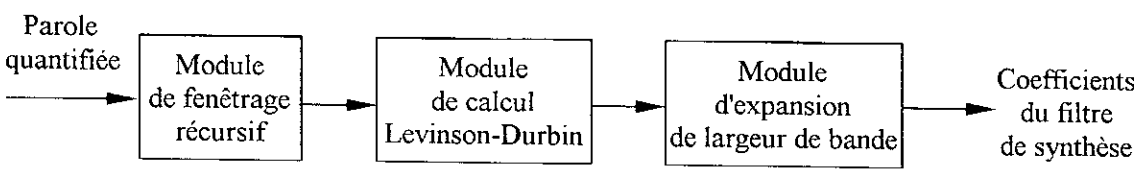



**Figure 4.3.** Adaptateur du filtre de synthèse en backward

Cet adaptateur (bloc 20) réactualise les coefficients des filtres de synthèses de 9 et 19. Il prend le vecteur parole quantifié comme entrée et donne des coefficients du filtre de synthèse comme sortie. Une version de cet adaptateur est en figure (4.3).
Les opérations du module de fenêtrage récursive et le module de récursion de Levinson-Durbin sont exactement les mêmes avec ceux de la figure 4.2, excepté pour ces 3 différences :

- le signal d'entrée est maintenant la parole quantifiée au lieu de la parole non quantifiée ;
- l'ordre du prédicteur est de 50 au lieu de 10 ;
- la longueur effective de la fenêtre de Barnwell est plus longue.

Soit $P(z)$ la fonction de transfert du prédicteur LPC d'ordre 50:

$$P(z) = \sum_{i=1}^{50} \bar{a}_i z^{-i} \tag{4.6}$$

Où $\bar{a}_i$ sont les coefficients du prédicteur. Pour augmenter la robustesse aux erreurs du canal, ces coefficients sont modifiés de façon que les pics dans le "spectre" LPC résultant ont des largeurs de bande un peu plus larges.
Le module d'expansion de la largeur de bande procède de la façon suivante : ayant les coefficients du prédicteur LPC $\bar{a}_i$, un nouveau ensemble de coefficients $a_i$ sont calculés selon :

$$a_i = \lambda^i \bar{a}_i \qquad i = 1,2,....,50 \tag{4.7}$$

Où $\lambda$ est calculé par [33] :

$$\lambda = e^{-(\frac{\pi\Delta\omega}{8000})} = 0.9883 \tag{4.8}$$

Ceci a pour effet de déplacer tous les pôles du filtre de synthèse radialement vers l'origine par un facteur $\lambda$. Vue que les pôles sont déplacés du cercle unité, les pics dans le spectre sont élargis. Après cette expansion, le prédicteur LPC modifié a une fonction de transfert :

$$P'(z) = \sum_{i=1}^{50} a_i z^{-i} \tag{4.9}$$

Les coefficients modifiés sont pris comme entrée aux filtres de synthèses 9 et 19 et au module de calcul du vecteur réponse impulsionnelle.

Ces filtres de synthèses ont comme fonction de transfert :

$$f(z) = \frac{1}{1 - P'(z)} \tag{4.10}$$

Comme au filtre perceptuel, les filtres de synthèses 9 et 19 sont réactualisés une fois tous les 4 vecteurs. Les actualisations sont basées sur la parole quantifiée antérieure.

### 4.1.7. Adaptateur du gain du vecteur d'excitation (Bloc 18)

Ce module réactualise le gain d'excitation σ(n) pour chaque vecteur à l'instant n. Le gain d'excitation σ(n) est un facteur d'échelle utilisé pour le vecteur d'excitation y(n) sélectionné. Ce module prend le vecteur d'excitation à gain mis à l'échelle c(n) comme entrée et donne σ(n) (gain d'excitation) comme sortie. Il « prédit » le gain de c(n) en se basant sur les gains de c(n-1), c(n-2), ... Une prédiction linéaire adaptative dans le domaine de gain logarithmique est utilisée.

L'adaptateur gain opère de la façon suivante : l'unité de décalage d'un vecteur fournit le vecteur d'excitation à gain mis à l'échelle antérieure c(n-1). Le module de calcul RMS calcule la valeur RMS du vecteur c(n-1). Puis, celle en dB du RMS c(n-1).

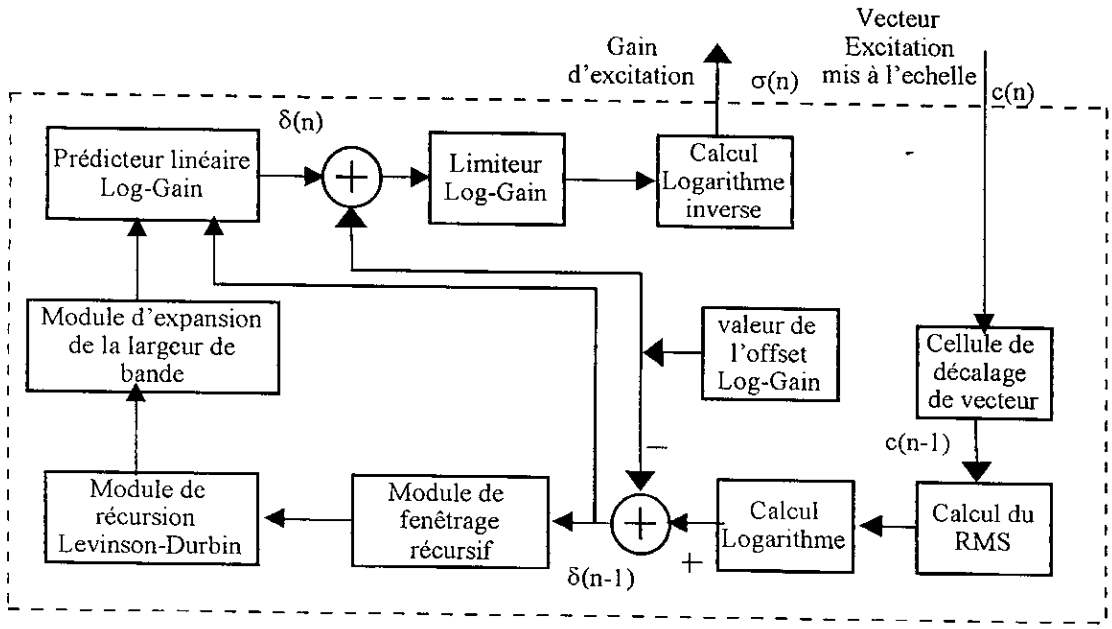



**Figure 4.4.** Schéma bloc du module de calcul du gain de mise à l'échelle [32].

Une valeur d'offset log-gain de 32 dB est retranchée de cette valeur (logarithme du

RMS). cette valeur est environ égale à l'énergie moyenne du gain d'excitation (en dB) durant la parole voisée.

Le gain logarithmique δ(n-1) ayant l'offset retranché est alors utilisé par le module de fenêtrage récursif et celui de calcul de récursion Levinson-Durbin. De même, ces blocs opèrent exactement de la même façon que ceux dans le module de l'adaptateur du filtre perceptuel de pondération, excepté que la longueur effective de la fenêtre est moins longue et que le signal analysé est un gain logarithmique au lieu de la parole d'entrée. Il est a noter qu'une seule valeur est calculée pour tous les 5 échantillons de parole.

Le module de récursion Levinson-Durbin donne les coefficients du prédicteur linéaire d'ordre 10 ayant comme fonction de transfert :

$$R(z) = \sum_{i=1}^{10} \overline{\alpha}_i z^{-i} \tag{4.11}$$

le module d'expansion de largeur de bande déplace les racines de ce polynôme radialement vers l'origine du plan z. Le prédicteur gain à largeur de bande élargie résultante a une fonction de transfert :

$$R(z) = \sum_{i=1}^{10} \alpha_i z^{-i} \tag{4.12}$$

Où les coefficients $\alpha_i$ sont : $\alpha_i = (0.9)^i \overline{\alpha}_i$

Cette expansion de largeur de bande permet à l'adaptateur gain d'être plus robuste aux erreurs de canal. Les $\alpha_i$ sont alors utilisés comme coefficients prédicteur linéaire log-gain.

Le prédicteur est actualisé une fois par cycle d'actualisation. Le prédicteur tend à prédire δ(n) en se basant sur une combinaison linéaire de δ(n-1), δ(n-2), ..., δ(n-10). La version prédite de δ(n) est notée $\overline{\delta}(n)$ est donnée par :

$$\overline{\delta}(n) = \sum_{i=1}^{10} \alpha_i \delta(n-i). \tag{4.13}$$

Après l'obtention de $\overline{\delta(n)}$, on ajoute la valeur d'offset log-gain de 32 dB. Le limiteur log-gain, vérifie alors, si sa valeur résultante n'est pas trop grande ou trop petite et limite cette valeur si nécessaire. Les limites inférieures et supérieures sont mises respectivement à 0 dB et 60 dB. La sortie de ce limiteur log-gain est linéarisée.
Le limiteur gain permet d'avoir un gain dans le domaine linéaire entre 1 et 1000.

## 4.2. Module de recherche dans le dictionnaire

Le module de recherche dans le dictionnaire formé de 1024 mot-codes a pour rôle l'identification de l'indice du meilleur mot code qui donne le vecteur parole synthétique le plus proche du vecteur parole d'entrée.

### 4.2.1. Recherche de l'excitation optimale dans le dictionnaire

**Principe de recherche dans le dictionnaire :** Pour réduire la complexité de recherche dans un dictionnaire de 10 bits (1024 mots codes), nous avons adopté la structure "gain-shape". Le dictionnaire est ainsi décomposé en deux dictionnaires: un de 7 bits (128 mots code) pour la quantification de la forme, et un autre de 3 bits (8 mots codes) pour la quantification du gain.

Ensuite, nous avons utilisé la technique du **ZIR** (Zero Input Response) qui consiste à prendre la mémoire des filtres en cascade (synthèse et perceptuel) et la soustraire de la branche haute (figure 4.5). Ceci, nous a permis d'effectuer les opérations de filtrage(filtres sans mémoire) sous forme de calcul matriciel.

En principe, le module de recherche dans le dictionnaire met à l'échelle chacun des 1024 mots codes candidats par le gain d'excitation courant $\sigma(n)$ et filtre les 1024 vecteurs obtenus par les 2 filtres en cascade F(z) et W(z). La mémoire du filtre est initialisée à zéro à chaque fois que le module présente un nouveau mot code au filtre résultant H(z):

$$H(z) = F(z).W(z). \qquad (4.14)$$

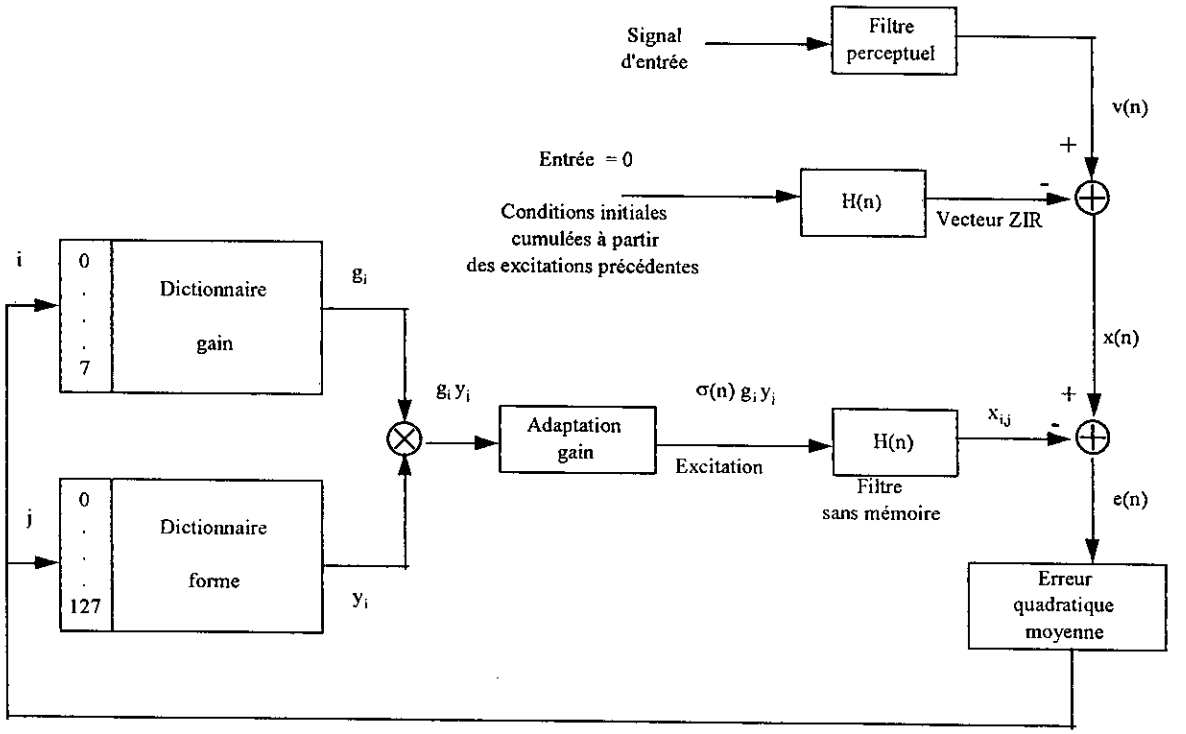



**Figure 4.5** : Introduction de la technique du vecteur ZIR et procédure de sélection de la meilleure excitation.

Soit $y_j$ le $j^{ieme}$ mot code dans le dictionnaire forme (j varie de 0 à 127), et soit $g_i$ le $i^{ieme}$ niveau dans le dictionnaire gain (i varie de 0 à 7).

Soit h(n) la réponse impulsionnelle du filtre H(z). Quand le mot code spécifié par les indices i et j des dictionnaires est à l'entrée du filtre en cascade H(z), la sortie du filtre est exprimée par :

$$x_{i,j} = H.\sigma(n).g_i y_j . \qquad (4.15)$$

où H est une matrice dont les éléments sont la réponse impulsionnelle des filtres en cascade.

$$H = \begin{bmatrix} h(0) & 0 & 0 & 0 & 0 \\ h(1) & h(0) & 0 & 0 & 0 \\ h(2) & h(1) & h(0) & 0 & 0 \\ h(3) & h(2) & h(1) & h(0) & 0 \\ h(4) & h(3) & h(2) & h(1) & h(0) \end{bmatrix} \tag{4.16}$$

La recherche dans le dictionnaire consiste à chercher la meilleure combinaison des indices i et j qui minimise la distorsion **MSE** (Mean Squared Error) suivante :

$$D = \left| x(n) - x_{ij} \right|^2 = \sigma^2(n) \left| \bar{x}(n) - g_i H y_j \right|^2 \tag{4.17}$$

Où $\bar{x}(n) = x(n) / \sigma(n)$ est le vecteur cible QV à gain normalisé.

Après développement on obtient :

$$D = \sigma^2(n) \left[ \left|\bar{x}(n)\right|^2 - 2 g_i \bar{x}^T(n) H y_j + g_i^2 \left|H y_j\right|^2 \right] \tag{4.18}$$

Les termes $\left|\bar{x}(n)\right|^2$ et la valeur de $\sigma^2(n)$ sont fixes durant la recherche. Minimisée D est équivalent à la minimisation de:

$$\overline{D} = -2 g_i p^T(n) y_j + g_i^2 E_j \tag{4.19}$$

Où

$$p(n) = H^T \bar{x}(n) \; ; \tag{4.19 a}$$

$$E_j = \left|H y_j\right|^2 \tag{4.19 b}$$

Notons que $E_j$ est actuellement l'énergie du $j^{eme}$ mot code forme filtré et ne dépend pas du vecteur cible QV $\bar{x}(n)$. Notons aussi que le mot code forme $y_j$ est fixe et que la matrice H dépend seulement des coefficients du filtre de synthèse et du filtre perceptuel, qui sont fixes durant une période de 4 vecteurs. Par conséquent, $E_j$ est aussi fixe durant cette même période.

Quand les deux filtres sont réactualisés, nous pouvons calculer et mémoriser les 128 termes d'énergie possible $E_j$, j = 0,1,...,127 (correspondants aux 128 mots code forme) et utiliser ces termes d'énergie d'une façon périodique pour la recherche dans le dictionnaire durant les 4 vecteurs parole suivants. Ceci, nous permet de réduire la complexité de recherche dans le dictionnaire.

Pour plus de réduction dans les calculs, nous pouvons pré-calculer et mémoriser les 2 vecteurs ; $b_i = 2g_i$ et $c_i = g_i^2$ pour i = 0,1,...,7. Ces deux vecteurs sont fixes vu que $g_i$ est fixe.

La relation (4.19) devient alors:

$$\overline{D} = -b_i p_j + c_i E_j \qquad (4.20)$$

Où

$$p_j = p^T(n)\, y_j. \qquad (4.20.a)$$

Nous remarquons qu'une fois les tables $E_j$, $b_i$ et $c_i$ sont pré-calculées et mémorisées, le produit $p_j = p^T(n)\, y_j$ , qui dépend de j, prend la majorité du taux de calcul dans la détermination de $\overline{D}$. La procédure de recherche dans le dictionnaire consiste alors à scruter dans le dictionnaire forme et identifier le meilleur indice i du gain pour chaque mot code forme $y_j$.

Il existe plusieurs manières de trouver le meilleur indice i du gain pour un mot code forme $y_j$ donné :

- la première, et la plus évidente manière, est d'évaluer les 8 valeurs possibles de i et sélectionner l'indice i qui correspond à la plus petite valeur D. Cependant, cela demande 2 multiplications pour chaque i ;
- une seconde manière de procéder consiste à évaluer le gain optimal $\bar{g} = \frac{p_j}{E_j}$ en premier, et quantifier ce gain $\bar{g}$ à l'un des 8 niveaux de gain $(g_0,.....,g_7)$ dans le dictionnaire gain de 3 bit. Le meilleur indice i est celui du niveau de gain $g_i$ qui est le plus proche de $\bar{g}$. Seulement cette manière de procédé nécessite une opération de division pour chacun des mots codes, et la division est très déconseillée à implanter dans les cartes DSP ;
- une troisième approche, qui est une modification de la seconde, est particulièrement efficace à l'implantation DSP. La quantification de $\bar{g}$ peut être une série de comparaison entre $\bar{g}$ et les «frontières de cellule de quantification», qui sont les points milieu entre les niveaux de gain adjacents. Soit $d_i$ le point milieu entre le niveau gain $g_i$ et $g_{i+1}$ qui ont le même signe. Alors tester «$g < d_i$» est équivalent à tester $p_j < d_i E_j$ . Ainsi, en utilisant ce test, on évite l'opération de division et on effectue qu'une

opération de multiplication pour chaque indice i. Cette approche est utilisée dans la recherche dans le dictionnaire. Les frontières de la cellule de quantification gain $d_i$ sont fixées et peuvent ainsi être pré-calculées et mémorisées dans une table. Pour 8 niveaux de gain, on a seulement 7 valeurs limites $d_0$, $d_1$, $d_2$, $d_3$, $d_4$ et $d_5$ et $d_6$ qui sont utilisées.

Une fois que les meilleurs indices "i" et "j" sont identifiés, ils sont concaténés pour former la sortie du module recherche dans le dictionnaire sous la forme d'un indice de 10 bits.

## 4.2.2. Mode opératoire du module recherche dans le dictionnaire

L'opération du module de recherche du dictionnaire est décrite ci dessous (fig.4.5bis). A chaque instant que les filtres de synthèse et perceptuel sont actualisés, le module "calcul du vecteur réponse impulsionnelle" calcule les 5 premiers échantillons de la réponse impulsionnelle des filtres en cascade F(z) et W(z).

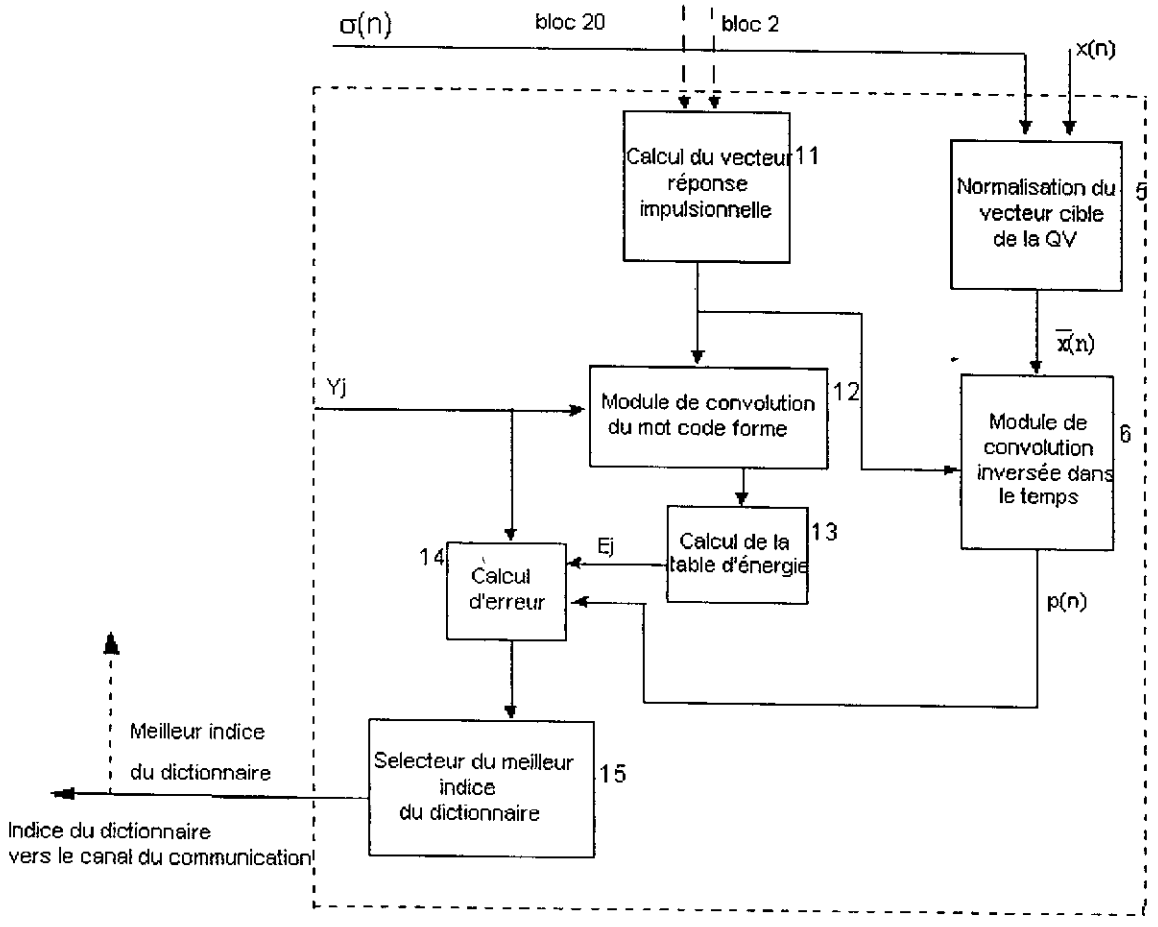



**Figure 4.5bis.** Module de recherche dans le dictionnaire

Pour calculer le vecteur réponse impulsionnelle, nous mettons d'abord la mémoire des filtres en cascade à zéro, puis nous excitons les filtres avec une séquence d'entrée (1, 0, 0, 0, 0 ). Les 5 échantillons de sortie correspondants du filtre sont h(0), h(1),..., h(4) qui constitue le vecteur réponse impulsionnelle désiré. Il est ensuite gardé constant et utilisé dans la recherche au niveau de dictionnaire pour les 4 vecteurs de parole suivants, jusqu'à la réactualisation des filtres 9 et 10.

Le module de calcul de la convolution du vecteur forme calcule ensuite les 128 vecteurs $E_{yj}$ , j = 0, 1, 2, ..., 127. En d'autres termes, il convolue chaque mot de code $y_j$ , j = 0, 1,...,127 avec la réponse impulsionnelle h(0), h(1).....h(4), où la convolution est seulement effectuée pour les 5 premiers échantillons. L'énergie des 128 vecteurs est calculée et mémorisée dans le module de calcul de table d'énergie (bloc 13) selon l'équation (4.19.b).
L'énergie d'un vecteur est définie comme étant la somme des carrées des composantes de chaque vecteur. Notons que les calculs dans les blocs 11, 12 et 13 sont effectués seulement une fois tous les 4 vecteurs de parole, alors que les autres blocs dans le module de recherche dans le dictionnaire effectuent les calculs pour chaque vecteur.

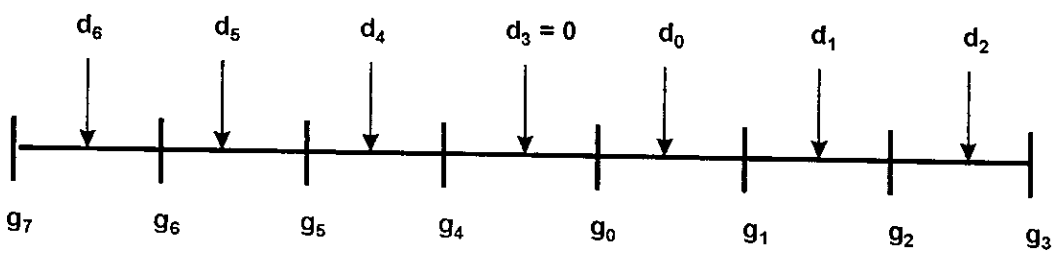
Notant aussi la réactualisation de la table $E_j$ est synchronisé avec les réactualisations des coefficients du filtre de synthèse. De là, la nouvelle table $E_j$ peut être utilisée en commençant par le 3$^{eme}$ vecteur parole pour chaque cycle d'adaptation.
Le module de normalisation du vecteur cible QV calcule le vecteur cible à gain normalisé

$$\bar{x}(n) = \frac{x(n)}{\sigma(n)}.$$

Après, le module de convolution temps-inversé calcule le vecteur $p(n) = H^T(n)\,\bar{x}(n)$
Cette opération est équivalente en premier à inverser l'ordre des composants de $\bar{x}(n)$, et convolué ensuite le vecteur résultant avec le vecteur réponse impulsionnelle, et ensuite inverser l'ordre des composants de la sortie (d'où le nom «time-reversed convolution»).
Une fois les tables $E_j$, $b_i$ et $c_i$ sont pré-calculés et mémorisés, le vecteur p(n) est aussi calculé, le module de calcule d'erreur et le sélecteur du meilleur indice du dictionnaire procèdent ensemble pour exécuter l'algorithme de recherche dans le dictionnaire [32] :

1. initialiser $D_{min}$ à un nombre de plus grande valeur possible de D ;
2. mettre l'indice du dictionnaire forme j = 0 ;
3. calculer le produit intérieur $p_j = p^T(n) .y_j$ ;
4. si $p_j < 0$ aller à l'étape 8 pour chercher dans les gains négatifs; Sinon aller à l'étape 5 pour chercher parmi les gain positifs ;
5. si $p_j < d_0 E_j$, mettre i = 0 et aller à l'étape 11, sinon aller à l'étape 6 ;
6. si $p_j < d_1 E_j$, mettre i = 1 et aller à l'étape 11, sinon aller à l'étape 7 ;
7. si $p_j < d_2 E_j$, mettre i = 2 et aller à l'étape 11, sinon mettre i = 3 et aller à l'étape 11
8. si $p_j > d_4 E_j$, mettre i = 4 et aller à l'étape 11, sinon aller à l'étape 9 ;
9. si $p_j > d_5 E_j$, mettre i = 5 et aller à l'étape 11, sinon aller à l'étape 10 ;
10. si $p_j > d_6 E_j$, mettre i = 6, sinon i = 7 ;
11. calculer $D = - b_i p_j + c_i E_j$ ;
12. si $D < D_{min}$, alors mettre $D_{min} = D$, $i_{min} = i$ et $j_{min} = j$ ;
13. si j < 127, mettre j = j + 1 et aller à l'étape 3 sinon aller à l'étape 14 ;
14. quand l'algorithme arrive ici toutes les 1024 combinaisons possibles de gain et forme ont été scrutés.

Les indices résultants $i_{min}$ et $j_{min}$ sont les indices du gain et la forme. Le meilleur indice du dictionnaire ( 10 bits ) est la concaténation de ces deux indices et correspond au meilleur mot code d'excitation $y(n) = g_{i\,min} \cdot y_{j\,min}$.
L'indice de 10 bits est envoyé a travers le canal de communication.




*Disposition des niveaux de gain et des points milieux tels qu'ils sont décrits dans la procédure de sélection de la meilleure excitation.*

## 4.3. Décodeur simulé

Bien que l'encodeur a identifié et a transmit le meilleur indice du dictionnaire,

certaines tâches additionnelles doivent être effectuées pour la préparation de l'encodage des vecteurs parole qui suivent.

Tout d'abord l'indice sélectionné servira à extraire à partir du dictionnaire QV d'excitation le meilleur mot code correspondant $y(n) = g_{i\ min} \cdot y_{j\ min,}$. Ce vecteur optimal est mis à l'échelle par le gain d'excitation courant $\sigma(n)$ pour donner le vecteur excitation :

$$c(n) = \sigma(n)\, y(n) \tag{4.21}$$

Ce vecteur c(n) est alors passé à travers le filtre de synthèse 19 pour obtenir le vecteur parole quantifié $c_p(n)$. Noter que les blocs 16 à 20 forment le décodeur simulé. En absence d'erreurs de canal, $c_p(n)$ est le vecteur parole quantifié.

Dans la figure (4.1), l'adaptateur du filtre de synthèse backward a besoin de ce vecteur $c_p(n)$ pour actualiser les coefficients du filtre de synthèse. De même, l'adaptateur du vecteur gain backward a besoin du vecteur excitation mis à l'échelle c(n) pour actualiser les coefficients du prédicteur linéaire log-gain. Ensuite, nous procédons à l'actualisation de la mémoire des filtres 9 et 10. Pour accomplir ceci :

nous sauvegardons la mémoire des filtres 9 et 10 qui s'est accumulée après avoir effectuer le calcul de la réponse à entrée nulle (ZIR) décrite dans le sous chapitre 4.1.4 .

Nous mettons les mémoires des filtres 9 et 10 à zéro et fermons l'interrupteur 5, i.e. le connecter au noeud 7. Le vecteur excitation mis à l'échelle c(n) est passé à travers des filtres à mémoire nulles.

Vue que la longueur du vecteur c(n) est de 5 échantillons seulement et la mémoire du filtre est vide, le nombre de (multiplication et addition) est seulement de 0 à 4 pour une période de 5 échantillons. Ceci est une réduction significative dans les calculs sachant qu'il aurait fallu 70 (multiplication et addition ) par échantillon si la mémoire du filtre n'était pas nulle. Ensuite, nous additionnons la mémoire originale sauvegardé du filtre avec la mémoire du filtre qu'on vient d'établir après filtrage de c(n).

Ceci, en effet additionne la réponse à entrée nulle avec la réponse à état sans mémoire des filtres 9 et 10. Le résultat de cette opération est la séquence mémoire désiré qui va être utilisé pour calculer le vecteur ZIR durant l'encodage du prochain vecteur parole.

L'opération de codage décrit la façon de coder un seul vecteur de parole. l'encodage de la forme d'onde entière de parole est achevée en répétant ces opérations pour chaque vecteur.

## 4.4. Décodeur LD-CELP

La figure 4.6 donne le schéma bloc du décodeur LD-CELP.

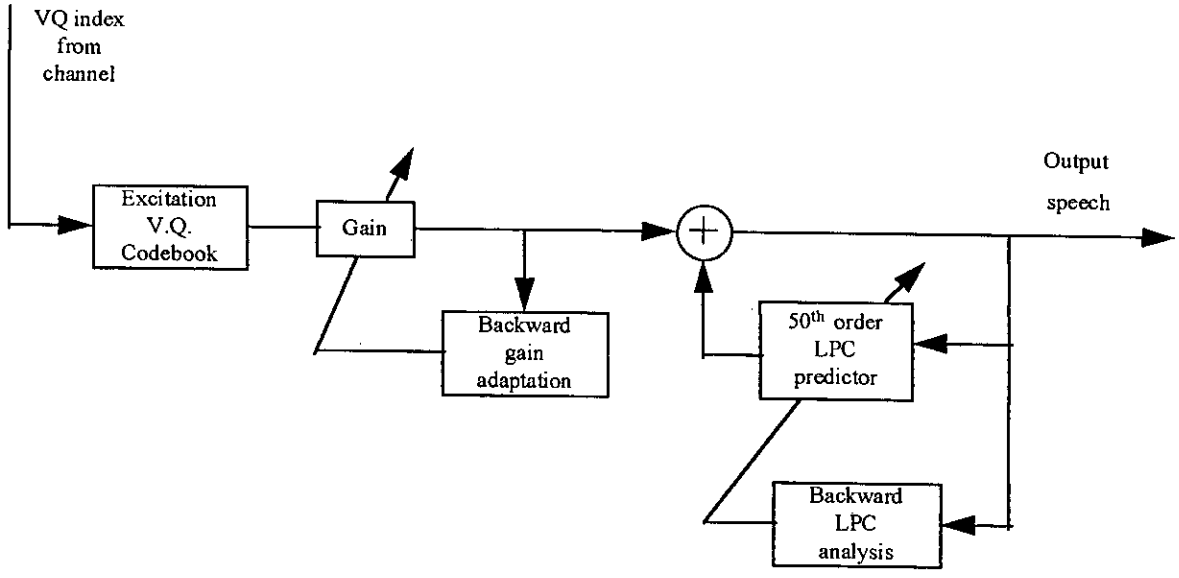



**Figure 4.6.** Décodeur LD-CELP

L'indice reçu passe par un bloc de décodage du mot code QV d'excitation. Ce bloc contient le dictionnaire de quantification vectorielle de l'excitation (dictionnaire forme et gain) identique à celui de l'encodeur. Il utilise le meilleur indice reçu pour extraire le meilleur mot code y(n) sélectionner au niveau de l'encodeur. Le vecteur excitation y(n) ainsi produit est mis à l'échelle par l'unité de mise à l'échelle gain en multipliant chaque composante par le gain $\sigma(n)$. Le vecteur d'excitation c(n) mis à l'échelle est appliqué au filtre de synthèse qui a la même fonction de transfert que le filtre de synthèse dans l'encodeur. Il filtre le vecteur c(n) pour donner le vecteur parole décodée $s_d(n)$.

L'adaptation du gain se fait de la même façon que celle décrite au niveau de l'encodeur. Il en est de même pour les coefficients du filtre de synthèse.

## 4.5. Quantification Vectorielle de l'Excitation

A un débit binaire de deux bits par échantillon, on a 10 bits pour coder chaque vecteur d'excitation de dimension 5. La complexité de recherche à travers un dictionnaire de 10 bits avec 1024 mots code indépendants est assez élevée. Pour réduire cette complexité, nous avons introduit la structure "gain-shape" et décomposé le dictionnaire

en un produit de dictionnaire gain à 3 bits et un dictionnaire forme (shape) à 7 bits.

Au lieu que le dictionnaire forme ne soit "peuplé" par des nombres aléatoires gaussiens comme dans le CELP conventionnel, ce dictionnaire est optimisé en boucle fermée par un algorithme de conception basé sur le critère de perception d'erreur pondérée utilisé par l'encodeur LD-CELP. De ce fait, les contributions d'adaptation du prédicteur LPC et du gain sont prises en considération dans la conception.
L'algorithme de conception est similaire à celui de LBG (LINDE BUZO GRAY) [15]. Mais, vue que la conception du dictionnaire est basée sur le critère perceptuel de pondération de l'erreur dans le LD-CELP, l'encodeur entier est utilisé à chaque itération pour encoder la séquence d'apprentissage.

Les vecteurs d'apprentissage sont classés selon la règle de codage d'erreur minimum, et les mots codes sont réactualisés selon la conditions de centroïde dérivée ci dessous.
Le dictionnaire "gain" consiste en 1 bit signe et 2 bits de niveaux d'amplitudes, le bit signe a pour rôle de doubler la taille du dictionnaire "forme" sans pour autant doubler la complexité de recherche dans celui-ci.

Soient g(n) et $\eta(n)$ le niveau d'amplitude de sortie et le multiplieur signe produit pour la recherche dans le dictionnaire à l'instant "n". Soit le $i^{ieme}$ dictionnaire "forme" de taille L. Et soit $S_j$ l'ensemble des indices temporels pour lesquels "$y_j$" est sélectionné comme meilleur mot de code du dictionnaire forme durant le codage de la séquence d'apprentissage des vecteurs de parole d'entrée $\{x_n ; n = 1,..., N\}$.
il s'agit de trouver le $(i+1)^{ieme}$ nouveau dictionnaire forme qui minimise la $i^{ieme}$ distorsion moyenne :

$$D(i) = \sum_{n=1}^{N} d(x_n, y_n(i)) \qquad (4..22)$$

La séquence de vecteurs cibles sont classés en L cellules, ($S_j$ ,j = 1, ..., L). L'équation est décomposée en L termes, un terme pour chaque cellule particulière ;

$$
\begin{aligned}
D = & \sum_{n \in S_1} \left\| x(n) - \eta(n)\sigma(n)g(n)H(n)y_1 \right\|^2 \\
& + \sum_{n \in S_2} \left\| x(n) - \eta(n)\sigma(n)g(n)H(n)y_2 \right\|^2 \\
& + \ldots + \sum_{n \in S_L} \left\| x(n) - \eta(n)\sigma(n)g(n)H(n)y_L \right\|^2
\end{aligned}
\tag{4.23}
$$

Chaque sommation englobe tout les vecteurs cibles qui sont classés dans le même mot code forme. La minimisation de l'équation selon le mot code correspondant est équivalente à minimiser séparément chaque terme de sommation, vu qu'un mot code forme n'apparaît seulement que dans une seule sommation particulière.

La distorsion totale accumulée de la $j^{ieme}$ classe correspondant à "$y_j$" est donnée par :

$$
D_j = \sum_{n \in S_j} \left\| x(n) - \eta(n)\sigma(n)g(n)H(n)y_j \right\|^2 \tag{4.24}
$$

En prenant la dérivée partielle selon $y_j$ et en l'annulant, nous obtenons :

$$
\begin{aligned}
\frac{\partial D_j}{\partial y_j} = & -2 \sum_{n \in S_j} \sigma(n)\eta(n)g(n)H^T(n)x(n) \\
& + 2 \sum_{n \in S_j} \sigma^2(n)g^2(n)H^T(n)H(n)y_j = 0
\end{aligned}
\tag{4.25}
$$

Alors, le centroïde $y_j^*$ de la $j^{ieme}$ classe qui minimise $D_j$ satisfait l'équation suivante :

$$
\begin{aligned}
& \left[ \sum_{n \in S_j} \sigma^2(n)g^2(n)H(n)^T H(n) \right] y_j^* \\
& \qquad = \sum_{n \in S_j} \eta(n)\sigma(n)g(n)H^T(n)x(n)
\end{aligned}
\tag{4.26}
$$

Les sommations de cette équation normale sont accumulées séparément pour chacun des 128 mots de codes "forme" pour toute la séquence d'apprentissage.
On résout les 128 équations normales résultantes pour obtenir les 128 nouveaux centroïdes qui remplaceront ceux du dictionnaire (ancien). La séquence d'entraînement est ensuite codée avec ce nouveau dictionnaire.

Cette procédure de réactualisation est répétée jusqu'à l'obtention d'un dictionnaire satisfaisant.

Dans l'algorithme LBG original pour la QV directe (sans boucle de prédiction), la distorsion globale et la réactualisation du dictionnaire sont garanties à converger vers un minimum local. Une exigence de base pour cette convergence est que les séquences d'apprentissage soient fixes durant toutes les itérations. C'est le cas pour la conception du dictionnaire pour une QV directe. Par contre, pour la conception en boucle fermée, l'ensemble des vecteurs du quantificateur changent d'une itération à l'autre ceci, est dû au fait que le quantificateur est à l'intérieur de la boucle de prédiction. Par conséquent, la conception en boucle fermée n'est pas garantie à converger.

En pratique, la distorsion globale décroît toujours durant les premières itérations, ce qui se traduit par une réduction significative dans la distorsion globale.

## 4.6. Mise en oeuvre de la conception du Dictionnaire "forme"

Lors de la conception du dictionnaire "forme", nous avons été amenés à choisir correctement le dictionnaire initial, ce dictionnaire joue un rôle essentiel dans la convergence de l'algorithme de conception vers un dictionnaire optimal en un nombre d'itérations non exhaustif.

Le dictionnaire d'excitation dont les vecteurs modélisent l'excitation du filtre de synthèse, lequel dans le cas idéal est représenté par le résiduel, a été peuplé par 128 vecteurs extraits des résiduels de la base de données (séquence d'apprentissage). La nouvelle technique d'initialisation de l'algorithme GLA a été utilisée pour obtenir les 128 vecteurs mots codes de dimension 5 [16].

L'expression du résiduel est :

$$r_n = \sum_{i=0}^{p} a_i s_{n-i} \qquad n = 0....N-1 \text{ et } a_0 = 1; \quad 1 \le i \le p \tag{4.27}$$

$s_n$ : signal parole (base de données) et $a_i$ sont les coefficients de prédiction.

Le calcul du résiduel a été fait en prenant un ordre de prédiction égale à 12, une trame de 240 échantillons sans recouvrement et une fenêtre de Hamming comme une fenêtre de pondération. Les coefficients de prédiction ont été calculés en utilisant l'algorithme de Levinson-Durbin.

Une fois la séquence résidu de la base de données parole a été obtenue, on applique la technique d'initialisation de l'algorithme GLA pour obtenir les 128 vecteurs mots code de dimension 5.

Ces 128 mots codes constitueront le dictionnaire initial utilisé pour démarrer la conception du dictionnaire "forme".

Tout l'encodeur LD-CELP est utilisé lors de l'encodage de la séquence d'apprentissage.

Les équations (4.26) à résoudre ont la forme suivante :

$$\mathbf{A.x = b} \tag{4.28}$$

Où A est une matrice carrée (5,5), b est un vecteur colonne (5,1) et x un vecteur colonne (5,1) de valeurs inconnues qu'il s'agit de déterminer.

La méthode d'obtention de la solution est de multiplier chaque coté de l'équation 4.28 par $A^{-1}$ :

$$\mathbf{x = A^{-1}.b} \tag{4.29}$$

Seulement pour la résolution des équations linéaires, cette solution est loin d'être la plus efficace et la plus stable numériquement.

En observant la structure de la matrice dans l'équation (4.26), on remarque que A peut se mettre sous la forme $H^T$ H où H est une matrice carrée, triangulaire ayant les éléments de la diagonale principale différents de zéro. La matrice A possède une structure d'une matrice hermitienne. Nous avons utilisé la décomposition de Cholesky pour résoudre ces équations. Tous les dictionnaires intermédiaires sont sauvegardés pour choisir le dictionnaire donnant les meilleures performances.

La base de donnée de parole est constituée d'une série de huit phrases phonétiquement équilibrées de langue anglaise et française (quatre phrases prononcées par des locuteurs masculins et les quatre autres par des locuteurs féminins) d'une durée totale 30 secondes , soit 240 000 échantillons.

Le meilleur dictionnaire est celui donnant le meilleur signal à bruit segmental.
Après 25 itérations, nous obtenons la figure 4.7 suivante :

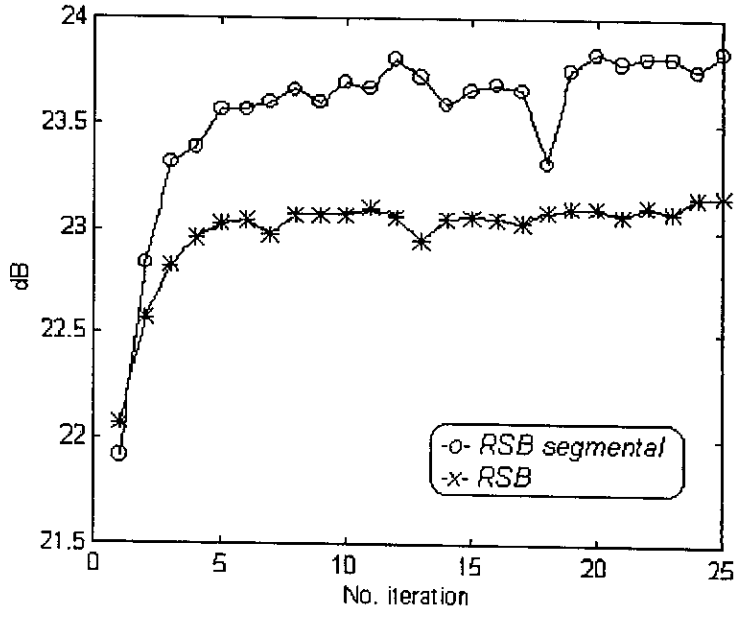



**Figure 4.7.** Variation du RSB segmental et du RSB en fonction des itérations durant la conception du dictionnaire "forme" avec gain optimal (non quantifié).

## 4.7. Conception du dictionnaire "gain"

Durant la conception du dictionnaire "forme", le gain utilisé pour la recherche du meilleur mot code est le gain optimal (non quantifié) déterminé par la formule suivante :

$$g_k = \frac{\left(\bar{x}^T H y_k\right)}{\left\|H y_k\right\|^2} \tag{4.30}$$

(Cette formule est obtenue en dérivant l'équation (4.24) selon $g_k$ et en l'annulant).
Pour procéder à la conception du dictionnaire du gain, nous avons crée une base de données constituée de gains optimaux obtenus lors du codage de la séquence d'apprentissage donnant le meilleur RSB segmental.

Vu que nous allons attribuer 3 bits pour la quantification du gain, nous avons le choix entre concevoir un dictionnaire :

- de 8 mots codes ($2^3$) distincts ;
- de 4 mots codes ($2^2$) avec un 1 bit pour le signe, nous aurons ainsi une symétrie par rapport a zéro.

Nous avons adopté la $2^{eme}$ approche, vu que l'histogramme des gains optimaux possède une symétrie. Nous appliquons l'algorithme GLA pour avoir un dictionnaire de 4 éléments, la séquence d'apprentissage étant les valeurs absolues des gains optimaux dont l'histogramme est en figure 4.8.

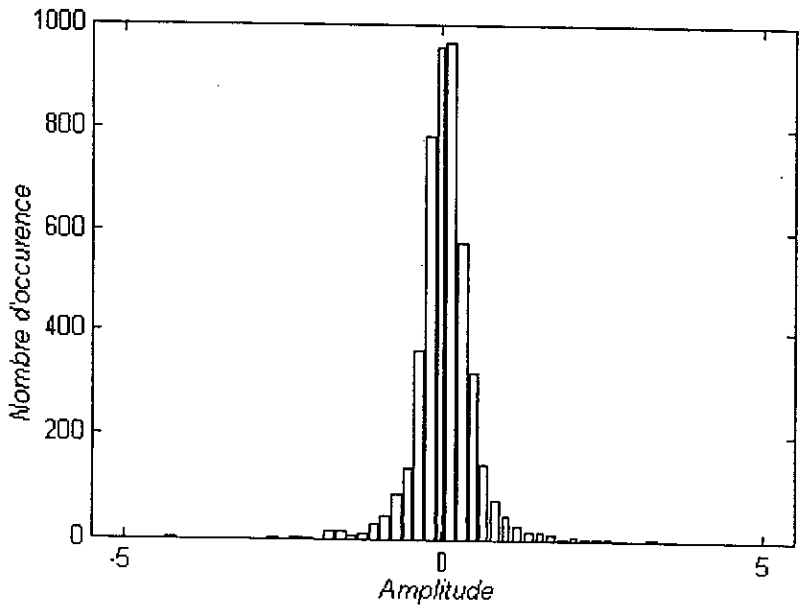



**Figure 4.8.** Histogramme du gain optimal obtenu durant la conception du dictionnaire "forme" donnant le meilleur RSB segmental.

Après plusieurs itérations, la convergence est atteinte. Les dictionnaires "gain" et "forme" obtenus sont sauvegardés dans un fichier de donnée et sont utilisés par l'encodeur LD-CELP.

## 4.8. Performances du codeur LD-CELP

### 4.8.1. Organisation du programme

Le programme du codeur est composé de :

- deux fichiers header, l'un contenant les définitions des constantes et des pré déclarations des fonctions et l'autre contenant les prototypes des fonctions de

traitement d'erreurs, d'allocation et désallocation d'espace mémoire ;

- un fichier de donnée contenant le dictionnaire stochastique ;
- 15 fichiers de code source.

Programme principal CMAIN.C

Le fichier principal cmain contient le programme principal de l'opération de codage, ces opérations consistent à l'ouverture des fichiers de données et de stockage du bitstream et l'initialisation des paramètres du codeur. Il appelle les fonctions qui réalisent les différentes tâches de traitement. Le paramètre - nom de fichier à coder - est utilisé comme argument d'entrée.

La structure du programme principal est :

Entête (Fichier header).
Début
Lecture des arguments
Initialisation des différents paramètres
Ouverture de fichiers de données : Fichier à coder (xxxx . dat) en lecture, Fichier d'écriture ou sera sauvegardé le train binaire (bitstream.dat).

Tant que :
Lecture du signal de la parole par vecteur de 5 échantillons.
Traitement des vecteurs (opération de codage).
Sauvegardes des meilleurs indices donnés pour le vecteur parole courant dans le fichier bitstream.dat après concaténation.
Fin tant que.
Fermeture des différents fichiers ouverts.
Fin.

## 4.8.2. Evaluation des performances

Afin d'évaluer les performances du codeur LD-CELP réalisé, nous avons procédé au codage d'une série de phrases phonétiquement équilibrées, échantillonnées à une fréquence de 8 kHz, et ayant une bande passante de 3.2 kHz. Ces phrases n'appartiennent pas à la séquence d'apprentissage utilisée pour la conception des

dictionnaires "forme" et "gain" : les phrases sont les suivantes :

**H1 :** "Annie s'ennuie loin de mes parents"

**H2 :** "Dés que le tambour bat, les gens accourent"

**H3 :** "Vous poussez, des cris de colère ?"

**H4 :** "Oh dear, the speaker apologized, I have been out of synchronization"

**H5 :** "Les deux camions se sont heurtés de face"

**H6 :** "Un loup s'est jeté immédiatement sur la petite chèvre"

**F1 :** "I gather you will be abandoning the major revisions"

**F2 :** "La bas il y a des mauvaises vagues très hautes"

**F3 :** "La vaisselle propre est mise sur l'évier"

**F4 :** "Quand il s'est réveillé, il était trop tard"

**F5 :** "Huit satellites ont été mobilisés"

**Hx** : désigne un locuteur masculin.

**Fx** : désigne un locuteur féminin.

La figure 4.9. nous donne une comparaison d'un segment de forme d'onde de parole tiré de la phrase H4.

La figure 4.10. nous donne un exemple de codage de la phrase F1, le signal synthétique obtenu, le signal excitation, le signal erreur ainsi que l'évolution du RSB obtenu par rapport à la puissance du signal parole.

Le tableau 4.1 nous donne les performances objectives du codeur réalisé. Il fournit une haute qualité de parole synthétisée. La moyenne des RSB segmentaux obtenus est proche de 20 dB. Les tests d'écoute ont confirmé l'efficacité du codeur/décodeur pour l'obtention de la qualité élevée.

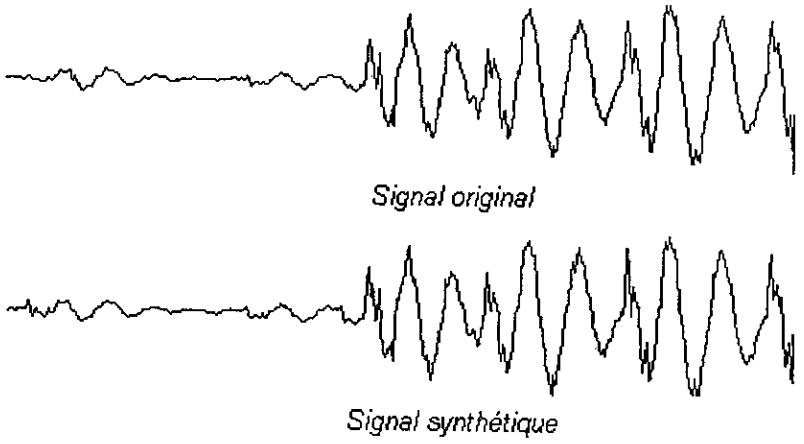



**Figure 4.9.** Comparaison de forme d'onde d'un segment de parole de la phrase H4.

| Phrases | RSB | RSB segmental |
|---|---|---|
| H1 | 20.48 | 18.77 |
| H2 | 20.04 | 20.18 |
| H3 | 17.38 | 19.25 |
| H4 | 21.67 | 18.76 |
| H5 | 16.34 | 15.85 |
| F1 | 21.20 | 19.91 |
| F2 | 22.31 | 21.12 |
| F3 | 20.65 | 18.88 |
| F4 | 22.16 | 21.22 |

**Tableau 4.1.** Performance du codeur LD-CELP pour une excitation stochastique. Les phrases n'appartiennent pas à la séquence d'apprentissage utilisée pour la conception du dictionnaire stochastique.

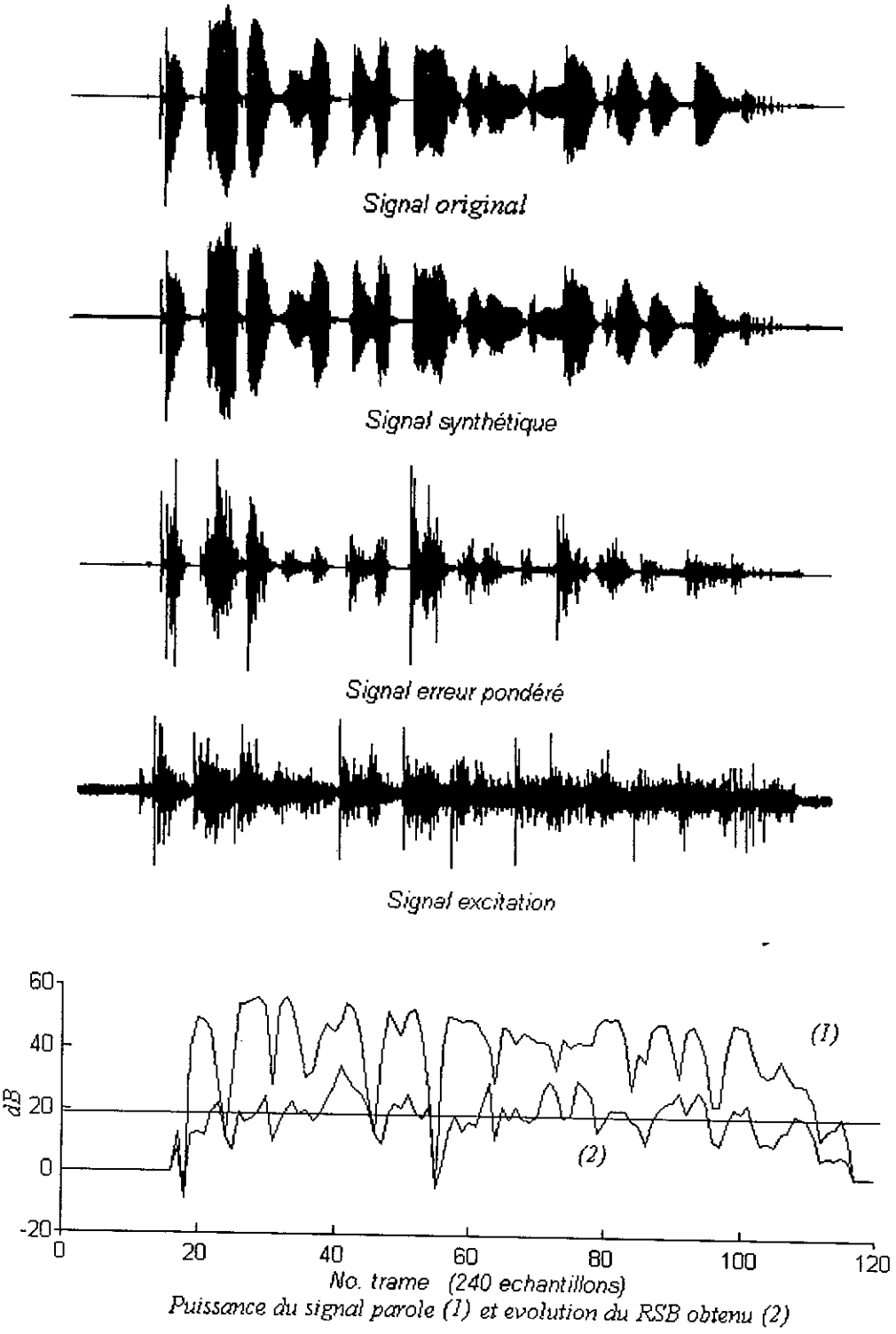



**Figure 4.10.** Exemple de phrase codée (F1) **a)** signal original **b)** signal synthétique **c)** signal erreur pondéré (multiplié par 4) **d)** signal excitation (multiplié par 20) **e)** évolution du RSB par rapport à la puissance du signal original.

# 4.9. Codeur LD-CELP à excitation ternaire

## 4.9.1. Réduction de complexité

Les propriétés algébriques de dictionnaire structuré permettent de construire des algorithmes efficaces pour le calcul de la distorsion associé à chacun des vecteurs d'excitation lors de la recherche exhaustive dans le dictionnaire.

Le critère à minimiser :

$$\overline{D} = E_j(n) - 2\ p^T(n)\, y_j \tag{4.31}$$

Avec : $p(n) = H^T(n)\, x(n)$, $E_j(n) = ||H(n).y_j||^2$

Le premier terme, qui est une énergie, ne dépend que du vecteur d'excitation et de la réponse impulsionnelle H, produite par la cascade du filtre de synthèse et du filtre perceptuel. Le deuxième terme est un produit scalaire du vecteur d'excitation du dictionnaire analysé et celui d'excitation naturelle.

Pour les dictionnaires structurés, le calcul de distorsion peut être simplifié davantage. Examinons le 1$^{ere}$ terme :

$$|Hy|^2 = \sum_{k=0}^{N-1}\left(\sum_{i=0}^{k} h(k-i)y(i)\right)^2 \tag{4.32}$$

$$\begin{aligned}|Hy|^2 = (h(0)y(0))^2 + (h(1)y(0) + h(0)y(1))^2 + \ldots + \\ (h(N-1)y(0) + \ldots + h(0)y(N-1))^2\end{aligned} \tag{4.33}$$

Dans le cas d'un dictionnaire non structuré, prenant h(0) = 1, le calcul de $|Hy|^2$ impose, pour chaque vecteur d'effectuer 1, 2, puis N-1 multiplications, avec N élévations au carré d'où un total de N(N+1)/2 multiplications par vecteur. De même, (N-1)(N+2)/2 additions par vecteur sont requises.

Considérons, maintenant, un dictionnaire structuré ternaire de dimension N [33], ensemble de ($3^N$-1) vecteurs non nuls dont les coordonnées sont dans {-1, 0, 1}. Pour ce dictionnaire, aucune multiplication n'est requise pour calculer toutes les sommes partielles de h(0),...,.h(N-1) multipliées chacun par 0,1,-1. Cette opération ne demande que des additions.

Donnons un exemple, soit un dictionnaire ternaire de dimension N = 3, nous avons 27 configurations différentes ; de [-1 -1 -1], [-1 -1 0], ..., [1 1 1]. Nous pouvons remarquer que chaque configuration possède son symétrique (sauf le vecteur [0 0 0]), par exemple le terme [-1 -1 -1] a son terme symétrique [1 1 1]. Si nous calculons les sommes partielles de $h_0$ , $h_1$ , $h_2$ ... chacun affecté d'un coefficient de {-1, 0, 1}, nous avons constaté qu'ils sont en fait la dernière coordonnée de Hy pour y décrivant le dictionnaire. Soit $b(y_k)$ cette dernière coordonnée. On notera pour simplifier b(m) au lieu de $b(y_m)$ :

$$b(m) = \sum_{i=0}^{N-1} h(N-1-i)y_m(i) \tag{4.34}$$

Etant donné que la configuration est symétrique , nous aurons à évaluer que $(3^3-1)/2 = 13$ termes d'énergie. Le tableau 4.2 nous donne les différents cas de figures, en affectant le numéro 1 au vecteur [0 0 1] et le numéro 13 au vecteur [-1 -1 -1].

| l<br>m | 0 | 1 | 2 | b(m) |
|---|---|---|---|---|
| 1 | 0 | 0 | -1 | $-h_0$ |
| 2 | 0 | -1 | 1 | $-h_1+h_0$ |
| 3 | 0 | -1 | 0 | $-h_1$ |
| 4 | 0 | –1 | –1 | $-h_1-h_0$ |
| 5 | -1 | 1 | 1 | $-h_2+h_1+h_0$ |
| 6 | -1 | 1 | 0 | $-h_2+h_1$ |
| 7 | -1 | 1 | -1 | $-h_2+h_1-h_0$ |
| 8 | -1 | 0 | 1 | $-h_2+h_0$ |
| 9 | -1 | 0 | 0 | $-h_2$ |
| 10 | -1 | 0 | -1 | $-h_0-h_2$ |
| 11 | -1 | -1 | 1 | $-h_2-h_1+h_0$ |
| 12 | -1 | -1 | 0 | $-h_2-h_1$ |
| 13 | -1 | -1 | -1 | $-h_0-h_1-h_2$ |

**Tableau 4.2.** Configuration ternaire et sommes partielles

On remarque que les vecteurs n'ayant qu'une coordonnée non nulle ont un indice qui est

une puissance de 3 ; b(1), b(3), b(9). Ces vecteurs (0, 0, -1), (0, -1, 0) et (-1, 0, 0) ont respectivement pour valeurs de $|Hy|^2$

$|Hy_0|^2 = h_0^2$

$|Hy_1|^2 = h_0^2 + h_1^2$

$|Hy_2|^2 = h_0^2 + h_1^2 + h_2^2$

Tous les autres vecteurs peuvent s'exprimer en fonction de ces derniers.
Généralement on a :

$$\begin{aligned} b(3^i) &= -h_i \qquad i = 0,\ldots,N-1 \quad \text{si } i \neq 0 \\ b(3^i - j) &= b(3^i) - b(j) \qquad j = 1,\ldots,(3^i - 1)/2 \\ b(3^i + j) &= b(3^i) + b(j) \end{aligned} \tag{4.35}$$

Soit maintenant, $g(i) = |Hy_i|^2$. Remarquons que les vecteurs sont engendrés par décalage vers la gauche et une concaténation d'un symbole -1, +0, +1. Par exemple, prenant [0 0 +1] un décalage à gauche donne 0 +1, une concaténation de -1 donne [0 +1 -1], qu'est le vecteur suivant. Puis une concaténation de 0, donne [0 +1 0], qu'est le vecteur suivant le vecteur [0 +1 -1], est ainsi de suite. Le décalage vers la gauche est l'inverse d'un retard temporel, soit une anticipation. Soit g(0), g(1) et g(2) la réponse du signal anticipé-concaténé. Un retard unitaire sur ce signal produit la réponse 0, g(0), g(1) d'où le calcul de l'énergie de la réponse $g(0)^2 + g(1)^2 + g(2)^2$ comme la somme de l'énergie retardée plus la dernière coordonnée élevée au carré : $(g(0)^2 + g(1)^2) + g(2)^2$. On appliquant ceci aux termes d'énergie, on aura :

$|Hy|^2 = |THy|^2 + b(y)^2$, ou T : opérateur décalage temporel.

$g(1) = 1$

$g(2) = g(1) + b(2)^2$

$g(3) = g(1) + b(3)^2$

$g(4) = g(1) + b(4)^2$

$g(5) = g(2) + b(5)^2$

$g(6) = g(2) + b(6)^2$

$g(7) = g(2) + b(7)^2$

$g(8) = g(3) + b(8)^2$...

jusqu'au dernier terme :

$g(13) = g(4) + b(13)^2$

Nous obtenons ainsi une nette réduction de coût de calcul, soit 2 additions et 1 multiplication par vecteur à comparer au cas non structuré (voir tableau 4.3).

| Opération par vecteur | Ternaire | Stochastique |
|---|---|---|
| multiplication | 1 | N(N+1)/2 |
| addition | 2 | (N-1)(N+2)/2 |

**Tableau 4.3.** Coût de calcul pour une excitation ternaire et une excitation stochastique.

Une fois que les termes d'énergies sont évalués, il nous reste le calcul des intercorrélations (deuxième terme de l'équation 5.1). Le terme $p^T(n)y_j$ avec $p(n) = H^T(n)$ x(n) , n'est autre que le produit scalaire du vecteur y et du produit de la transposée de la matrice H avec le vecteur x(n). D'où l'adaptation de la procédure de calcul de "b" pour celui de $p^T(n)y_j$.

## 4.9.2. Application

Un dictionnaire ternaire de dimension N = 5 a été appliqué pour l'encodeur LD-CELP. Les sept bits d'excitation donnent droit à un dictionnaire de 128 vecteurs d'excitations. La structure ternaire donne $[3^5 - 1]/2 = 121$ vecteurs ternaires.
L'algorithme de détermination des termes d'énergie pour N = 5 est donné ci-dessous:

Initialisation : g(1) =1 ;

Récursion : pour i = 0, ..., 5

pour j = 1, ..., $(3^i-1)/2$

$b(3^i) = -h_i$

$b(3^i - j) = b(3^i) - b(j)$

$b(3^i + j) = b(3^i) + b(j)$

pour j = 2, ..., 121

pour i = 1, ..., M M étant le nombre de vecteurs à décaler $M = (3^N-1)/3$

$g(j) = g(i) + b(j)^2$

### 4.9.3. Effet du facteur d'échelle

Ce dictionnaire ternaire entier est mis à l'échelle. En effet, pour chaque vecteur de coordonnée entière du dictionnaire considéré, on détermine par apprentissage statistique [15] un coefficient multiplicateur. Ce coefficient permet de mieux décrire la répartition énergétique statistique, fonction de l'information temps fréquence du signal d'excitation naturel.

Pour la conception des facteurs d'échelle, nous procédons par apprentissage statistique. Soit $\alpha_j$ le facteur d'échelle correspondant au $j^{ieme}$ mot code du dictionnaire "forme" "$y_j$". Nous procédons au codage de la séquence d'apprentissage, La distorsion totale accumulée de la $j^{ieme}$ classe correspondant à "$\alpha_j$" est donnée par :

$$D_j = \sum_{n \in S_j} \left\| x(n) - \alpha_j \eta(n)\sigma(n)g(n)H(n)y_j \right\|^2 \qquad (4.36)$$

Le nouveau facteur d'échelle est obtenu en minimisant l'expression suivante :

$$D_j = \sum_{n \in S_j} (x(n)^T x(n) - 2\alpha_j \eta(n)\sigma(n)g(n)x(n)^T H(n)y_j + \alpha_j^2 \sigma(n)^2 g(n)^2 \left\| H(n)y_j \right\|^2) \qquad (4.37)$$

Les éléments de l'équation sont ceux définis dans le chapitre 4.2.

En prenant la dérivée partielle selon $\alpha_j$ et en l'annulant, nous obtenons :

$$\alpha_j^{new} = \frac{\sum_{n \in N_i} \eta(n)\sigma(n)g(n)x(n)^T H(n)y_j}{\sum_{n \in N_i} \sigma(n)^2 g(n)^2 \left\| H(n)y_j \right\|^2} \qquad (4.38)$$

Après plusieurs itérations, nous obtenons les facteurs d'échelle et nous associons à chaque mot code forme son propre facteur. Ces facteurs sont sauvegardés dans un fichier de donné.

Nous avons procédé ensuite au codage de plusieurs phrases n'appartenant pas à la séquence d'apprentissage. Pour comparer les performances du codeur LD-CELP ternaire sans facteur d'échelle et avec facteur d'échelle, nous avons calculé le RSB et le RSB segmental. Nous avons obtenu le tableau de valeurs 4.4; des tests d'écoute ont

montré que les phrases synthétiques obtenues avec le codeur à excitation ternaire sans facteur d'échelle présentent un bruit audible.

L'introduction du facteur d'échelle améliore considérablement la qualité du signal de parole.

| | Excitation ternaire avec facteur d'échelle | | Excitation ternaire sans facteur d'échelle | |
|---|---|---|---|---|
| Phrases | RSB | RSB seg. | RSB | RSB seg. |
| H1 | 20.03 | 17.99 | 17.54 | 13.54 |
| H2 | 20.36 | 18.49 | 19.03 | 13.77 |
| H3 | 15.70 | 15.39 | 15.50 | 12.88 |
| H4 | 21.34 | 18.41 | 16.68 | 15.39 |
| F1 | 21.35 | 18.90 | 20.23 | 15.70 |
| F2 | 21.69 | 20.61 | 19.97 | 16.99 |

**Tableau 4.4.** Performance du codeur LD-CELP à excitation ternaire (avec et sans facteur d'échelle).

Donc, l'excitation est [33 et 34] le produit de :

- une amplitude de référence, obtenue par une prédiction linéaire des logarithmes des excitations précédentes ;
- un gain, choisi dans un tableau fixe, qui est le même pour toutes les directions ;
- un vecteur de composantes entières, mis à l'échelle par un coefficient fixe, ne dépendant que de la direction du vecteur.

## 4.9.4. Evaluation des performances

Afin d'évaluer les performances du codeur en utilisant les deux types d'excitation (statistique et ternaire), nous avons codé une série de phrases n'appartenant pas à la séquence d'apprentissage utilisée lors de la conception du dictionnaire stochastique. Les résultats montrent que le dictionnaire ternaire et le dictionnaire stochastique produisent la même qualité de parole (tableau 4.5). Le premier permet d'avoir une réduction significative de la complexité.

| | RSB | | RSB segmental | |
|---|---|---|---|---|
| Phrases | Dictionnaire stochastique | Code ternaire | Dictionnaire stochastique | Code ternaire |
| H1 | 20.48 | 20.03 | 18.77 | 17.99 |
| H2 | 20.04 | 20.36 | 20.18 | 18.49 |
| H3 | 17.38 | 15.70 | 19.25 | 15.39 |
| H4 | 21.67 | 21.34 | 18.76 | 18.41 |
| F1 | 21.20 | 21.35 | 19.91 | 18.90 |
| F2 | 22.31 | 21.69 | 21.12 | 20.61 |
| F3 | 20.65 | 20.35 | 18.88 | 18.44 |

**Tableau 4.5.** Performance du codeur LD-CELP pour une excitation statistique et une excitation ternaire.

La figure 4.11 nous donne une comparaison de forme d'onde d'un segment de parole de la phrase F2 pour une excitation ternaire et stochastique.

La figure 4.12 représente l'évolution du RSB obtenu après codage de la phrase (F6) par rapport à l'évolution temporelle du signal parole codé.

La figure 4.13 nous donne un exemple de codage de la phrase (F6), le signal synthétique obtenu, le signal excitation, le signal erreur pondéré, les deux derniers signaux sont multipliés par des facteurs de 4 et 20 respectivement.

La figure 4.14 est une comparaison de forme d'ondes de signaux synthétiques obtenus avec les dictionnaires stochastique et ternaire.

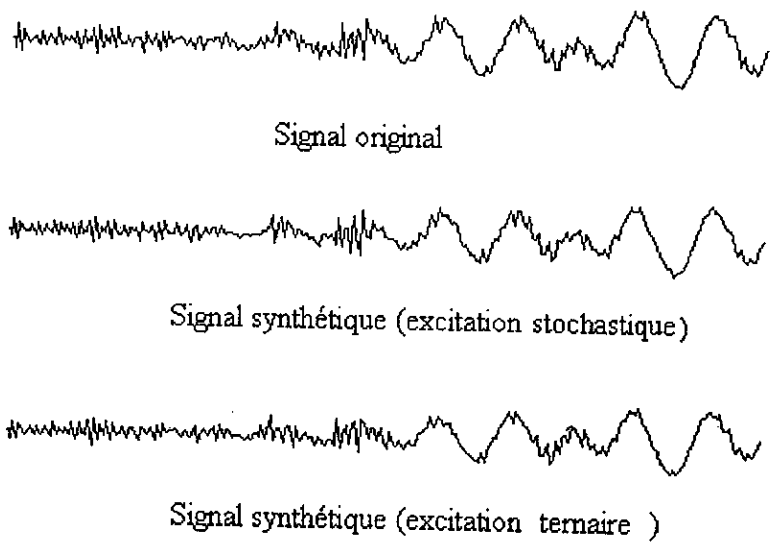



**Figure 4.11.** Comparaison de forme d'onde d'un segment de parole de la phrase F2.

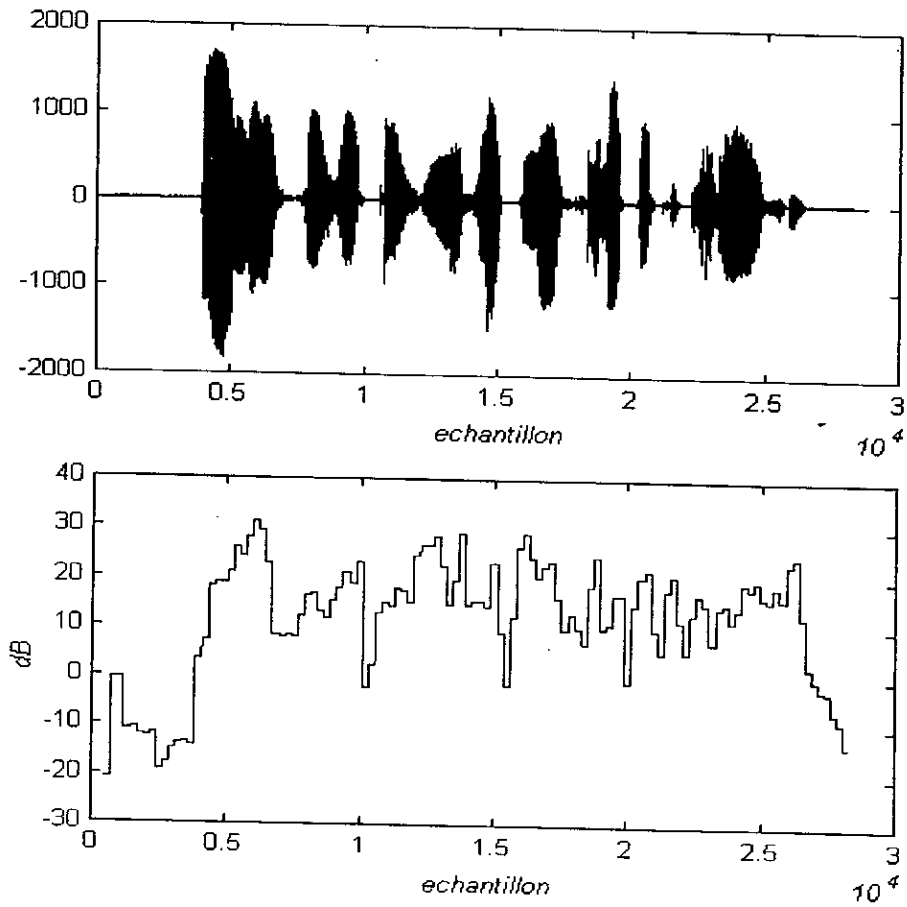



**Figure 4.12.** Représentation temporelle du signal de parole codé (phrase F6) ainsi que le représentation du RSB obtenu correspondant.

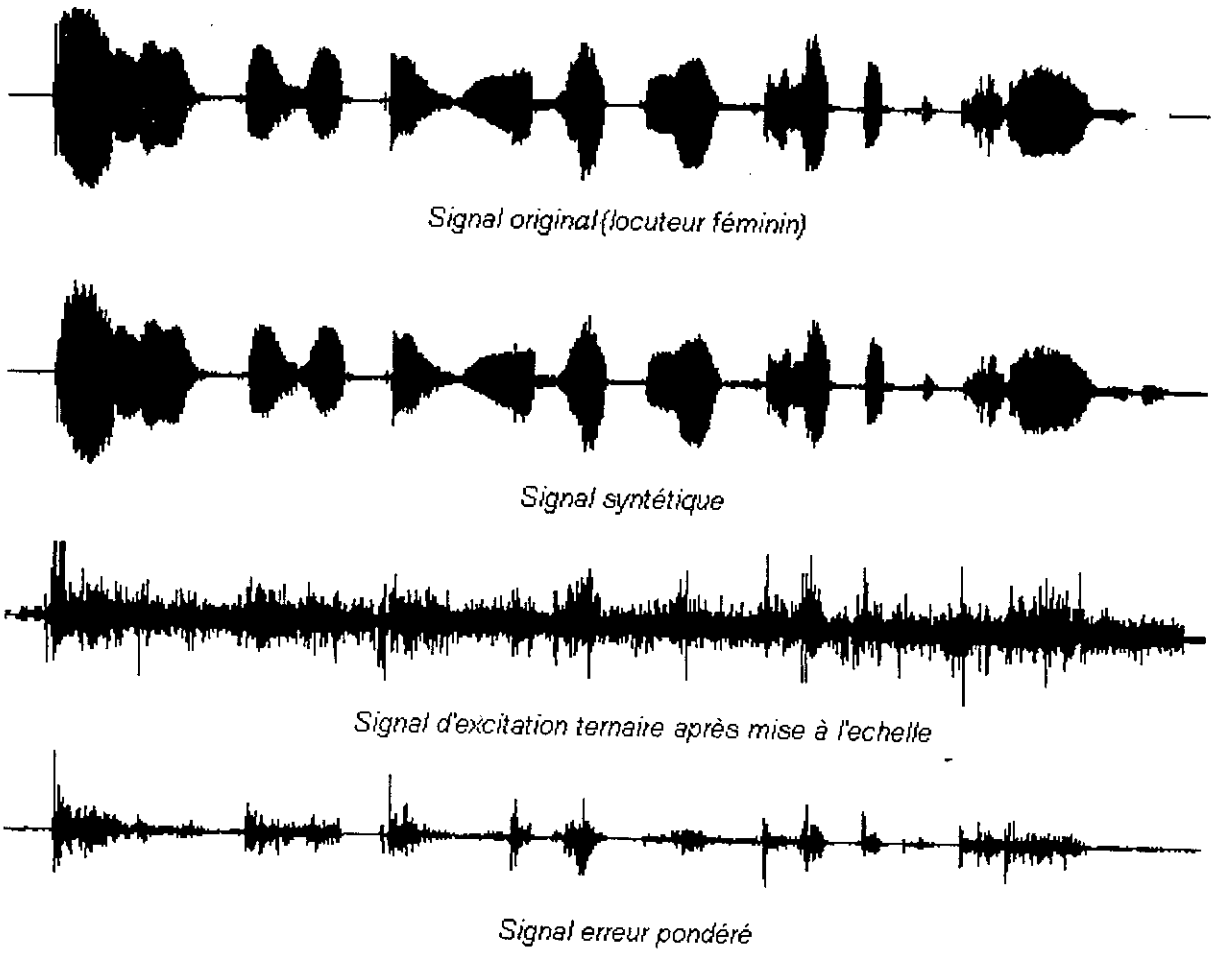



**Figure 4.13.** Exemple de phrase codée (F6) et les signaux obtenus. Le signal d'erreur pondéré et le signal d'excitation sont multipliés respectivement par 4 et 20.

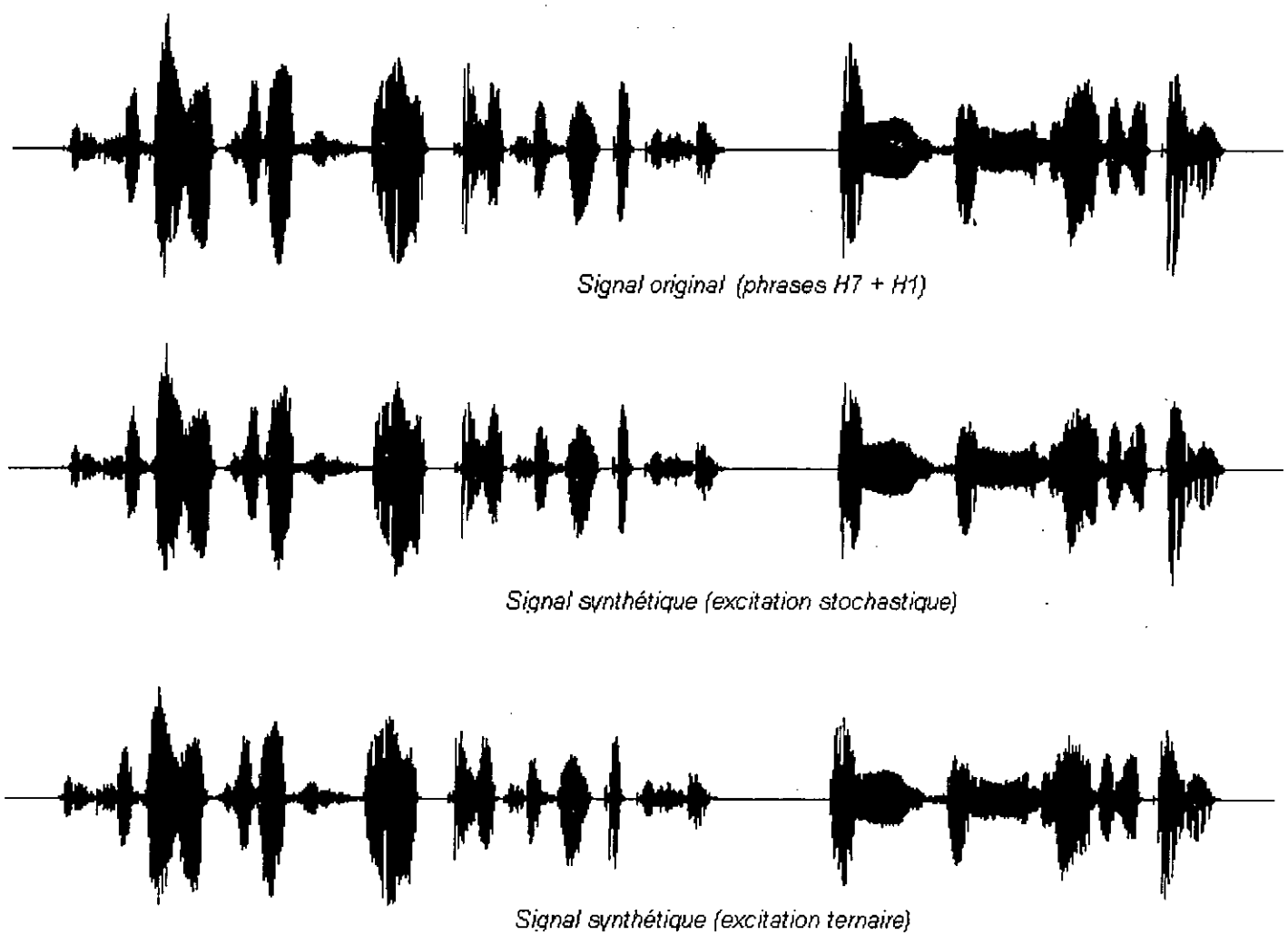



**Figure 4.14.** Comparaison de formes d'ondes de signaux synthétiques obtenus en utilisant des excitations ternaire et stochastique.

## 4.10. Codeur LD-CELP à large bande

Les codeurs CELP sont utilisés comme codeur (en bande étroite) à faible et moyen débits dans plusieurs applications. Ces codeurs fournissent une bonne qualité de la parole aux débits typiques de 4.8 à 8 Kbits/s et fournissent une excellente qualité aux débits supérieurs à 8 Kbits/s.

Cependant, il y a des applications où il n'est pas nécessaire de coder à de faibles débits. Une augmentation de la largeur de bande est généralement souhaitable. Par exemple, dans les applications de vidéo conférence.

Si des signaux de parole large bande, sont utilisés, la qualité perceptuelle globale peut être améliorée. Il a été établi que l'addition de la bande passante de 50 à 200 Hz ainsi que celle de 3.4 à 7 kHz améliore l'agrément d'écoute [35].

Appliquons l'algorithme LD-CELP réalisé pour développer un codeur de la parole large bande à un débit inférieur à 64 Kbits/s (32 Kbits/s).

L'approche employée (la plus simple) est de doubler la fréquence d'échantillonnage et d'utiliser le codeur LD-CELP pour coder la totalité de la bande 0 à 7 kHz en modifiant l'ordre de prédiction du filtre de synthèse [36]. La figure 4.15 nous montre l'évolution du RSB segmental en fonction de l'ordre de prédiction du filtre de synthèse. Nous remarquons qu'il n'est pas nécessaire de garder l'ordre de prédiction aussi élevé que p = 50. Afin de réduire la complexité, l'ordre de prédiction a été diminué à p = 20 ; la perte de performance est peu significative.

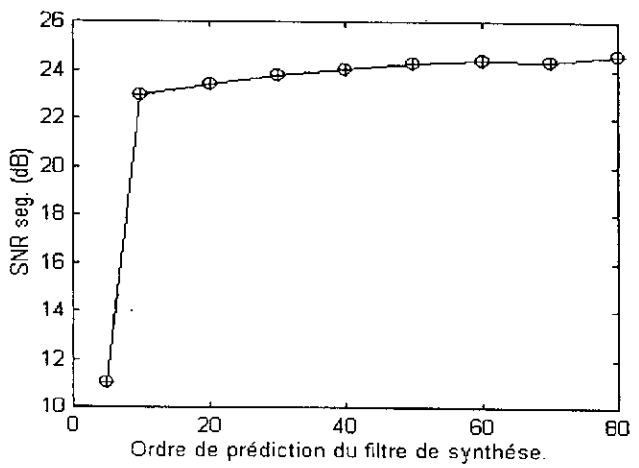



**Figure 4.15.** Evolution du RSB en fonction de l'ordre de prédiction du filtre de synthèse.

### 4.10.1. Evaluation des performances

Afin d'évaluer les performances de ce codeur, nous procédons au codage d'une série de phrases (large bande) phonétiquement équilibrées. La fréquence d'échantillonnage est de 16 kHz.

Les phrases utilisées sont les suivantes:

**LF1**: "Quand il s'est réveillé, il était trop tard. Huit satellites ont été mobilisés"

**LF2**: "La pirogue se mit en travers du courant"

**LF3**: "I gather you will be abandoning the major revisions"

**LH1**: "The other memorable event in that conference was the worst presentation I ever heard"

**LH2**: "I must have reread that article three times before I realized what was bothering me"

**LH3**: "Oh dear, the speaker apologized, I have been out of synchronization'

Les trois premières phrases sont prononcées par des locuteurs féminins, les trois dernières sont prononcées par des locuteurs masculins.

| Phrases | RSB | RSB segmental |
|---|---|---|
| LF1 | 22.56 | 21.65 |
| LF2 | 23.13 | 22.35 |
| LF3 | 21.72 | 21.65 |
| LH1 | 23.42 | 23.14 |
| LH2 | 23.17 | 23.85 |
| LH3 | 22.08 | 22.41 |

**Tableau 4.6.** Performances du codeur LD-CELP large bande.

La qualité de la parole obtenue est excellente, comme le montre le tableau 4.6, les tests d'écoute ont permis de confirmer la transparence de la qualité. En effet, aucune différence entre les signaux orignaux et ceux synthétisés n'est décelable.

La figure 4.16 nous montre une illustration de l'évolution du RSB obtenu pour une phrase codée (LH3).

La figure 4.17 nous donne une comparaison de forme d'onde d'un segment de parole de la phrase (LF2).

La figure 4.18 nous donne un exemple de codage de la phrase (LF1), le signal synthétique, le signal excitation et le signal erreur pondéré ; les deux derniers signaux sont multipliés par des facteurs de 4 et 10 respectivement.

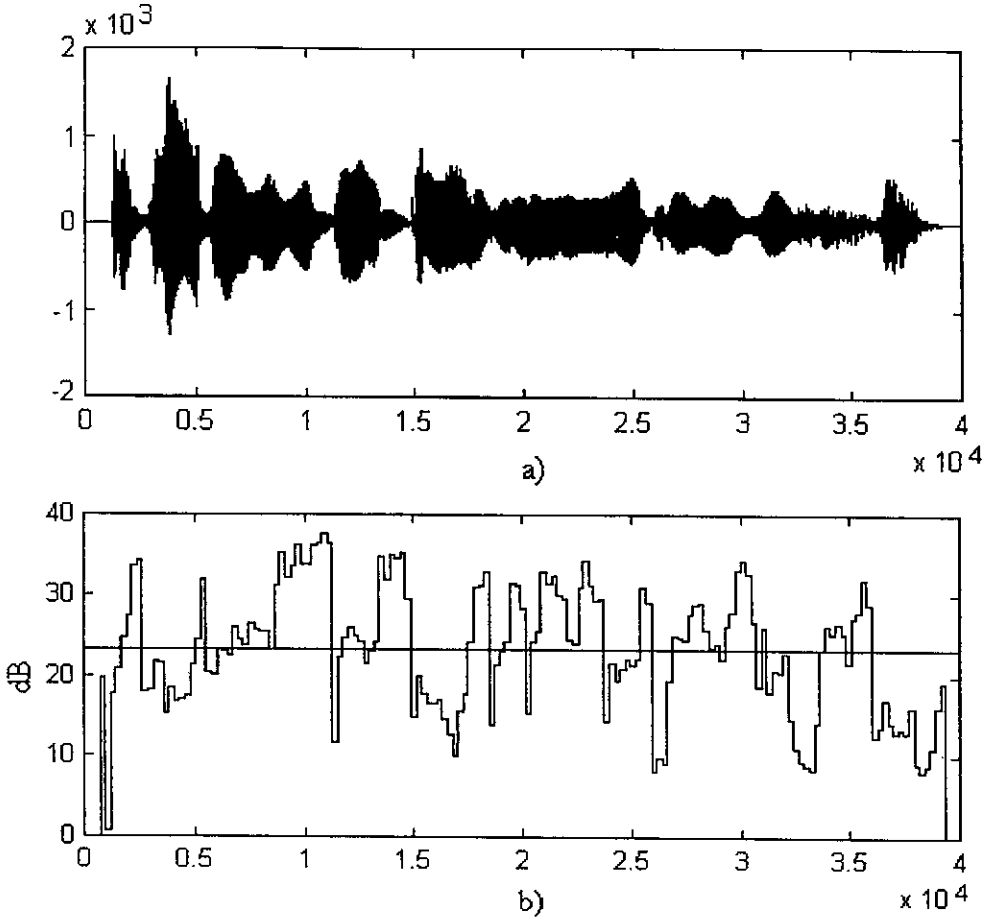



**Figure 4.16. a)** Signal de la Parole (phrase **LH3**) **b)** Evolution du RSB obtenu en fonction du temps (évalué sur des trames de 240 échantillons).

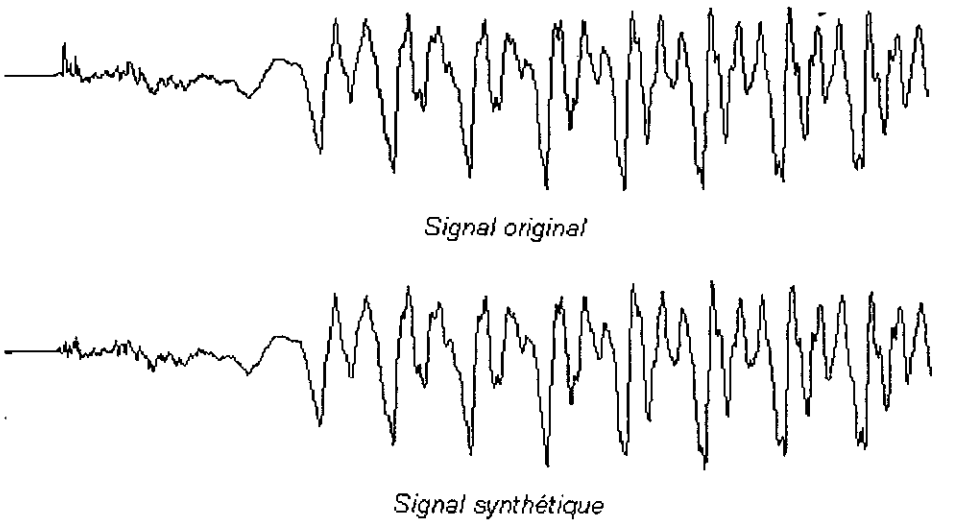



**Figure 4.17.** Comparaison de forme d'onde d'un segment de parole de la phrase (LF2) et le signal synthétique correspondant.

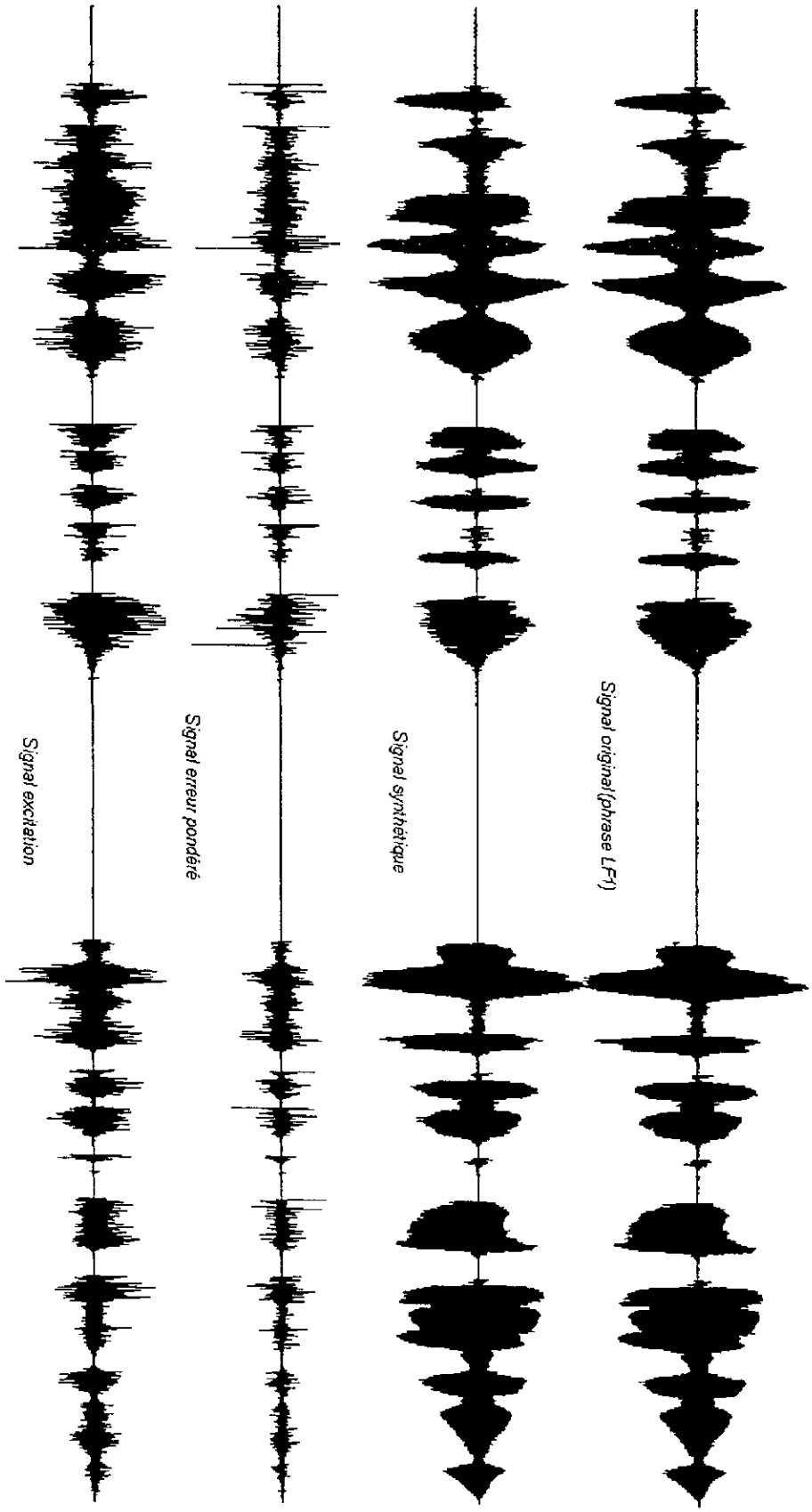



**Figure 4.18.** Exemple de phrase large bande codée et les signaux obtenus. Le signal erreur pondéré est multiplié par un facteur 4 et le signal d'excitation par un facteur 10.

## 4.11. Conclusion

Dans ce chapitre, nous avons donné une description technique de la mise en oeuvre des différents blocs du codeur/décodeur LD-CELP. L'opération de recherche dans le dictionnaire de QV de l'excitation est décrite en détail et est optimisée en réduisant le nombre d'opération arithmétique nécessaire dans l'opération de recherche du meilleur mot code correspondant au vecteur parole à coder.

Les performances d'un codeur stochastique dépendent essentiellement de la qualité du dictionnaire d'excitation. Pour cela, nous avons décrit la procédure de conception de dictionnaire "forme" et "gain" pour le codeur LD-CELP. En faisant la conception en boucle fermée, les contributions d'adaptation des prédicteurs LPC et gain sont prises en compte dans la conception. Nous avons présenté les performances du codeur réalisé en procédons au codage d'une série de phrases phonétiquement équilibrées. Le codeur LD-CELP nous permet d'avoir un signal de parole synthétique d'excellente qualité.

Les dictionnaires structurés nous permettent d'exploiter leur homogénéité afin de trouver des relations entre leurs différents éléments. Ces relations permettent de réduire la complexité de calcul par rapport au dictionnaire stochastique. Comme extension de notre travail, nous avons utilisé un générateur de code ternaire comme source d'excitation, ces vecteurs a composantes entières sont mis à l'échelle par un facteur dédié à chaque vecteur. Le coût de calcul de la distorsion est réduit par l'utilisation d'une récurrence entre les éléments des mot codes ternaire. La qualité de la parole obtenue pour ce type de dictionnaire est équivalente à celle obtenue avec un dictionnaire stochastique.

Une autre extrapolation est faite en appliquant au codeur réalisé un signal de parole large bande pour obtenir un codeur à 32 Kbits/s. Les performances obtenues nous montre que les signaux de parole reconstruits ont une qualité transparente. Néanmoins la complexité a pratiquement doublée.

# Conclusions générales

Notre travail consistait à réaliser un codeur/décodeur de la parole en bande étroite (bande téléphonique) à un débit de 16 Kbits/s ayant un faible retard (< 5 ms).

Nous avons commencé par décrire l'algorithme de codage CELP, cette technique se base sur le principe d'analyse par synthèse. Elle consiste à faire passer successivement des séquences d'excitation issues d'un dictionnaire ou d'un générateur de codes à travers une succession de filtres qui modélisent le conduit vocal puis choisir l'excitation donnant la plus petite distorsion moyenne quadratique après pondération de l'erreur par un filtre perceptuel.

Ensuite, le codeur LD-CELP a été décrit. Ce codeur fournit une combinaison de retard de codage assez faible et une haute qualité de la parole ce qui n'était atteint dans le passé qu'avec les standards G.721 32 Kb/s ADPCM ou le G.711 64 Kb/s PCM. Les principales caractéristiques du codeur sont :

- un prédicteur LPC d'ordre supérieur et une analyse LPC adaptative en backward ;
- adaptation régressive du gain d'excitation au travers un prédicteur log-gain adaptative ;
- l'opération de fenêtrage pour les analyses LPC est assurée par la fenêtre de BARNWELL modifiée qui donne les coefficients d'autocorrélations d'une façon récursive. Cette fenêtre est la réponse impulsionnelle d'un filtre IIR d'ordre 2 ;
- le dictionnaire d'excitation est optimisé en boucle fermée ;
- un filtre perceptuel de pondération est utilisé pour le masquage du bruit de codage.

L'opération de recherche dans le dictionnaire de QV de l'excitation a été décrite. Elle a été optimisée en réduisant le nombre d'opération arithmétique lors de la recherche du meilleur mot code correspondant au vecteur parole à coder. En effet, le cycle

d'adaptation des coefficients du filtre de synthèse d'ordre 50 et du filtre perceptuel est de 4 vecteurs (soit 20 échantillons) toute en gardant un vecteur de base de bufferisation de 5 échantillons seulement. Ce cycle nous permet de pré calculer une table d'énergie fixe pendant l'encodage de 4 vecteurs. De même l'utilisation des frontières des cellules de quantification permet de réduire le nombre de multiplications requises lors de la recherche de la distorsion minimale. Les indices correspondants aux meilleurs mots codes "forme" et "gain" sont concaténés en 1 mot de 10 bits.

L'opération de conception du dictionnaire d'excitation est une partie très importante dans le processus de conception de n'importe quel codeur de la parole. Pour cette raison, l'opération de conception du dictionnaire d'excitation est décrite en détail. Ce dictionnaire est composé de deux dictionnaires, l'un de "forme" sur 7 bits et le deuxième de "gain" sur 3 bits.

L'algorithme de conception du dictionnaire forme est décrit. Il est similaire à l'algorithme LBG. La convergence n'étant pas garantie car l'optimisation se fait en boucle fermée (la séquence d'apprentissage change d'un itération à une autre). Après plusieurs itérations, on choisit le dictionnaire donnant le meilleur RSB segmental. Un dictionnaire initial est appliqué afin d'accélérer la convergence de la conception. Le dictionnaire gain est composé de 3 bits, 2 bits pour les niveaux d'amplitudes et 1 bit de signe ; Il a été conçu en appliquant l'algorithme LBG. La séquence d'apprentissage est composé de gains optimaux (non quantifiés) obtenus par application du meilleur dictionnaire "forme" pour le codage de la base de donnés de parole.

Les mesures objectives obtenues et les tests d'écoute confirment la qualité transparente de la parole synthétique obtenue à la sortie du décodeur.

Après avoir conçu le dictionnaire stochastique, ce dictionnaire a été remplacé par un générateur (dictionnaire algébrique) de codes ternaire [-1, 0, +1] dont les mots codes entiers sont mis à l'échelle par des facteurs non entiers. La structure algébrique nous permet de réduire la complexité de calcul en remarquant l'existence d'une structure qui permet d'avoir les termes d'énergies dans l'expression de la distorsion en fonction de termes particuliers "directeurs" d'une façon ne nécessitant pas d'opérations de multiplication mais seulement des additions.

La qualité de la parole obtenue est assez proche de celle obtenue avec un dictionnaire stochastique mais avec une réduction de complexité de calcul et de stockage.

Nous avons extrapolé notre travail en adaptant le codeur réalisé pour des signaux de parole large bande (50 - 7000 Hz ) et à un débit de 32 Kbits/s.

La qualité de la parole obtenue est excellente mais l'inconvénient majeur de l'approche utilisée est que la complexité de l'algorithme a pratiquement doublée.

Le programme réalisé est opérationnel en temps différé. On peut l'exploiter pour la compression des fichiers de paroles dans les supports de stockage tels les CD ROMs ou les supports de stockage miniaturisés. Un taux de compression de 8 est atteint pour les échantillons de parole quantifiés sur 16 bits.

Nous souhaitons que le travail réalisé soit suivi par une implantation de l'algorithme LD-CELP sur une carte DSP pour les applications en temps réel. Des contraintes liées au codage en temps réel peuvent apparaitre. Pour réduire la complexité, il est préférable d'utiliser le dictionnaire ternaire.

# Références Bibliographiques


[1] M. R. Schroeder and B. S. Atal, " Code Excited Linear Prediction (CELP) : High Quality Speech at Very Low Bit Rates," in Proc. Int. Conf. ASSP, pp. 937-940, Apr. 1985.

[2] B. S. Atal and J. R. Remde, "A New Model of LPC Excitation for Producing Natural Sounding-Speech at Low Bit Rates," in Proc. IEEE Int. Conf. Acoust. Speech and Signal Process., Apr. 1982, pp. 614-617.

[3] B. S. Atal, "Predictive Coding of Speech at Low Bit Rates," IEEE Trans. Commun., Vol. COM-30, No.4, Apr. 82.

[4] R. Zelinski and P. Noll, "Adaptive Transform Coding of Speech Signals, " IEEE Trans. Acoust., Speech, Signal Processing. ,vol. ASSP-25, pp. 299-309, may 1977.

[5] F.K. Soong, R. V. Cox, and N. S. Jayant, "A High Quality Subband Speech Coder with Backward Adaptive Predictor and Optimal Time-Frequency Bit Assignment," in Proc. IEEE Int. Conf. Acoust. Speech and Signal Process., Apr. 86 pp. 2387-2390,

[6] R. Boite et M. Kunt, "Traitement de la parole,"Presses Polytechniques Romandes, 1987.

[7] Calliope, "La parole est son traitement automatique," Collection Technnique et Scientifique des télécommunications, MASSON 1989.

[8] J. Stachurski, "A Pitch Pulse Evolution model for Linear Predictive Coding of Speech,"PhD's Thesis . McGill University Montreal, Canada. May 1997.

[9] N. Jayant and P. Noll, "Digital Coding of Waveforms : Principles and applications to speech and video," Englewood Cliffs, New Jersy : Prentice-Hall, 1984.

[10] N. Jayant, "Signal Compression : Technology Targets and Research Directions," IEEE Journal on Selected Areas in Communications.Vol. 10, No. 5, June 92.

[11] J. H. Chen, R. V. Cox, Y. C. Lin, N. Jayant and M. J. Melchner, "A Low-Delay CELP Coder for the CCITT 16 kb/s Speech Coding Standard," IEEE Journal on selected areas in communications, Vol. 10. No.5. June 1992.

[12] J. MacQueen, "Some Methods for Classification and Analysis of multivariate observations," In Proc. Of Fifth Berkeley Symposium on Math. Stat. And Prob., Vol. 1, pp. 281-296, 1967.

[13] R. M. Gray, "Vector Quantization," IEEE Trans. Acoustics, Speech, Signal Processing, Vol. ASSP-34, April 1984.

[14] J. Makhoul, S. Roucos and H. Gish, "Vector Quantization in Speech Coding," Prcoc. IEEE, Vol. 73, pp. 1551-1588, November 1985.

[15] Linde, Buzo and Gray, "An Algorithm for Vector Quantization Design, " IEEE Trans. Comm., jan. 1980.

[16] I. Katsavounidis, C.-C. Jay Kuo, and Zhen Zhang, " A New Initialisation Technique for Generalized Lloyd Iteration," IEEE Signal Processing Letters. Vol. 1 No. 10 october 1994.

[17] J. Makhoul, "Linear Prediction : a Tutorial Revue," Proc. IEEE, Vvol. 63. Pp 561-580, 1975.

[18] J.D. Markel, A. H. Gray, "Linear Prediction of Speech," Springer Verlag, Berlin 1976.

[19] N. Moreau, "Codage Préditif du Signal de Parole à Débit Réduit : une Présentation Unifiée," Annales Télécommunications, 46 No. 3-4, 1991.

[20] M. R. Schroeder and B. S. Atal, "Stochastic Coding of Speech Signals at Very Low Bit Rates: The Importance of Speech Perception," Speech Communications Vol. 4, Aug 1985, pp. 155-162.

[21] L.Watts and V. Cuperman, "A Vector ADPCM Analysis by Synthesis Configuration for 16 kb/s Speech Coder," in Proc. IEEE Global Comm. Conf. Dec. 1988, pp. 275-279.

[22] S. Singhal and B.S. Atal, "Improving Performance of Multi-Pulse LPC Coders at Low Bit Rates,". In Proc. Intern. Conf. Acoust. Speech, Signal Process., vol. 1, paper No. 1.3, March 1988

[23] R. P. Ramachandran and P. Kabal, "Pitch Prediction Filters in Speech Coding, ". Vol. ASSP 37 No. 5, May1989, pp. 642-650.

[24] R. P. Ramachandran and P. Kabal, "Stability and Performance Analysis of Pitch Filters in Speech Coders," IEEE Trans. Acoustics, Speech, Signal Proess., Vol. ASSP-35, July 1987, pp. 937-946.

[25] James H. Y. Loo, "'Intraframe and Interframe Coding of Speech Spectral Parameters," Master's Thesis, McGill University, Montreal, Canada, September 1996.

[26] B. Atal, S. Shroeder, "Predictive Coding of Speech Signals and Subjective Error Criteria," IEEE Trans. on ASSP, vol. ASSP- 27, No. 3 pp. 247-254, June 1979.

9[27] J.H. Chen, "High Quality 16 kb/s Speech Coding with a One Way Delay Less than 2 ms," in Proc. Int. Conf. Acoust. Speech, Signal Process. Apr. 1990, pp. 453-456.

[28] J.D. Gibson and G. B. Haschee, "Backward Adaptive Tree Coding of Speech at 16 kbps ".In Proc. IEEE in Proc. InternConf. Acoust. Speech, Signal Process., Apr. 1988, pp. 251-254

[29] V. Cuperman, A. Gercho, R. Pettigrew, Jey- Hsin Yao, "Low Delay Vecctor Excitation Coding of Speech at 16 kb/s," IEEE Transactions On Communications, Vol. 40, No. 1, January 1992.

[30] T. P. Barnwell III, "Recursive Windowing for Generating Autocorrelation Coefficients for LPC Analysis," in IEEE Trans. Acoust. Speech, Signal Process., pp. 1062-1066, Oct 81.

[31] J.H. Chen and A. Gercho, "Gain-Adaptive Vector Quantization with Application to Speech Coding," IEEE Trans. Comm., Sep. 1987, pp. 918-930.

[32] "Detailed Desccription of AT & T's LD-CELP Algorithm" Technical Report.

[33] R. Di Francesco, "Codage Algébrique de la parole : Prédiction linéaire à excitation par code ternaire," Annales des Télécom., Vol. 47, No. 5-6, 1992.

[34] Gerson, Jasiuk, "Vector Sum Excited Linear Prediction (VSELP) Speech Coding at 8 kb/s," ICASSP 1990.

[35] C. McElroy, B. Murray, A. D. Fagan, "Widband Speech Coding in 7.2 kb/s," Proc. of ICASSP 1993 pp. II-624-II-627.

[36] A. Fuldseth, E. Harborg, F. T. Johansen and J. E. Knudsen, "Wideband Speech at 16 kbits/s for a Videophone Application,". Speech Communication, Vol. 11, June 1992, No, 2-3, pp 143-148.

[37] V. Ramamoorthy and N. S. Jayant, "Enhancement of ADPCM Speech by Adaptive Posfiltering," Bell Syst. Tech. J., vol. 63, no. 8, pp. 1465-1475, Oct. 1984.

[38] M. Mauc and G. Baudoin, "Reduced Complexity CELP Coder," Proc. of ICASSP 92 pp. I-53-I-56.

[39] P. Dymarski and N. Moreau, "Algorithms for the CELP Coder with Ternary Excitation," EUROSPEECH, pp. 241-244, 1993.

[40] V. Iyengar and P. Kabal, "A Low Delay 16 kb/s Speech Coder," in Proc. IEEE Int. Conf. Acoust. Speech and Signal Process, Apr. 1988 pp. 243-246.

[41] M. Djeddou and M. Halimi, " A Low delay 16 Kbits / s speech coder," In Proceding IEEA International Annual Conference, Vol. 2, University of Batna December 1997.

[42] N. S. Jayant and V. Ramamoorthy, "Adaptive Posfiltering of 16 kb/s ADPCM Speech," in Proc. IEEE ICASSP, Apr. 1986, pp. 829-832.

[43] R. V. Cox ,S. L. Gay , Y. Shoham ,S. R. Quackenbush, N. Seshadri and. S. Jayant, " New Directions in Subband Coding," IEEE J. Selec. Areas Comm., vol. 6, pp. 391-904 Feb 1988.

[44] N. M. Berouti, J. Jachner, D. Sloan and P. Mermelstein, "Reducing Signal Delay in Multi Pulse Coding at 16 kb/s," in Proc. IEEE Int. Conf. ASSP., Apr. 1986 pp. 3043-3046.

[45] T. Taniguchi, S.Unagami, K. Iseda, Y. Moshida and S. Tominaga, "A 16 kbps ADPCM with Multi-Quantizer (ADPCM-MQ) Codec and its Implementation by Digital Signal Processor," In Proc. IEEE ICASSP., Apr. 1987, pp. 1340-1343.

[46] M. W.Marcellin, T. R. Fischer, and J. D. Gibson, "Predictive Trellis Coded Quantization of Speech," in Proc. IEEE ICASSP Apr.1988 pp. 247-250.

[47] P. Kroon , ED F. Deprettere, R. J. Sluyter, "Regular Pulse Excitation. A Novel Approach to Effective and Efficient Multipulse Coding of Speech," IEEE Transactions on Acoustics Speech and Signal Processing Trans. ASSP Vol. 34 pp. 1054-1063 Oct. 86.

[48] I.M. Trancoso and B. S. Atal, "Efficient Procedures for Finding the Optimum Innovation in Stochastic Coders," Proc. of Intern. Conf. on Acoust. Speech and Signal Processing ICASSP 86 Tokyo pp. 2375-2378.

[49] J.H. Chen and A. Gersho, "Real Time Vector APC Speech Coding at 4800 bps with Adaptive Postfiltering," Proc. of Intern. Conf. on Acoust. Speech and Signal Processing ICASSP 87 pp. 2185-2188.

[50] J. P. Adoul, P. Mabileau, M.Delprat and S. Morisette,"Fast CELP Coding Based on Algebraic Codes," IEEE ICASSP 1987 pp. 1957-1960.

[51] Y. M. Cheng, D.O. Shaughnessy and P. Mermelstein, "Statistical Recovery of Widband Speech from Narrrowband Speech," IEEE, Transactions on Speech and Audio Processing October 1994 Vol.2 No.4, pp 544-548.

[52] V. Cuperman and R. Peng, "Lattice Low Delay Vector Excitation Coding of Speech at 8-16 Kb/s," IEEE Trans. On Communications.Vol. 42 No. 6 pp 2219-2223 June 1994.

[53] R. Soheili, A.M. Kondoz and B.G. Evans, "An 8 Kb/s LD-CELP with Improved Excitation and Perceptual Modelling," Proc. of ICASSP 93. pp II-620 II-627.

[54] R. Soheili, A.M. Kondoz and B.G. Evans, "Techniques for Improving the Quality of LD-CELP Coders at 8 Kb/s," Proc. of ICASSP 92. pp. I-49 I-52.

[55] A. Gercho, "Asymptotically Optimal Block Quantization," IEEE. Trans. IT-28, 157-166.

[56] I. M. Troncoso and B. S. Atal, "Efficient Procedures for Selecting the Optimum Innovation in Stochastic Coders," IEEE Trans. Acoustics, Speech, Signal Processing, Vol. 38, pp. 385-396, March. 1990.

[57] Hui Guanghui, T. Wenshun, N. Weizhen, W. Dejun, "Real Implementation of 16 kb/s Low Delay CELP Speech Algorithm on a TMS320C30," IEEE TENCOM' 93. BEIJING.

[58] V. Cuperman, R. Pettigrew, "Robust Low-Complexity Backward Adaptive Pitch Predictor for Low-Delay Speech Coding," IEE Procedings-I, Vol. 138, No. 4, August 1991.

[59] S. Marlow, B. Buggy, "Classified Vector Excitation Speech Coding," IEE Procedings, Vol, 136, Pt. I, No. 5, October 1989.

[60] J.L. Flanagan, "Speech analysis, synthesis and perception," Springer-Verlag, 1972.

[61] S. M. Kay, "Modern Spectral Estimation, Theory and Application," Prentice Hall, Englewood Cliffs, New Jersy 07632, 1988.

[62] W. H. Press, B. P. Flannery, S. A. Teukolsky, W. T. Vetterling, "Numerical Recipes in C. The Art of Scientific Computing," Cambridge University Press.

[63] J.R. Deller, Jr, J. G. Proakis, and J. L. Hall, "Discrete - time Processing of Speech Signals," New York : MacMillan, 1993.

[64] I. A. Gerson and M. A. Jasiuk, "Techiques for Improving the Performance of CELP-Type Speech Coders," IEEE Journal on Selected Areas in Communications. Vol. 10, No. 5 June 1992.

[65] N. Jayant, J. Johnston and R. Safranek, "Signal Compression Based On Models of Human Perception," Proc. IEEE, Vol. 81, pp. 1385-1422, April 1993.

[66] W. B. Kleijn and D. J. Krasinski, "Fast Methods for the CELP Speech Coding Algorithm," IEEE Trans. Acoustics, Speech, Signal Processing, Vol. 73, pp. 1330-1342, Aug. 1990.

[67] N. Moreau and P. Dymarski, "Selection of Excitation Vectrors for the CELP Coders," IEEE Trans. Speech and Audio Processing, Vol. 2, pp. 29-41, Jan. 1994.

[68] B. S. Atal and J. R. Remde, "A NEW Model of LPC Excitation for Producing Natural-Sounding Speech at Low Bit Rates," in Proc. IEEE Int. Conf. On Acoustics, Speech, Signal Processing, pp. 614-617, Paris 1982.

[69] M. Elshafei-Ahmed and M. I. Al Suwaiyel, "Fast Methods for Code Search in CELP," IEEE Trans. Speech and Audio Processing, Vol. 1, pp. 315-325, July. 1993.

[70] M. Johnson and T. Taniguchi, "On Line and Off Line Computational Reduction Techniques Using Backward Filtering in CELP Speech Coders," IEEE Transactions On Signal Processing, Vol. 40, No. 8. August 1992.

[71] M. Copperi, D. Sereno, "CELP Coding of Speech at 9.6 kb/s," IEEE-ICASSP, pp. 1685-1688, 1986.

[72] J. H. Chen and A Gersho, "Vector Adaptive Coding of Speech at 9.6 kb/s," IEEE-ICASSP, pp. 1693-1696, 1986.

[73] B. S. Atal, V. Cuperman and A. Gersho, "Advances in Speech Coding," Kluwer Academic Publishers 1991.

[74] C. R. Galand, J. E. Menez and M. M. Rosso, "Adaptive Code Excited Predictive Coding," IEEE Transactions On Signal Processing, Vol. 40, No. 6, June 1992.

[75] S. Wang, A. Sekey and A. Gersho, "An Objective Measure for Predicting Subjective Quality of Speech Coders," IEEE Journal On Selected Areas In Communications, Vol. 10, No. 5, June 1992.

[76] W. Y. Chan, S. Gupta and A. Gersho,"Enhanced Multistage Vector Quantization Design," IEEE Transactions on Communications, Vol. 40, No. 11, November 1992.

[77] T. V. Ramabadran and C. D. Lucck, "Complexity Reduction of CELP Speech Coders Through The Use of Phase Information," IEEE Transactions on Communications, Vol. 42, No. 2/3/4, February/March/April 1994.

[78] J. H. Chen and A. Gersho, "Covariance and Autocorrelation Methods for Vector Linear Prediction," in Proc. Int. Conf. Acoust., Speech, Signal Procedding, (Dallas), pp. 1545-1548, April 1987.

[79] J. Grass, "Quantization of Predictor Coefficients In Speech Coding," Master's Thesis, McGill University, Montreal, Canada, September 1990.

[80] Y. Shoham, " Vector Predictive Quantization of The Spectral Parameters For Low Rate Speech Coding," in Proc. Int. Conf. Acoust., Speech, Signal Proceding, (Dallas), pp. 2181-2184, April 1987.

[81] C. Laflamme, J. P. Adoul, R. Salami, S. Morisette and P. Mabilleau, "16 kbps Wideband Speech Coding Technique Based On Algebraic CELP," IEEE-ICASSP, pp. 13-16, 1991.

[82] L. A. Hernandez-Gomez, F. J. Casajus-Quiros, A. R. Figueras-Vidal and R. Garcia-Gomez, "Reducing Complexity on a Code Excited Linear Predictor," Signal Processing III, EURASIP, pp. 481-483, 1986.

[84] C. Galand, D. Esteban et Menez, "Techniques de Codages de la parole à Débit Moyen (5 à16 Kbits/s)," L'onde Electrique, Vol. 61. No. 8-9, pp. 38-53. 1981.

[85] S. P. Lloyd, "Least Squares Quantization in PCM," IEEE Trans. On Information Theory, Vol. IT 28, No. 2, March 1982.

[86] T. V. Ramabdran and D. Sinha, "Speech Data Compression Through Sparse Coding of Innovation," IEEE Transactions on Speech and Audio Processing, Vol. 2, No. 2 pp. 350-352. April 1994.

[87] V. Cuperman and R. Peng, "Lattice Low Delay Vector Excitation Coding of Speech at 8-16 Kb/s," IEEE Transactions on Communications Vol. 42, No.6, pp. 2219-2223. June 1994.

[88] H. C. Woo and J. D. Gibson,"Low Delay Tree Coding of Speech at 8 Kbits/s," IEEE Transactions on Speech and Audio Processing, Vol. 2, No. 3 pp. 361-370. July 1994.

[89] X. Wu and L. Guan, "Acceleration of the LBG Algorithm," IEEE Transactions on Communications Vol. 42, No.2/3/4 pp. 1512-1517 Feb/March/April 1994.

[90] C. Nassar and M. R. Soleymani, "Codebook Design For Quantization Using Simulated Annealing," IEEE Transactions on Speech and Audio Processing, Vol. , No. 4, pp. 400-404, Oct. 1993.

[91] S. Dimolitsas, F. L. Corcoran, M. Baraniecki and J. G. Philips Jr, "Use of Low Delay Code Excited Linear Prediction Technology in Circuit Multiplexed Networks," Proc. Of Int. Conf. On Acoustics, Speech and Signal Processing, ICASSP 93, pp. II-608 II-611.

[92] R. Soheili, A. M. Kondoz and A. D. Evans, "An 8 KB/S LD-CELP with Improved Excitation and Perceptaul Modelling," Proc. Of Intern. Conf. On Acoustics, Speech and Signal Processing ICASSP 93 pp. II-616 II-619.

[93] T. Eriksson and J. Sjoberg, "Dynamic Bit Allocation in CELP Excitation Coding," Proc. Of Intern. Conf. On Acoustics, Speech and Signal Processing ICASSP 93 pp. II-171 II-174.

[94] D. Sen and D. H. Irving and W. H. Holmes, "Use of an Auditory Model to Improve Speech Coders," Proc. Of Intern. Conf. On Acoustics, Speech and Signal Processing ICASSP 93 pp. II-411 II-414.

[95] A. Kataoka, T. Moriya and S. Hayashi, "An 8 KB/S Speech Coder Based on Conjugate Structure CELP," Proc. Of Intern. Conf. On Acoustics, Speech and Signal Processing ICASSP 93 pp. II-592 II-595.

[96] Y. Tanaka and T. Tanaguchi, "Efficient Coding of LPC Parameters Using Adaptive Prefiltring and MSVQ With Partially Adaptive Codebook," Proc. Of Intern. Conf. On Acoustics, Speech and Signal Processing ICASSP 93 pp. II-5 II-8.

[97] H. S Wang and N. Moayeri, "Trellis Coded Vector Quantization," IEEE Transactions on Communications, Vol. 40 No. 8, pp. 1273-1276, Aug. 1992.

[98] T. K. Wang, J. Foster and S. Ardalan, "Adaptive Vector Quantization for Waveform Coding," Proc. Of Intern. Conf. On Acoustics, Speech and Signal Processing, ICASSP 93 pp. I-101 I-104

[99] Y. J. Liu, "On Reducing the Bit Rate of a Celp-Based Speech Coder," Proc. Of Intern. Conf. On Acoustics, Speech and Signal Processing ICASSP 92 pp. I-49 I-52.

[100] N. Moreau and P. Dymarsky, "Successive Orthogonalization in Multistage Celp Coder," Proc. Of Intern. Conf. On Acoustics, Speech and Signal Processing ICASSP 92 pp. I-611-64.

[101] J. H. Yao, J. J. Shynk and A Gersho, "Low Delay VXC at 8 KB/S With Interframe Coding," Proc. Of Intern. Conf. On Acoustics, Speech and Signal Processing ICASSP 92 pp. I-45 I-48.

[102] M. Delprat, M. Lever and C. Gruet, "Efficient Excitation and Fast Selectioning CELP Coding of Speech," Eurospeech 89.

[103] A. M. Kondoz, K. Y. Lee, and B. G. Evans, "Speech Coding at 9.6 KB/S and Below Using Vector Quantized Transform Coder," EUROCON 88 8th European Conference On Electrothecnics Conference Procedings on Areas Communications, Stockholm Sweden, June 13-17, pp. 36-39.

[104] A. M. Kondoz, K. Y. Lee, and B. G. Evans, " A Robust Vector Quantized Sub Band Coder for Good Quality Speech Coding at 9.6 KB/S," EUROCON 88 8th European Conference On Electrothecnics Conference Procedings on Areas Communications, Stockholm Sweden, June 13-17, pp. 44-47.

[105] AL. Spataru, "Théorie de la transmission de l'information," Editions Masson.

---