République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

ECOLE NATIONALE POLYTECHNIQUE

Département d'Electronique

Mémoire de **MAGISTER**

Option : ***Traitement du signal & Communications***

Présenté par

Elias BENAMIRA
Ingénieur d'Etat en Electronique

Codeur de la Parole par Interpolation de la Forme d'Onde

Soutenu le 06/11/2003 devant le jury composé de :

Président	:	D^r	R. AKSAS ..... Professeur à l'ENP
Promoteur	:	D^r	D. BERKANI ..... Professeur à l'ENP
Examinateurs	:	D^r	M. HALIMI ..... Chargé de recherches au CSC
		D^r	L. HAMAMI ..... Maître de conférences à l'ENP
		D^r	M. GUERTI ..... Maître de conférences à l'ENP

A ma mère

et à la mémoire

de mon père

Remerciements

J'aimerais exprimer ma sincère reconnaissance à mon promoteur, Professeur Daoud Berkani, pour son support durant la période de préparation de ce travail et pour ses conseils précieux.
Mes remerciements à tous les membres du jury qui ont contribué à améliorer la qualité de ce travail par leurs remarques constructives.
Mes vifs remerciements au chef du département d'Electronique, Docteur Trabelsi, pour ses encouragements et sa compréhension ainsi qu'à tous mes profs de Post-Graduation.
Toute ma gratitude aux membres de ma famille et mes amis qui n'ont pas cessé de fournir l'aide matérielle et morale nécessaires à l'accomplissement de ce projet, en particulier, Hassan B. et Amine B..

Résumé

Le codage de la parole à des débits voisins de 4 kbps promet d'être largement employé dans des applications telles que la téléphonie visuelle et les communications mobiles et personnelles. Le présent travail a pour but de développer un codeur de la parole basé sur le schéma d'interpolation de la forme d'onde (WI :*Waveform Interpolation*), avec comme objectif une reconstitution fidèle de la parole à un débit proche de 4 kbps. Pour mettre en évidence les performances du model WI, un codeur a été implémenté (sans quantification) en utilisant le langage de programmation C (Borland C++ 3.1) en arithmétique flottante.

La bonne performance du modèle a été vérifiée par des tests d'écoute (tests subjectifs).

La simulation du codeur WI a donné une parole de bonne qualité avec un degré élevé d'intelligibilité et de naturel comparé aux codeurs conventionnels travaillant au même débit.

Mots clés : *codage de la parole, prédiction linéaire, interpolation de la forme d'onde, quantification.*

Abstract

Speech coding at bit rates near 4 kbps is expected to be widely deployed in applications such as visual telephony, mobile and personal communications. This research focuses on developing a speech coder based on the waveform interpolation (WI) scheme, with an attempt to deliver near toll-quality speech at rates around 4 kbps. A WI coder has been simulated in floating-point using the C programming language. The high performance of the WI model (without quantization) has been confirmed by subjective listening tests.

The implementation of the coder using the language Borland C++ 3.1 has produced a good quality coded speech with a high degree of intelligibility and naturalness when compared to the conventional coding schemes operating in the neighbourhood of 4 kbps.

Key words : *speech coding, linear prediction ,waveform interpolation , quantization.*

ملخص

تشفير الكلام بتدفقات تقارب 4 ك.ب./ثا.. يعد باستعمال واسع في مجالات مثل الهاتف المرئي و الاتصالات المتنقلة و الشخصية. هذا العمل يهدف إلى تحقيق نظام تشفير الكلام يعتمد على مبدأ استكمال شكل الموجة مع محاولة إعادة تشكيل الكلام بوفاء و بتدفق يجاور 4 ك.ب./ثا.. لا ثبات فعالية هذا النظام، تم تجريبه من خلال برنامج يؤدي مهمة التشفير بدون تكميم باستخدام لغة البرمجة C. تم التأكد من كفاءة هذا النموذج من خلال اختبارات السمع. النموذج غير المكمم لهذا النظام أنتج كلاما طبيعيا ذا نوعية جيدة بالمقارنة مع أنظمة التشفير المعتادة و التي تعمل بنفس التدفق.

كلمات مفتاحية. *تشفير الكلام، التنبؤ الخطي، استكمال شكل الموجة، التكميم.*

Table des matières

Liste des abréviations

ADPCM : Adaptive Differential Pulse Code Modulation
CDMA : Code Division Multiple Access
CELF : Code-Excited Linear Prediction
CODEC : COder and DECoder
CW : Characteristic Waveform
DCVQ : Dimension Conversion Vector Quantization
DOD : Department of Defense (U.S.)
DSP : Digital Signal Processor
DTFS : Discrete-Time Fourier Series
EVRC : Enhanced Variable Rate Codec
FBR : Fixed Bit-Rate
FS : Federal Standard (U.S.)
GLA : Generalized Lloyd Algorithm
IMBE : Improved Multi-Band Excitation
ITU : International Telecommunication Union
ITU-T : ITU - Telecommunication standardization sector
LD-CELP : Low-Delay Code Excited Linear Prediction
LP : Linear Prediction
LPC : Linear Prediction Coding
LSF : Line Spectral Frequency
LSP : Line Spectral Pair
MBE : Multi-Band Excitation
MELP : Mixed Excitation Linear Prediction
MIPS : Million Instructions Per Second
MOS : Mean Opinion Score
MSE : Mean Square Error
PCM : Pulse Code Modulation
PWI : Prototype Waveform Interpolation
REW : Rapidly Evolving Waveform
SEW : Slowly Evolving Waveform
SNR : Signal-to-Noise Ratio
V/UV : Voiced /UnVoiced
VBR : Variable Bit-Rate
VDVQ : Variable Dimension Vector Quantization
VQ : Vector Quantization
WI : Waveform Interpolation

INTRODUCTION

Dans les systèmes numériques modernes, un signal de parole est représenté dans un format numérique ; c'est à dire une séquence binaire de bits. Il est souvent préférable pour le signal d'être représenté avec le minimum de bits possible. Pour les applications de stockage, moins de bits signifie moins d'espace mémoire requis. Pour les applications de transmission, moins de bits signifie une diminution de la bande passante, de la puissance et de la mémoire. Il est, donc, très rentable d'utiliser un algorithme efficace de compression de la parole dans un système de transmission ou de stockage de la parole numérique. Le codage de la parole est la technologie qui offre de tels algorithmes.

Bien que des bandes passantes très larges sont devenues possibles dans les communications à support filaire grâce au développement rapide des moyens de transmission optiques, il y a toujours un besoin croissant de conservation et de réduction de la bande passante, plus particulièrement, dans les transmissions sans fil et par satellites. D'un autre côté, avec la tendance croissante des communications multimédia et autres applications liées à la parole telles que les répondeurs automatiques, la demande de réduction de la mémoire dans les systèmes de stockage de la voix est de plus en plus présente. Ces exigences font que le codage de la parole reste un domaine de recherche et de développement très vivant.

De surcroît, l'émergence de processeurs spécialisés de traitement du signal (DSP) de plus en plus rapides fournit aux chercheurs en codage de la parole plus d'encouragements pour penser à des algorithmes nouveaux et plus améliorés, des algorithmes qui peuvent nécessiter un effort de calcul plus important qu'avant.

Chapitre 1

APERCU SUR
LE CODAGE DE LA PAROLE

1.1 Supports de codage de la parole

Dans ce paragraphe, on va définir les composants d'un codeur et énoncer quelques concepts liés au codage de la parole.

1.1.1 Composants d'un codeur de la parole

Un codeur de la parole (appelé aussi **codec** de la parole) est toujours constitué d'un codeur et d'un décodeur. Le codeur réalise la fonction de compression tandis que le décodeur réalise celle de décompression. Ces deux fonctions coexistent dans les systèmes de

transmission / stockage . La figure 1.1 illustre un exemple d'un tel système. A l'étage de compression, le codeur prend le signal numérique original et en tire une suite de bits à débit réduit. Cette suite de bits est, alors, transmise au récepteur ou à un support de stockage. A l'étage de décompression, le décodeur va faire l'opération inverse de celle du codeur afin de reconstituer une approximation du signal original à partir de la suite de bits compressée. Donc, le décodeur doit approximativement avoir la structure inverse du codeur.

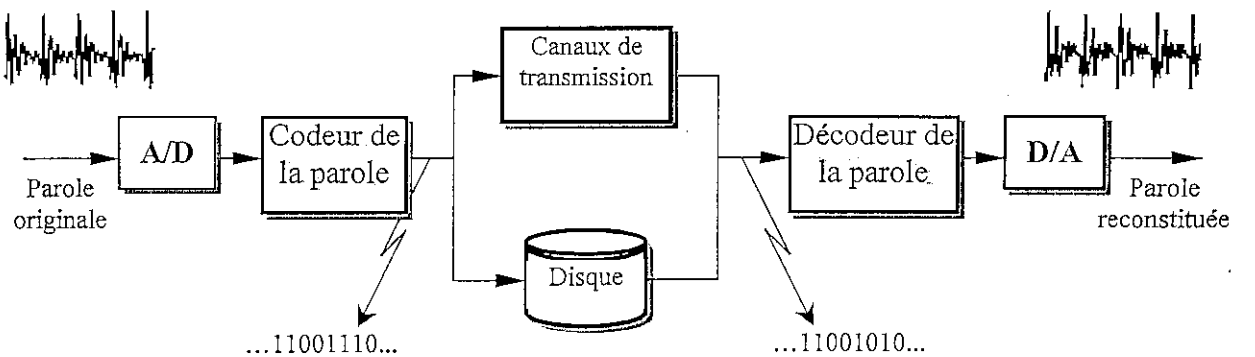

Fig. 1.1 Schéma bloc d'un système de transmission / stockage de la parole.

1.1.2 Concepts de trames et de sous - trames

La parole est un signal variant dans le temps [1]. Afin de pouvoir bien l'analyser, on divise le signal en blocs successifs de manière à ce que les échantillons se trouvant à l'intérieur de chaque bloc puissent être considérés stationnaires. Ces blocs sont appelés *trames*. En outre, quelques étapes peuvent nécessiter une plus grande résolution dans le temps et ont besoin d'être réalisées sur des blocs plus petits. Ces derniers sont appelés *sous-trames*.

1.1.3 Critères de performance

Pendant la sélection d'un codeur de la parole, certains aspects de performance doivent être pris en considération et on se trouve habituellement obligé de faire des compromis. Différentes applications exigent que les codeurs soient optimisés selon différents critères. Nous avons choisi huit critères importants dont on décrit brièvement chacun comme suit :

(1) le débit moyen : ce paramètre est généralement mesuré en bit par seconde (bps). Le mot « moyen » est utilisé ici car quelques codeurs travaillent à des débits variables par opposition aux débits fixes. Il est à noter que tous les débits mentionnés dans cette thèse n'incluent aucun débit additionnel pour la correction d'erreurs.

(2) la qualité de la parole : une méthode populaire pour évaluer la qualité de la parole est l'échelle MOS (Mean Opinion Score) qui est une évaluation subjective. Des « listeners » chevronnés donnent leurs évaluations de la qualité de la parole basées sur une échelle de cinq points –mauvaise (1), pauvre (2), moyenne (3), bonne (4) et excellente (5). A cause d'une large variation entre les listeners , le test du MOS exige un grand nombre de phrases (données du signal parole), de locuteurs et de listeners pour donner une évaluation précise d'un codeur. En Amérique du nord, un MOS de 4.5 et 5 signifie, généralement, une bonne qualité tandis que pour une parole de qualité synthétique, le MOS descend à moins de 3.5. Il existe, aussi, des tests objectifs tels que le SNR (Signal to Noise Ratio), c'est le rapport signal sur bruit. Généralement, les mesures objectives ne sont pas aussi longues et coûteuses que les subjectives mais elles ne sont pas suffisantes pour les propriétés de perception du système auditoire humain.

(3) le retard algorithmique : comme on l'a déjà mentionné, la majorité des codeurs de la parole traitent les échantillons par blocs, ainsi, un retard temporel est souvent introduit entre la parole originale et la parole reconstituée. Dans le contexte du codage de la parole, ce retard est appelé retard algorithmique et il est, généralement, défini comme étant la somme de la longueur du bloc de parole en cours de traitement et la celle du bloc futur nécessaire pour le traitement des échantillons du bloc courant. Dans des applications comme la téléphonie, il y a souvent une limitation très stricte du retard temporel. Dans d'autres comme le stockage de la voix, le retard est toléré.

(4) la complexité de calcul : les algorithmes de codage sont souvent destinés pour être implémentés sur des cartes DSP. La mémoire et la vitesse sont donc les deux éléments les plus importants qui définissent la complexité. Le premier est spécifié par la taille de la RAM utilisée pour l'exécution de l'algorithme. Le deuxième est mesuré en millions d'instructions par seconde, communément connu par MIPS. Ce dernier peut être mesuré dans un processeur à virgule fixe ou flottante. Un algorithme à complexité élevée nécessite une carte plus rapide pour une implémentation en temps réel et une consommation plus élevée d'énergie, ce qui est extrêmement désavantageux pour des systèmes portables.

(5) la sensibilité aux erreurs du canal : ce paramètre sert à mesurer la robustesse des codeurs de parole contre les erreurs du canal qui sont souvent causées par la présence de bruit du canal, d'affaiblissement du signal et d'interférences inter-symboles. Le problème des erreurs du canal devient de plus en plus important dans le codage de la parole depuis que plusieurs codeurs récents sont utilisés dans les communications sans fil. Dans de tels systèmes, le codeur doit être capable de donner une qualité raisonnable avec des pourcentages d'erreurs d'au plus 10%.

(6) la robustesse contre le bruit acoustique environnant : dans les applications réelles, on se trouve en face de plusieurs sources de bruit acoustique telles que les voitures, les rues et les bureaux. Il est, donc, essentiel que la performance de l'algorithme de codage de la parole ne souffre pas à cause de ces environnements. La question du bruit acoustique environnant devient particulièrement cruciale quand il s'agit d'applications militaires ou en communication mobile. En effet, la compétition de vocodeurs à 2.4 kbps organisée par le US DoD en 1996 exigeait que les algorithmes aient de bonnes performances dans les milieux calmes et dans les milieux bruités [2].

(7) la largeur de bande de la parole codée : c'est à dire la largeur de bande du signal parole que le codeur doit traiter. On trouve des codeurs à bande étroite dans la téléphonie qui exige une largeur de bande de 200 à 3400 Hz. D'autre part, on trouve des applications de codage à large bande allant de 7 à 20 kHz dans les transmissions audio, les téléconférences et les télé-enseignements.

(8) les propriétés acoustiques additionnelles : quelques codeurs sont capables de fournir aussi bien la compression que d'autres opérations acoustiques additionnelles. Ils peuvent, par exemple, modifier le pitch et les formants, accélérer ou ralentir la parole sans affecter l'estimation du pitch.

1.1.4 Quantification

En théorie, une représentation numérique précise d'une seule ou d'un ensemble de valeurs nécessite un nombre infini de bits, ce qui n'est pas réalisable en pratique. Par conséquent, la différence entre la valeur originale et sa version digitalisée est toujours présente lorsqu'un signal est stocké ou transmis numériquement. Le but de la quantification est de minimiser cette différence appelée, aussi, bruit de quantification ou erreur de quantification.

Il existe deux types de quantification : la quantification scalaire et la quantification vectorielle (QV). Un quantificateur scalaire ramène une valeur numérique unique à la valeur approximative la plus proche à partir d'un ensemble de valeurs possibles prédéfinis [3]. La quantification vectorielle, quant à elle, opère sur un ensemble de valeurs. Au lieu de quantifier chacune de ces valeurs indépendamment, elle traite tout l'ensemble comme étant une seule entité ou vecteur et le représente par un seul indice et, en même temps, elle minimise la distorsion introduite. De cette manière, l'efficacité du codage peut être considérablement améliorée s'il existe une information redondante dans ce bloc de valeurs .

Dans la contexte de la QV, l'ensemble de vecteurs représentants possibles est appelé « *codebook* », c'est à dire le dictionnaire des codes. Chacun des vecteurs représentant du codebook définit un « *codeword* », c'est à dire un mot code ou vecteur code. Le nombre de mots codes dans un dictionnaire définit la taille du dictionnaire et le nombre d'éléments dans chaque mot code définit la dimension du dictionnaire.

Selon le type d'application, il y a plusieurs mesures de distorsion qui peuvent être adoptées pour évaluer et/ou concevoir un quantificateur. La mesure la plus sollicitée est celle de la distance Euclidienne. Les mesures basées sur la perception humaine sont très appropriées .Ces dernières sont très avantageuses pour les codeurs de la parole, en particulier pour le codage de vecteurs de paramètres spectraux étant donné que l'oreille humaine est sensible aux variations des fréquences et des intensités. Les détails sur la sensibilité de la perception humaine seront mieux décrits au paragraphe1.4.

Vu sa grande efficacité dans le codage, la QV fait l'objet de beaucoup de recherches. Plusieurs algorithmes de quantification vectorielle ont été développés pour la recherche et la conception efficace des dictionnaires tels que la Gain - Shape VQ, la QV par division ou segmentation et la QV à étapes multiples [4] .

Récemment, la quantification vectorielle à dimension variable (VDVQ : Variable Dimension Vector Quantization) a fait l'objet d'une attention particulière. Contrairement à la QV conventionnelle, la QVDV peut traiter des vecteurs à dimension variable et chaque vecteur à l'entrée peut être quantifié à l'aide d'un seul dictionnaire universel [5].

1.2 Production et propriétés de la parole

Pour trouver le meilleur algorithme de codage de la parole, il faut d'abord avoir une bonne connaissance du système humain de production de la parole avec toutes ses propriétés et ses limites.

Du point de vue physiologique, la parole humaine est produite par la sortie de l'air à partir des poumons et à travers les cordes vocales et conduit vocal qui se termine par l'ouverture de la bouche.

Du point de vue traitement du signal, le mécanisme de production de la parole peut être modélisé par un signal d'excitation qui attaque un filtre variant dans le temps (le conduit vocal) qui sert à amplifier ou à atténuer certaines fréquences du son dans le signal d'excitation. Le conduit vocal est modélisé par un système variant dans le temps car il consiste en une combinaison de la gorge, la bouche et les lèvres qui ont un certain mouvement pendant la génération de la parole.

Les propriétés du signal d'excitation dépendent énormément du type de son émis (voisé ou non voisé). Les voyelles sont des exemples de parole voisée (/a/, /i/, /o/, /u/) tandis que les occlusives comme /p/ et /k/ sont des exemples de sons non voisés. L'excitation pour une parole voisée est un signal quasi-périodique généré par l'ouverture et la fermeture de la glotte pendant le passage de l'air venant des poumons. Cette excitation est souvent appelée excitation glottale. Le conduit vocal, étant généralement considéré de nature linéaire, ne peut altérer la périodicité de l'excitation glottale. Par conséquent, les sons voisés sont, également, de nature quasi-périodique.

Pour les sons non voisés, la glotte reste ouverte et l'excitation est formée par le passage de l'air à travers une constriction étroite à un certain point du conduit vocal créant une certaine turbulence. La parole non voisée et son signal d'excitation sont similaires à un bruit avec une énergie plus faible que celle du cas de voisement. La figure 1.2 illustrent un exemple de segments voisé et non voisé dans le domaine temporel et leurs spectres de puissances correspondants.

Dans le domaine spectral, du fait de sa quasi-périodicité, la parole voisée possède une structure en lignes harmoniques. L'espacement entre les harmoniques est appelée fréquence fondamentale. L'enveloppe du spectre, appelée aussi structure des formants, est caractérisée par un ensemble de pics appelés formants. La structure des formants (pôles et zéros de l'enveloppe) est essentiellement attribuée à la forme du conduit vocal. Ainsi, par le

mouvement de la langue, la mâchoire et les lèvres, la structure subit un changement en conséquence. Contrairement au spectre voisé, le segment non voisé contient moins d'informations spectrales utiles. Il ne contient pas d'harmoniques, il est plat et à large bande.

1.3 Perception auditive humaine

Pour atteindre le maximum de performance dans un codeur de la parole, il est essentiel de bien connaître les caractéristiques du système auditif humain. L'exploitation des propriétés de la perception de l'oreille humaine peut amener à une importante amélioration des performances d'un codeur de la parole. Ceci est particulièrement vrai lorsqu'on essaie de réaliser des codeurs à des débits très réduits tout en évitant une dégradation audible majeure. Une des propriétés les plus importantes est le masquage auditif qui a un grand effet sur la perception d'un signal en présence d'un autre [6].

Le bruit a moins de chance d'être perçu aux fréquences de la parole ayant une énergie importante (les formants) et a plus de chances d'être perçu aux fréquences de la parole de faible énergie (les vallées par exemple). Le masquage spectral est une technique populaire qui exploite cette limitation de la perception par la concentration du maximum de bruit (résultant de la compression) dans les régions de hautes énergies du spectre où il sera moins audible.

Les sons voisés et non voisés sont perçus différemment par l'oreille humaine. Pour les segments voisés, le degré de périodicité et la continuité temporelle [7,8,9] sont d'une grande importance (bien qu'une périodicité excessive peut engendrer le phénomène de réverbération). Dans le domaine spectral, les amplitudes et les positions des trois premiers formants (généralement en dessous de 3 kHz) et l'espacement entre les harmoniques sont importants [10].

Pour la parole non voisée, il a été montré dans [11] que les segments non voisées peuvent être remplacés par un signal bruit ayant une enveloppe spectrale similaire sans altérer la qualité de perception du signal parole. Dans les deux cas de voisement et de non voisement, l'enveloppe temporelle contribue à l'intelligibilité et au naturel [12,13] de la parole.

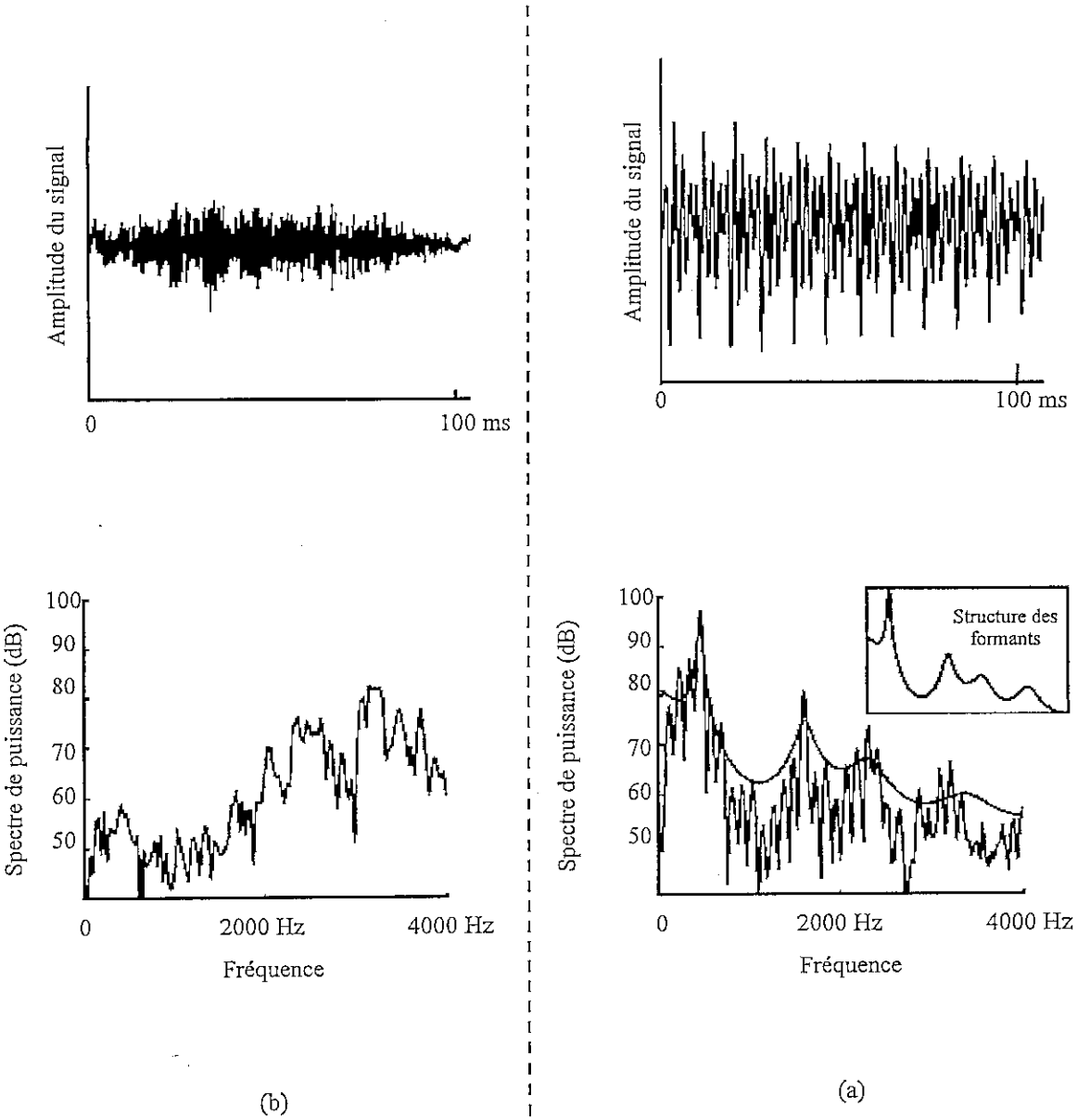

Fig. 1.2 (a)Tranche de parole voisée (en haut) et son spectre de puissance.
(b)Tranche de parole non voisée (en haut) et son spectre de puissance.

1.4 Standardisation des codeurs de la parole

La standardisation des codeurs de la parole de haute qualité à bande étroite (c. à. d. la bande de la téléphonie : de 200 Hz à 3400 Hz échantillonnée à 8 kHz et représentée par le format PCM uniforme 16 bits) et à débits réduits s'est intensifiée durant la dernière décennie.

En 1994, l'Union Internationale des Télécommunications (UIT) a adopté l'algorithme LD-CELP (Low Delay Code Excited Linear Predictive) [14] pour le codage de haute qualité (Toll quality) de la parole à 16 kbps qu'elle a appelé G.728. Peu de temps après, un autre codeur de la parole basé sur la technique CELP travaillant à un débit de 8 kbps a été développé par l'Université de Sherbrooke [15]. C'était un codeur toll-quality (la qualité grand public) dont les performances sont comparables à celles du LD-CELP à 16 kbps. En 1996, ce codeur fit son entrée parmi les standards de l'UIT avec l'appellation G.729. Durant la même année, le Département de la Défense US standardisait un vocodeur à 2.4 kbps avec la qualité de communications pour remplacer les FS1015 et FS1016. Parmi les sept candidats à cette standardisation, le vainqueur était le vocodeur MELP (Mixed-Excitation Linear Predictive) développé par Texas Instruments [16].

Actuellement, l'UIT est à la recherche d'un standard de codage de la parole travaillant à un débit autour de 4 kbps et produisant une parole de qualité équivalente au standard existant (G.729 à 8 kbps). Ce standard sera destiné à plusieurs applications comme la téléphonie visuelle, les applications multimédia et la téléphonie IP. Des efforts sont en cours à travers le monde pour la préparation de ce standard.

1.5 Objectifs de ce travail

L'un des candidats prometteurs de cette standardisation à 4 kbps de l'UIT est le codeur par *interpolation de la forme d'onde* (WI :Waveform Interpolation). Ce codeur a été développé, pour la première fois, par AT&T à la fin des années 80 [17] et depuis, plusieurs améliorations lui ont été ajoutées [18, 19, 20, 21, 22]. Le but principal de ce travail est de montrer l'efficacité de cet algorithme en proposant un schéma de quantification qui permet d'atteindre un débit de 4.25 kbps. Cet objectif est atteint grâce à la simulation d'un codeur WI (sans quantification) en langage C. Cette présente thèse peut servir de référence pour ceux qui souhaitent implémenter un codeur WI. Pour chaque composant du codeur WI, les descriptions fonctionnelles et les calculs mathématiques correspondants sont fournis.

Chapitre 2

CODAGE DE LA PAROLE

PAR PREDICTION LINEAIRE

Dans ce chapitre, on se penche sur l'analyse du codage par prédiction linéaire (LPC) qui constitue un composant indispensable dans la plupart des algorithmes de codage. Plus précisément, nous allons examiner l'analyse LPC à court-terme dont l'objectif est d'enlever la corrélation (redondance) dans un signal de parole par l'emploi d'un filtre de prédiction linéaire (LP) variant dans le temps. Les coefficients du filtre sont appelés coefficients LP et sa sortie est le signal d'excitation ou signal résiduel. Ces coefficients LP caractérisent l'enveloppe spectrale du signal parole contrôlée par le conduit vocal humain tandis que le signal résiduel décrit l'excitation du conduit vocal.

Un avantage clé de l'analyse LP est que la parole est décomposée en deux composants hautement indépendants l'un de l'autre, les paramètres du conduit vocal (les coefficients LP) et l'excitation glottale (l'excitation LP). Ces deux paramètres sont très différents du point de vue quantification. Par conséquent, chacun aura son propre schéma d'analyse et de

quantification, ce qui améliore l'efficacité du codage. La dernière décennie a vu le développement de schémas de quantification très efficaces pour les coefficients LP [23], cependant, la représentation du signal d'excitation reste encore problématique. Plusieurs techniques prometteuses ont été proposées ces dernières années afin de remédier à ce problème, l'une d'elles est la technique WI.

On va procéder comme suit. D'abord, on expose les bases fondamentales de l'analyse LPC court-terme et on discute la manière de calculer les coefficients LP. Puis, on introduit une représentation populaire des coefficients LP (les paires de raies spectrales) qui offrent de meilleures propriétés de quantification et d'interpolation. A la fin, on discute les concepts d'extension de la largeur de bande et de préaccentuation.

2.1 La prédiction linéaire dans le codage de la parole

Comme on l'a déjà vu au paragraphe 1.3, la production de la parole est le résultat du passage de l'excitation glottale à travers le conduit vocal. Dans le codage par prédiction linéaire, ce processus peut être modélisé par l'excitation d'un filtre linéaire variant dans le temps par un signal résiduel, comme le montre la figure 2.1. Le filtre est un tout - pôle d'ordre

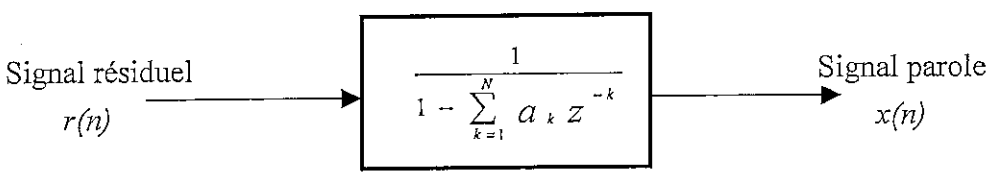

Fig. 2.1 Filtre de synthèse LP

N. Puisque ce filtre synthétise la parole, il est appelé *filtre de synthèse LP* et ses coefficients a_1 , a_2 , ... , a_N sont dits coefficients LP.

Le filtre de synthèse modélise l'effet du conduit vocal imposé à l'excitation glottale. Donc, sa réponse fréquentielle correspond à l'enveloppe spectrale (corrélation court - terme) du signal parole de l'entrée. En d'autres termes, les fréquences de résonance du filtre doivent correspondre aux positions des formants du signal parole comme indiqué dans la figure 1.2a. Par conséquent, l'ordre N du filtre doit être choisi de manière à ce qu'il y ait deux pôles associés à chaque formant. Pour un signal parole échantillonné à 8 kHz, il est souvent suffisant de mettre $N=10$.

Le filtre inverse du filtre de synthèse est appelé filtre d'analyse LP. Sa fonction principale est d'extraire le signal résiduel du signal parole comme le montre la figure 2.2.

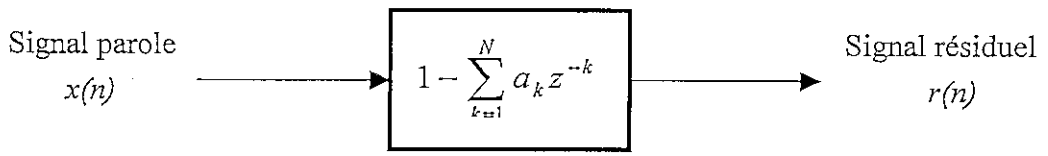

Fig. 2.2 Filtre d'analyse LP

Les équations aux différences qui relient $x(n)$ et $r(n)$ sont :

$$\left.\begin{aligned} r(n) &= x(n) - \sum_{k=1}^{N} a(n)x(n-k) \\ x(n) &= r(n) + \sum_{k=1}^{N} a(n)x(n-k) \end{aligned}\right\} \qquad (2.1)$$

Puisque la forme du conduit vocal change dans le temps, les filtres d'analyse et de synthèse sont variants dans le temps et, par conséquent, les coefficients $\{a_k\}$ changent dans le temps. Néanmoins, dans un codeur pratique, ces coefficients sont calculés une fois par trame. Le prochain paragraphe traite les procédures d'estimation des $\{a_k\}$.

2.2 Estimation des coefficients LP

Il y a deux approches pour l'estimation des coefficients $\{a_k\}$ qui sont la méthode d'auto corrélation et la méthode de covariance. Les deux méthodes utilisent le principe classique des moindres carrés et donnent l'ensemble $\{a_k\}$ qui minimisent l'énergie du signal résiduel résultant.

2.2.1 Méthode d'auto corrélation

Le signal parole est, d'abord, multiplié par une fenêtre d'analyse $w(n)$ de longueur finie L_w pour obtenir une fenêtre de parole $x_w(n)$:

$$x_w(n) = w(n)x(n) \tag{2.2}$$

La fenêtre $w(n)$ la plus utilisée pour minimiser l'énergie aux bords est celle de Hamming décrite par :

$$w(n) = \begin{cases} 0.54 - 0.46\cos\left(\frac{2\pi n}{L_w - 1}\right), & \text{pour } 0 \le n < L_w \\ 0, & \text{ailleurs.} \end{cases} \tag{2.3}$$

Puis, on trouve l'expression de l'énergie de l'erreur de prédiction E.
A partir de (2.1), on peut avoir

$$E = \sum_{n=-\infty}^{\infty} r^2(n) = \sum_{n=-\infty}^{\infty} \left[x_w(n) - \sum_{k=1}^{N} a_k x_w(n-k) \right]^2 \tag{2.4}$$

L'ensemble de coefficients $\{a_k\}$ qui minimise E est calculé en posant

$$\frac{\partial E}{\partial a_k} = 0 \qquad \text{pour } k=1, 2, \ldots, N \tag{2.5}$$

ce qui mène à un système linéaire de N équations

$$\sum_{-\infty}^{\infty} x_w(n)x_w(n-i) = \sum_{k=1}^{N} a_k \sum_{n=-\infty}^{\infty} x_w(n-i)x_w(n-k) \quad \text{pour } i=1, 2, \ldots, N \tag{2.6}$$

En définissant la fonction d'auto corrélation du signal fenêtré $x_w(n)$ par

$$R(i) = \sum_{n=-\infty}^{\infty} x(n)x(n-i) = \sum_{n=1}^{L_w-1} x_w(n)x_w(n-i) \tag{2.7}$$

et sachant que la fonction d'auto corrélation est une fonction paire où $R(n) = R(-n)$, le système d'équations (2.6) peut être mis sous la forme matricielle :

$$\begin{bmatrix} R(0) & R(1) & \dots R(N-1) \\ R(1) & R(0) & \dots R(N-2) \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ R(N-1) R(N-2) \dots & & R(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \cdot \\ \cdot \\ \cdot \\ a_N \end{bmatrix} = \begin{bmatrix} R(1) \\ R(2) \\ \cdot \\ \cdot \\ R(N) \end{bmatrix} \tag{2.8}$$

Puisque la matrice de (2.8) a une structure de Toeplitz, les coefficients $\{a_k\}$ peuvent être calculés de manière plus efficace et plus rapide en utilisant la récursion de Levinson-Durbin [24]. De plus, la structure de Toeplitz garantie que les pôles du filtre de synthèse résultant seront à l'intérieur du cercle unité et, par voie de conséquence, la stabilité du filtre est toujours satisfaite.

2.2.2 Méthode de covariance

C'est une autre méthode de calcul des coefficients $\{a_k\}$. Bien que les deux approches soient similaires, elles diffèrent par l'emplacement de la fenêtre d'analyse. La méthode de covariance applique le fenêtrage au signal d'erreur et non pas au signal parole. Dans ce cas, l'énergie de l'erreur de prédiction devient

$$E = \sum_{n=-\infty}^{\infty} r^2(n)\, w(n) \tag{2.9}$$

La résolution de (2.9) de la même manière que la méthode d'auto corrélation permet d'obtenir un système de N équations linéaires qu'on peut mettre sous la forme matricielle :

$$\begin{bmatrix} \varphi(1,1) \varphi(1,2) \dots \varphi(1,N) \\ \varphi(2,1) \varphi(2,2) \dots \varphi(2,N) \\ \cdot \\ \cdot \\ \cdot \\ \varphi(N,1) \varphi(N,2) \dots \varphi(N,N) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \cdot \\ \cdot \\ \cdot \\ a_N \end{bmatrix} = \begin{bmatrix} \varphi(0,1) \\ \varphi(0,2) \\ \cdot \\ \cdot \\ \varphi(0,N) \end{bmatrix} \tag{2.10}$$

où $\varphi(i, j)$ est la fonction de covariance pour $x(n)$ définie par

$$\varphi(i, j) = \sum_{n=-\infty}^{\infty} x(n-i)\, x(n-j)\, w(n) \tag{2.11}$$

La matrice de covariance n'a pas la structure de Toeplitz, mais elle possède la propriété importante d'être symétrique définie positive, ce qui implique que les $\{a_k\}$ peuvent être calculés de manière efficace par la décomposition de Cholesky [24].

La méthode de covariance n'applique pas le fenêtrage au signal parole d'entrée, ce qui la rend très avantageuse pour les applications d'estimation spectrale à haute résolution. Cependant, elle ne garantie pas la stabilité du filtre tout - pôle de synthèse LP ; les pôles des coefficients estimés peuvent se retrouver à l'extérieur du cercle unité. La méthode de covariance impose, donc, l'utilisation d'un algorithme de stabilisation pour ramener les pôles à l'intérieur du cercle unité, ce qui augmente la complexité du codeur. Pour cette raison, cette méthode ne sera pas adoptée dans notre implémentation.

2.3 Interpolation des coefficients LP

Les coefficients de prédiction $\{a_k\}$ sont estimés à la fréquence des trames. Afin d'éviter les changements rapides des coefficients entre deux trames successives, les coefficients sont interpolés à la fréquence des sous – trames pour assurer que leur évolution à travers les trames soit lente. Autrement, une grande variation des coefficients entre trames peut engendrer des transitions indésirables, de la rugosité et même des discontinuités audibles dans le signal parole reconstitué.

Comme il est bien connu, l'interpolation directe des coefficients $\{a_k\}$ peut causer l'instabilité du filtre d'analyse. Par conséquent, ces coefficients sont, généralement, transformés dans un autre domaine, interpolés puis reconvertis au domaine $\{a_k\}$. Le domaine le plus utilisé à cette fin est celui des LSF (Line Spectrum Frequency) appelé aussi LSP (Line Spectrum Pair). Il fournit, non seulement la stabilité des coefficients LP interpolés, mais aussi de meilleures propriétés de quantification.

La conversion des coefficients LP du domaine $\{a_k\}$ au domaine des LSF peut être effectuée comme suit [26]. Posons d'abord

$$A(z) = 1 - \sum_{k=1}^{N} a_k z^{-k} \tag{2.12}$$

Notons que les zéros de $A(z)$ sont les pôles du filtre de synthèse LP et les zéros du filtre d'analyse LP. Ces zéros sont projetés sur le cercle unité à travers les deux transformations en z : $P(z)$ et $Q(z)$ d'ordre $(N+1)$:

$$P(z) = A(z) + z^{-(N+1)}A(z^{-1})$$
$$Q(z) = A(z) - z^{-(N+1)}A(z^{-1}) \qquad (2.13)$$

Les zéros de $P(z)$ et $Q(z)$, situés sur le cercle unité, sont entrelacés. Les coefficients LSF sont les positions angulaires $\{\omega_i\}$ de ces zéros entre 0 et π. Plus précisément, les LSF peuvent être écrits dans l'ordre croissant suivant:

$$0 = \omega_0 < \omega_1 < \ldots < \omega_N < \omega_{N+1} = \pi \qquad (2.14)$$

Les ω_0 et ω_{N+1} sont toujours égaux à 0 et π respectivement et n'ont pas besoin d'être codés. En outre, l'ordre croissant des LSF comme indiqué dans (2.14) assure la stabilité du filtre de synthèse. Ce type de contrôle très simple de la stabilité n'existe pas pour les coefficients $\{a_k\}$.

Une autre propriété très importante des coefficients LSF est la sensibilité spectrale localisée. Pour les coefficients $\{a_k\}$, une petite erreur dans un des coefficients peut dramatiquement altérer l'enveloppe spectrale et même déstabiliser le filtre de synthèse. Alors que si un des coefficients LSF est déformé, l'altération spectrale n'aura effet que dans le voisinage de ce LSF.

Les zéros des polynômes (2.13) peuvent être calculés par la méthode décrite dans [27] où les polynômes de Chebyshev sont utilisés pour trouver les racines dans le domaine des cosinus.

2.4 Extension de la largeur de bande

Occasionnellement, l'analyse LP génère un filtre de synthèse dont la structure des formants possède des pics très pointus. Cela implique que les pôles du filtre sont très proches du cercle unité et, donc, le filtre se trouve marginalement stable. Une telle stabilité marginale dans les filtres LP peut multiplier les chances d'avoir un croisement dans la quantification des LSF qui peut, en retour, causer des pépiements (chirps) dans la parole reconstituée. Une solution à ce problème est l'utilisation d'une extension de la largeur de bande pour dilater la largeur de bande dans la réponse fréquentielle du filtre.

Dans cette procédure, chaque coefficient a_k est remplacé par $\gamma^k a_k$, où $k = 1, 2, \ldots, N$. Une telle multiplication a pour effet de déplacer les pôles du cercle unité vers l'origine par un facteur γ. Cela donne comme résultat des pics moins pointus et un élargissement de la bande dans la réponse fréquentielle du filtre d'analyse et, par conséquent, le filtre devient plus stable. Cela réduit, par la même, les croisements dans la quantification des LSF étroitement espacés.

Ce γ, appelé *facteur d'extension de la largeur de bande*, contrôle le déplacement des pôles vers l'intérieur. Les valeurs typiques de γ sont entre 0.996 et 0.988, ce qui correspond à une extension de la largeur de bande entre 10 et 30 Hz respectivement

2.5 Pré accentuation

Dans la procédure de conversion A/D conventionnelle, un signal parole analogique est filtré passe-bas avant d'être échantillonné. Une telle opération empêche le recouvrement spectral dans la parole numérisée mais, en même temps, réduit l'énergie des composantes de haute fréquence. Ceci est indésirable dans l'analyse LP car une énergie relativement faible dans les hautes fréquences peut engendrer une matrice d'auto corrélation (2.8) mal conditionnée et, par voie de conséquence, affecte la précision numérique des coefficients LP [28]. Pour remédier à ce problème, l'énergie des composantes haute fréquence de la parole est amplifiée avant le calcul des coefficients LP. Cette opération est effectuée en faisant passer le signal parole $x(n)$ à travers le filtre

$$H(z) = 1 - \alpha z^{-1} \tag{2.15}$$

où α détermine la fréquence de coupure du filtre tout - zéro d'ordre 1. De cette manière, l'énergie relative du spectre de haute fréquence peut être augmentée. Cette procédure est appelée *préaccentuation* et α, appelé *facteur de préaccentuation,* sert à contrôler le degré de préaccentuation. La valeur typique de α est autour de 0.1 [6]. Pour éliminer l'effet de la préaccentuation, un filtre de *désaccentuation* (l'inverse de $H(z)$) est utilisé au décodeur.

Chapitre 3

INTERPOLATION

DE LA FORME D'ONDE

3.1 Origines et principes du codage WI

L'importance de la perception de la périodicité dans la parole voisée est à l'origine du développement de la technique de codage par interpolation de la forme d'onde. Cette technique a été introduite, en premier lieu, par W. B. Kleijn [7] et la première version était appelée *Prototype Waveform Interpolation* (PWI). La PWI codait les segments voisés seulement et, par conséquent, elle était utilisée en combinaison avec d'autres codeurs tels que le CELP pour coder les segments non voisés.

La PWI exploite le fait que les formes d'ondes de longueur égale à la période du pitch (période fondamentale) évoluent lentement dans la temps. Cette évolution lente des formes d'ondes suggère qu'on n'a pas besoin de transmettre toutes les périodes de la trame au décodeur ; au lieu de cela, on peut les transmettre à des intervalles réguliers. Au décodeur, les formes d'ondes non transmises sont retrouvées au moyen d'une interpolation. De cette manière, le degré de périodicité de la parole voisée sera mieux contrôlé et, par conséquent, on obtient une parole voisée reconstituée de haute qualité [9]. Dans la PWI, les périodes du signal sélectionnées pour être transmises sont dites formes d'ondes prototypes (*Prototype Waveforms*).

Bien que la PWI travaille remarquablement bien avec les segments voisés, elle a le défaut de ne pas pouvoir être appliquée aux segments non voisés. En d'autres termes, elle doit toujours être utilisée avec une autre méthode de codage de la parole pour manipuler les segments non voisés. Ainsi, la commutation entre les codeurs devient inévitable et réduit considérablement la robustesse du codeur. En 1994, la PWI a été raffinée pour devenir la WI qui est capable de prendre en charge les sons voisés et non voisés [18, 29]. Similaire à la PWI, la WI représente un signal parole avec une séquence de formes d'ondes. Pour la parole voisée, ces formes d'ondes sont simplement de longueurs égales à la période du pitch (pitch cycles). Pour la parole non voisée et le bruit de fond, les formes d'ondes sont de différentes longueurs et contiennent des signaux assimilables à du bruit. Puisque les formes d'ondes ne sont plus limitées à la période du pitch, il n'est plus approprié d'utiliser le terme forme d'onde *prototype* ou pitch-cycle. A la place, on adopte le terme forme d'onde *caractéristique* (*Characteristic Waveform*) qui sera abrégé par CW par la suite.

Une différence clé entre la WI et la PWI est que les formes d'ondes dans la WI sont prélevées à une fréquence plus grande. Cependant, une augmentation de la fréquence de prélèvement des formes d'ondes entraînera une augmentation du débit. Pour contrer ce problème, la WI décompose la CW en une forme d'onde à évolution lente (SEW) et une forme d'onde à évolution rapide (REW). La SEW représente la composante quasi-périodique du signal parole tandis que la REW représente la composante non périodique et le bruit restants dans le signal. Puisque les deux formes d'ondes ont des propriétés différentes du point de vue perception, elles sont quantifiées séparément pour améliorer l'efficacité du codage.

3.2 Vue d'ensemble du codeur WI

La figure 3.1 présente une schéma bloc du codeur WI. On peut le diviser en deux couches : la couche d'analyse-synthèse et la couche de quantification. Dans la première couche, le bloc d'analyse (processeur **100**) exécute, d'abord, l'analyse LPC sur le signal parole entrant et fournit le signal résiduel. Puis, le pitch est estimé et le signal résiduel est, alors, décomposé en une suite de CW. Ces CW sont, alors, alignées et normalisées en puissance pour donner une surface (signal à deux dimension) qui illustre l'évolution des formes d'ondes à travers la trame. L'étage de synthèse (processeur **200**) effectue l'opération

inverse de celle de l'analyse. Le signal résiduel est reconstruit à partir des CW et envoyé au filtre de synthèse LP où le signal parole est, finalement, reconstitué.

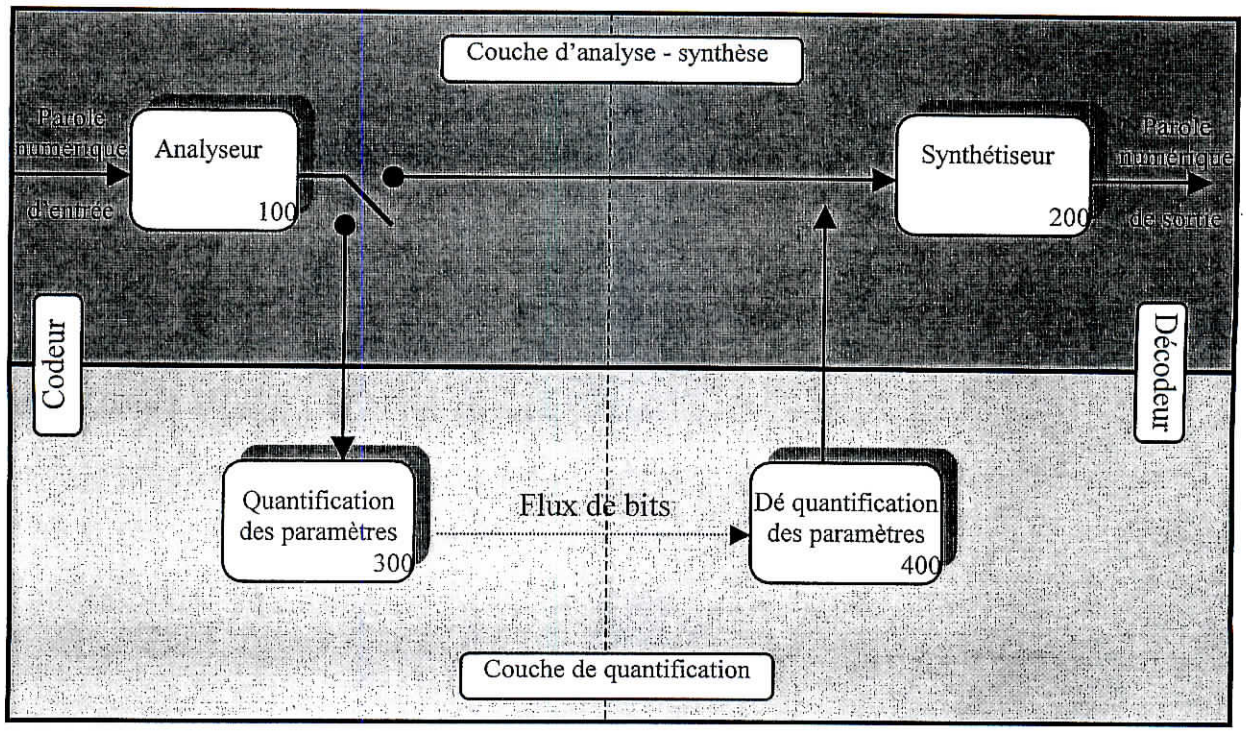

Fig. 3.1 Schéma bloc d'un système de codage WI. Le commutateur permet au codeur d'éviter la couche de quantification et nous permet de mesurer la performance de la couche d'analyse – synthèse.

Le processeur **300** dans la couche quantification exécute la décomposition en SEW/REW et la quantification des paramètres. Le processeur **400** au récepteur dé-quantifie et reconstitue les CW à partir des SEW et REW transmises.

Dans ce chapitre, on va discuter la couche d'analyse – synthèse qui comprend les éléments clés de la technique WI comme l'extraction du pitch, l'extraction des CW, leur alignement et leur interpolation. Cette étude est basée largement sur le travail de W. B. Kleijn sur la WI.

Pour chaque processeur dans la couche, on donnera les détails d'implémentation avec les calculs mathématiques appropriés. Pour faciliter la discussion, on donnera les schémas détaillés des processeurs sélectionnés. Les processeurs **300** et **400** de la couche quantification seront examinés dans le chapitre suivant.

3.3 Représentation des formes d'ondes caractéristiques

Avant de rentrer dans les détails de chaque processeur, on commence, d'abord, par choisir une représentation mathématique appropriée pour les CW. Comme on va le voir au fur et à mesure, la majorité des calculs dans la WI sont associés aux CW, il est donc crucial d'avoir la meilleure représentation des CW qui permet de réduire la complexité du codeur.

Les CW sont, finalement, utilisées pour construire une surface bidimensionnelle décrivant l'évolution des formes d'ondes du signal résiduel. Ainsi, la représentation des CW recherchée doit permettre d'avoir un signal bidimensionnel.

Pour commencer, on considère une seule CW unidimensionnelle. La CW est une séquence de valeurs réelles à temps discret de longueur égale à la période du pitch. Donnons la notation $s(m)$ à la CW de longueur P (*Pitch period*) :

$$s(m) \in \mathbf{R} \qquad m = 0, 1, \ldots, P-1 \qquad (3.1)$$

Une partie du traitement dans la WI est faite dans la domaine fréquentiel. Ceci implique qu'une représentation temps – fréquence serait très favorable. Nous avons, donc, choisi le représentation en série de Fourier à temps discret (DTFS : Discrete Time Fourier Series) où $s(m)$ peut être exprimée par :

$$s(m) = \sum_{k=0}^{\lfloor P/2 \rfloor} \left[A_k \cos\left(\frac{2\pi km}{P}\right) + B_k \sin\left(\frac{2\pi km}{P}\right) \right] \qquad 0 \le m < P \quad (3.2)$$

où $\{A_k\}$et $\{B_k\}$sont les coefficients de Fourier à temps discret (DTFS) calculés à l'aide d'un ensemble d'équations de transformation. Plus précisément, si P est pair :

$$\left.\begin{aligned} A_k &= \frac{2}{P}\sum_{m=0}^{P-1}\left[s(m)\cos\left(\frac{2\pi km}{P}\right)\right] \\ B_k &= \frac{2}{P}\sum_{m=0}^{P-1}\left[s(m)\sin\left(\frac{2\pi km}{P}\right)\right] \end{aligned}\right\} \quad \text{pour } k = 1, 2, \ldots, P/2 - 1$$

$$\left.\begin{aligned} A_k &= \frac{1}{P}\sum_{m=0}^{P-1}\left[s(m)\cos\left(\frac{2\pi km}{P}\right)\right] \\ B_k &= \frac{1}{P}\sum_{m=0}^{P-1}\left[s(m)\sin\left(\frac{2\pi km}{P}\right)\right] \end{aligned}\right\} \quad \text{pour } k = 0 \text{ et } P/2 \qquad (3.3)$$

Quand P est impair :

$$\left.\begin{aligned} A_k &= \frac{2}{P}\sum_{m=0}^{P-1}\left[s(m)\cos\left(\frac{2\pi km}{P}\right)\right] \\ B_k &= \frac{2}{P}\sum_{m=0}^{P-1}\left[s(m)\sin\left(\frac{2\pi km}{P}\right)\right] \end{aligned}\right\} \quad \text{pour } k = 1, 2, \ldots, (P-1)/2$$

$$\left.\begin{aligned} A_k &= \frac{1}{P}\sum_{m=0}^{P-1}\left[s(m)\cos\left(\frac{2\pi km}{P}\right)\right] \\ B_k &= \frac{1}{P}\sum_{m=0}^{P-1}\left[s(m)\sin\left(\frac{2\pi km}{P}\right)\right] \end{aligned}\right\} \quad \text{pour } k = 0 \qquad (3.4)$$

La forme d'une CW peut, maintenant, être décrite par un ensemble de coefficients DTFS $\{A_k, B_k\}$. Notons que l'indice m dans (3.2) n'est pas nécessairement entier ; il peut prendre n'importe quelle valeur réelle dans l'intervalle $0 \le m < P$. En d'autres termes, les valeurs situées entre deux instants discrets ($s(1.3)$, par exemple) peuvent être calculées aisément par (3.2).

Après avoir obtenu la représentation pour une CW, nous somme, maintenant, prêts à construire une représentation bidimensionnelle pour une séquence de CW. En fait, cette représentation est simplement obtenue en ajoutant une modification à (3.2). Ainsi, on attache un indice de temps discret n à tous les paramètres dans (3.2) qui varient dans le temps. Ces paramètres sont $\{A_k\}$, $\{B_k\}$ et P. L'équation (3.2) peut donc être écrite comme suit :

$$s(n, m) = \sum_{k=1}^{\lfloor P(n)/2 \rfloor}\left[A_k(n)\cos\left(\frac{2\pi km}{P(n)}\right) + B_k(n)\sin\left(\frac{2\pi km}{P(n)}\right)\right] \qquad 0 \le m < P(n) \quad (3.5)$$

où les coefficients $\{A_k(n)\}$ et $\{B_k(n)\}$ sont, maintenant, variants dans le temps de même que la valeur du pitch $P(n)$. Il faut noter que nous avons ignoré les coefficients A_0 et B_0 dans l'équation (l'indice k commence à partir de $k = 1$ au lieu de $k = 0$). Ceci est du au fait que B_0 dans (3.3) et (3.4) est un coefficient redondant ($\sin(0) = 0$). D'un autre côté, A_0 représente la composante DC du signal et n'a aucune importance vis à vis de la perception. Par conséquent, ces deux coefficients peuvent être ignorés.

L'équation 3.5 est, à présent, la représentation d'un signal bidimensionnel où m et n sont les variables courantes. Chaque CW évolue le long de l'axe m et la forme des CW évolue à travers le temps le long de l'axe n.

Cependant, la longueur de la CW dans (3.5) dépend du pitch $P(n)$ variant dans le temps ; les CW à des instants différents peuvent avoir des longueurs différentes. Il est, généralement, plus convenable de normaliser toutes les CW à une longueur commune. Cette normalisation peut être accomplie en substituant

$$\phi = \phi(m) = \frac{2\pi m}{P(n)} \tag{3.6}$$

dans (3.5) et on peut obtenir

$$s(n, \phi) = \sum_{k=1}^{\lfloor P(n)/2 \rfloor} [A_k(n)\cos(k\phi) + B_k(n)\sin(k\phi)] \qquad 0 \le \phi(.) < 2\pi \tag{3.7}$$

De cette manière, toutes les CW ont la même longueur 2π. La figure 3.2 donne une illustration de cette normalisation et un exemple d'une surface bidimensionnelle représentée par (3.7).

Remarques sur la représentation en DTFS

- A première vue, $B_{P/2}$ dans (3.3) semble être un coefficient redondant puisque $\sin(m\pi)=0$ pour tout entier m. En fait, ce n'est pas entièrement vrai. Comme nous le verrons au paragraphe 3.4.5, ce coefficient particulier ne sera plus égal à zéro quand le signal subit un décalage dans le temps dans le processeur d'alignement.
- Généralement, représenter un signal par ses coefficients DTFS implique que le signal est répété de façon périodique. De même, représenter une CW par les DTFS signifie qu'elle est extraite d'un signal périodique.
- Les représentations dans le domaine temps peuvent réduire la complexité du codeur dans une certaine mesure en évitant les transformations en DTFS directe et inverse. Néanmoins, elles peuvent être problématiques dans les traitements liés à la fréquence [25].

3.4 Etage d'analyse

Pour commencer, on va se concentrer sur le processeur d'analyse **100**. Comme il est déjà mentionné, le but fondamental de ce processeur est de décomposer le signal parole en une série de CW (une surface bidimensionnelle) et d'extraire d'autres paramètres orthogonaux tels que les LSF, l'énergie et le pitch. La figure 3.3 montre tous les processeurs que comprend la couche d'analyse.

Cette thèse suppose que la parole à l'entrée et celle reconstituée sont dans un format numérique (16 bits par échantillon) avec une fréquence d'échantillonnage de 8 kHz. La taille de la trame L_f est de 160 échantillons (20 ms) et la longueur de la sous – trame L_{sf} est de 20 échantillons (ce choix sera justifié plus loin dans l'étude de l'extraction des CW).

3.4.1 Analyse LP

Chaque trame de parole entrante est, tout d'abord, envoyée au processeur **130** où elle subit une analyse LP d'ordre 10 pour en extraire l'ensemble $\{a_k\}$ des coefficients LP. Avant cela, le signal parole subit une pré-accentuation en utilisant $\alpha = 0.1$ dans (2.15). Cette opération a pour but de compenser la perte de l'énergie des composantes haute fréquence due au filtrage passe – bas pendant la conversion A/D. La parole pré-accentuée est, alors, fenêtrée en utilisant la fenêtre de Hamming définie dans (2.3) avec $L_w = 240$. Le centre de le fenêtre coïncide avec l'extrémité droite de la trame courante. En d'autres termes, la fenêtre couvre 120 échantillons de la trame courante et 120 de la trame future. Ces 120 échantillons futurs provoquent un retard algorithmique de 15 ms. La méthode d'auto-corrélation est appliquée à cette fenêtre de parole pour générer les coefficients du filtre $\{a_k\}$. Ces $\{a_k\}$ sont modérés en utilisant $\gamma = 0.98829$ ce qui est équivalent à une extension de la largeur de bande égale à 30 Hz. Les coefficients résultants sont convertis en coefficients LSF et envoyés au processeur **120**. Toutes les opérations précédentes sont effectuées une fois par trame, d'où la fréquence de mise à jour des coefficients LP vaut 50 Hz dans notre implémentation.

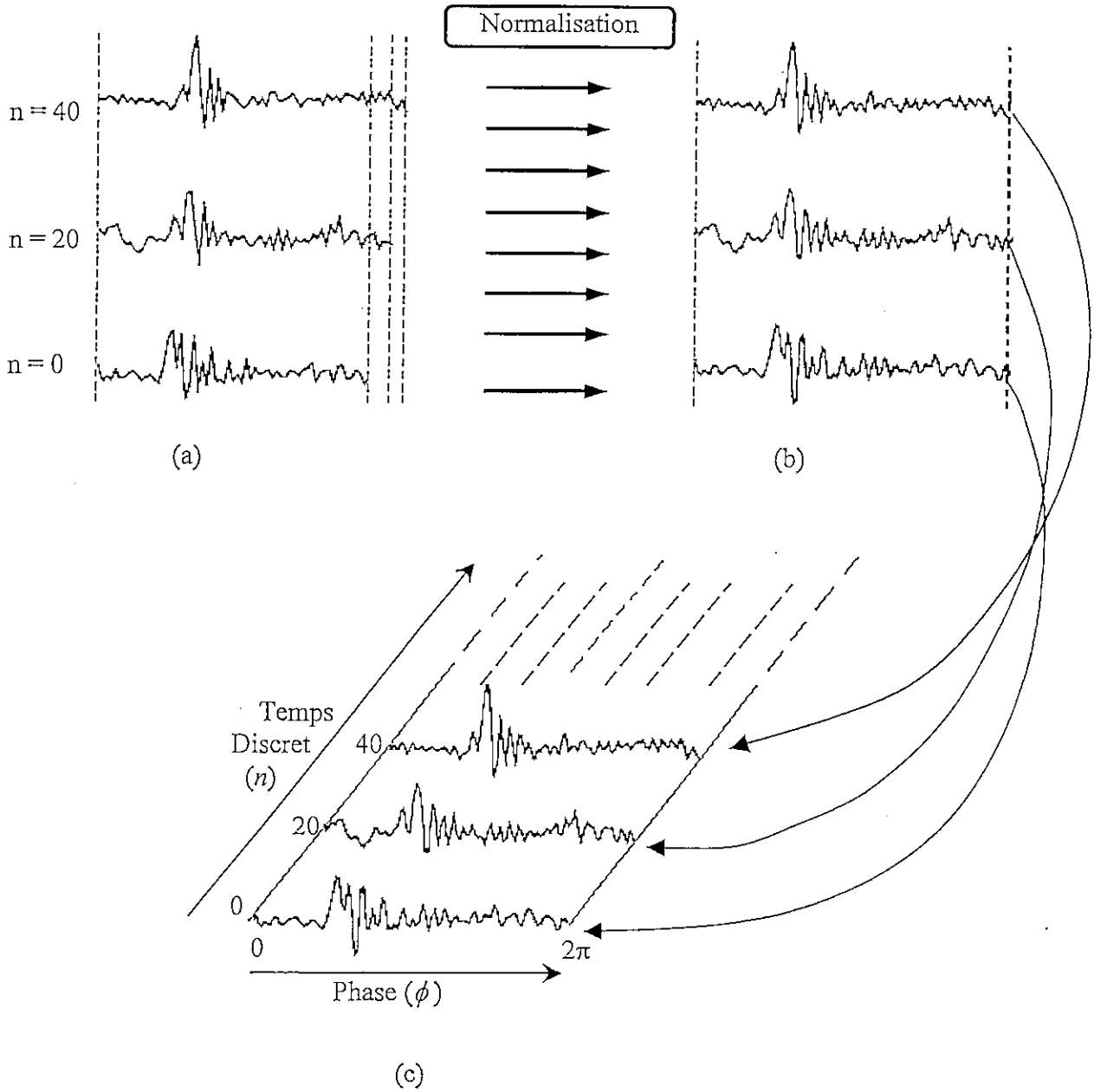

Fig. 3.2 Exemple d'une surface de formes d'ondes caractéristiques. (a) Les CW (pré alignées) sont prélevées aux instants n = 1, 9, 17. On remarque qu'elles ont des longueurs différentes. (b) Les CW après normalisation. (c) Formation de la surface d'évolution des CW. Chaque CW évolue le long de l'axe de ϕ et l'évolution des CW dans le temps se fait sur l'axe n.

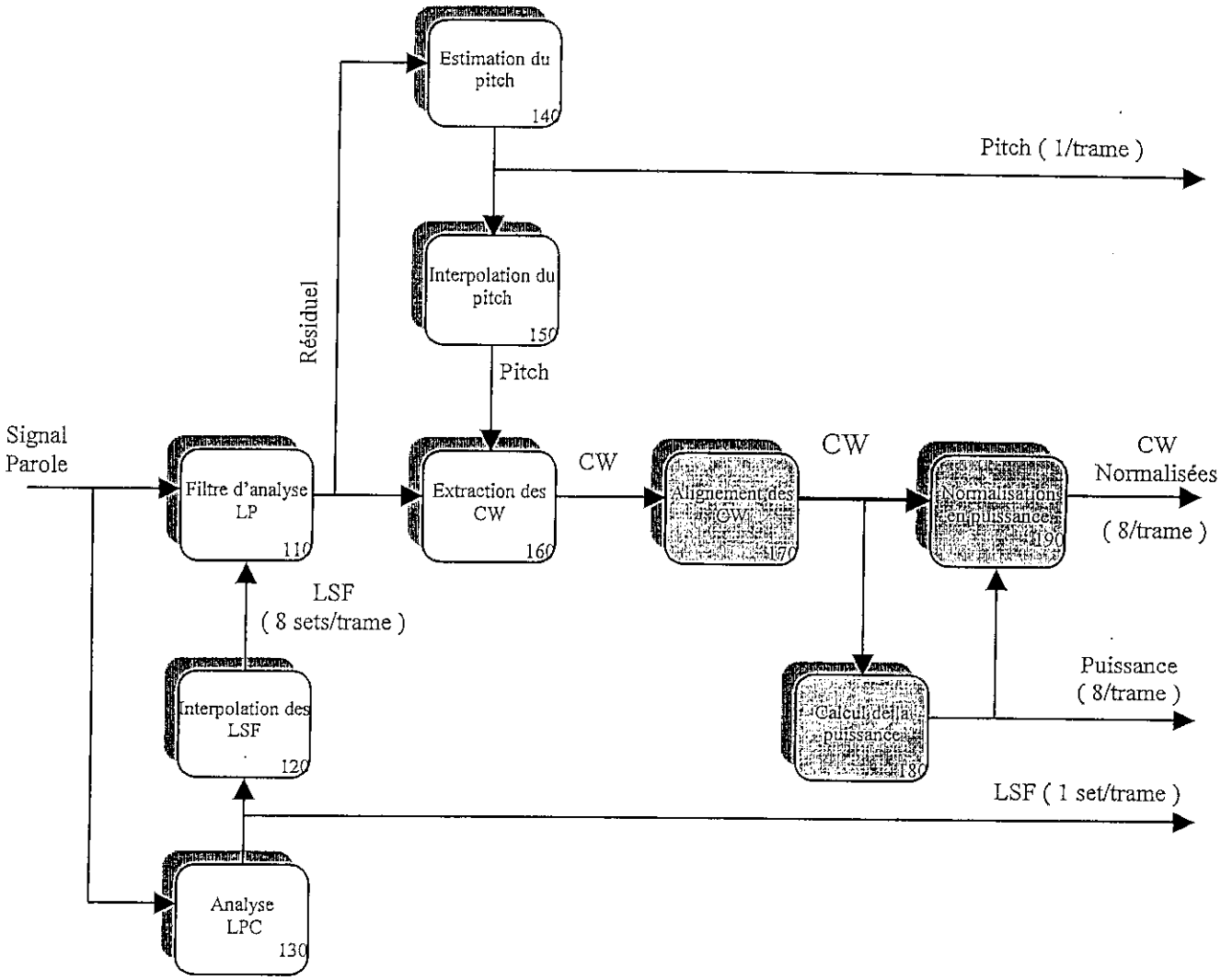

Fig. 3.3 Schéma bloc de la couche d'analyse de la WI (processeur **100**).
Les processeurs colorés travaillent à la fréquence des sous – trames
tandis que les autres travaillent à celle des trames.

3.4.2 Estimation du pitch

Les échantillons du signal résiduel (y compris les 120 échantillons de la trame future) sont envoyés au processeur **140** qui effectue l'estimation du pitch. Dans la technique WI, la précision de l'estimateur du pitch est très cruciale pour la performance du codeur. En particulier, l'opération d'extraction au codeur (processeur **160**) et l'interpolation (processeurs **230** et **250**) au décodeur reposent lourdement sur la valeur estimée du pitch.

Il existe plusieurs procédures d'estimation du pitch. Quelques unes sont basées sur la localisation des « marqueurs de pitch » (le pic dominant dans chaque période pitch du signal résiduel) tandis que d'autres sont basées sur la recherche de la position du maximum d'auto-corrélation ou du gain de prédiction pour une trame d'échantillons. Dans cette implémentation de la WI, on adopte l'algorithme tiré du EVRC (Enhanced Variable Rate Codec) [32] qui appartient à la seconde catégorie. On donne une description brève de cet algorithme dans ce qui suit.

L'estimation du pitch est effectuée une fois par trame. Pour chaque trame de données, l'estimateur fait deux calculs indépendants sur deux fenêtres qui se recouvrent. La première comprend la trame courante entière et la deuxième fenêtre comprend la seconde moitié de la trame courante et la première moitié de la trame future. Ces échantillons futurs ont déjà été calculés dans le processeur 110. Donc, l'estimation du pitch n'introduit aucun autre retard au codeur.

Puis, les calculs des gains de prédiction pour toutes les valeurs possibles du retard sont faits séparément pour chaque fenêtre. Ce gain de prédiction, noté β, est défini par :

$$\beta = \max\left\{0, \min\left\{\frac{\sum_{i=0}^{L_f-d-1} r(i)r(i+d)}{\sqrt{\sum_{j=0}^{L_f-d-1} r^2(j)\sum_{k=0}^{L_f-d-1} r^2(k+d)}}, 1.0\right\}\right\}, \quad P_{\min} \le d \le P_{\max} \qquad (3.8)$$

où d est un entier qui représente le retard et $r(.)$ est le signal résiduel. Le dénominateur sert comme facteur de normalisation et les fonction max et min permettent de garder β dans l'intervalle [0, 1]. Si le retard d correspond à la vraie valeur du pitch du signal ou à son multiple entier, le β correspondant sera proche de 1.0. Par contre, β tend à être considérablement inférieur à l'unité pour toutes les valeurs du retard si le signal ne présente aucun caractère périodique (parole non voisée). Ainsi, dans le but de retrouver le meilleur pitch, on cherche le retard d qui fournit un β maximum. Ce retard sera appelé *retard optimal*.

Après avoir trouvé le retard optimal pour chaque fenêtre, on utilise quelques seuils pour combiner les retards optimaux des deux fenêtres afin d'obtenir le retard le plus fiable dans la trame courante. Soit (d_0, β_0) le retard optimal et le gain correspondant de la première fenêtre et (d_1, β_1) ceux de la deuxième fenêtre, le retard final estimé d_{opt} est obtenu par :

$$
\begin{aligned}
&\text{Si } (\beta_0 > \beta_1 + 0.4) \\
&\quad \{ \\
&\qquad \text{si } (|d_0 - d_1| > 15) \\
&\qquad\quad d_{opt} = d_0 \\
&\qquad \text{sinon} \\
&\qquad\quad d_{opt} = \left\lceil (d_0 + d_1) / 2.0 \right\rceil \\
&\quad \} \\
&\text{sinon} \\
&\quad d_{opt} = d_1
\end{aligned}
$$

β_0 et β_1 sont des fonctions de confiance qui indiquent le degré de fiabilité des pitchs estimés (d_0 et d_1). Par exemple, si β_0 est plus grand que β_1, cela indique que d_0 est plus fiable que d_1. Il faut noter que les valeurs de d dans (3.8) sont entières. Donc, l'estimateur de pitch décrit par cette équation donne des valeurs entières du pitch. En effet, des valeurs entières du pitch (avec une résolution de 1 échantillon pour une fréquence de 8 kHz) sont suffisantes pour l'implémentation de notre codeur WI.

L'équation (3.8) travaille avec deux paramètres $P_{\min}$ et $P_{\max}$ qui sont les valeurs minimale et maximale de la période du pitch. Dans notre implémentation, elles sont égales à 20 et 120 respectivement. On pouvait étendre cet intervalle de 20 à 147 puisque, de toute manière, on alloue 7 bits pour quantifier le pitch ($147 - 20 + 1 = 128 = 2^7$). Cependant, un intervalle plus large de valeurs du pitch peut mener à plusieurs apparition de doublement / triplement de pitch dont on parlera plus en détail dans le paragraphe 3.4.3. Le codage du pitch sera détaillé au chapitre 4.

Remarques sur l'estimation du pitch

- Ce processeur donne toujours une période du pitch même si le signal n'est pas périodique. Dans le cas de parole non voisée où β est faible, la période du pitch varie. Dans ce cas, le pitch est fixé à la valeur minimale $P_{\min}$ afin de réduire la charge de calcul du codeur. Comme on va le voir dans le paragraphe 3.4.4, cette valeur du pitch sera utilisée pour fixer la longueur des CW extraites dans le processeur **160**. Les plus courtes CW permettent de réduire la complexité (les calculs), spécialement dans la transformation en DTFS et dans le processus d'alignement.

- Le calcul du gain de prédiction sur tout l'intervalle des retards (de P_{min} à P_{max}) est très onéreux.

3.4.3 Interpolation du pitch

Comme déjà mentionné dans 3.4.2, le pitch est estimé une seule fois par trame. Cependant, la WI exige une valeur de la période du pitch à chaque point d'extraction dans le processeur **160** pour exécuter l'extraction. Pour résoudre ce problème tout en gardant le même degré de complexité, on utilise un interpolateur de pitch (processeur **150**) pour calculer les pitchs intermédiaires. Bien qu'il existe plusieurs algorithmes d'interpolation du pitch, la technique d'interpolation linéaire classique est suffisante pour la WI.

Si on définit $P(n_1)$ et $P(n_2)$ comme étant les valeurs des pitchs aux extrémités de la trame courante telles que $n_1 < n_2$, alors, le pitch peut être linéairement interpolé par :

$$P(n)=\frac{(n_2-n)P(n_1)+(n-n_1)P(n_2)}{n_2-n_1}\ ,\ n_1 \le n \le n_2 \qquad (3.9)$$

où n_1 - $n_2 = L_f = 160$ échantillons dans notre implémentation.

Néanmoins, dans la parole naturelle, plus spécialement, au début et à la fin d'un segment voisé, la valeur du pitch peut doubler, tripler ou diminuer de la moitié [9]. En plus, les estimateurs de pitch souffrent souvent des erreurs fréquentes où le pitch estimé est un multiple entier du vrai pitch. Si on ne fait pas attention et qu'on effectue l'interpolation linéaire à travers ces déviations de la vraie valeur du pitch, le signal parole reconstitué contiendra des pépiements audibles.

Pour corriger ce problème, on interpole les valeurs du pitch comme suit.
Pour le cas où $P(n_1) < P(n_2)$:

$$P(n) = \begin{cases} \dfrac{C(n_2-n)P(n_1)+(n-n_1)P(n_2)}{C(n_2-n_1)} & \text{pour } n_1 \le n < \dfrac{n_1+n_2}{2}, \\ \dfrac{C(n_2-n)P(n_1)+(n-n_1)P(n_2)}{n_2-n_1} & \text{pour } \dfrac{n_1+n_2}{2} \le n < n_2. \end{cases} \qquad (3.10)$$

où la constante C est définie comme étant le rapport $P(n_2)$ sur $P(n_1)$ arrondi au plus proche entier.

Pour $P(n_1) > P(n_2)$:

$$P(n) = \begin{cases} \dfrac{(n_2-n)P(n_1)+C(n-n_1)P(n_2)}{n_2-n_1}, & \text{pour } n_1 \le n < \dfrac{n_1+n_2}{2}, \\ \dfrac{(n_2-n)P(n_1)+C(n-n_1)P(n_2)}{C(n_2-n_1)}, & \text{pour } \dfrac{n_1+n_2}{2} \le n < n_2. \end{cases} \quad (3.11)$$

où C est le plus proche entier rapport de $P(n_1)$ sur $P(n_2)$.

Le facteur C peut être considéré comme un indicateur qui nous informe si le pitch est multiple ou sous-multiple du précédent. Quand C est égal à 1, ceci indique qu'il n'y a aucun doublement ou triplement du pitch et les formules précédentes effectueront une simple interpolation linéaire (3.9). D'autre part, quand C est supérieur à 1, ça implique que le pitch est un multiple ou sous-multiple du précédent et l'interpolation décrite par (3.10, 3.11) est réalisée de manière à ce que le pitch change de façon discontinue au point milieu par le facteur C. La figure 3.4 illustre un exemple d'une telle interpolation dans le cas d'un doublement du pitch et dans celui d'une diminution de moitié du pitch.

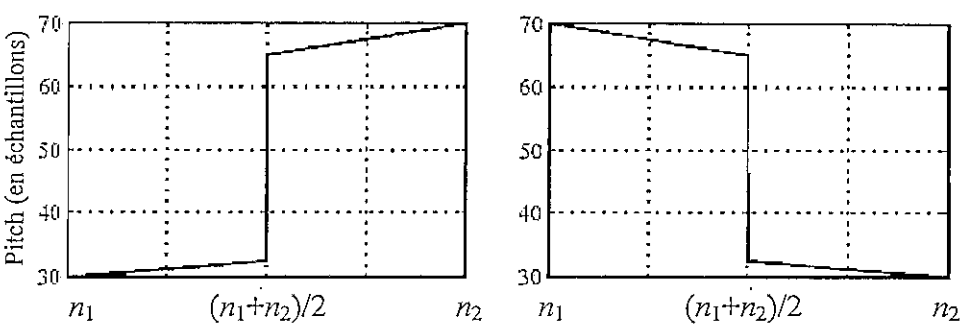

Fig. 3.4 Interpolation du pitch dans le cas d'un doublement de sa valeur. A gauche : interpolation entre 30 et 70 en utilisant (3.10). A droite : vice-versa en utilisant (3.11).

Les valeurs fractionnelles du pitch n'étant pas importantes dans la WI, toutes les valeurs résultantes de (3.9) et (3.10) ou (3.11) sont arrondies aux valeurs entières les plus proches.

3.4.4 Extraction des CW

Après avoir estimé et interpolé le pitch, on passe à l'extraction des CW dans le processeur **160**. L'opération d'extraction est effectuée une fois par sous-trame à une fréquence déterminée par le débit d'extraction R_{extr}. En fait, ce débit est lié aux limites de la fréquence fondamentale (donc de la période du pitch). Comme la limite inférieure de la longueur du pitch est égale à 20 échantillons, le nombre de CW à extraire dans une trame de 160 échantillons ne doit pas être inférieur à 160/20 = 8 CW. Nous l'avons, donc, fixé à 8/trame, c'est à dire 400 Hz dans notre simulation de la WI.

Dans le processus d'extraction, on commence par diviser la trame courante en huit intervalles de même longueur. Le point situé sur l'extrémité droite de chaque intervalle sera un *point d'extraction* comme illustré dans la figure 3.6a. Donc, deux points d'extraction adjacents seront séparés de 20 échantillons. Cet intervalle définit la longueur L_{sf} de notre sous-trame.

A chaque point d'extraction, on prend le pitch interpolé dans le processeur et on forme une *fenêtre d'extraction* de cette longueur. La fenêtre d'extraction est centrée au point d'extraction et le signal résiduel contenu dans cette fenêtre formera notre CW extraite. Par conséquent, la CW extraite a toujours la longueur de la période du pitch.

Les CW sont étendues périodiquement pendant la conversion au domaine DTFS. Par conséquent, si aucune attention n'est observée vis à vis des extrémités de la CW pendant l'extraction, cela peut mener à des discontinuités importantes dans la CW périodique (à l'endroit où l'extrémité droite rencontre l'extrémité gauche). De telles discontinuités peuvent causer des distorsions audibles dans la parole reconstituée. Pour éviter cela, le point d'extraction de chaque CW est laissé libre de balayer une certaine plage ε de positions à droite et à gauche de sa position initiale. La position qui donne la plus petite énergie du signal autour des deux extrémités de la fenêtre d'extraction est choisie. La figure 3.5 montre un exemple de l'opération d'extraction. Dans notre implémentation, ε peut prendre des valeurs entre $-\varepsilon_{max}$ et $+\varepsilon_{max} = 15$. Des expériences ont montré que ε_{max} peut aller jusqu'à 16 échantillons sans affecter la qualité de la parole reconstituée.

Pour calculer efficacement l'énergie des extrémités, on crée d'autres fenêtres appelées *fenêtres d'énergie des extrémités* centrées sur les deux points extrémités de la fenêtre d'extraction, comme montré sur la figure 3.6. L'énergie des extrémités pour une fenêtre d'extraction est la somme des énergies des échantillons qui entourent les deux extrémités de

cette fenêtre. La longueur de la fenêtre d'énergie de chaque extrémité est notée δ qu'il est suffisant de mettre égale à 10 échantillons.

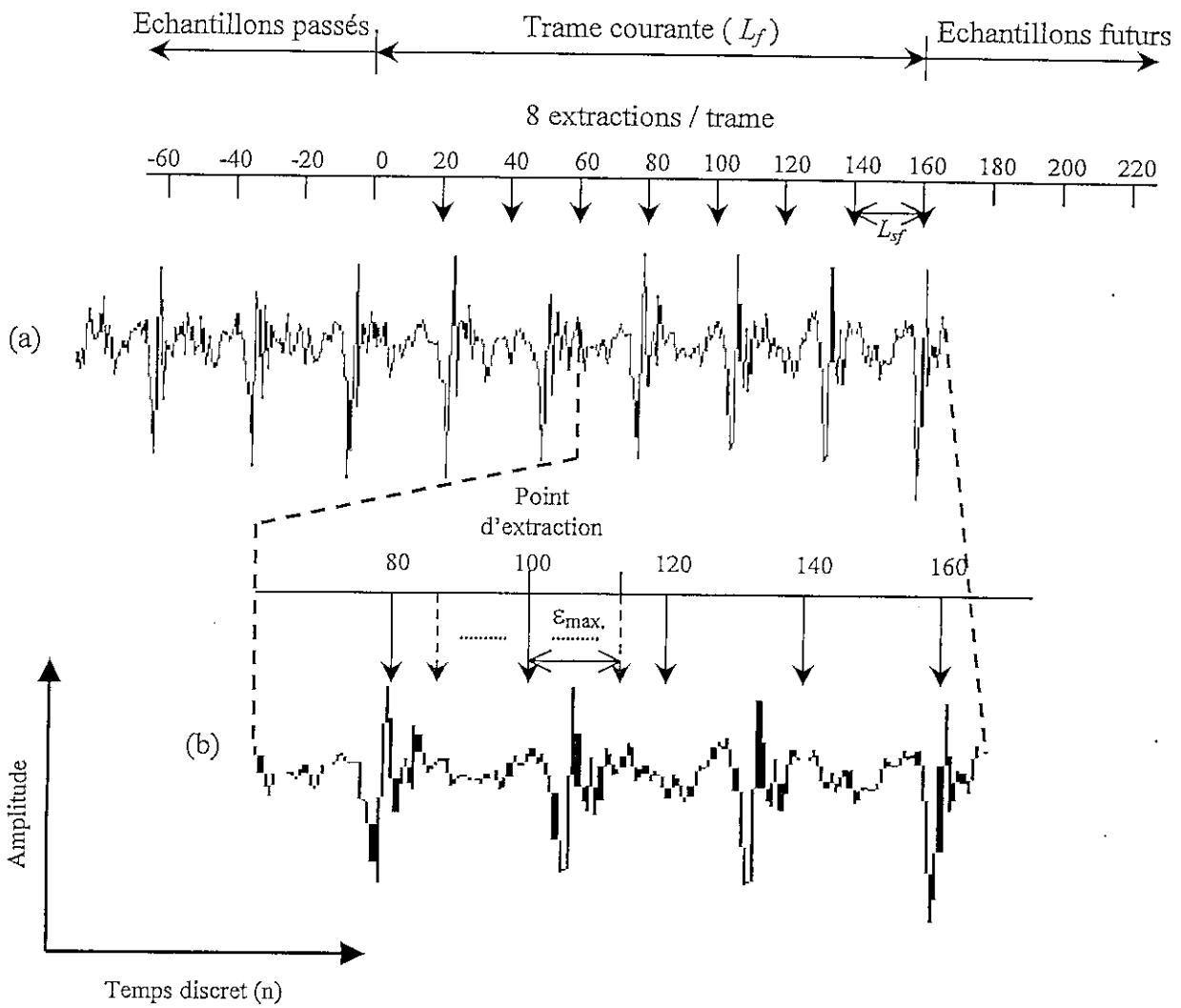

Fig. 3.5 Exemple d'un point d'extraction libre. (a) Les positions originales des points d'extraction des 8 CW. Chaque point d'extraction peut être déplacé légèrement jusqu'à ce que les extrémités de la fenêtre d'extraction soient dans des régions de faible énergie. (b) Illustration détaillée pour le point d'extraction à n = 100.

En plus de l'extraction, le processeur **160** effectue la transformation des CW au domaine DTFS en utilisant les équations (3.3) et (3.4). Il est à rappeler que les coefficients A_0 et B_0 peuvent être ignorés.

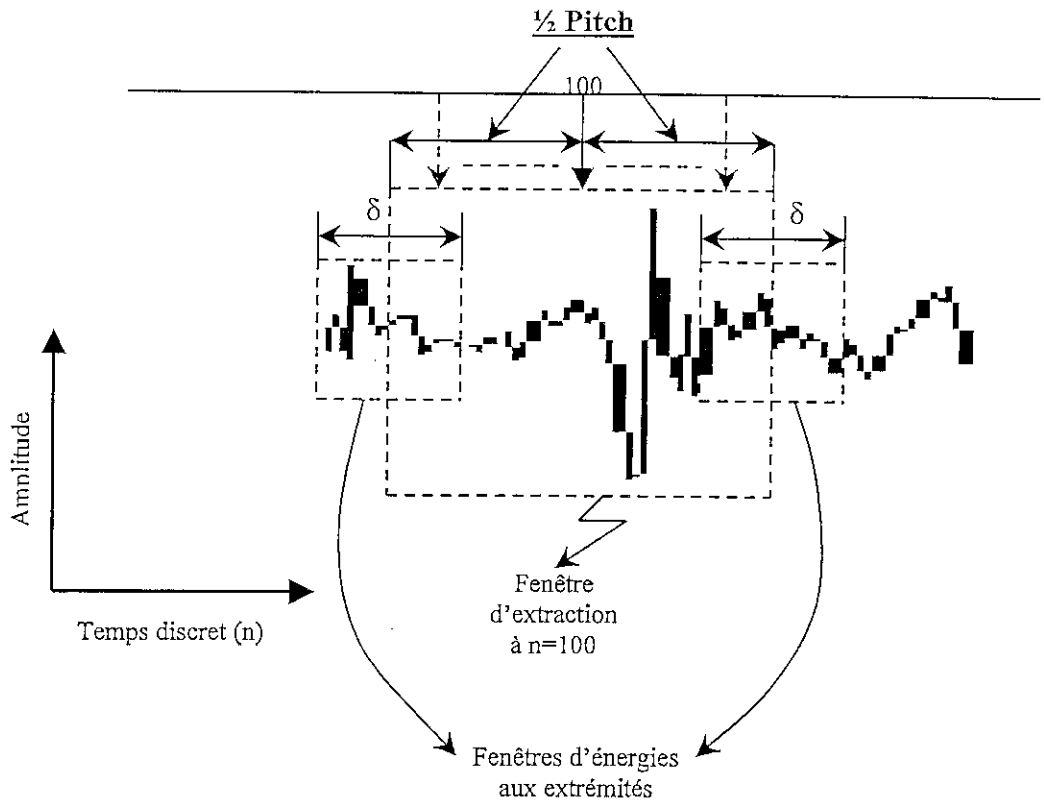

Fig. 3.6 La fenêtre d'extraction au point n=100. Ses deux fenêtres d'énergie aux extrémités sont illustrées clairement et sont de longueur δ. La fenêtre d'extraction est de longueur égale à la période du pitch.

Remarques sur l'opération d'extraction

- Les fenêtres d'extraction peuvent sortir en dehors des extrémités de la trame, d'où la nécessité d'avoir un certain nombre d'échantillons passés et futurs. Puisque la plus grande longueur d'une CW est P_{max}, le nombre d'échantillons passés nécessaires doit être au moins égal à $P_{max} \div 2 = 60$. Même chose pour le nombre d'échantillons futurs.
- Les fenêtres d'extraction successives se recouvrent presque tout le temps. En d'autres termes, deux CW adjacentes peuvent partager les mêmes segments du signal résiduel. Plus encore, puisque chaque point d'extraction peut avoir un déplacement de ε (entre –16 et 16), deux CW adjacentes peuvent même être identiques. Un exemple d'un tel cas est donné dans la figure 3.7 où les CW extraites aux points n = 120 et n = 140 sont les mêmes. Même chose pour n = 40 et n = 60.

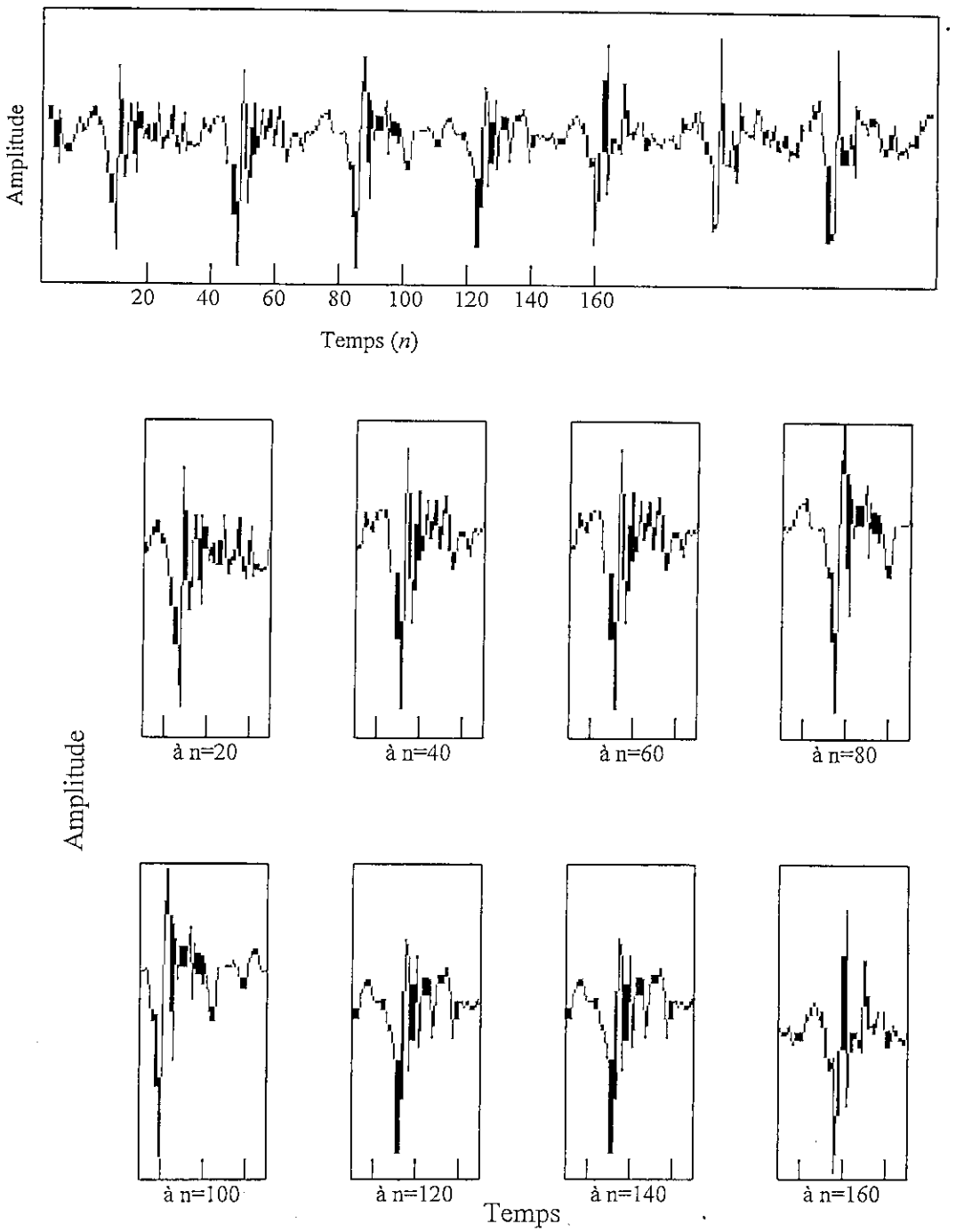

Fig. 3.7 Exemple d'extraction de 8 CW à partir d'une trame de signal résiduel. En haut : le signal résiduel original avec les points d'extraction situés à n=20, 40, ..., 160. En bas : les 8 CW extraites. Puisque la position de chaque point d'extraction peut être décalée de ε (entre −16 et 16), les CW adjacentes peuvent être extraites du même segment résiduel. La CW extraite à n=40 est la même que celle extraite à n=60. Même chose pour celle à n=80 et n=100 avec un léger décalage. Celles extraites à n=120 et à n=140 sont identiques aussi.

- Pour la parole voisée, chaque CW extraite peut être considérée comme une période individuelle du pitch. Pour la parole non voisée, les CW sont assimilables à des segments de bruit de longueurs variables.
- Dans notre implémentation, la taille d'une trame est de 160 échantillons. Puisqu'on a 8 extractions dans chaque trame et chaque extraction contient au minimum 20 échantillons (P_{min}), alors, chaque échantillon dans une trame appartient au moins à une CW si ε est à zéro.

3.4.5 Alignement des CW

La procédure d'extraction dans le processeur **170** donne une description en DTFS pour chaque CW. En général, ces CW ne sont pas en phase, ceci dit, les caractéristiques principales dans les formes d'ondes ne sont pas alignées. Afin d'avoir une description précise des CW et de leur évolution dans la trame (comme celle illustrée dans la figure 3.2c), on doit établir un *alignement* de ces CW.

Dans notre implémentation, cet alignement est réalisé dans le processeur **170** à la fréquence des sous-trames. Plus précisément, cela se fait pour chaque deux CW successives (la CW courante et la CW précédente). Le processeur aligne la CW courante avec celle précédente en introduisant un décalage temporel circulaire à la trame courante. Puisque la représentation en DTFS nous permet de considérer la CW comme une seule période d'un signal périodique, ce décalage temporel circulaire est, en réalité, équivalent à l'adition d'une phase linéaire aux coefficients DTFS.

La figure 3.8 montre un schéma bloc du processeur d'alignement **170**. Pour faciliter la compréhension de ce schéma, on va séparer la discussion du processus d'alignement en trois scénarios différents. Dans le premier scénario, on va supposer que les deux CW sont de même longueur. On va, donc, discuter le critère d'alignement (processeur **173**) et l'opération de décalage dans le temps (processeur **174**). Le premier processeur détermine la longueur du décalage temporel nécessaire à la CW courante pour être alignée avec la précédente. Le deuxième décale la CW courante en introduisant le décalage circulaire calculé par le processeur **173** aux coefficients DTFS. Les processeurs **171** et **172** ne sont pas nécessaires dans ce scénario car les CW ont la même longueur.

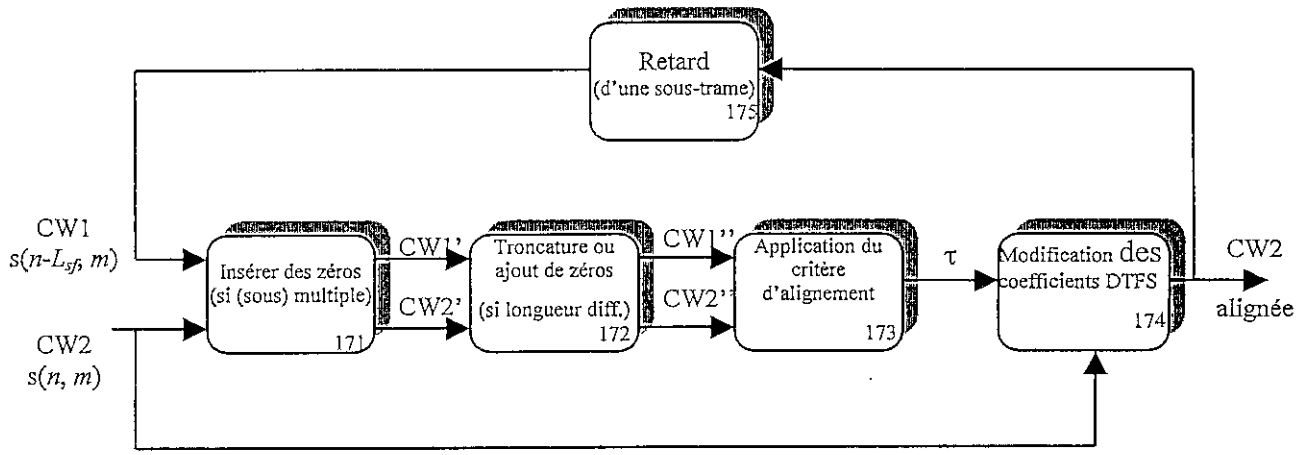

Fig. 3.8 Schéma bloc du processeur d'alignement **170**

Dans le deuxième scénario, les deux CW sont supposées de longueurs différentes (sans que l'une soit multiple de l'autre). Au lieu de chercher une nouvelle version du processeur **173**, on ajoute le processeur **172** pour résoudre le problème de la différence en longueur.

Dans le dernier scénario, on étudiera un autre processeur qui traite le cas où la longueur de l'une est multiple de celle de l'autre.

Scénario 1 : Alignement avec même dimension

Commençons avec le premier scénario où les CW courante et précédente ont la même longueur. En utilisant (3.5), les représentations en DTFS de deux CW successives sont :

$$s(n_0,m)=\sum_{k=1}^{M}\left[A_k(n_0)\cos\left(\frac{2\pi km}{P}\right)+B_k(n_0)\sin\left(\frac{2\pi km}{P}\right)\right]$$
$$s(n_1,m)=\sum_{k=1}^{M}\left[A_k(n_1)\cos\left(\frac{2\pi km}{P}\right)+B_k(n_1)\sin\left(\frac{2\pi km}{P}\right)\right] \qquad (3.12)$$

où n_0 et n_1 sont les positions dans le temps des CW précédente et présente respectivement. En plus, pour une meilleure commodité de notation,

$$P=P(n)=P(n-1)$$
$$M=\lfloor P(n)/2 \rfloor=\lfloor P(n-1)/2 \rfloor \qquad (3.13)$$

P représente la longueur (pitch) des CW et M est le nombre d'harmoniques du spectre. Dans notre implémentation, puisque le processeur **170** travaille à la fréquence des sous-trames, $n_1 - n_0 = L_{sf} = 20$.

Supposons, maintenant, qu'un décalage circulaire de T échantillons est appliqué à la CW courante, $s(n_1, m)$ devient

$$s(n_1,m-T)=\sum_{k=1}^{M}\left[A_k(n_1)\cos\left(\frac{2\pi k(m-T)}{P}\right)+B_k(n_1)\sin\left(\frac{2\pi k(m-T)}{P}\right)\right] \tag{3.14}$$

Il est clair que le décalage circulaire T dans le temps est équivalent à l'adition d'une phase linéaire $\frac{2\pi T}{P}$ dans le domaine DTFS. Pour trouver la valeur du décalage temporel T nécessaire à l'alignement de la CW1 avec la CW0, on utilise leur inter-corrélation comme suit :

$$T=\underset{0\le T'<P}{\operatorname{argmax}}\sum_{k=1}^{M}\left\{\left[A_k(n_0)A_k(n_1)+B_k(n_0)B_k(n_1)\right]\cos\left(\frac{2\pi kT'}{P}\right)+\left[B_k(n_0)A_k(n_1)+A_k(n_0)B_k(n_1)\right]\sin\left(\frac{2\pi kT'}{P}\right)\right\} \tag{3.15}$$

Le terme de droite de (3.15) est l'inter-corrélation entre les deux CW exprimée en terme de coefficients DTFS. Cette équation peut être exprimée en terme du décalage temporel normalisé τ. En substituant

$$\tau=\frac{2\pi T}{P} \tag{3.16}$$

dans (3.15), on obtient

$$\tau=\underset{0\le\tau'<2\pi}{\operatorname{argmax}}\sum_{k=1}^{M}\left\{\left[A_k(n_0)A_k(n_1)+B_k(n_0)B_k(n_1)\right]\cos(k\tau')+\left[B_k(n_0)A_k(n_1)+A_k(n_0)B_k(n_1)\right]\sin(k\tau')\right\} \tag{3.17}$$

Cette équation représente le critère d'alignement et forme la base du processeur **173**.

Un avantage immédiat de l'exécution de l'alignement dans le domaine DTFS est que cela permet un alignement fractionnel sans calcul additionnel tout en évitant les sur-échantillonnage et sous-échantillonnage conventionnels. Cet alignement fractionnel se fait à n'importe quelle résolution désirée (τ peut prendre toutes les valeurs réelles entre 0 et 2π). Une résolution de ¼ d'un échantillon pour τ (pour une fréquence d'échantillonnage de 8000 Hz) donne de bons résultats.

La prochaine étape dans l'alignement est d'incorporer le décalage temporel τ dans les coefficients DTFS de la CW courante $s(n_1, m)$. Cela se fait en développant les sinus et cosinus de (3.14) en utilisant les identités trigonométriques fondamentales. En regroupant les termes significatifs, on obtient un nouveau ensemble de DTFS :

$$\left.\begin{aligned} A_k'(n_1) &= A_k(n_1)\cos\left(\frac{2\pi kT}{P}\right) - B_k(n_1)\sin\left(\frac{2\pi kT}{P}\right) \\ B_k'(n_1) &= A_k(n_1)\sin\left(\frac{2\pi kT}{P}\right) + B_k(n_1)\cos\left(\frac{2\pi kT}{P}\right) \end{aligned}\right\} \quad \text{pour } k = 1, 2, \ldots, M \qquad (3.18)$$

d'où

$$s(n_1, m - T) = \sum_{k=1}^{M}\left[A_k'(n_1)\cos\left(\frac{2\pi kT}{P}\right) + B_k'(n_1)\sin\left(\frac{2\pi kT}{P}\right)\right] \qquad (3.19)$$

$\{A_k'(n_1)\}$ et $\{B_k'(n_1)\}$ sont les nouveaux coefficients DTFS de la CW décalée de T échantillons à droite. L'équation (3.18) peut être exprimée en terme du décalage temporel normalisé τ en utilisant (3.16) :

$$\left.\begin{aligned} A_k'(n_1) &= A_k(n_1)\cos(k\tau) - B_k(n_1)\sin(k\tau) \\ B_k'(n_1) &= A_k(n_1)\sin(k\tau) + B_k(n_1)\cos(k\tau) \end{aligned}\right\} \quad \text{pour } k = 1, 2, \ldots, M \qquad (3.20)$$

En résumé, le processeur **173** utilise (3.17) pour trouver le τ optimal et le processeur **174** utilise, alors, (3.20) pour incorporer τ dans les coefficients DTFS. La figure 3.9 montre un exemple d'une séquence de CW alignées.

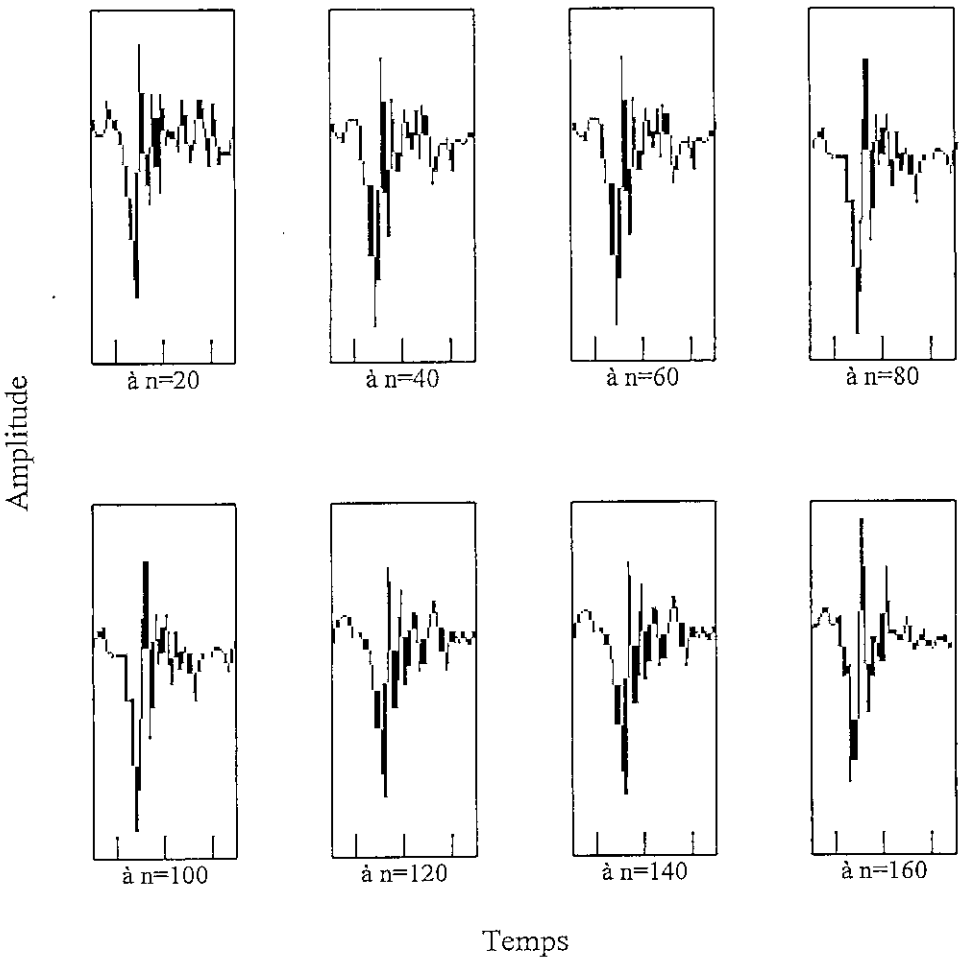

Fig. 3.9 Illustration d'une trame de CW alignées.
La version non alignée des mêmes CW est dans la fig. 3.7

Scénario 2 : Alignement avec dimensions différentes

Dans le premier scénario, on a supposé que les deux CW ont la même longueur, ce qui, en général, n'est pas le cas. En d'autres termes, le critère d'alignement (3.17), qui est basé sur cette supposition d'égalité de dimension, n'est plus applicable directement. Pour éviter de calculer un nouveau critère d'alignement, on dédie le processeur **172** pour un pré - traitement des CW en appliquant une des deux opérations suivantes afin d'égaliser leur dimensions avant de passer au critère d'alignement :

1. dans le domaine fréquentiel, on tronque la CW la plus longue jusqu'à ce qu'elle ait la même longueur que l'autre.
2. dans le domaine fréquentiel, on remplit de zéros la plus courte CW jusqu'à ce qu'elle ait la même longueur que l'autre.

Dans la première approche, abandonner les harmoniques de haute fréquence aura pour effet de rétrécir la CW dans le temps. Bien que la CW peut perdre quelques détails temporels dans ce processus, les harmoniques à l'extrémité haute fréquence du spectre tendent à avoir relativement une faible énergie. Par conséquent, la forme de la CW tronquée se rapproche, généralement, très bien la forme originale.

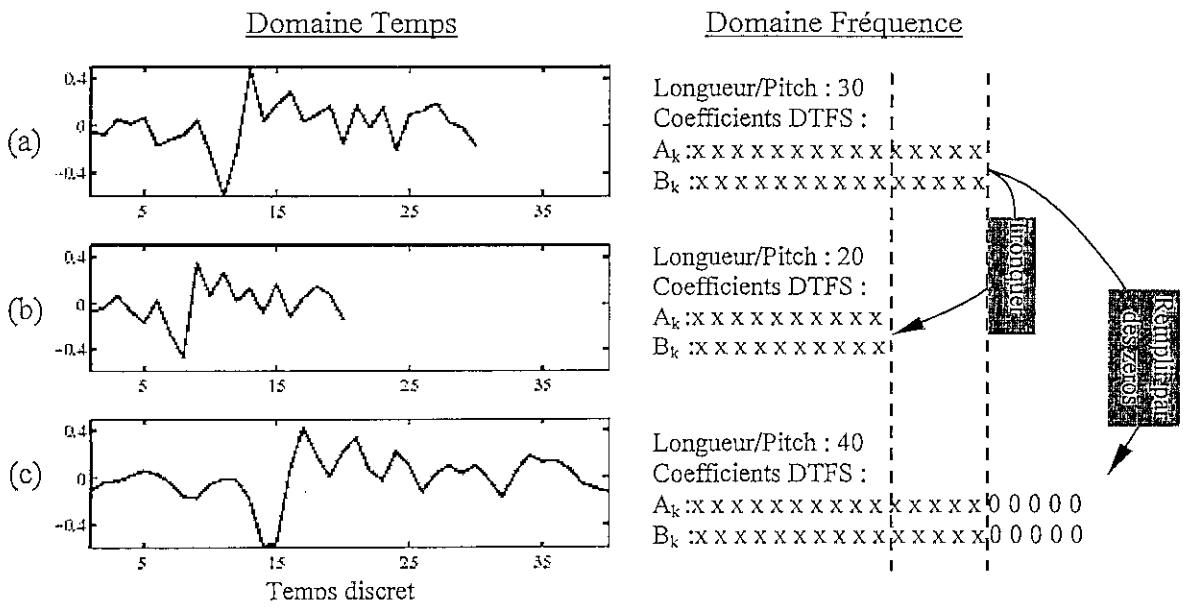

Fig. 3.10 Echelonnage temporel des CW. (a) Une CW de longueur 30 échantillons et ses coefficients DTFS correspondants 15 $\{A_k\}$ + 15 $\{B_k\}$. (b) La version tronquée de longueur 20 échantillons. La forme temporelle globale est préservée mais les détails sont plus ou moins perdus. (c) La version étirée après avoir ajouté des zéros aux coefficients DTFS. Cet étirement n'introduit aucune information nouvelle à la séquence temporelle, mais il offre une meilleure résolution.

Dans la seconde option, le remplissage par des zéros dans le domaine spectral provoque un allongement temporel de la CW pour qu'elle ait la même longueur que la CW précédente. Cette opération, équivalente à une interpolation à bande limitée dans la domaine temporel, n'introduit aucune information temporelle nouvelle à la séquence, mais elle offre une résolution plus élevée. La figure 3.10 montre l'exemple d'une CW contractée et étirée dans le temps.

Scénario 3 : Alignement avec longueur (sous-)multiple du pitch

Comme on l'a déjà mentionné au paragraphe 3.4.3, le pitch peut, occasionnellement, doubler, tripler ou diminuer de moitié dans la parole naturelle. Donc, des périodes de pitch multiples ou sous-multiples peuvent apparaître dans une CW extraite. Afin d'éviter les complications dans l'alignement, la plus courte CW est dupliquée un nombre entier de fois dans le processeur **171** de manière à ce que sa longueur atteigne celle de la plus longue CW. Dans le domaine fréquentiel, ceci est équivalent à l'insertion d'harmoniques d'amplitude nulle entre les harmoniques de la plus courte CW. La figure 3.11 montre comment les zéros sont insérés entre les coefficients DTFS $\{A_k, B_k\}$ et le résultat correspondant dans le domaine temporel.

Pour détecter l'apparition de (sous-) multiple du pitch, on opère de la même manière que celle du paragraphe 3.4.3 en utilisant l'indicateur C. Si cet indicateur est différent de l'unité, alors, il y a eu division ou multiplication du pitch. $C = 2$, signifie que la valeur du pitch a doublé et on insère un zéro entre chaque deux coefficients DTFS adjacents pour que la CW soit dupliquée une fois (Fig. 3.11b). $C = 3$, signifie que le pitch a triplé et on insère, alors, deux zéros entre chaque deux coefficients DTFS adjacents de la plus courte CW pour qu'elle soit dupliquée deux fois (Fig. 3.11c). On procède de la même manière pour les autres multiples.

Remarques sur le processus d'alignement

- L'alignement est réalisé entre deux CW successives. Pour aligner entre deux trames successives, on applique la même règle entre la première CW de la trame courante et la dernière CW de la trame passée.
- Dans la figure 3.10b, il est clair que la puissance du signal (énergie par échantillon) diminue après l'avoir tronqué dans le domaine spectral. Par contre, le remplissage par zéros et l'insertion de zéros (Figures 3.10c et 3.11) préserve la puissance du signal. La raison pour cela se fera plus claire au paragraphe 3.4.6. Le fait que la puissance diminue après avoir tronqué quelques coefficients DTFS est transparent pour l'opérateur arg max de (3.17) et, par conséquent, il n'y aura aucun effet sur la valeur finale de τ.

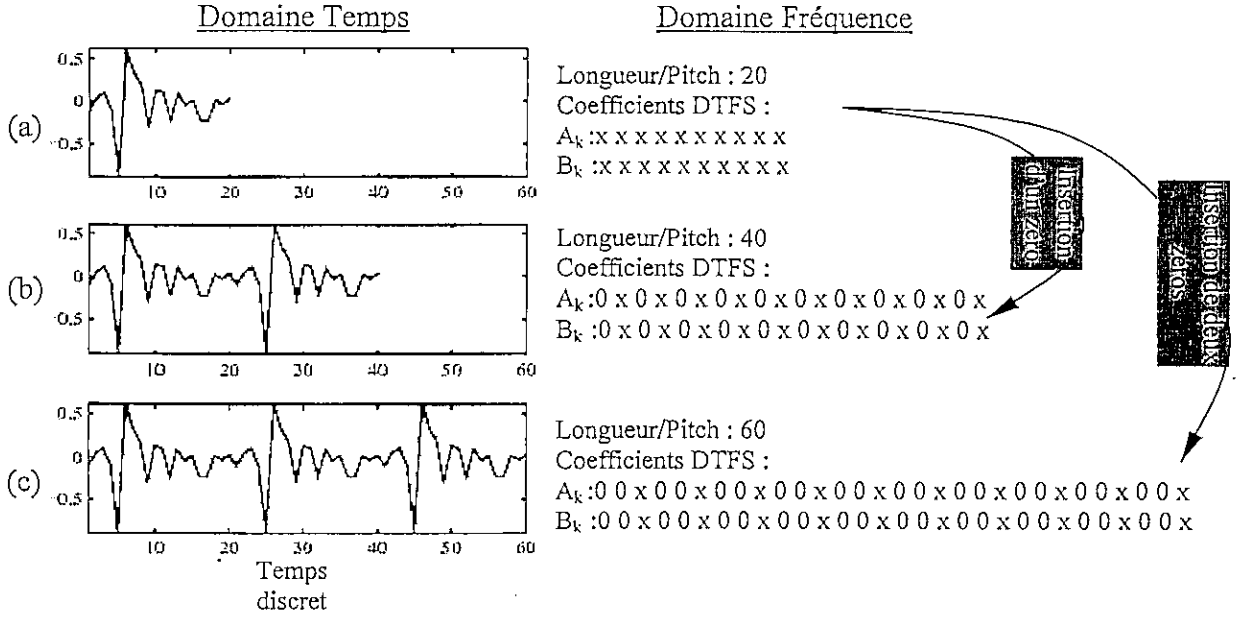

Fig. 3.11 Illustration de l'insertion de zéros entre les composantes spectrales. (a) Une CW de longueur 20 échantillons :10 $\{A_k\}$+10 $\{B_k\}$. (b) La forme d'onde de (a) est dupliquée une fois après insertion d'un zéro entre deux harmoniques adjacentes. (c) La forme d'onde de (a) est dupliquée deux fois après insertion de deux zéros entre chaque deux harmoniques adjacentes.

- Une simple évaluation du critère d'alignement (3.17) peut être très coûteuse du point de vue calcul, particulièrement pour les CW longues. Par exemple, si $s(n_0,.)$ et $s(n_1,.)$ sont de longueur 90 échantillons, on aura $90 \times 4 = 360$ inter - corrélations à calculer (en supposant que la résolution de l'alignement est de ¼ échantillon). Chacune de ces inter - corrélations nécessite au moins $90 \times 2 = 180$ multiplications selon (3.17). Ainsi, le coût total du calcul nécessaire au critère d'alignement tout seul est d'environ

$$360 \times 180 = 28800 \text{ multiplications !.}$$

- Les CW (comme on l'a déjà mentionné au § 3.4.4) sont extraites de manière à éviter une grande énergie aux extrémités ; cependant, vue la nature du décalage circulaire, le processus d'alignement peut engendrer des CW à énergie élevée aux extrémités. Toutefois, cela ne causera aucune discontinuité dans la parole reconstituée puisque les CW ont été étendues périodiquement avant l'alignement.

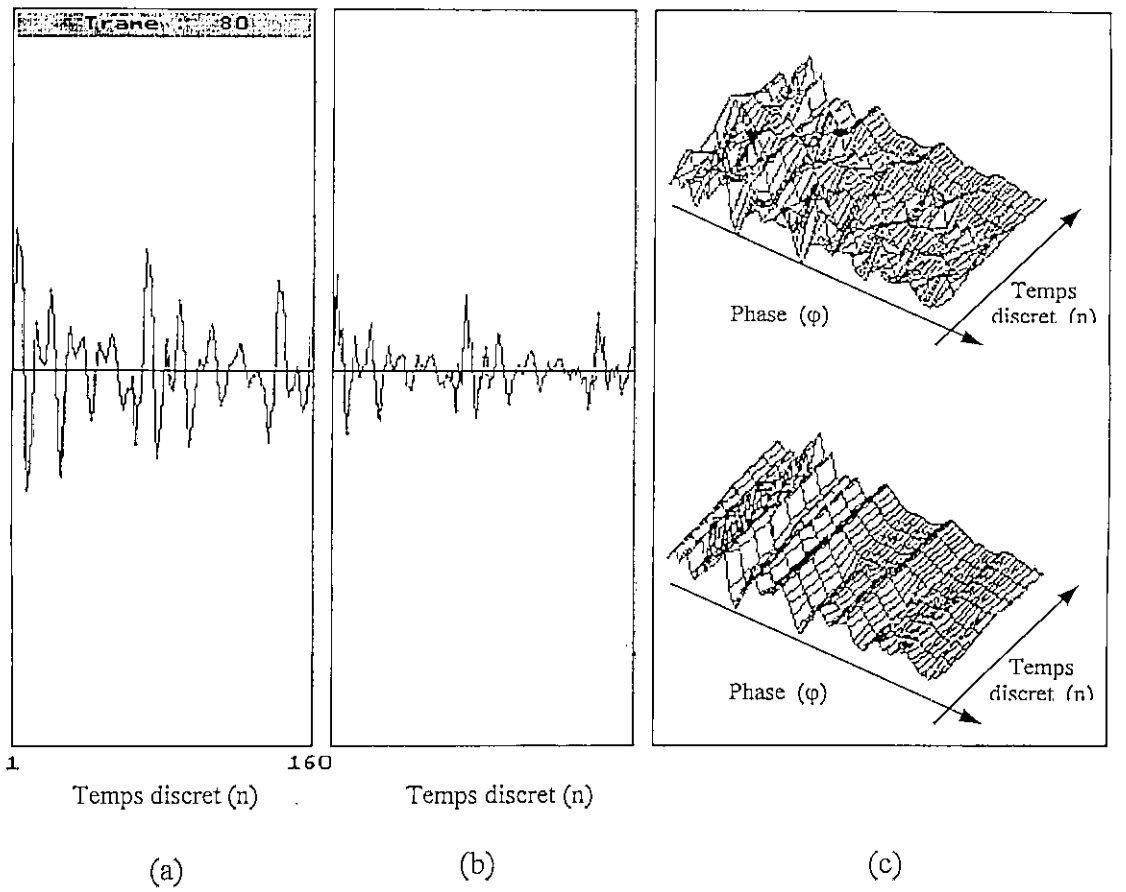

Fig. 3.12 Illustration de l'opération d'alignement.
(a) Une trame (20 ms) de parole d'une voix masculine (fichier m28b.wav).
(b) La trame du signal résiduel correspondant.
(c) En haut : Extraction et formation de la surface d'évolution de 8 CW. En bas : La surface d'évolution des CW après alignement. Le pitch vaut 68 échantillons.

3.4.6 Calcul de la puissance et normalisation des CW

Après l'extraction et l'alignement des CW, leurs puissances sont normalisées (dans cette thèse, la puissance d'une CW est définie comme étant l'énergie moyenne par échantillon sur une période du pitch). Ainsi, la relation entre une CW normalisée et sa version non normalisée est exprimée en terme de puissance. La but principal de cette normalisation est de séparer la puissance et la forme des CW afin de les quantifier séparément pour avoir une meilleure efficacité du codage.

Le processeur d'extraction de la puissance **180** calcule la puissance de chaque CW. Cette puissance et sa CW sont introduites au processeur **190** qui effectue la normalisation. Puisque toutes les CW ont déjà été converties en coefficients DTFS, le calcul de la puissance et la normalisation sont réalisés sur les coefficients DTFS $\{A_k, B_k\}$.

La puissance moyenne d'une CW à l'instant n, notée $\psi(n)$, peut être exprimée par :

$$\psi(n) = \frac{1}{P(n)} \sum_{m=0}^{P(n)-1} |s(n,m)|^2 \qquad (3.21)$$

où $P(n)$ est la longueur de la CW. En combinant (3.5) et (3.21), on obtient :

$$\psi(n) = \frac{1}{P(n)} \sum_{m=0}^{P(n)-1} s(n,m)\, s^*(n,m)$$

$$= \frac{1}{P(n)} \sum_{m=0}^{P(n)-1} s(n,m) \sum_{k=1}^{\lfloor P(n)/2 \rfloor} \left[A_k^*(n) \cos\left(\frac{2\pi km}{P(n)}\right) + B_k^* \sin\left(\frac{2\pi km}{P(n)}\right) \right]$$

Puisque nous traitons un signal parole réel et des échantillons du signal résiduel, les $\{A_k(n)\}$ et les $\{B_k(n)\}$ sont toujours réels, ce qui implique :

$$A_k(n) = A_k^*(n)$$

$$B_k(n) = B_k^*(n) \qquad (3.22)$$

Aussi, pour des besoin de simplicité, on va omettre l'indice n dans l'expression de ψ puisqu'on fait le traitement pour une seule position n. $\psi(n)$ devient :

$$\psi = \frac{1}{P} \sum_{m=0}^{P-1} s(m) \sum_{k=1}^{\lfloor P/2 \rfloor} A_k \cos\left(\frac{2\pi km}{P}\right) + \frac{1}{P} \sum_{m=0}^{P-1} s(m) \sum_{k=1}^{\lfloor P/2 \rfloor} B_k \sin\left(\frac{2\pi km}{P}\right)$$

En inter changeant l'ordre des deux sommes dans chaque terme, on obtient

$$\psi = \frac{1}{P} \sum_{k=1}^{\lfloor P/2 \rfloor} A_k \sum_{m=0}^{P-1} s(m) \cos\left(\frac{2\pi km}{P}\right) + \frac{1}{P} \sum_{k=1}^{\lfloor P/2 \rfloor} B_k \sum_{m=0}^{P-1} s(m) \sin\left(\frac{2\pi km}{P}\right)$$

Maintenant, on peut utiliser (3.3) et (3.4) pour construire

$$\psi = \begin{cases} \frac{1}{2} \sum_{k=1}^{P/2-1} \left(A_k^2 + B_k^2\right) + A_{P/2}^2 + B_{P/2}^2 & \text{pour P pair,} \\ \frac{1}{2} \sum_{k=1}^{\lfloor P/2 \rfloor} \left(A_k^2 + B_k^2\right) & \text{pour P impair.} \end{cases} \qquad (3.23)$$

L'équation (3.23) est la formule utilisée par le processeur **180** pour déterminer la puissance de la CW à partir de ses coefficients DTFS.

Pour le processeur **190** qui réalise la normalisation de la puissance, la formule est obtenue en divisant (3.23) par ψ. On obtient une puissance unité à gauche :

$$1.0 = \begin{cases} \dfrac{1}{2\psi}\displaystyle\sum_{k=1}^{P/2-1}\left(A_k^2 + B_k^2\right) + \dfrac{A_{P/2}^2}{\psi} + \dfrac{B_{P/2}^2}{\psi} & \text{pour } P \text{ pair,} \\ \dfrac{1}{2\psi}\displaystyle\sum_{k=1}^{\lfloor P/2 \rfloor}\left(A_k^2 + B_k^2\right) & \text{pour } P \text{ impair.} \end{cases} \quad (3.24)$$

En introduisant ψ à chaque coefficient DTFS, (3.24) devient

$$1.0 = \begin{cases} \dfrac{1}{2}\displaystyle\sum_{k=1}^{P/2-1}\left[\left(\frac{A_k}{\sqrt{\psi}}\right)^2 + \left(\frac{B_k}{\sqrt{\psi}}\right)^2\right] + \left(\frac{A_{P/2}}{\sqrt{\psi}}\right)^2 + \left(\frac{B_{P/2}}{\sqrt{\psi}}\right)^2 & \text{pour P pair,} \\ \dfrac{1}{2}\displaystyle\sum_{k=1}^{\lfloor P/2 \rfloor}\left[\left(\frac{A_k}{\sqrt{\psi}}\right)^2 + \left(\frac{B_k}{\sqrt{\psi}}\right)^2\right] & \text{pour P impair.} \end{cases} \quad (3.25)$$

Donc, la normalisation consiste à diviser chaque coefficient DTFS par la racine carrée de la puissance moyenne : $\sqrt{\psi}$.

En résumé, pour chaque CW entrante, le processeur **180** utilise (3.23) pour calculer la puissance ψ à partir des coefficients DTFS. Le processeur **190** divise, alors, chaque coefficient DTFS par le facteur $\sqrt{\psi}$ suivant (3.25) afin d'obtenir la version normalisée en puissance (puissance unité) de la CW.

Remarques sur la normalisation de la puissance

L'équation (3.23) indique que la puissance moyenne de la CW est directement proportionnelle à la somme des énergies des composantes harmoniques. Ceci explique pourquoi l'addition d'harmoniques d'amplitude zéro au spectre de la CW préserve la

puissance originale (Fig. 3.10c, 3.11b et 3.11c). Ceci explique, aussi, la réduction de puissance résultante de la troncature des harmoniques dans la figure 3.10b.

3.4.7 Sortie de l'étage d'analyse

En résumé, l'étage d'analyse décompose un segment de parole en quatre paramètres (le pitch, les coefficients LSF, les puissances et les CW). Les deux premiers ont une fréquence de calcul égale à celle des trames, tandis que les deux derniers sont calculés une fois par sous-trame (huit fois par trame). Il faut mettre en évidence le fait que ces CW forment une surface de formes d'ondes évoluant à deux dimensions.

Normalement, les quatre paramètres sont envoyés aux processeurs **300** et **400** pour la quantification et la dé-quantification qui seront traitées au chapitre 4. Cependant, si le codeur exécute la couche d'analyse – synthèse seulement, ces paramètres seront directement envoyés au processeur **200** (Fig. 3.1).

3.5 Etage de synthèse

A partir des LSF, pitch, puissances et CW normalisées, le signal parole peut être reconstitué dans le processeur de synthèse **200**. D'autre part, si le codeur travaille avec la couche de quantification, le bloc de synthèse reçoit les versions quantifiées de ces paramètres.

Le schéma bloc du processeur de synthèse est donné dans la figure 3.13. Similaires aux processeurs du codeur, la fréquence d'exécution varie d'un processeur à un autre dans la couche de synthèse. Quelques processeurs opèrent à la fréquence des trames, d'autres sont exécutés à la fréquence des sous-trames ou même à la celle des échantillons.

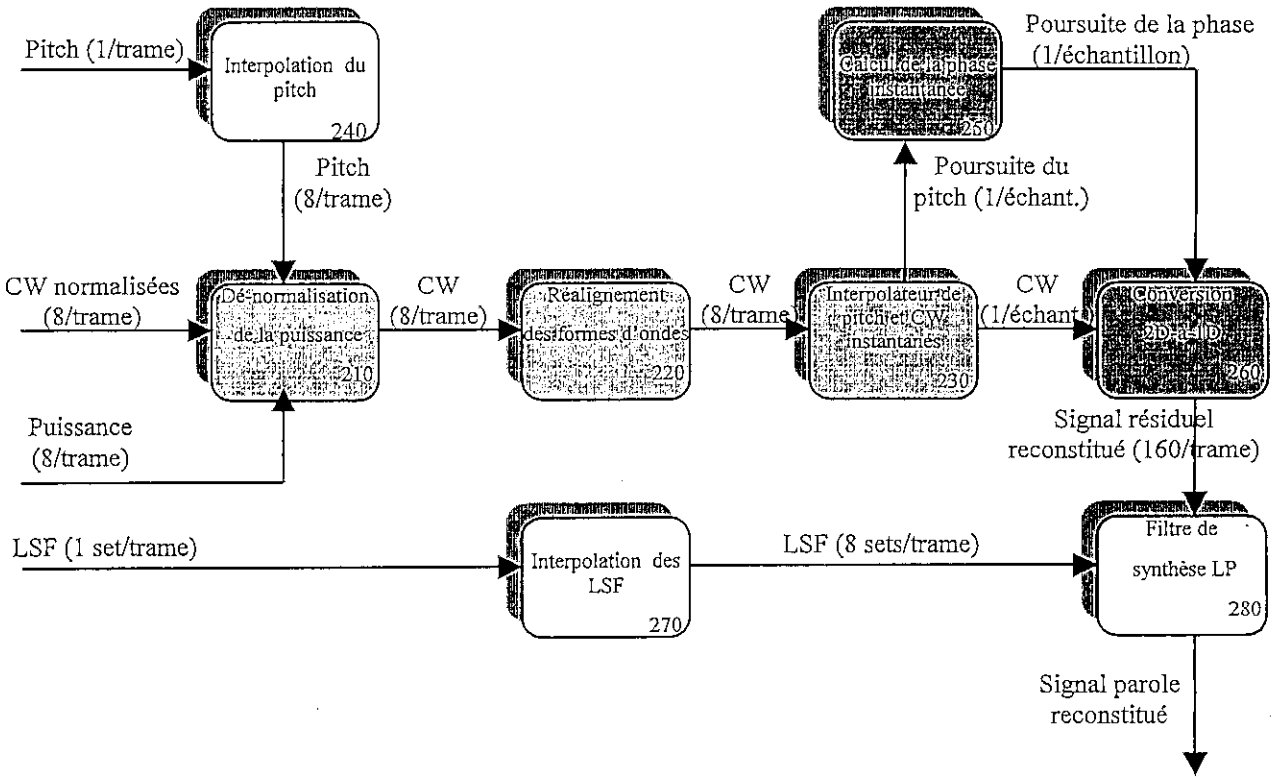

Fig. 3.13 Schéma bloc d'un décodeur WI (version détaillée du processeur **200** de la figure 3.1). Les processeurs colorés en gris clair sont exécutés une fois par trame tandis que ceux colorés en gris foncé sont exécutés à la fréquence des échantillons. Les autres sont exécutés une fois par trame.

Puisque les processeurs **240** et **270** sont identiques aux processeurs **150** et **120** respectivement, leurs descriptions fonctionnelles ne seront pas répétées.

4.5.1 Dé-normalisation en puissance et réalignement des CW

En premier lieu, chaque CW entrante est dé-normalisée dans le processeur **210**. Ce processeur utilise l'information de pitch fournie par l'interpolateur **240** pour déterminer la longueur de la CW. Pour restaurer la puissance de la CW, on fait l'opération inverse de (3.25) en multipliant chaque coefficient DTFS par $\sqrt{\psi}$.

Après avoir dé-normalisé les CW, elles sont envoyées au processeur **220** de réalignement. Si le codeur travaille sans quantification, ce processeur n'est pas nécessaire car les CW ont déjà été alignées au codeur par le processeur **170**. D'autre part, si les paramètres du codeur ont été quantifiés, les CW successives peuvent ne plus être bien alignées une fois dé-quantifiées.

3.5.2 Génération des pitchs et CW instantanés

Maintenant, nous avons une CW reconstruite et alignée dans chaque sous – trame. Dans la technique WI, il est nécessaire d'avoir une CW et une valeur du pitch à chaque point d'échantillonnage pour reconstruire le signal résiduel unidimensionnel. Ces CW et pitchs instantanés sont générés dans le processeur **230**.

Une interpolation linéaire peut servir à sur – échantillonner les CW. Quand ce sur – échantillonnage est exécuté entre deux CW de même longueur, une interpolation directe est appliquée. Cependant, si les CW ont des dimensions différentes ou si l'une a une longueur multiple de celle de l'autre, des calculs supplémentaires seront nécessaires pour assurer une bonne interpolation.

L'interpolation est linéaire mais n'emploie pas les équations (3.10) et (3.11) de pitch (sous-) multiple. Il faut bien s'assurer que les valeurs de pitch générées dans cet interpolateur correspondent aux longueurs des CW instantanées.

La figure 3.14 montre le schéma bloc de l'interpolateur **230** qui peut prendre en charge l'interpolation des CW et du pitch dans trois scénarios possibles : (i) dimensions égales, (ii) dimensions différentes et (iii) dimensions (sous-) multiples du pitch.

Scénario 1 : interpolation avec dimensions égales

En premier lieu, on suppose que les deux CW ont la même longueur P. Ainsi, les processeurs **231**, **232** et **234** ne seront pas exécutés. Si on note par n_0 et n_1 les instants des extrémités de l'intervalle d'interpolation, alors, la CW instantanée $s(n, m)$ à l'instant n peut être calculée par interpolation entre $s(n_0, m)$ et $s(n_1, m)$. Dans le domaine temporel, cette opération est exprimée par :

$$s(n,m)=\left(\frac{n_1-n}{n_1-n_0}\right)s(n_0,m)+\left(\frac{n-n_0}{n_1-n_0}\right)s(n_1,m) \qquad n_0\le n\le n_1, 0\le m<P \qquad (3.26)$$

En substituant (3.5) dans (3.26), on obtient

$$\left.\begin{aligned} A_k(n)&=\left(\frac{n_1-n}{n_1-n_0}\right)A_k(n_0)+\left(\frac{n-n_0}{n_1-n_0}\right)A_k(n_1)\\ B_k(n)&=\left(\frac{n_1-n}{n_1-n_0}\right)B_k(n_0)+\left(\frac{n-n_0}{n_1-n_0}\right)B_k(n_1)\end{aligned}\right\} \quad \text{pour } k=1, 2, \ldots, \lfloor P/2\rfloor \qquad (3.27)$$

En d'autres termes, l'interpolation linéaire entre les deux CW dans le temps est équivalente à celle de leurs coefficients DTFS. L'interpolation est exécutée une fois par sous – trame,

$$n_1-n_0=L_{sf}=20.$$

Puisque les deux CW sont de même longueur, les CW interpolées auront la même longueur également. Par conséquent, le processeur **235** délivrera un contour constant du pitch.

Scénario 2 : interpolation avec dimensions différentes

En général, le pitch varie dans l'intervalle d'une sous - trame et, donc, les CW aux extrémités auront des longueurs différentes (différents nombres de coefficients $\{A_k, B_k\}$). Pour faciliter l'interpolation dans un cas pareil, on peut allonger dans le temps la plus petite CW pour qu'elle ait la même longueur que la plus longue avant de passer à l'interpolation. Comme déjà fait au paragraphe 3.4.5, un tel allongement dans le temps est équivalent à l'addition d'harmoniques nulles dans la représentation DTFS. Ainsi, le processeur **232** est exécuté pour ajouter des zéros à la plus petite CW avant d'interpoler entre les deux CW (processeur **233**) de la même manière que dans le scénario 1 pour obtenir les CW instantanées, le processeur **231** restant toujours inactif.

Dans le processeur **235**, l'équation d'interpolation linéaire conventionnelle (3.9) peut être utilisée pour sur-échantillonner le pitch. Cependant, les valeurs du pitch sur-échantillonnées résultant de ce processeur peuvent ne pas coïncider avec les longueurs des CW interpolées (au processeur **233**). Pour éviter un tel problème, on utilise le processeur **234** pour faire coïncider les longueurs des CW avec le contour du pitch.

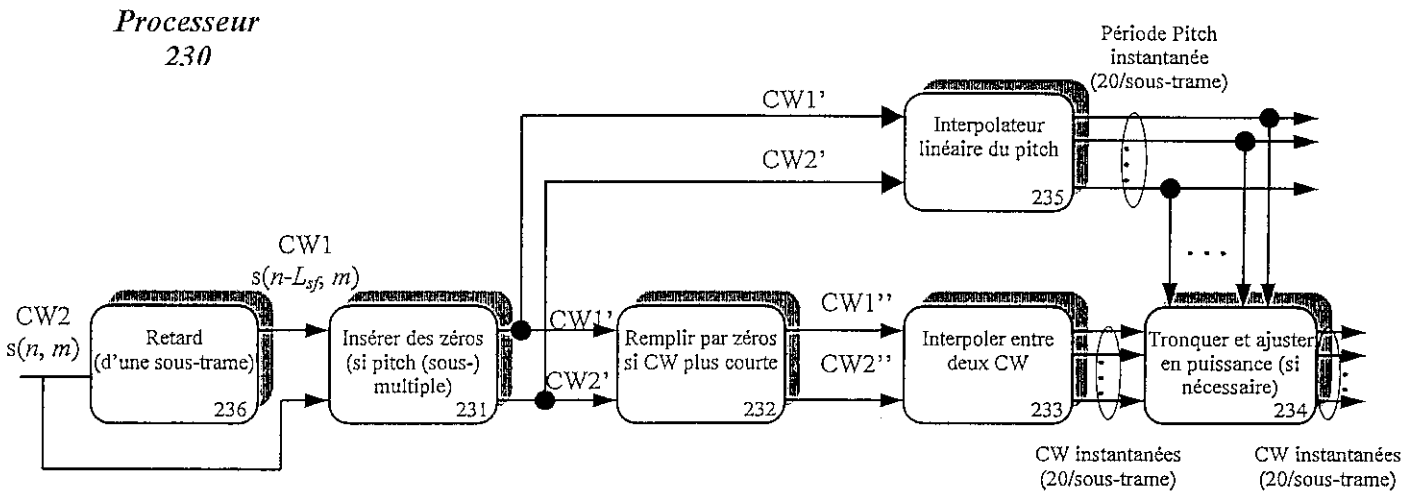

Fig. 3.14 Schéma bloc du processeur d'interpolation

Scénario 3 : interpolation avec des longueurs (sous-) multiples du pitch

Si la CW courante est considérablement plus longue ou plus courte que la précédente, cela implique que le pitch actuel est certainement multiple ou sous-multiple, respectivement, du précédent. Le processeur **231** sert à traiter un tel cas. De même que le processeur **170**, ce processeur utilise l'indication C comme détecteur de (sous-) multiple de pitch. Si le pitch a été multiplié ou divisé ($C > 1$) sur l'intervalle de la sous - trame, le processeur duplique la plus petite CW un nombre entier de fois de manière à ce que leur longueur totale soit très proche ou égale à celle de la plus longue CW. Cette procédure est équivalente à l'insertion d'harmoniques nulles entre les harmoniques originales de la représentation DTFS. Une fois que ces zéros sont insérés, les CW sont envoyées au processeur **232** et traitées de la même manière que dans le scénario 2.

Remarques sur l'interpolation du pitch et des CW

- A l'extrémité d'une trame, l'interpolation est effectuée entre la dernière CW de la trame passée et la première CW de la trame courante.
- Les valeurs instantanées du pitch envoyées au processeur **250** ne sont pas arrondies en valeurs entières afin d'avoir une meilleure précision de la poursuite de phase.

3.5.3 Estimation de la phase instantanée

On va étudier le processeur **250** dont l'objectif est de convertir les valeurs du pitch en une poursuite de la phase. Ce contour de la phase sera utilisé par le processeur **260** qui permet de retrouver le signal résiduel unidimensionnel à partir de la surface bidimensionnelle de CW. Puisqu'on a une valeur du pitch à chaque instant d'échantillonnage, la poursuite de la phase peut être calculée en faisant la sommation de l'air se trouvant en dessous du contour de la fréquence $F(n)$. La relation entre $F(n)$ et le pitch $P(n)$ est exprimée par :

$$P(n) = \frac{1}{F(n)} \tag{3.28}$$

Si on désigne par $\phi(.)$ le contour de la phase, la phase en chaque point d'échantillonnage peut être calculée par

$$\phi(n) = \phi(n-1) + \int_{n-1}^{n} \frac{2\pi}{P(n')} dn' \tag{3.29}$$

où $\phi(n)$ et $\phi(n-1)$ sont les phases courante et précédente. L'intégrale correspond à l'aire de l'intervalle entre les points $n-1$ et n. En supposant que le pitch évolue linéairement sur l'intervalle d'intégration, (3.29) peut être écrite sous la forme :

$$\phi(n) = \phi(n-1) + \int_{n-1}^{n} \frac{2\pi}{(n-n')P(n-1) + (n'+n+1)P(n)} dn' \tag{3.30}$$

Une évaluation rapide de cette intégrale mène à :

$$\phi(n) = \begin{cases} \phi(n-1) + \dfrac{2\pi}{P(n)-P(n-1)} \ln\left[\dfrac{P(n)}{P(n-1)}\right] & \text{Pour } P(n) \neq P(n-1), \\ \phi(n-1) + \dfrac{2\pi}{P(n)} & P(n) = P(n-1). \end{cases} \tag{3.31}$$

En exécutant (3.31) échantillon par échantillon, le processeur **250** peut convertir le pitch instantané $P(.)$ en une phase instantanée $\phi(n)$. La phase $\phi(n)$ est une fonction croissante puisque l'intégrale dans (3.30) est toujours positive. Afin de réduire la complexité du calcul, une technique similaire à la somme de Riemann peut être adoptée pour approcher l'intégrale de (3.29). Plus précisément, l'intégrale $\int_{n-1}^{n} \frac{dn'}{P(n')}$ peut être évaluée approximativement par le rectangle de hauteur

$$\frac{1}{2}\left(\frac{1}{P(n-1)}+\frac{1}{P(n)}\right) \tag{3.32}$$

et de largeur égale à celle de l'échantillon. Remarquons que (3.32) représente la moyenne des deux fréquences successives $F(n\text{-}1)$ et $F(n)$. Il résulte de cette approximation que (3.29) peut être écrite :

$$\phi(n) \approx \phi(n-1) + \pi\left(\frac{1}{P(n-1)}+\frac{1}{P(n)}\right) \tag{3.33}$$

Pour une implémentation en pratique, la relation (3.33) est une approximation fiable de (3.31). La phase initiale $\phi(0)$ au début de chaque trame peut être fixée à une valeur arbitraire (aléatoire) car elle n'affecte pas la qualité de perception de la parole reconstituée.

3.5.4 Transformation 2D à 1D

Le processeur **260** convertie la surface bidimensionnelle des CW $(A_k(.), B_k(.))$ en un signal résiduel unidimensionnel $r(.)$. L'opération de conversion est effectuée échantillon par échantillon comme on peut le voir graphiquement par l'exemple de la figure 3.15 qui montre le processus de reconstitution d'une voix masculine. La poursuite de la phase instantanée correspondant à cette trame (dans le processeur **250**) est illustrée dans la figure 3.15a. La figure 3.15b montre la surface des CW interpolées (dans le processeur **230**) où chaque CW est normalisée à la longueur 2π. La transformation se fait en superposant les deux graphes. La projection de leur intersection (points de rencontre des droites de poursuite de la phase avec la surface de CW) donne le signal résiduel $r(n)$ (figure 3.15c). Cette transformation est implémentée par l'opération inverse de la décomposition en DTFS. Plus précisément, $r(n)$ est déterminé en utilisant (3.7) où ϕ est une fonction de n :

$$r(n) = s(n,\phi(n)) = \sum_{k=1}^{\lfloor P(n)/2 \rfloor}\left[A_k(n)\cos(k\phi(n)) + B_k(n)\sin(k\phi(n))\right] \qquad 0 \le \phi(.) < 2\pi \tag{3.34}$$

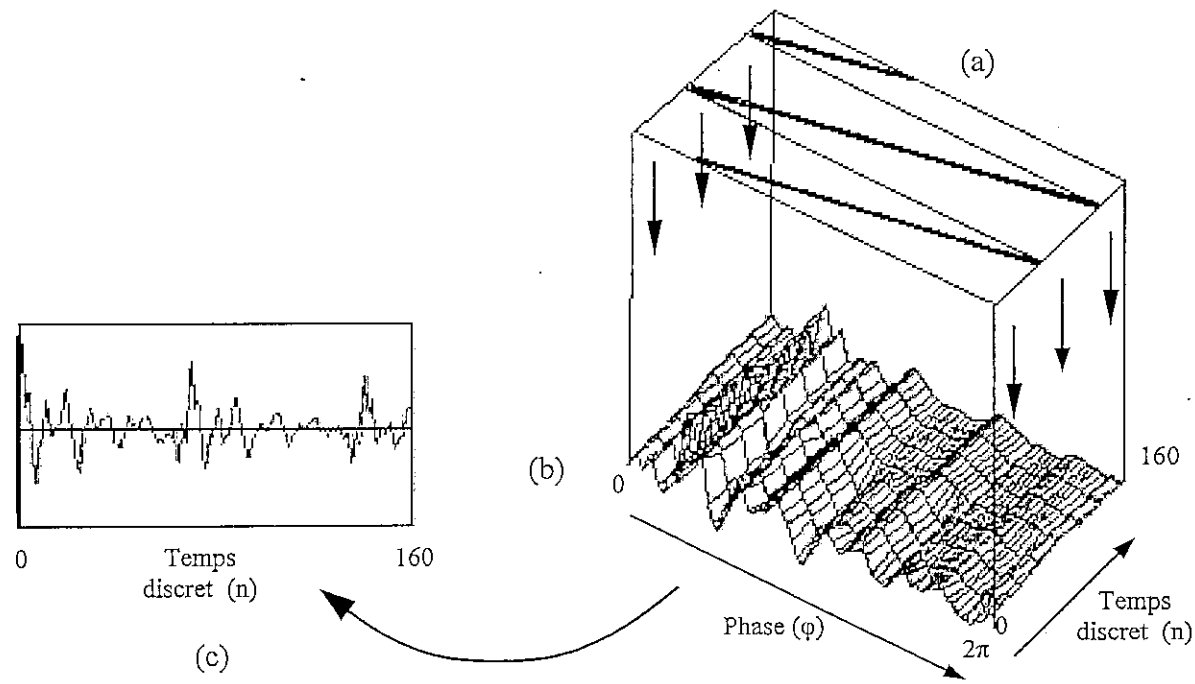

Fig. 3.15 Transformation de la surface de CW en signal résiduel. (a) les droites indiquant la poursuite de la phase interpolée instantanée pour un signal voisé de pitch égal à 68. (b) la surface de CW sur laquelle seront projetées les droites de (a). (c) le signal résiduel résultant de l'intersection.

3.5.5 Synthèse LP

Le signal résiduel reconstitué est utilisé comme signal d'excitation du filtre de synthèse LP dans le processeur **280** pour obtenir le signal parole final. La fonction de transfert du filtre est équivalente à celle de la figure 2.1 et les coefficients du filtre sont donnés par la conversion des LSF en coefficients LP. La parole ainsi reconstituée est *désaccentuée* en utilisant le filtrage inverse de celui de la pré-accentuation avec la même valeur α utilisée dans le processeur **130**.

3.6 Performances de la couche d'analyse – synthèse

La couche d'analyse - synthèse a été simulée (sans quantification) à travers un programme en langage C. Dans cette section, on va exposer les différentes questions

concernant la performance de ce système. Les résultats de l'évaluation subjective seront présentés également.

3.6.1 Désynchronisation dans le temps (Time Asychrony)

La méthode WI décrite dans ce chapitre ne maintient, généralement, pas la synchronisation entre le signal parole original et reconstitué. Ceci est, principalement, dû au manque d'exactitude dans l'estimation du contour de la phase et à l'opération d'échelonnage temporel pendant l'interpolation des formes d'ondes (processeur **230**). Une parfaite reconstruction dans la méthode WI serait possible si les conditions suivantes étaient satisfaites :

1- Effectuer l'extraction des CW une fois par échantillon au lieu de une fois par sous - trame (éliminant, ainsi, l'interpolation des CW et l'opération d'échelonnage dans le temps).

2- Atteindre une poursuite exacte de la phase $\phi(n)$.

3- Obtenir la valeur exacte de la phase initiale $\phi(0)$.

4- Fixer les points d'extraction, c. à. d. $\varepsilon = 0$.

5- Préserver la composante continue (DC) dans chaque CW.

Il faut noter qu'il est très difficile, en pratique, d'avoir la valeur exacte du pitch à chaque instant d'échantillonnage.

Il résulte de la désynchronisation dans le temps de la WI que la mesure du SNR entre la parole originale et celle codée ne peut pas être utilisée comme critère de fidélité ; par conséquent, on utilise les tests subjectifs pour évaluer la qualité de la parole reconstituée de la technique WI.

3.6.2 Evaluation subjective de la qualité

Il a été prouvé qu'une parole presque transparente peut être générée par le schéma non quantifié de la WI [19]. Pour vérifier la précision de notre système d'analyse – synthèse, on a procédé à un test d'écoute comparant la parole originale à celle reconstituée (sans quantification).

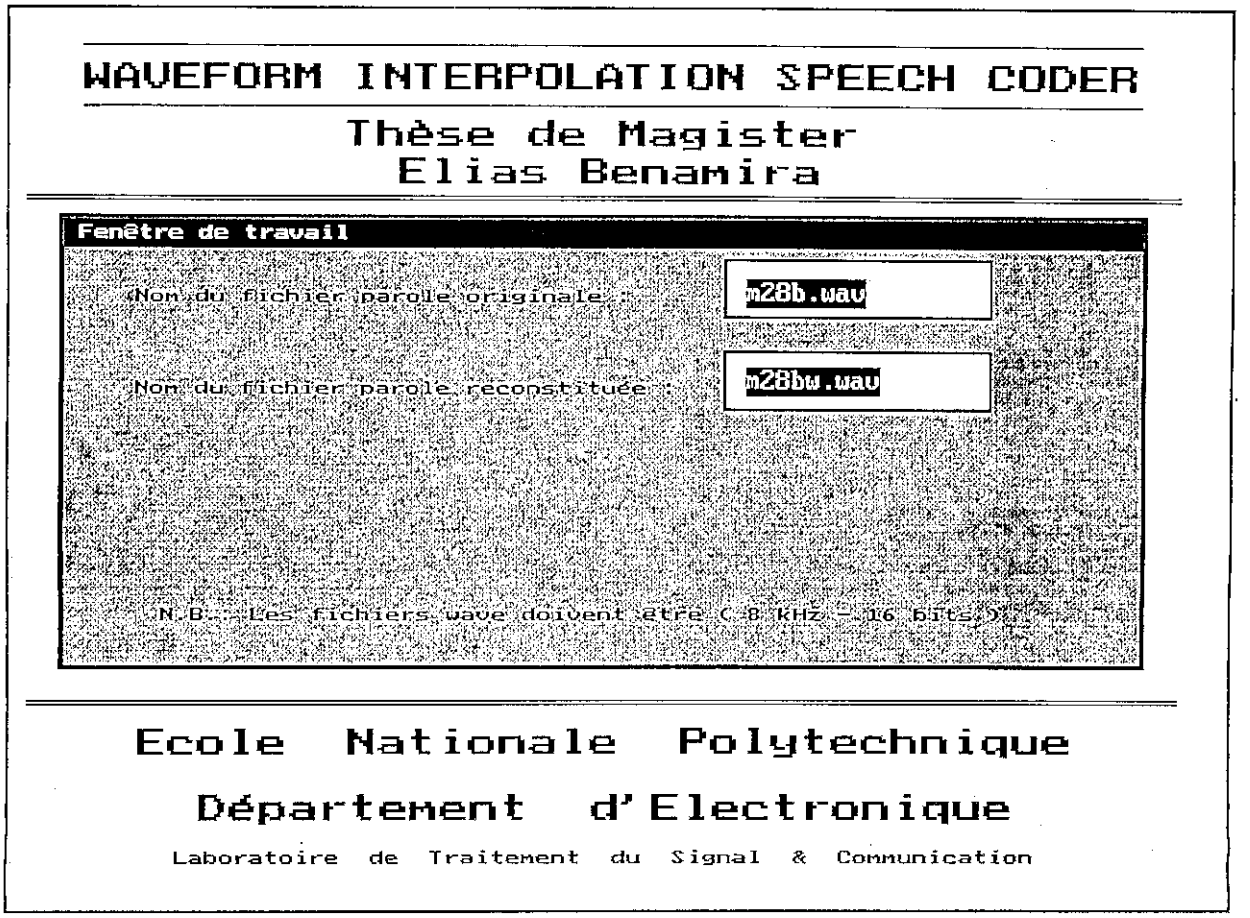

Fig. 3.16 Interface graphique du programme du codeur WI avec Borland C++ 3.1.

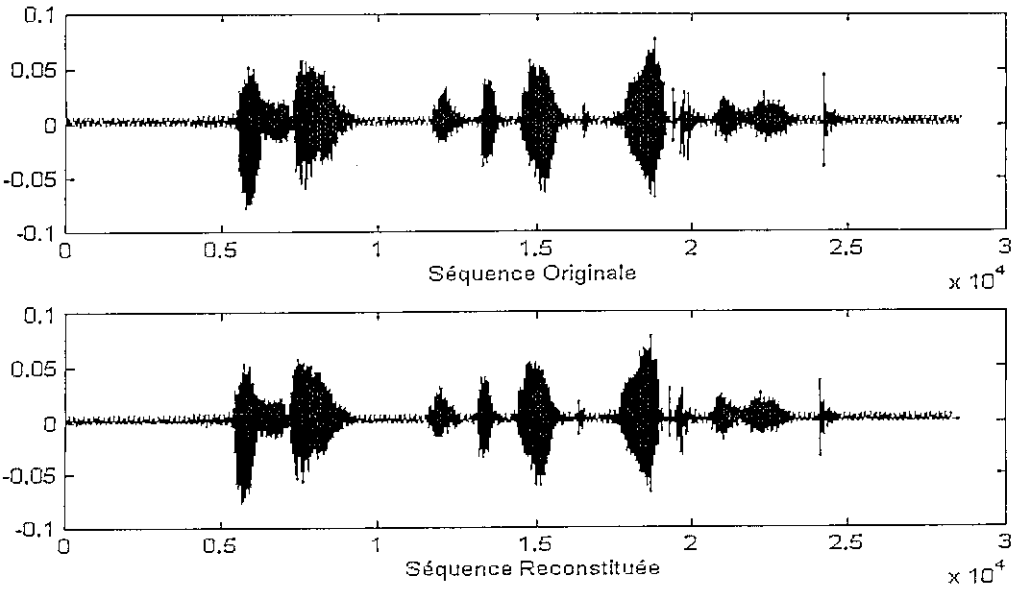

Fig. 3.17 Graphes original (en haut) et reconstitué (en bas) d'une des phrases du test d'évaluation subjectif prononcée par une femme : *« Sunday is the best part of the week»*. La visualisation ainsi que l'écoute sont réalisés grâce à un petit programme MATLAB.

Le test consistait à faire écouter 10 phrases, dont 6 sont prononcées par des hommes et 4 par des femmes, par une vingtaine de « listeners ». Ces derniers devaient qualifier la parole reconstituée en la comparant à la phrase originale selon l'échelle de cinq points du test du MOS décrit au chapitre 1. La moyenne des évaluations de tous les listeners était de 4/5 ce qui correspond à une « bonne » qualité de la parole reconstituée. La figure 3.17 donne un résultat des tests dans lequel on voit et on écoute les phrases originale et reconstituée grâce à un petit code écrit en Matlab. La figure 3.16 montre l'interface graphique du programme de codage/décodage.

3.7 Importance de l'extension de la largeur de bande

L'extension de la largeur de bande est très bénéfique à l'opération LP car elle assure la stabilité des filtres LP. Dans la simulation, on a remarqué que l'extension de la largeur de bande peut améliorer d'autres aspects du codeur WI. Il s'agit d'un phénomène observé dans le signal résiduel LP qui est la disparition des impulsions du pitch (pitch pulse disappearance) [15]. Ce phénomène peut affecter la performance du codeur. L'extension de la largeur de bande permet de résoudre ce problème.

Il a été observé que quelques segments de parole humaine ont une forme sinusoïdale. Ceci apparaît surtout dans les sons nasalisés car les zéros du spectre dans ces sons tendent à supprimer les énergies dans les second et troisième formants. Par conséquent, il n'y aura qu'un ou deux harmoniques dominantes dans le spectre du signal, ce qui entraîne des formes d'ondes presque sinusoïdales dans le signal parole. Ces forme d'ondes sinusoïdales, qui ont des corrélations court- terme élevées, donnent un gain de prédiction élevé dans l'analyse LP. Ces gains élevés, à leur tour, font que le signal résiduel a une énergie faible, d'où les impulsions des périodes pitch commencent à disparaître. Cela peut affecter les performances de l'estimation du pitch et augmente les chances d'avoir un mauvais alignement des CW. Une solution à ce problème et d'ajuster le facteur γ d'extension de la largeur de bande dans l'analyse LP. En diminuant γ, les impulsions des pitchs commencent à réapparaître dans le signal résiduel. Plus en diminue γ, plus les impulsions des pitchs deviennent claires. Cependant, plus γ diminue, plus l'énergie du signal résiduel augmente, d'où une quantification des CW moins efficace. La valeur de γ peut descendre jusqu'à 0.9.

Chapitre 4

QUANTIFICATION DES PARAMETRES DU CODEUR

La couche d'analyse - synthèse de la WI (en l'absence de quantification) fournit une parole de qualité transparente et, donc, ferait l'objet d'une excellente base pour le développement d'un codeur de la parole. Dans ce chapitre, on va discuter la couche de quantification et présenter une conception initiale d'un codeur de la parole à 4 kbps.

Il y a quatre paramètres à transmettre dans le schéma de la WI : les paramètres LP (LSF), le pitch, l'énergie et les CW. La figure 4.1 montre les processeurs sollicités pour le codage des paramètres. Notons que ces processeurs font partie des processeurs **300** et **400** de la figure 3.1.

On va commencer par le codage des LSF et du pitch. Puis, on va procéder à la quantification de l'énergie et des CW qui nécessite d'autres traitements avant le codage.

4.1 Quantification des LSF

Dans l'allocation de bits du codeur WI à 4 kbps, on alloue 30 pour la quantification de chaque ensemble de LSF dont la fréquence de mise à jour et de transmission est de 50 Hz (un ensemble par trame).

La quantification vectorielle par division SVQ (Split Vector Quantization) est employée avec un vecteur de 10 coefficients LSF divisé en trois sous - vecteurs de dimensions 3, 3 et 4. Ces sous – vecteurs sont quantifiés séparément en utilisant 10 bits pour chacun. Le meilleur dictionnaire (codebook) pour chaque sous - vecteur est sélectionné sur la base de la mesure de la distorsion pondérée moyenne spécifiée par [23].

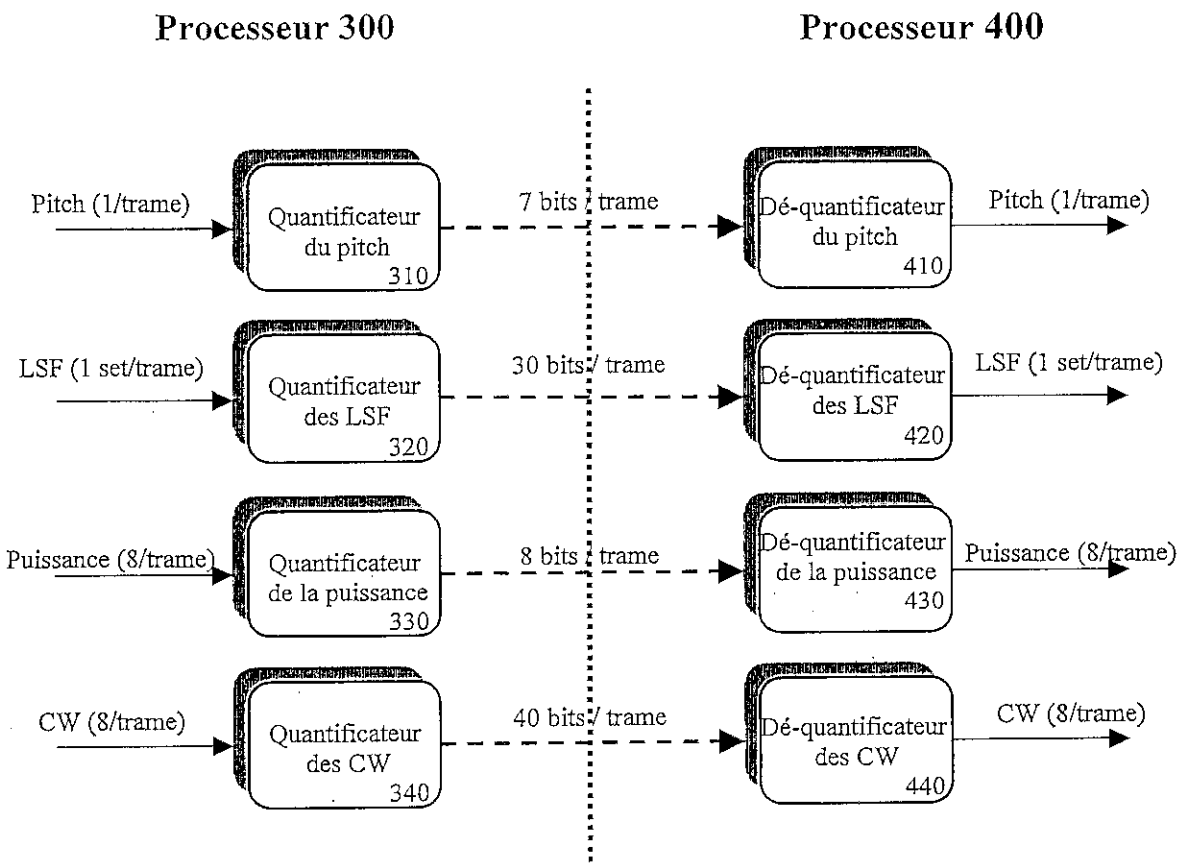

Fig. 4.1 Schéma bloc d'un quantificateur WI.

Dans cette mesure particulière de la distorsion, la pondération assignée à un coefficient LSF donné est proportionnelle à la sensibilité spectrale et la valeur de l'enveloppe spectrale à ce LSF. Les dictionnaires sont réalisés en utilisant l'algorithme de Lloyd généralisé (GLA) avec le critère de l'erreur quadratique moyenne (MSE) pour la mesure de la distorsion.

4.2 Quantification du Pitch

La fréquence de transmission de pitch est de 50 Hz (un par trame). Puisque l'estimateur de pitch (processeur **140**) fournit des valeurs entières, nous avons un total des 101 valeurs possibles ($P_{\max} - P_{\min} + 1$) qu'on peut coder à l'aide de 7 bits.

4.3 Quantification de la puissance

La quantification et la dé-quantification de la puissance sont réalisées par les processeurs **330** et **430** respectivement. Contrairement aux LSF, la puissance nécessite un traitement supplémentaire avant la quantification. La figure 4.2 montre les schémas blocs de **330** et **430**.

Etant donné que le logarithme du signal puissance est plus significatif que le signal puissance lui-même, les valeurs entrantes de la puissance sont d'abord, transformées au domaine logarithmique. Puis, elles sont filtrées passe - bas et sous - échantillonnées de 400 Hz à 100 Hz (2 valeurs / trame) [29]. Les valeurs échantillonnées sont codées par un quantificateur scalaire différentiel non adaptatif en utilisant un dictionnaire de 4 bits. Au récepteur, le signal puissance est décodé et sur - échantillonné à la fréquence de 400 Hz par interpolation dans le processeur **433**. C'est une interpolation linéaire exécutée directement sur les valeurs du logarithme de la puissance. Une fois le contour de la puissance est sur - échantillonné, le signal puissance est obtenu par l'opération exponentielle.

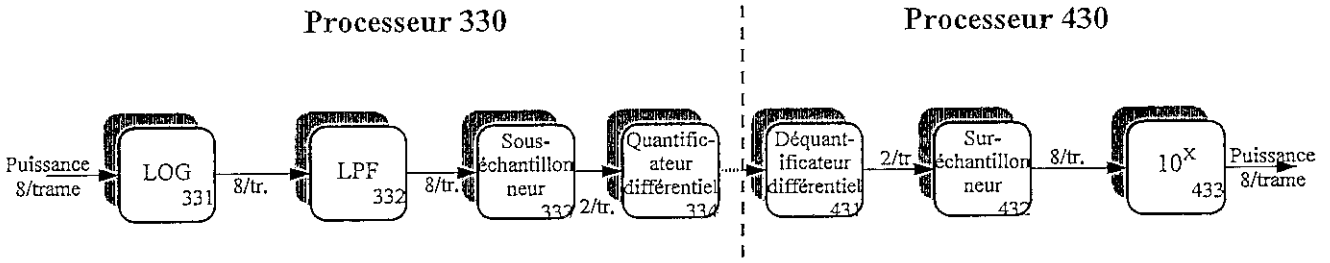

Fig. 4.2 Schémas des quantificateur et dé quantificateur de la puissance.

4.3.1 Conception du filtre passe-bas

L'objectif du filtrage passe – bas dans le processeur **332** est d'éviter le recouvrement spectral du sous – échantillonnage dans le processeur **333**. Puisque le facteur de sous – échantillonnage est de 4 (de 400 Hz à 100 Hz), la fréquence de coupure du filtre doit être de

50 Hz ou son équivalente normalisée 0,25. On utilise un filtre FIR anti-repliement de réponse impulsionnelle, notée $h_{Gain}(m)$, calculée par fenêtrage de la réponse d'un filtre passe-bas idéal (coupure à 50 Hz) avec une fenêtre de Hamming de longueur 17 échantillons. Finalement, on peut obtenir $h_{Gain}(m)$ en normalisant la réponse fenêtrée de sorte que :

$$\sum_{i=-8}^{8} h_{Gain}\ (i) = 1 \tag{4.1}$$

Les réponses en amplitude et en phase de h_{Gain} et sa réponse impulsionnelle sont tracées dans la figure 4.3.

Dans le processeur **332**, la procédure de filtrage passe – bas est réalisée dans le domaine temporel par une convolution linéaire qu'on peut exprimer par :

$$\tilde{\psi}_{\log}(kL_{sf}) = \sum_{i=-8}^{8} \psi_{\log}(kL_{sf} - iL_{sf}) h_{Gain}(i) \qquad \text{k=1, 2, ...} \tag{4.2}$$

où $\psi_{\log}(.)$ et $\tilde{\psi}_{\log}(.)$ représentent le contour du logarithme de la puissance et sa version filtrée passe – bas.

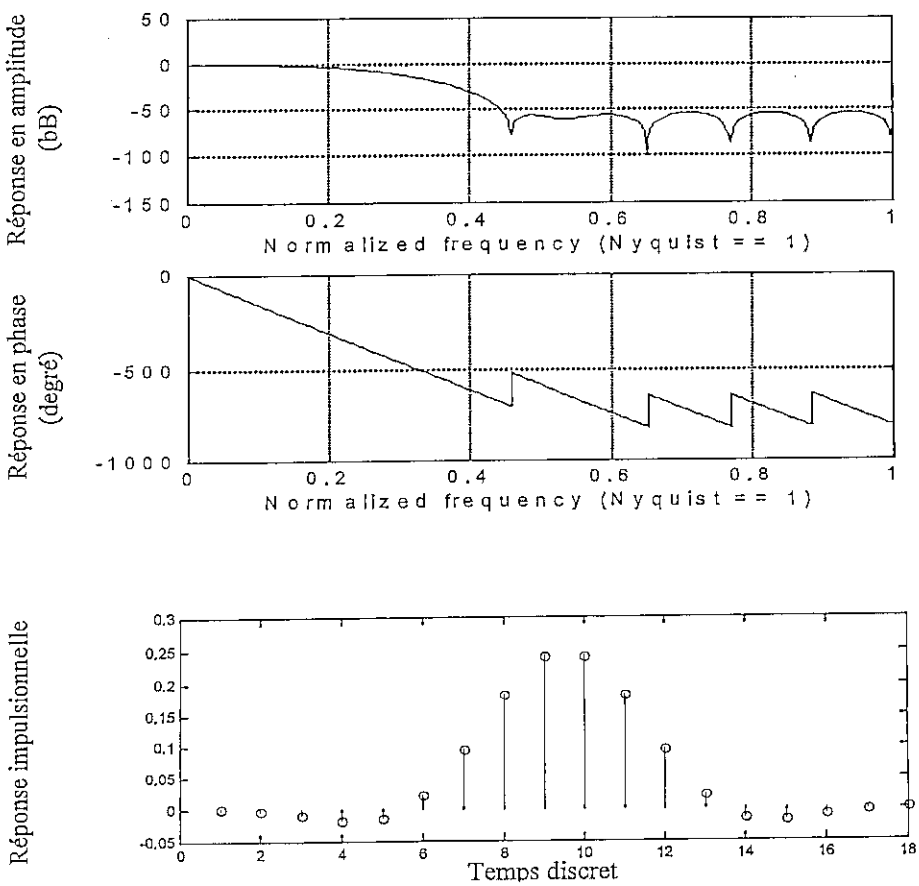

Fig. 4.3 Caractéristiques du filtre anti-repliement utilisé avant le sous-échantillonnage (processeur **332**). En haut : la réponse en amplitude du filtre. Au milieu : sa réponse en phase. En bas : sa réponse impulsionnelle $h_{Gain}(m)$. La fréquence de coupure normalisée du filtre est de égale à 0.25.

La nature non – causale du filtre impose l'utilisation de quelques échantillons futurs pour la procédure de convolution. Si on examine (4.2) de très près, on s'aperçoit que la convolution nécessite, réellement, huit échantillons de la trame future et huit échantillons de la trame passée pour le calcul des valeurs filtrées $\widetilde{\psi}_{\log}$ de la trame courante. Ceci est traduit par un retard algorithmique d'une trame (20 ms) dans le codeur. La figure 4.4 donne une illustration détaillée de cette nécessité des échantillons passés et futurs.

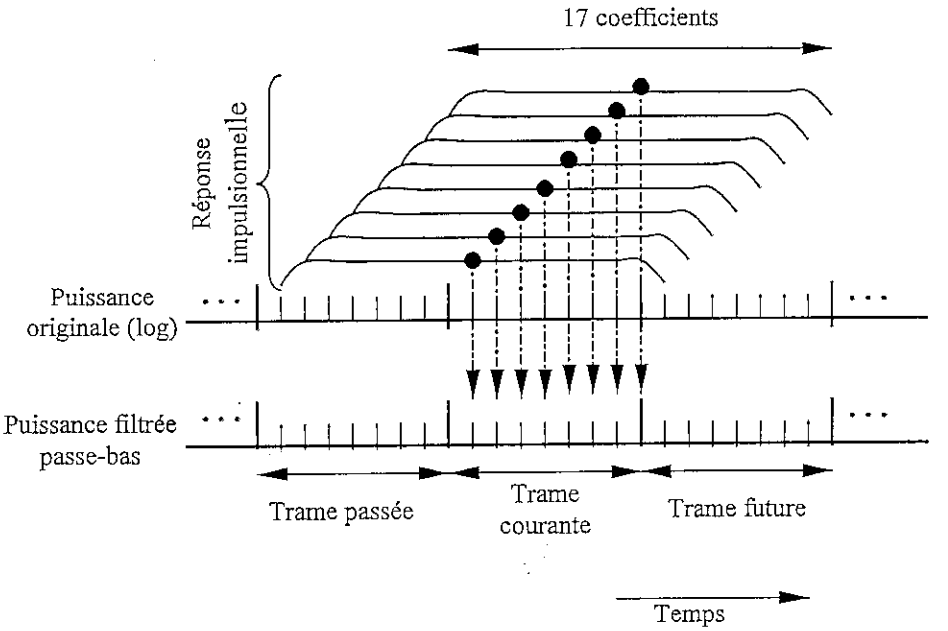

Fig. 4.4 Procédure de convolution pour le filtrage passe-bas du contour de puissance. Chaque point sur l'axe du temps représente une valeur $\psi_{\log}$ du logarithme de la puissance. Les points sur l'axe inférieur représentent les valeurs filtrées $\widetilde{\psi}_{\log}$ dans la trame.

Remarques sur la quantification de la puissance

- Comme on peut le voir sur la figure 4.4, un filtre FIR plus précis (plus de coefficients) nécessite plus d'échantillons futurs. Cela mène à un compromis entre la précision du filtre et le retard du codeur.
- Si la fréquence d'extraction des CW était plus grande, $\psi_{\log}(.)$ aurait une plus haute résolution. Donc, le filtre aura un plus grand nombre de coefficients sans ajouter de retard au codeur. Mais puisque l'augmentation de la fréquence d'extraction est associée à

l'augmentation de la complexité du codeur, la précision du filtre et la complexité du codeur forment un compromis.

- Puisque le processeur **333** effectue un sous – échantillonnage, on ne transmet que deux valeurs $\tilde{\psi}_{\log}$ dans chaque trame. En d'autres termes, le processeur **332** est sollicité pour calculer seulement deux valeurs filtrées par trames au lieu de huit.

4.4 Quantification des CW

Dans cette section, on va discuter le quantificateur (processeur **340**) et le dé–quantificateur (processeur **440**) des CW : Les figures 4.5a et 4.5b donnent les schémas blocs des deux processeurs. Les CW, comme la puissance, nécessitent un traitement supplémentaire avant la quantification. Plus précisément, chaque CW est décomposée en deux formes d'ondes qui seront quantifiées séparément. L'objectif et les détails de cette décomposition seront examinés dans le paragraphe suivant.

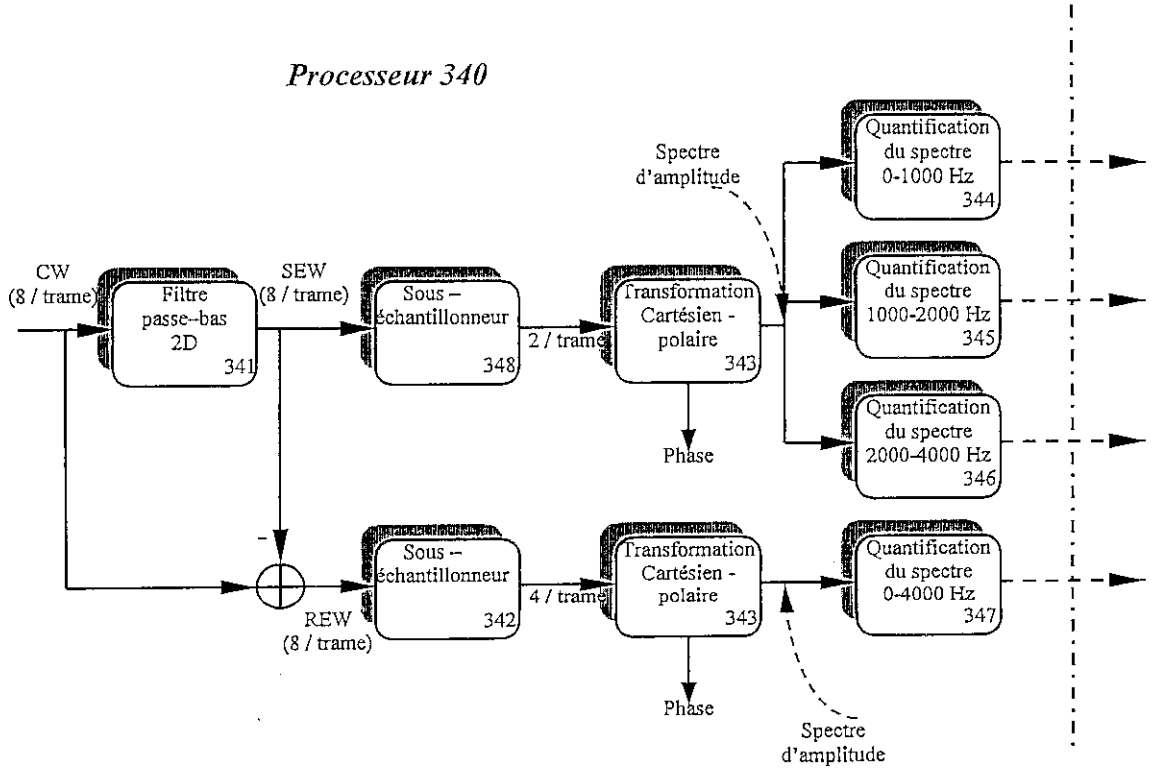

Fig. 4.5a Schéma bloc du quantificateur des CW

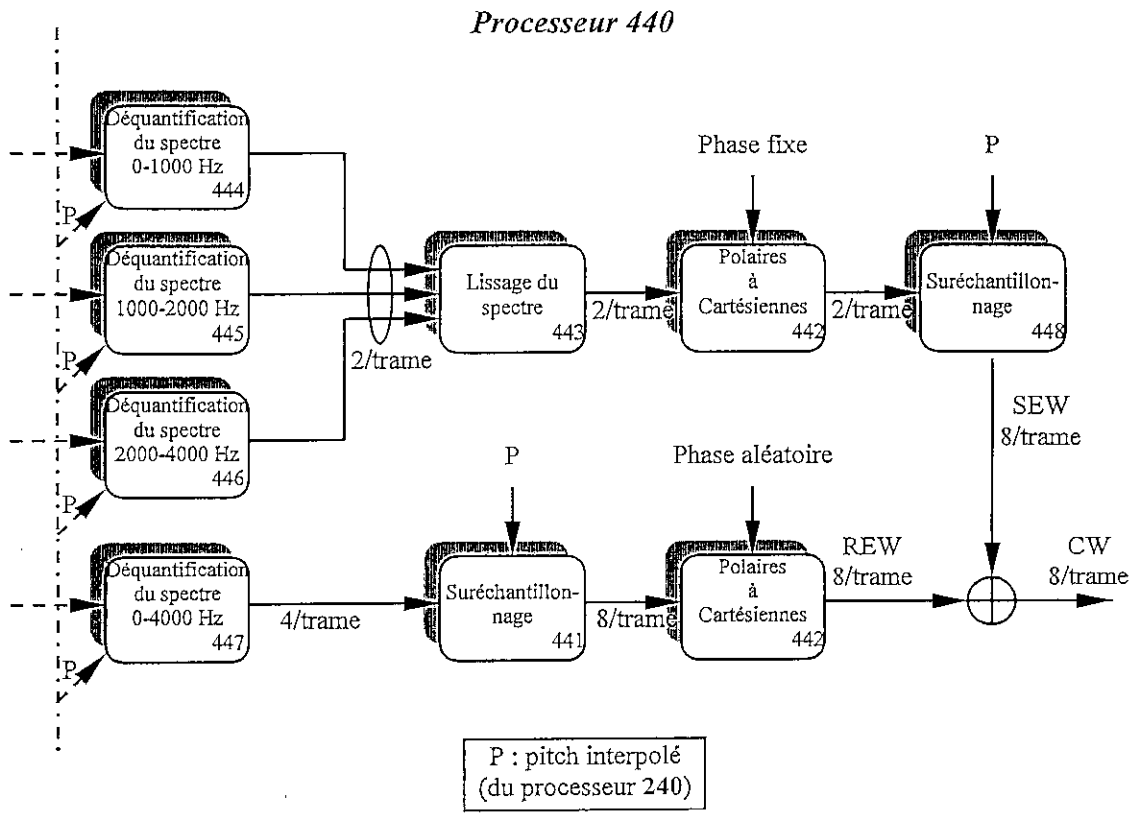

Fig. 4.5b Schéma bloc du dé-quantificateur des CW

4.4.1 Décomposition en SEW – REW

A première vue, il apparaît qu'une représentation précise des CW (une surface évoluant dans le temps) nécessite un débit de transmission très élevé, plus particulièrement pour les segments non voisés qui possèdent un plus grand débit d'information. Heureusement, l'oreille humaine n'est pas sensible à toute l'information contenue dans cette surface. Comme on l'a déjà mentionné au chapitre 1.4, la perception humaine des sons voisés est très différente de celle des sons non voisés, ce qui suggère la possibilité d'exploiter une telle différence pour quantifier les CW avec une meilleure précision du point de vue perception.

Au lieu d'adopter une classification voisé / non voisé, [29, 18, 19] proposent une nouvelle technique de décomposition dans laquelle chaque CW est séparée en deux composantes avant la quantification. Ces deux composantes sont : une forme d'onde à évolution lente (SEW : Slowly Evolving Waveform) et une forme d'onde à évolution rapide (REW : Rapidly Evolving Waveform) représentant les composantes périodique et non

périodique (bruit) du signal parole. En exploitant les différences dans la perception humaine de ces deux formes d'ondes, une meilleure efficacité de codage est possible en les quantifiant séparément.

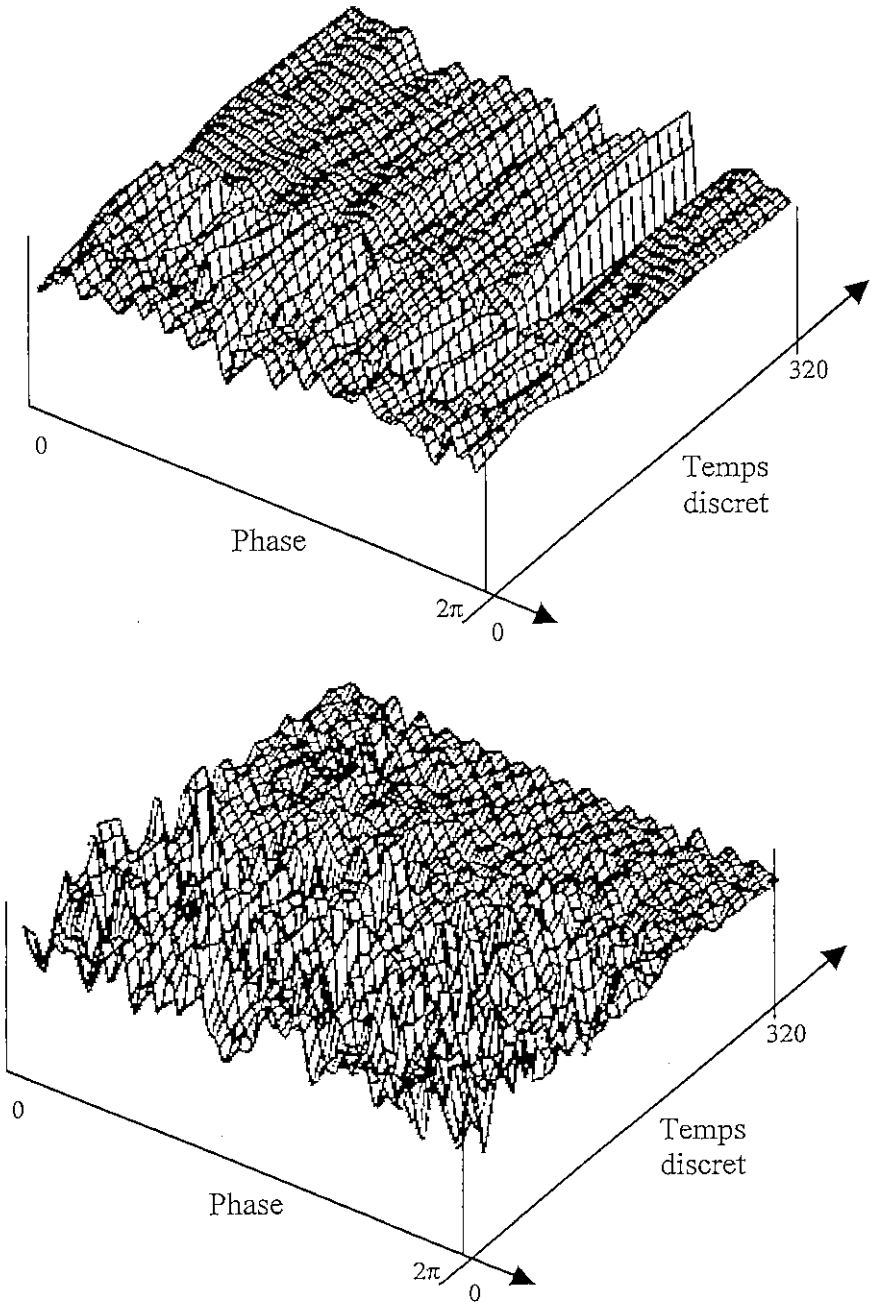

Fig. 4.6 Décomposition d'un segment de longueur 40 ms (320 échantillons) en surfaces SEW (en haut) et REW (en bas). La fréquence de coupure du filtre passe-bas est de 25 Hz.

La SEW est obtenue en filtrant passe – bas la surface des CW le long de l'axe du temps discret dans le processeur **341** et la REW peut être obtenue en retranchant la SEW de la CW.

Pour une parole voisée, le SEW et le REW représentent, respectivement, une forme d'onde ressemblant à une impulsion et une composante de bruit. Vu la présence de périodicité dans les régions voisées, la SEW possède, généralement, un niveau d'énergie plus élevé que la REW. Inversement, pour la parole non voisée où le signal évolue plus rapidement et où il n'y a aucune périodicité apparente, la décomposition distribue la plus grande partie de l'énergie de la CW à la REW. La figure 4.6 illustre un exemple de décomposition en deux surfaces SEW et REW.

Pour éviter d'introduire d'avantage de retard au codeur, le filtre passe – bas dans le processeur **341** utilise le même nombre de coefficients (17 coefficients). C'est aussi un filtre non causal à phase linéaire. Cependant, ce filtre nécessite une fréquence de coupure de 25 Hz équivalente à la fréquence normalisée 0.125. Sa réponse impulsionnelle, notée $h_{CW}(m)$, est calculée de la même manière que h_{Gain}.

La figure 4.7 trace la réponse en amplitude et en phase de $h_{CW}(m)$ et sa réponse impulsionnelle. Notons que la réponse en fréquence possède une bande de transition assez large. Ceci est dû, principalement, au fait que le filtre FIR possède seulement 17 coefficients. On peut augmenter le nombre de coefficients pour avoir une meilleure précision du filtre, mais aux dépends d'un retard algorithmique plus important.

Sachant que la transformation en DTFS est une opération linéaire, le filtrage passe – bas des CW dans le temps est équivalent au filtrage passe – bas de leurs coefficients DTFS. Pour cette raison, le processeur **341** réalise le filtrage directement sur les coefficients A_k et B_k (pour tout k). De manière plus précise, pour calculer la CW filtrée passe – bas à l'instant n, on peut utiliser la formule suivante :

$$\left.\begin{aligned} \widetilde{A}_k(n) &= \sum_{i=-8}^{8} A_k(n - iL_{sf})h_{CW}(i) \\ \widetilde{B}_k(n) &= \sum_{i=-8}^{8} B_k(n - iL_{sf})h_{CW}(i) \end{aligned}\right\} \quad pour \quad k = 1,2,...,\lfloor P(n)/2 \rfloor \qquad (4.3)$$

Puisque les CW ont un débit de 400 Hz, l' indice n de temps discret doit être multiple de L_{sf}. Les $\widetilde{A}_k(n)$ et $\widetilde{B}_k(n)$ sont les coefficients DTFS de la CW filtrée passe-bas (SEW) à l'instant n.

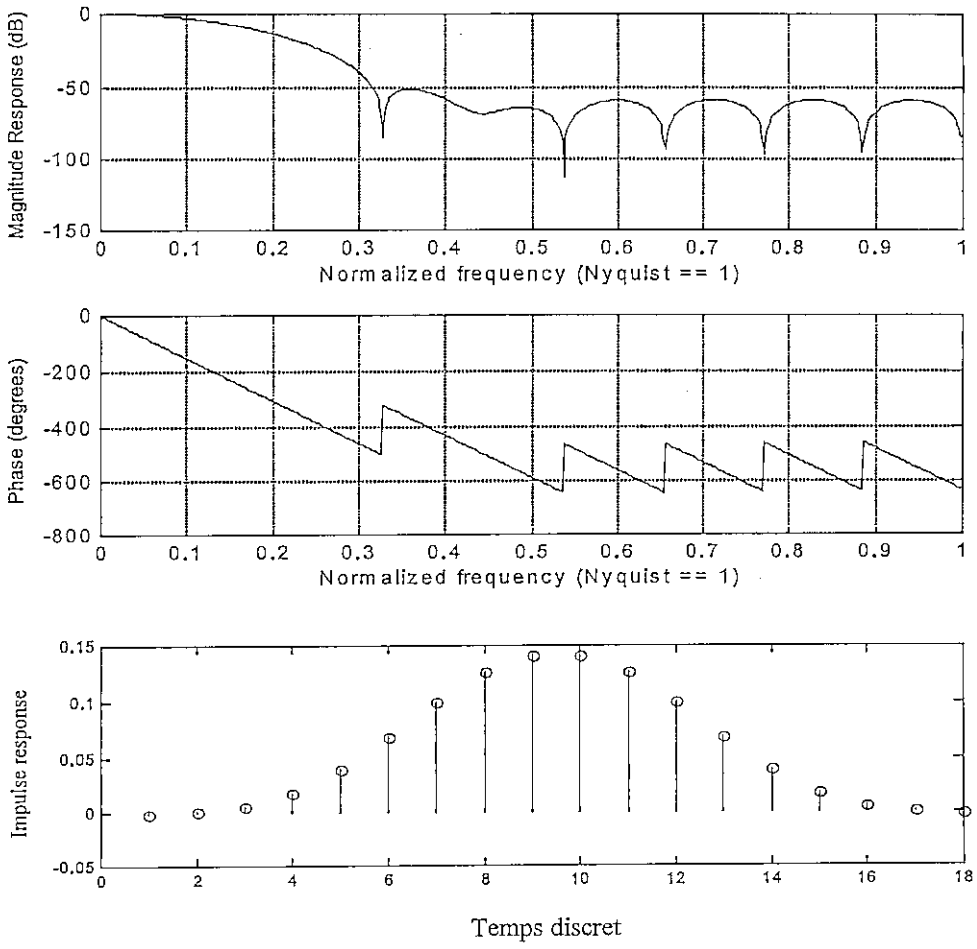

Fig. 4.7 Caractéristiques du filtre passe-bas de décomposition en SEW-REW. En haut : sa réponse en amplitude. Au milieu : sa réponse en phase. En bas : sa réponse impulsionnelle $h_{CW}(m)$. La fréquence de coupure normalisée est égale à 0.125.

Or, la dimension des CW varie avec le pitch. Pour faciliter le filtrage, les mêmes techniques du paragraphe 3.4.5 sont utilisées pour allonger ou contracter les CW de manière à ce que toutes les CW à l'intérieur de le fenêtre de filtrage aient la même longueur avant le filtrage. Rappelons que la première opération correspond à remplir par des zéros (zero padding) dans le domaine DTFS et la seconde correspond à la troncature spectrale suivie d'un ajustement de la puissance (pour compenser la perte en puissance). L'opération de filtrage avec ses pré–traitements est illustrée par un exemple dans la figure 4.8.

Remarques sur la décomposition SEW – REW

Comme illustré dans la figure 4.6, l'énergie des REW domine clairement dans les régions non voisées, alors que les SEW dominent dans les régions voisées. En fait, on peut

utiliser cette propriété pour obtenir une estimation approximative sur l'information de voisement (détecteur de voisement). Le degré de voisement est proportionnel au rapport d'énergie SEW/CW (ou inversement proportionnel au rapport REW/CW).

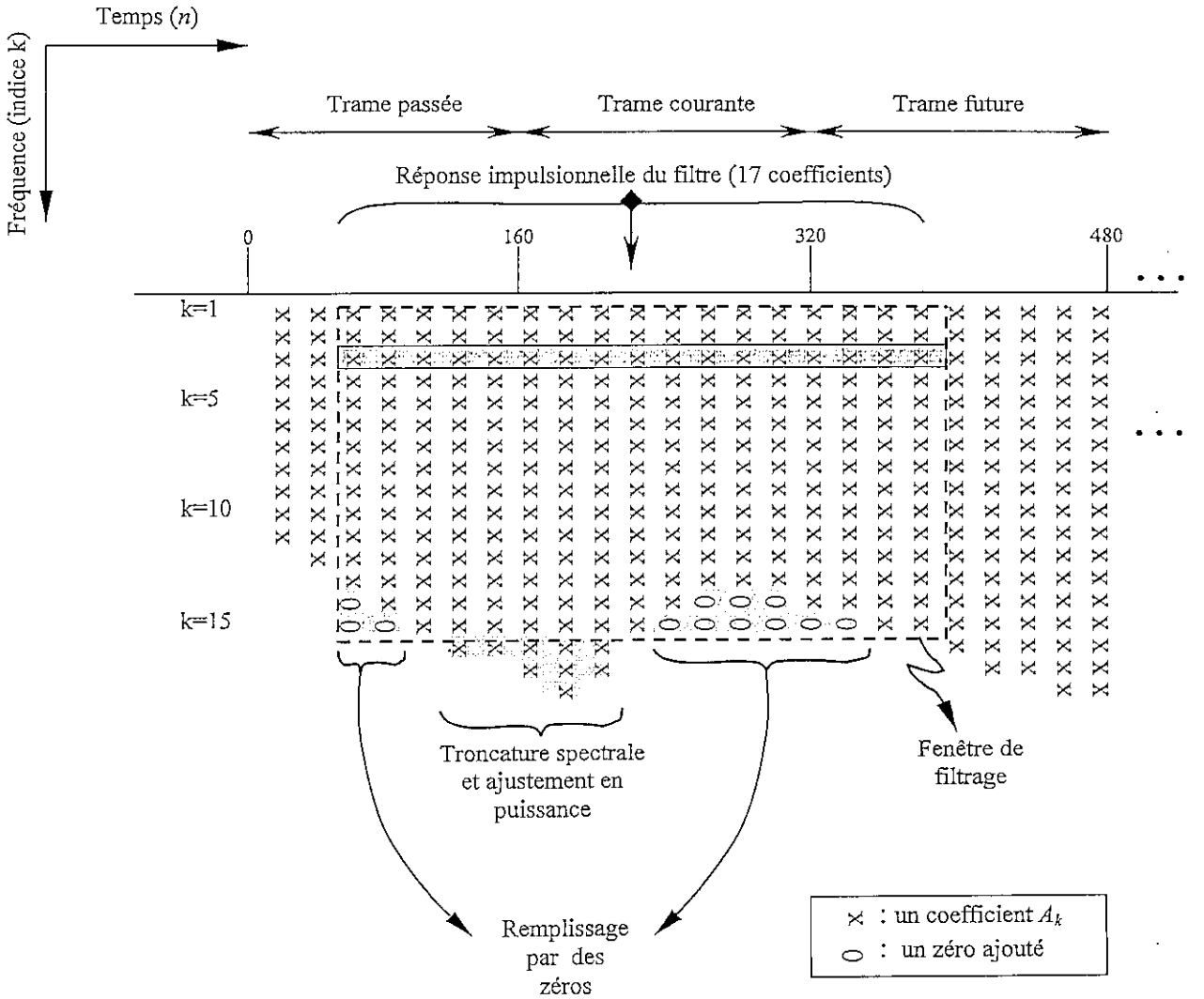

Fig. 4.8 Opération de filtrage passe-bas pour la décomposition en SEW-REW. Le schéma montre 24 CW successives couvrant trois fenêtres. Le filtre est centré à la CW du point 220. Puisque la longueur originale de cette CW est de k = 15, toutes les CW dans la fenêtre de filtrage doivent avoir la même longueur avant le filtrage. Les plus courtes CW de longueur < 15 seront étendues par ajout de zéros tandis que les plus longues (longueur > 15) seront tronquées puis ajustées en puissance. Ensuite, la procédure de filtrage est exécutée k-par-k (ligne par ligne). La ligne ombrée montre les coefficients impliqués dans le filtrage pour k = 3. La valeur filtrée passe-bas résultante sera $\widetilde{A}_3(220)$.

4.4.2 Quantification des REW

Dans ce paragraphe, on va focaliser sur la quantification des REW de manière très précise du point de vue perception.

Commençons d'abord par lister trois conclusions importantes des expériences faites dans [18]:

1. Une faible dégradation dans la qualité de la parole est observée si le spectre de phase des REW est remplacé par un spectre de phase aléatoire.
2. Aucune détérioration n'est observée dans la parole résultante si chaque spectre d'amplitude d'une REW est lissé par une fenêtre carrée de 1000 Hz.
3. Une très petite détérioration audible est produite si le spectre d'amplitude d'une REW est moyenné sur tous les REW dans un intervalle de 5 ms.

La première conclusion montre que le spectre des REW comporte quelques informations perceptibles et ne doit pas être transmis avec un faible débit. Les deuxième et troisième impliquent que la résolution dans le temps du spectre d'amplitude des REW est, nettement, plus importante que sa résolution en fréquence. En d'autre termes, les REW nécessitent un débit élevé et une technique de quantification grossière.

Pour exploiter ces résultats, le processeur **342** sous - échantillonne la REW entrante à un débit de 200 Hz qui est en accord avec la résolution dans le temps suggérée par la troisième conclusion (intervalle de 5 ms). Chaque REW sous - échantillonnée est, alors, convertie vers sa représentation polaire où le spectre de phase est complètement écarté. Le spectre d'amplitude est quantifié vectoriellement en utilisant un dictionnaire de huit vecteurs. On utilise un dictionnaire de taille aussi petite car une description grossière du spectre d'amplitude des REW est suffisante pour avoir une bonne qualité de codage selon la deuxième conclusion. L'indice du vecteur résultant est transmis et la consommation totale de bits par REW est égale à :

$$200 \times 3 = 600 \text{ bits/s} = 12 \text{ bits/trame.}$$

Au récepteur, les spectres des REW sont décodés et sur - échantillonnés par un facteur de 2 dans le processeur **441** du débit 200 Hz à 400 Hz (même débit que celui des CW). On réalise cela en insérant un nouveau spectre après chaque spectre reçu. Ces nouveaux spectres sont obtenus par interpolation linéaire des spectres adjacents ou en choisissant le spectre précédent. Des expériences ont indiqué qu'il n'y a aucune différence perceptible entre les résultats de ces deux méthodes. Finalement, chaque spectre d'amplitude d'une REW sur - échantillonnée est combiné avec un spectre de phase aléatoire et puis reconverti en

coordonnées rectangulaires. Les valeurs dans les spectres de phase sont indépendantes et uniformément réparties dans [0 , 2π]. Il est à noter que les spectres de phase aléatoire sont ajoutés aux REW à la fréquence des sous - trames.

Recherche et conception du dictionnaire des spectres des REW

La dimension du spectre d'amplitude des REW est proportionnelle à la période du pitch. Par conséquent, le spectre a une dimension variable et doit être décrit à l'aide d'un quantificateur vectoriel à dimension variable (VDVQ) approprié. Rappelons que, dans notre implémentation, le pitch varie de 20 à 120 échantillons, ce qui est équivalant à 10 à 60 harmoniques dans le spectre de la REW (la composante DC étant exclue).

Le dictionnaire des REW est conçu par une méthode appelée quantification vectorielle à dimension variable (VDVQ) [5]. Elle set basée sur la supposition que la génération d'un vecteur à dimension variable est le résultat d'un échantillonnage uniforme d'un autre vecteur à dimension fixe et large. Cette technique fonctionne comme suit.

Avant la formation (training) du dictionnaire, chaque spectre REW dans la séquence d'entraînement est, d'abord, interpolé à bande limitée en un vecteur à dimension fixe. Le choix naturel de cette dimension est le nombre maximal d'harmoniques dans le spectre (60 harmoniques). Une fois que tous les vecteurs d'entraînement sont convertis à la même dimension, on applique la technique GLA conventionnelle pour former le dictionnaire. Par conséquent, le dictionnaire résultant aura la dimension uniforme de 60.

Pendant le codage d'un spectre dans **347**, le dictionnaire est, avant tout, sous - échantillonné pour avoir la longueur du spectre donné. Après, le vecteur le plus proche est retrouvé à l'aide du critère MSE et son indice dans le dictionnaire est transmis au récepteur.

Après réception de l'indice dans le processeur **447**, le spectre de la REW est reconstitué en sous - échantillonnant le spectre quantifié. Le facteur de sous - échantillonnage dépend, évidemment, de la valeur du pitch fournie par le processeur **240**; le pitch détermine le nombre d'harmoniques dans le spectre. La taille du dictionnaire des REW est de huit seulement. Les formes du spectre sont lissées et peuvent être approximées par des polynômes d'ordre réduit [20]. Une telle représentation polynomiale réduit considérablement la taille de la mémoire consommée par le dictionnaire puisque, pour chaque mot codé, on stocke seulement quelques coefficients du polynôme au lieu du spectre entier (60 harmoniques). La correspondance entre un spectre et un polynôme est réalisée au sens de moindres carrés et il a été prouvé qu'un polynôme d'ordre cinq est adéquat pour la représentation d'un spectre REW.

4.4.3 Quantification des SEW

Puisque la fréquence de coupure du filtre de décomposition est égale à 25 Hz seulement, les SEW ont une largeur de bande d'évolution très petite (leur évolution est très lente). Cela suggère qu'on peut les sous - échantillonner de 400 Hz à environ 50 Hz. Cependant il est plus avantageux de les sous - échantillonner à une fréquence un peu plus grande de 100 Hz (deux SEW par trame) afin de compenser l'imprécision du filtre de décomposition dans le processeur **341**.

Chaque SEW sous - échantillonnée est convertie en notation polaire dont on écarte le spectre de phase. Le spectre d'amplitude est divisé en trois sous - bandes sans recouvrement : 0 - 1000 Hz, 1000 - 2000 Hz, 2000 - 4000 Hz . Ces sous - bandes sont quantifiées séparément où la bande de base est quantifiée avec 8 bits tandis que les deux autres sous - bandes sont quantifiées avec 3 bits chacune. Une telle allocation de bits est due à la grande capacité de résolution de l'oreille humaine pour les basses fréquences [21]. La consommation totale de bits pour chaque SEW est de 14 bits et le débit de transmission est :

$$100 \times 14 = 1400 \text{ bits/s} = 28 \text{ bits / trame.}$$

Au récepteur, après décodage et combinaison des sous - bandes, on applique une interpolation linéaire pour ajuster le spectre combiné aux extrémités des sous - bandes (c.a.d. à 1000 Hz et 2000 Hz) dans le processeur **443**. Un changement brusque ou une discontinuité importante dans le spectre peut causer des réverbérations dans la parole reconstituée.

Après avoir reconstitué et lissé le spectre d'amplitude, on lui associe un spectre de phase fixe et on le retransforme en coordonnées rectangulaires. Ce spectre de phase fixe est donné à partir d'un segment voisé d'une voix d'homme à pitch élevé (maximum d'harmoniques). Apres quoi, les SEW sont sur - échantillonnées dans le processeur **449** du débit 100 Hz à 400 Hz (le même que celui des REW reconstituées). Puisque ces SEW peuvent avoir des longueurs différentes (nombres différents de coefficients A_k et B_k), la procédure du scénario 2 du paragraphe 3.5.2 est adoptée pour l'accomplissement de ce sur - échantillonnage.

Recherche et conception du dictionnaire des SEW

De même que les REW, la dimension des SEW varie selon le pitch. Par conséquent, les procédures de recherche et de conception du dictionnaire des SEW sont similaires à celles des REW.

Les spectres d'amplitude des SEW dans la séquence d'entraînement sont, d'abord, interpolés à bande limitée à la longueur de 60 qui correspond au nombre maximum d'harmoniques. Chaque vecteur d'entraînement est divisé en trois sous - bandes. Les premier 25% des harmoniques (15 harmoniques) constituent la bande de base. Les 25% suivants constituent la deuxième sous - bande et les harmoniques restants constituent la dernière sous - bande qui correspond à la gamme de fréquence 2000 – 4000 Hz. Pour former les dictionnaires des trois sous - bandes, on adopte la technique GLA.

Quant aux procédures de recherche des dictionnaires, elles sont identiques à celles des REW à l'exception de celle concernant la bande de base qui est basée sur le critère de l'erreur modérée par perception (perceptually weighted error). Ce critère est très utilisé dans les codeurs basés sur la technique CELP et sera discuté dans le paragraphe suivant.

Critère de l'erreur modérée par perception

Dans le processeur **344**, le spectre SEW de la bande de base est sélectionné à partir d'un dictionnaire en minimisant l'erreur modérée par perception entre le spectre original et le spectre quantifié. La modération par perception est calculée à partir de la structure des formants du signal parole de manière à permettre plus de bruit de quantification dans les régions des formants que dans les vallées entre formants. Cela sert à exploiter la propriété de masquage spectral dans le système auditif humain. Puisque notre CW est définie dans le domaine résiduel, la modération peut être induite dans le filtre de synthèse :

$$H_w(z) = \frac{1}{1 - \sum_{k=1}^{N} a_k \gamma_w^k z^{-k}} \qquad 0 < \gamma_w \leq 1 \qquad (4.4)$$

où γ_w est le facteur de modération et est typiquement égal à 0.8.

De plus, les a_k sont les coefficients LP interpolés non quantifiés issus du processeur **120**.

Ainsi, le spectre de l'erreur modérée par perception dans **344** peut être obtenu en multipliant la réponse en amplitude $|H_w(z)|$ par le spectre de l'erreur qui est égal au carré de la différence entre les spectres original et quantifié.

Le tableau 4.1 donne l'allocation de bits pour un codeur WI travaillant avec un débit de 4.25 kbps dont les paramètres sont quantifiés selon la description présentée dans ce chapitre.

Tableau 4.1 *Allocation de bits d'un codeur WI à 4.25 kbps*

Paramètres WI	*Bits par paramètre*	*Fréquence de calcul du paramètre (Hz)*	*Bits par trame*	*Bits par seconde*
Pitch	7	50	7	350
Puissance	4	100	8	400
LSF	30	50	30	1500
SEW (amplitude)	14	100	28	1400
REW (amplitude)	3	200	12	600
SEW (phase)	0	100	0	0
REW (phase)	0	400	0	0
		Débit binaire total :	**85**	**4250**

Chapitre 5

CONCLUSIONS

Un codeur WI a été simulé par un programme en langage C.
La qualité de la parole résultante a été évaluée par le test subjectif (test d'écoute) utilisant la mesure du MOS (Mean Opinion Score).
Dans ce chapitre, on va résumer le travail réalisé et revoir quelques propriétés importantes de la technique de codage WI.

Résumé du travail réalisé

Dans le premier chapitre, on a fourni quelques informations de base sur le codage de la parole telles que les propriétés du signal parole et les composants de base des codeurs de la parole. l'objectif et le plan de notre travail ont été mentionnés à la fin du chapitre. Le but principal était de développer un codeur de la parole basé sur la technique WI avec l'espoir d'atteindre la qualité grand public (toll quality) à des débits autour de 4 kbps.

Le chapitre 2 donne un rappel bref sur le codage par l'analyse LP à court- terme. On a introduit, aussi, les concepts des paires de raies spectrales (LSF ou LSP), l'expansion de la largeur de bande et la pré-accentuation.

Dans le chapitre 3, on commence par introduire les origines et le concept du schéma de codage WI. On a donné un exposé approfondi sur l'implémentation de l'algorithme avec tous les détails concernant la couche d'analyse- synthèse (c. à. d. le model non quantifié), les calculs mathématiques pour chaque processeur étant formulés.

L'étage d'analyse décompose une trame de parole en quatre paramètres qui sont : les coefficients LSF, le pitch, la puissance et les CW. L'étage de synthèse reconstruit le signal parole à partir de ces paramètres. Les deux étages ont été conçus de manière à ne pas souffrir des apparitions de multiples ou sous-multiples de pitch.

La précision de la couche d'analyse - synthèse à été vérifiée par un test d'évaluation subjectif. La qualité de la parole reconstituée à été jugée bonne en moyenne sur plusieurs phrases (voix féminines et masculines) prononcées dans différents milieux (silence, bruit de voiture, bruit de la foule). Ce résultat est équivalent à la note 4/5 avec 1/5 pour « qualité mauvaise » et 5/5 pour « qualité excellente ».

Dans le chapitre 4, on a présenté un schéma de quantification pour le codeur WI visant un débit d'environ 4 kbps. On a décrit le schéma de quantification de chaque paramètre. L'objectif et l'implémentation de la décomposition en SEW- REW ont été donnés.

Puissance de la technique WI

Comparativement parlant, le codeur WI offre plusieurs qualités non communes aux codeurs à débit réduit conventionnels. Quelques unes de ces qualités sont :

- le succès du schéma WI est dû, en grande partie, à sa capacité de produire un niveau précis de périodicité pour les sons voisés, même à des débits très réduits. A l'inverse des codeurs basés sur la technique CELP qui ne peuvent maintenir la périodicité appropriée en travaillant au même débit.
- le model non quantifié du codeur WI produit une qualité de parole presque transparente. En d'autres termes, la performance du codeur WI est limitée par les quantificateurs et non par le model.
- un avantage important qu'offre la technique WI est qu'elle décompose la parole en paramètres relativement découplés (les coefficients LP, le pitch, la puissance, les SEW et les REW). Une telle indépendance permet une

quantification plus efficace des paramètres. Elle permet, aussi, aux paramètres d'être manipulés et contrôlés séparément. En fait, c'est cette indépendance qui rend possible les modifications de la longueur des CW (time - scaling) dans la WI (paragraphe 3.9) .

- le codage WI évite la classification binaire voisé / non voisé (V/UV). Ceci permet plus de robustesse aux milieux bruyants et aux erreurs du canal.
- en réalité, toutes les trames ne sont pas complètement voisées ou non voisées ; quelques trames contiennent les deux types de sons. Les codeurs à débit réduit traditionnels qui utilisent la classification V/UV ne sont pas capables de faire face à ces situations. Cependant, la décomposition en SEW / REW donne à la WI la possibilité de prendre en charge de tels segments du signal parole. Par conséquent, le codeur WI se comporte de manière plus robuste, plus particulièrement, quand le signal parole est affecté de bruits acoustiques environnants.
- bien que cela ne soit pas le but principal du codeur WI, il peut donner une information sur le voisement du signal parole (un détecteur de voisement). On peut avoir cette information grâce au rapport d'énergie entre la REW (ou la SEW) et la CW.

Appendice A

Constantes utilisées dans le codeur WI

Table A.1 Constantes utilisées dans la simulation

Symbole	Valeur	Description
L_f	160	Longueur de la trame
L_{sf}	20	Longueur de la sous - trame $= L_f \div R_{extr}$
N	10	Ordre du filtre LP
γ	0.98829	Facteur d'extension de la largeur de bande du filtre LP
L_w	240	Longueur de la fenêtre d'analyse LP
$P_{\min}$	20	Valeur minimale du pitch
$P_{\max}$	120	Valeur maximale du pitch
R_{extr}	8	Nombre d'extractions de CW par trame
δ	10	Longueur de la fenêtre d'énergie à chaque extrémité de la CW
α	0.1	Facteur de pré - accentuation
$\varepsilon_{\max}$	16	Décalage maximal du point d'extraction

Bibliographie

[1] L. R. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals*. Prentice-Hall, 1978.

[2] V. C. Welch and T. E. Tremain, " A new government standard 2400 bps speech coder," *Proc. IEEE Workshop on Speech Coding for Telecom.* (Sainte-Adèle, Québec), pp. 41-42, Oct. 1993.

[3] A. Gersho and R. M. Gray, Vector Quantization and Signal Compression. Kluwer Academic Publishers, 1992.

[4] R. M. Gray, " Vector quantization," IEEE Trans. Acoustics, Speech, Signal Processing, vol. ASSP-34, Apr. 1984.

[5] A. Das, A. V. Rao, and A. Gersho, " Variable-dimension vector quantization," IEEE Signal Processing Letters, vol. 3, pp. 200-202, July 1996.

[6] D. O'Shaughnessy, Speech Communication: Human and Machine. Addison-Wesley Publishing Company, 1987.

[7] W. B. Kleijn, " Continuous representations in linear predictive coding," Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing (Toronto), pp. 201-204, May 1991.

[8] W. B. Kleijn and W. Granzow, " Methods for waveform interpolation in speech coding," Digital Signal Processing, vol. 1, pp. 215-230, Jan. 1991.

[9] W. B. Kleijn, " Encoding speech using prototype waveforms," IEEE Trans. Speech and Audio Processing, vol. 1, pp. 386-399, Oct. 1993.

[10] A. S. Spanias, " Speech coding: A tutorial review," Proc. IEEE, vol. 82, pp. 1541-1582, Oct. 1994.

[11] G. Kubin, B. S. Atal, and W. B. Kleijn, " Performance of noise excitation for unvoiced speech," Proc. IEEE Workshop on Speech Coding for Telecom. (Sainte-Adèle, Québec), pp. 35-36, Oct. 1993.

[12] I. A. Atkinson, A. M. Kondoz, and B. G. Evans, " Time envelope vocoder, a new LP based coding strategy for use at bit rates of 2.4 kb/s and below," IEEE J. Selected Areas Commun., vol. 13, pp. 449-457, Feb. 1995.

[13] I. A. Atkinson, A. M. Kondoz, and B. G. Evans, " Time envelope LP vocoder : A new coding technique at very low bit rates," Proc. European Conf. on Speech Commun. and Technology (Madrid), pp. 241-244, Sept. 1995.

[14] 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 J. Selected Areas Commun., vol. 10, pp. 830-849, June 1992.

[15] R. Salami, C. Laflamme, J.-P. Adoul, A. Kataoka, S. Hayashi, C. Lamblin, D. Massaloux, S. Proust, P. Kroon, and Y. Shoham, " Description of the proposed ITU-T 8-kb/s speech coding standard," Proc. IEEE Workshop on Speech Coding for Telecom. (Annapolis), pp. 3-4, Sept. 1995.

[16] A. McCree, K. Truong, E. B. George, T. P. Barnwell III, and V. Viswanathan, " A 2.4 kbit/s MELP coder candidate for the new U.S. Federal Standard," Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing (Atlanta), pp. 200-203, May 1996.

[17] W. B. Kleijn, Analysis-by-Synthesis Speech Coding Based on Relaxed Waveform Matching Constraints. PhD thesis, Delf University of Technology, Delf, The Netherlands, Dec.1991.

[18] W. B. Kleijn and J. Haagen, " A general Waveform-Interpolation structure," Proc. European Signal Processing Conf. (Edinburg), pp. 1665-1668, Sept. 1994.

[19] W. B. Kleijn and J. Haagen, " Speech coder based on decomposition of characteristic waveforms," Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing (Detroit), pp. 508-511, May 1995.

[20] W. B. Kleijn, Y. Shoham, D. Sen, and R. Hagen, " A low-comlexity Waveform Interpolation coder," Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing (Atlanta), pp. 212-215, May 1996.

[21] J. Thyssen, W. B. Kleijn, and R. Hagen, " Using a perception-based frequency scale in Waveform Interpolation," Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing (Munich, Germany), pp. 1595-1598, Apr. 1997.

[22] H. Yang and W. B. Kleijn, " Pitch-synchronous subband representation of the linear prediction residual of speech," Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing (Seattle), pp. 529-532, May 1998.

[23] K. K. Paliwal and B. S. Atal, " Efficient vector quantization of LPC parameters at 24 bits/frame," IEEE Trans. Speech and Audio Processing, vol. 1, pp. 3-14, Jan. 1993.

[24] G. H. Golub and C. F. V. Loan, Matrix Computations. The John Hopkins University Press, second ed., 1989.

[25] W. B. Kleijn and K. K. Paliwal, eds., Speech Coding and Synthesis. Elsevier, 1995.

[26] F. Itakura, " Line spectrum representation of linear prediction coefficients of speech signals," Journal Acoustical Society America, vol. 57, p. 535, 1975. (abstract).

[27] P. Kabal and R. P. Ramachandran, " The computation of line spectral frequencies using Chebyshev polynomials," IEEE Trans. Acoustics, Speech, Signal Processing, vol. ASSP-34, pp. 1419-1426, Dec. 1986.

[28] J. R. Deller Jr., J. G. Proakis, and J. H. L. Hansen, Discrete-Time Processing of Speech Signal. Macmillan, 1993.

[29] W. B. Kleijn and J. Haagen, " Transformation and decomposition of the speech signal for coding," IEEE Signal Processing Letters, vol. 1, pp. 136-138, Sept. 1994.

[30] W. B. Kleijn and J. Haagen, " Waveform interpolation for coding and synthesis," in Speech Coding and Synthesis (W. B. Kleijn and K. K. Paliwal, eds.), pp. 175-208, Elsevier, 1995.

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

[32] Telecommunications Industry Association, TIA/EIA/PN-3292, EIA/TIA Interim Standard, Enhanced Variable Rate Codec (EVRC), Mar. 1996.

[33] J. Stachurski, A Pitch Pulse Evolution Model for Linear Predictive Coding of Speech. PhD thesis, McGill University, Montreal, Canada, May 1997.

[34] M. Leong, " Representing voiced speech using protoype waveform interpolation for low rate speech coding," Master's thesis, McGill University, Montreal, Canada, Nov. 1992.

[35] M. Leong and P. Kabal, " Smooth speech reconstruction using Prototype Waveform Interpolation," Proc. IEEE Workshop on Speech Coding for Telecom. (Sainte-Adèle, Québec), pp. 39-41, Oct. 1993.

[36] K. Yaghmaie and A. M. Kondoz, " Multiband prototype waveform analysis-synthesis for very low bit rate speech coding," Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing (Munich, Germany), pp. 1571-1574, Apr. 1997.

[37] D. Marston and F. Plante, " PWI speech coder in the speech domain," Proc. IEEE Workshop on Speech Coding for Telecom. (Pennsylvania), pp. 31-32, Sept. 1997.

[38] J. Stachurski and P. Kabal, " A pitch pulse evolution model for a dual excitation linear predictive speech coder," Proc. Seventeenth Biennial Symposium on Communications (Kingston), pp. 107-110, May 1994.

[39] D. W. Griffin and J. S. Lim, " Multiband excitation vocoder," IEEE Trans. Acoustics, Speech, Signal Processing, vol. 36, pp. 1223-1235, Aug. 1988.

[40] J. C. Hardwick and J. S. Lim, " A 4800 bps improved multi-band excitation speech coder," Proc. IEEE Workshop on Speech Coding for Telecom. (Vancouver), Sept. 1989.

[41] Y. Shoham, " Low-rate speech coding based on time-frequency interpolation," Proc. Int. Conf. on Spoken Language Processing, pp. 37-40, Oct. 1992.

[42] A. McCree and W. B. Kleijn, " Mixed Excitation Prototype Waveform Interpolation for low bit rate speech coding," Proc. IEEE Workshop on Speech Coding for Telecom. (Sainte-Adèle, Québec), pp. 51-52, Oct. 1993.

[43] Y. Tanaka and H. Kimura, " Low-bit-rate speech coding using a two-dimensional transform of residual signals and Waveform Interpolation," Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing (Adelaide), pp. 173-176, Apr. 1994.

[44] D. Sen and W. B. Kleijn, " Synthesis methods in sinusoidal and Waveform-Interpolation coders," Proc. IEEE Workshop on Speech Coding for Telecom. (Annapolis), Sept. 1995.

[45] M. R. Zad-Issa and P. Kabal, " A new LPC error criterion for improved pitch tracking," *IEEE Workshop on Speech Coding* (Pocono Manor, PA), pp. 1-2, 1997.

[46] B. Sylvestre, " Time-scale modification of speech: A time-frequency approach," Master's thesis, McGill University, Montreal, Canada, Apr. 1991.

[47] Y. Jiang and V. Cuperman, " Encoding prototype waveforms using a phase codebook," *Proc. IEEE Workshop on Speech Coding for Telecom.* (Annapolis), pp. 21-22, Sept. 1995.

[48] M. Festa and D. Sereno, " A speech coding algorithm based on prototype interpolation with critical bands and phase coding," *Proc. European Conf. on Speech Commun. And Technology* (Madrid), pp. 229-232, Sept. 1995.

[49] I. S. Burnett and G. J. Bradley, " Low complexity decomposition and coding of prototype waveforms," *Proc. IEEE Workshop on Speech Coding for Telecom.* (Annapolis), pp. 23-24, Sept. 1995.

[50] I. S. Burnett and G. J. Bradley, " New techniques for multi-prototype waveform coding at 2.84 kb/s," *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing* (Detroit), pp. 261-264, May 1995.

[51] I. S. Burnett and J. Ni, " Waveform Interpolation and analysis-by-synthesis | a good match," *IEEE Workshop on Speech Coding* (Pocono Manor, PA), pp. 29-30, Sept. 1997.