UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T

G.729

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

(03/96)

ASPECTS GÉNÉRAUX DES SYSTÈMES DE TRANSMISSION NUMÉRIQUES

CODAGE DE LA PAROLE À 8 kbit/s PAR PRÉDICTION LINÉAIRE AVEC EXCITATION PAR SÉQUENCES CODÉES À STRUCTURE ALGÉBRIQUE CONJUGUÉE

Recommandation UIT-T G.729

(Antérieurement «Recommandation du CCITT»)

AVANT-PROPOS

L'UIT-T (Secteur de la normalisation des télécommunications) est un organe permanent de l'Union internationale des télécommunications (UIT). Il est chargé de l'étude des questions techniques, d'exploitation et de tarification, et émet à ce sujet des Recommandations en vue de la normalisation des télécommunications à l'échelle mondiale.

La Conférence mondiale de normalisation des télécommunications (CMNT), qui se réunit tous les quatre ans, détermine les thèmes d'études à traiter par les Commissions d'études de l'UIT-T lesquelles élaborent en retour des Recommandations sur ces thèmes.

L'approbation des Recommandations par les Membres de l'UIT-T s'effectue selon la procédure définie dans la Résolution n° 1 de la CMNT (Helsinki, 1er-12 mars 1993).

La Recommandation UIT-T G.729, que l'on doit à la Commission d'études 15 (1993-1996) de l'UIT-T, a été approuvée le 19 mars 1996 selon la procédure définie dans la Résolution n° 1 de la CMNT.

NOTE

Dans la présente Recommandation, l'expression «Administration» est utilisée pour désigner de façon abrégée aussi bien une administration de télécommunications qu'une exploitation reconnue de télécommunications.

© UIT 1996

Droits de reproduction réservés. Aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de l'UIT.

TABLE DES MATIÈRES

Page

1 Introduction 1

2 Description générale du codeur 1

2.1 Codeur 2

2.2 Décodeur 3

2.3 Délai 4

2.4 Description du codeur de signaux vocaux 4

2.5 Conventions de notation 4

3 Description fonctionnelle du codeur 7

3.1 Prétraitement 7

3.2 Analyse par prédiction linéaire et quantification 7

3.3 Pondération perceptive 14

3.4 Analyse tonale en boucle ouverte 15

3.5 Calcul de la réponse impulsionnelle 16

3.6 Calcul du signal cible 16

3.7 Exploration du répertoire codé de séquences adaptatives 17

3.8 Structure et exploration du répertoire codé de séquences fixes 19

3.9 Quantification des gains 22

3.10 Mise à jour de la mémoire 24

4 Description fonctionnelle du décodeur 25

4.1 Procédure de décodage des paramètres 25

4.2 Post-traitement 28

4.3 Initialisation du codeur et du décodeur 30

4.4 Masquage des trames effacées 30

5 Description au bit près du codeur CS-ACELP 32

5.1 Utilisation du logiciel de simulation 32

5.2 Organisation du logiciel de simulation 32

CODAGE DE LA PAROLE À 8 kbit/s PAR PRÉDICTION LINÉAIRE AVEC EXCITATION PAR SÉQUENCES CODÉES À STRUCTURE ALGÉBRIQUE CONJUGUÉE

(Genève, 1996)

1 Introduction

La présente Recommandation décrit un algorithme pour le codage de signaux vocaux à 8 kbit/s au moyen de la prédiction linéaire à excitation par séquences codées à structure algébrique conjuguée (CS-ACELP) (*conjugate-structure algebraic-code-excited linear-prediction*).

Ce codeur est conçu pour fonctionner avec un signal numérique que l'on obtient en effectuant d'abord un filtrage du signal analogique d'entrée dans la bande téléphonique (Recommandation G.712) puis en l'échantillonnant à 8000 Hz et en le convertissant en signal MIC linéaire à mots de 16 bits, qui est injecté dans le codeur. Inversement, on reconvertira le signal de sortie du décodeur en signal analogique. D'autres éléments caractéristiques d'entrée ou de sortie, comme ceux qui sont spécifiés par la Recommandation G.711 pour les données MIC à 64 kbit/s, seront convertis en mots MIC linéaires de 16 bits avant codage, ou reconvertis après décodage en format approprié à partir des mots MIC linéaires de 16 bits. Le flux binaire passant du codeur au décodeur est défini dans le cadre de la présente Recommandation.

La présente Recommandation se présente comme suit: l'article 2 donne une vue d'ensemble de l'algorithme de prédiction CS-ACELP. Les articles 3 et 4 analysent, respectivement, les principes de codage et de décodage par l'algorithme CS-ACELP. L'article 5 décrit le logiciel qui définit ce codeur en arithmétique à virgule fixe sur 16 éléments binaires.

2 Description générale du codeur

Le codeur de prédiction CS-ACELP est fondé sur le modèle de codage prédictif linéaire à excitation par code (CELP) (*code-excited linear-prediction*). Le codeur opère sur des trames vocales de 10 ms correspondant à 80 échantillons à raison de 8000 échantillons par seconde. Toutes les trames de 10 ms, le signal vocal est analysé pour en extraire les paramètres du modèle de prédiction CELP (coefficients du filtre de prédiction linéaire, index et gains de répertoire codé adaptatif et de répertoire codé fixe). Ces paramètres sont codés et transmis. L'affectation des positions binaires aux paramètres de codage est représentée dans le Tableau 1. Dans le décodeur, ces paramètres servent à récupérer les paramètres d'excitation et du filtre de synthèse. On reconstitue la parole en filtrant ce flux d'excitation dans le filtre de synthèse à court terme, comme indiqué sur la Figure 1. Ce filtre fait appel à une prédiction linéaire (LP) (*linear prediction*) du 10e ordre. Le filtre à long terme ou de synthèse tonale, est mis en œuvre par la méthode dite du répertoire codé adaptatif. Le signal vocal, après avoir été reconstitué par calcul, est encore amélioré par un postfiltre fondamental.

TABLEAU 1/G.729

Affectation des bits dans l'algorithme de codage CS-ACELP à 8 kbit/s (trames de 10 ms)

Paramètre	Mot de code	Sous-trame 1	Sous-trame 2	Total par trame
Paires de raies spectrales	$L0$, $L1$, $L2$, $L3$			18
Délai du répertoire codé adaptatif	$P1$, $P2$	8	5	13
Parité du délai tonal	$P0$	1		1
Index de répertoire codé fixe	$C1$, $C2$	13	13	26
Signe de répertoire codé fixe	$S1$, $S2$	4	4	8
Gains de répertoire (étape 1)	$GA1$, $GA2$	3	3	6
Gains de répertoire (étape 2)	$GB1$, $GB2$	4	4	8
Total				80

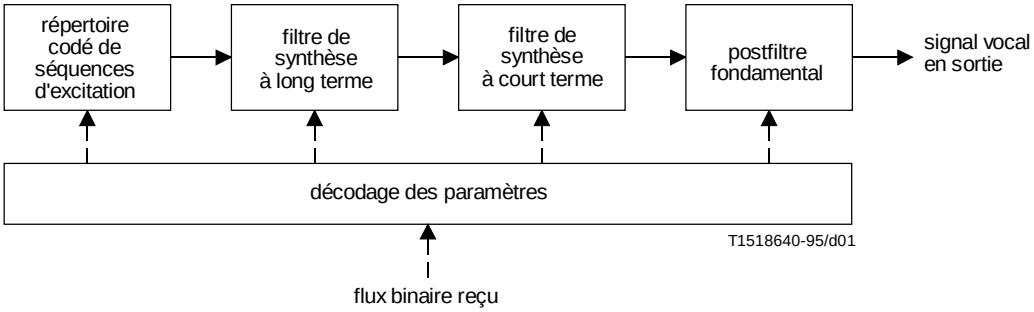

FIGURE 1/G.729

Schéma fonctionnel du modèle théorique de la synthèse par algorithme CELP

2.1 Codeur

Le principe du codage est schématisé sur la Figure 2. Le signal d'entrée subit un filtrage passe-haut et une normalisation dans le bloc de prétraitement. A la sortie du bloc de prétraitement, le signal est utilisé comme entrée pour toutes les analyses suivantes. L'analyse prédictive linéaire est effectuée toutes les trames de 10 ms afin de calculer les coefficients de filtrage prédictif linéaire. Ceux-ci sont convertis en paires de lignes spectrales (LSP) (*line spectrum pairs*) et numérisés sur 18 éléments binaires par quantification vectorielle (VQ) (*vector quantization*) prédictive en deux étapes. Le signal d'excitation est choisi au moyen d'une procédure de recherche par analyse et synthèse dans laquelle l'erreur entre le signal vocal original et le signal vocal reconstitué est minimisée en fonction d'une mesure de distorsion pondérée par la perception. A cette fin, le signal d'erreur passe par un filtre de pondération perceptive dont les coefficients sont déduits du filtre de prédiction linéaire avant quantification. Les poids de la pondération perceptive sont rendus adaptatifs afin d'améliorer la qualité des signaux d'entrée ayant une réponse en fréquence uniforme.

Les paramètres d'excitation (par répertoire codé fixe et par répertoire codé adaptatif) sont déterminés à chaque sous-trame de 5 ms (soit 40 échantillons). Les coefficients du filtre de prédiction linéaire, quantifiés et non quantifiés, sont utilisés pour la deuxième sous-trame, alors que la première utilise une interpolation des coefficients du filtre de prédiction linéaire (aussi bien quantifiés que non quantifiés). Le délai tonal en boucle ouverte est estimé toutes les trames de 10 ms, sur la base du signal vocal issu du pondérateur perceptif. Les opérations suivantes sont reprises pour chaque sous-trame. Le signal cible $x(n)$ est calculé par filtrage de l'énergie résiduelle du codage prédictif linéaire dans le filtre de synthèse pondérée $W(z)/\hat{A}(z)$. Les états initiaux de ces filtres sont mis à jour par filtrage de l'erreur mesurée entre l'énergie résiduelle du codage prédictif linéaire et l'excitation. Cela équivaut au procédé courant consistant à soustraire – du signal vocal pondéré – la réponse à entrée nulle du filtre de synthèse pondérée. La réponse impulsionnelle $h(n)$ du filtre de synthèse pondérée est calculée. Une analyse tonale en boucle fermée est ensuite effectuée (afin de déterminer le délai et le gain par répertoire codé adaptatif) au moyen du signal cible $x(n)$ et de la réponse impulsionnelle $h(n)$, par recherche autour de la valeur du délai tonal en boucle ouverte. On utilise un délai tonal fractionnaire, de résolution 1/3. Ce délai tonal est codé sur 8 éléments binaires dans la première sous-trame et codé différentiellement sur 5 éléments binaires dans la deuxième sous-trame. Le signal cible $x(n)$ est mis à jour par soustraction de la contribution (filtrée) du répertoire codé adaptatif et ce nouveau signal cible, $x'(n)$, est utilisé lors de l'exploration du répertoire codé fixe afin de déterminer l'excitation optimale. On fait appel à un répertoire algébrique de mots de 17 éléments binaires pour l'excitation par répertoire codé fixe. Les gains des contributions par répertoire codé adaptatif et par répertoire codé fixe sont quantifiés vectoriellement sur 7 éléments binaires (avec application au gain par répertoire codé fixe d'une prédiction par analyse à moyenne mobile). Finalement, les mémoires des filtres sont mises à jour au moyen du signal d'excitation ainsi déterminé.

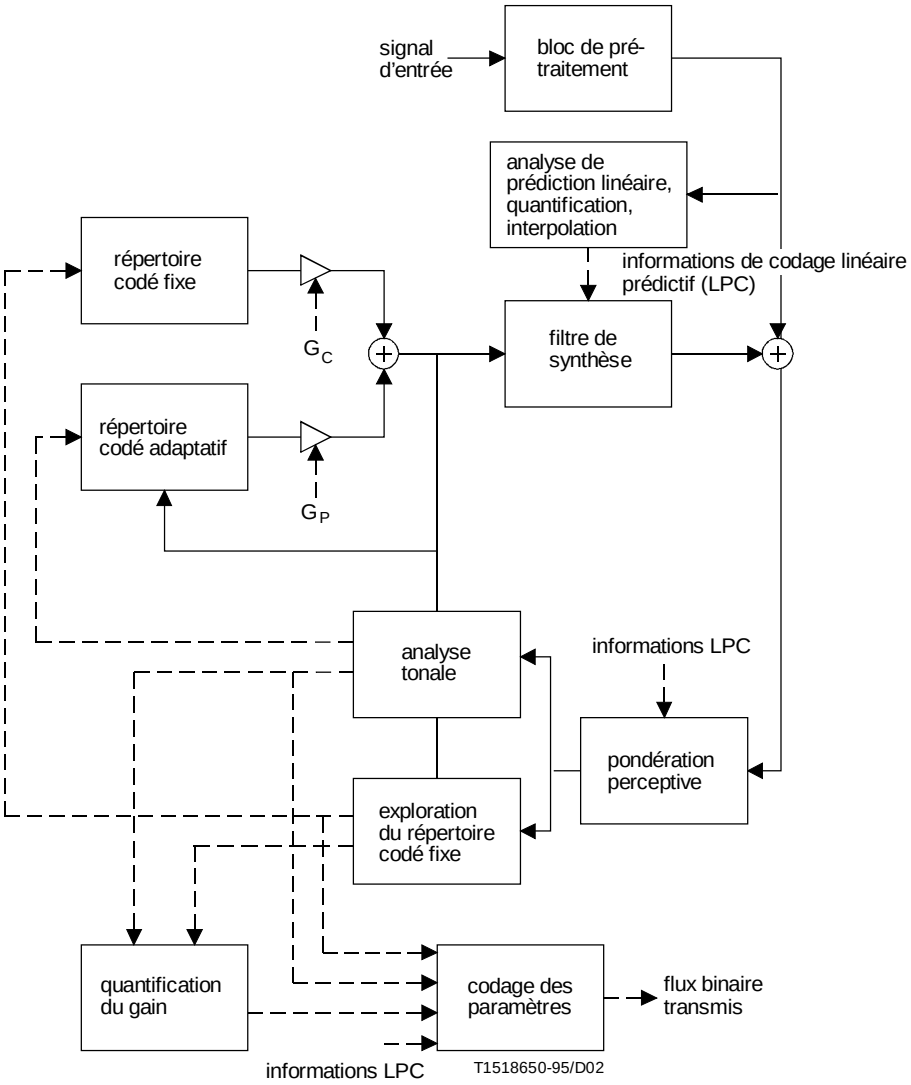

FIGURE 2/G.729

Principe de codage dans l'algorithme CS-ACELP

2.2 Décodeur

Le principe du décodeur est représenté sur la Figure 3. Les index paramétriques sont d'abord extraits du flux binaire reçu. Ces index sont ensuite décodés pour obtenir les paramètres de codage correspondant à une trame vocale de 10 ms. Ces paramètres sont les coefficients convertis en paires de raies spectrales (LSP), les 2 délais tonaux fractionnaires, les 2 vecteurs de répertoire codé fixe et les deux séries de gains par répertoire codé adaptatif et par répertoire codé fixe. Les coefficients en paires LSP sont interpolés et reconvertis en coefficients de filtre de prédiction linéaire pour chaque sous-trame de 5 ms, qui passe par les étapes suivantes:

- l'excitation est construite par combinaison des codes vectoriels adaptatifs et fixes, normalisés par leur gain respectif;
- le signal vocal est reconstitué par filtrage de l'énergie d'excitation dans le filtre de synthèse du codage prédictif linéaire;
- le signal vocal reconstitué est envoyé dans un bloc de post-traitement, qui comprend un postfiltre adaptatif utilisant la sortie des filtres de synthèse à court et à long terme, suivi d'un filtre passe-haut et d'un échantillonneur-normalisateur.

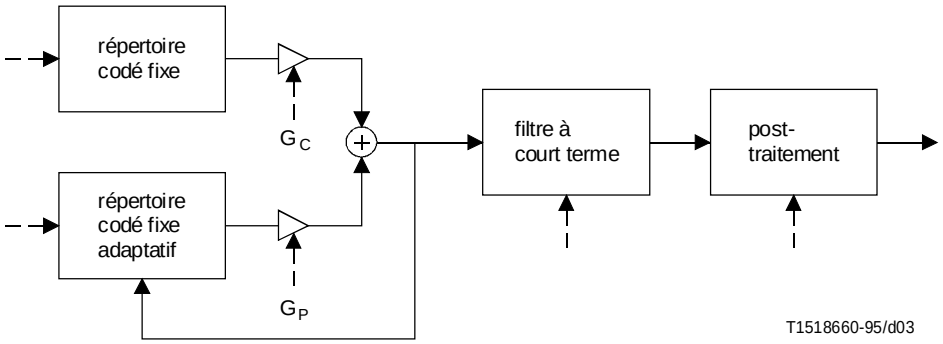

FIGURE 3/G.729

Principe du décodeur CS-ACELP

2.3 Délai

Ce codeur numérise les signaux audio, en particulier vocaux, sous la forme de trames de 10 ms. Il s'y ajoute un délai d'exploration de 5 ms, ce qui porte le délai algorithmique total à 15 ms. Tous les délais additionnels d'une mise en œuvre concrète de ce codeur sont dus à ce qui suit:

- temps de traitement nécessaire pour les opérations de codage et de décodage;
- temps de transmission dans la liaison de communication;
- délai de multiplexage lors de la combinaison de données audio avec d'autres données.

2.4 Description du codeur de signaux vocaux

La description donnée dans la présente Recommandation pour l'algorithme de codage des signaux vocaux est exprimée en termes d'opérations mathématiques en virgule fixe, exactes au bit près. Le code C de l'ANSI (*american national standards institute*), mentionné à l'article 5, fait partie intégrante de la présente Recommandation. Il reflète ce mode de description en virgule fixe au bit près. Les développements mathématiques décrivant le fonctionnement du codeur (article 3) et du décodeur (article 4) peuvent être mis en œuvre de diverses autres façons, ce qui n'exclut pas la possibilité d'une réalisation de codec non conforme à la présente Recommandation. La description de l'algorithme en code C ANSI à l'article 5 a donc priorité sur les descriptions mathématiques des articles 3 et 4, en cas de divergence. On peut se procurer, auprès de l'UIT, un ensemble non exhaustif de séquences de test pouvant être utilisées conjointement avec le code C.

2.5 Conventions de notation

Dans l'ensemble de la présente Recommandation, l'on a essayé de respecter les conventions de notation suivantes:

- les répertoires de séquences codées sont repérés par des caractères calligraphiques (cursifs) (par exemple $\mathcal{C}$);
- les signaux temporels sont repérés par leur symbole, assorti d'un index d'échantillon entre parenthèses [par exemple $s(n)$]. Le symbole n est utilisé comme index de pointage sur un échantillon;
- les indices supérieurs entre parenthèses (par exemple $g^{(m)}$) sont utilisés pour indiquer que des variables dépendent du temps. La variable m désigne, selon le contexte, soit un index de trame soit un index de sous-trame; la variable n désigne un index d'échantillon;
- les indices de récurrence sont repérés par un indice supérieur entre crochets (par exemple $E^{[k]}$);
- les indices inférieurs désignent un élément particulier dans une table de coefficients;
- le symbole ^ désigne une version quantifiée d'un paramètre (par exemple $\hat{g}_{\mathcal{C}}$);
- les étendues des paramètres sont indiquées entre crochets et comprennent les valeurs limites (par exemple [0,6, 0,9]);

- la fonction *log* correspond à un logarithme décimal;
- la fonction *int* correspond à la troncature d'un nombre jusqu'à sa valeur d'entier;
- les nombres décimaux à virgule flottante sont des versions arrondies des valeurs en virgule fixe sur 16 bits qui sont utilisées dans la mise en œuvre du code C de l'ANSI.

Le Tableau 2 énumère les symboles les plus couramment utilisés dans la présente Recommandation. Le Tableau 3 est un glossaire des signaux les plus utiles et le Tableau 4 résume les variables applicables ainsi que leurs dimensions. Les constantes sont énumérées dans le Tableau 5. Le Tableau 6 résume les acronymes utilisés dans la présente Recommandation.

TABLEAU 2/G.729

Glossaire des symboles les plus utiles

Désignation	Référence	Description
$1/\hat{A}(z)$	Equation (2)	Filtre de synthèse en codage prédictif linéaire
$H_{h1}(z)$	Equation (1)	Filtre passe-haut d'entrée
$H_p(z)$	Equation (78)	Postfiltre à long terme
$H_f(z)$	Equation (84)	Postfiltre à court terme
$H_t(z)$	Equation (86)	Filtre de compensation d'écart de niveau
$H_{h2}(z)$	Equation (91)	Filtre passe-haut de sortie
$P(z)$	Equation (46)	Préfiltre pour répertoire codé fixe
$W(z)$	Equation (27)	Filtre de pondération

TABLEAU 3/G.729

Glossaire des signaux les plus utiles

Désignation	Référence	Description
$c(n)$	3.8	Contribution du répertoire codé fixe
$d(n)$	3.8.1	Corrélation entre signal cible et signal $h(n)$
$ew(n)$	3.10	Signal d'erreur
$h(n)$	3.5	Réponse impulsionnelle des filtres de pondération et de synthèse
$r(n)$	3.6	Signal résiduel
$s(n)$	3.1	Signal vocal prétraité
$\hat{s}(n)$	4.1.6	Signal vocal reconstitué
$s'(n)$	3.2.1	Signal vocal fenêtré
$sf(n)$	4.2	Sortie postfiltrée
$sf'(n)$	4.2	Sortie postfiltrée et normalisée par le gain
$sw(n)$	3.6	Signal vocal pondéré
$x(n)$	3.6	Signal cible
$x'(n)$	3.8.1	Deuxième signal cible
$u(n)$	3.10	Excitation vers filtre de synthèse LP
$v(n)$	3.7.1	Contribution du répertoire codé adaptatif
$y(n)$	3.7.3	Convolution $v(n) * h(n)$
$z(n)$	3.9	Convolution $c(n) * h(n)$

TABLEAU 4/G.729

Glossaire des variables les plus utiles

Désignation	Longueur	Description
g_p	1	Gain par répertoire codé adaptatif
g_c	1	Gain par répertoire codé fixe
g_l	1	Terme de gain pour le postfiltre à long terme
g_f	1	Terme de gain pour le postfiltre à court terme
g_t	1	Terme de gain pour le postfiltre de compensation des écarts de niveau
G	1	Gain pour la normalisation par le gain
T_{op}	1	Délai tonal en boucle ouverte
a_i	11	Coefficients LP ($a_0 = 1{,}0$)
k_i	10	Coefficients de réflexion
k'_1	1	Coefficient de réflexion pour le postfiltre de compensation des écarts de niveau
o_i	2	Coefficients des écarts d'aire logarithmique (LAR)
ω_i	10	Fréquences normalisées des raies spectrales (LSF)
$\hat{p}_{i,j}$	40	Prédicteur à moyenne mobile pour la quantification des fréquences LSF
q_i	10	Coefficients des paires de lignes spectrales (LSP)
$r(k)$	11	Coefficients d'autocorrélation
$r'(k)$	11	Coefficients d'autocorrélation modifiés
w_i	10	Pondérateurs pour la quantification des coefficients LSP
$\hat{l}_i$	10	Sortie du quantificateur de coefficients LSP

TABLEAU 5/G.729

Glossaire des constantes les plus utiles

Désignation	Valeur	Description
f_s	8000	Fréquence d'échantillonnage
f_0	60	Extension de la largeur de bande
γ_1	0,94/0,98	Pondérateur du filtre de pondération perceptive
γ_2	0,60/[0,4 – 0,7]	Pondérateur du filtre de pondération perceptive
γ_n	0,55	Pondérateur de postfiltre
γ_d	0,70	Pondérateur de postfiltre
γ_p	0,50	Pondérateur du postfiltre fondamental
γ_t	0,90/0,2	Pondérateur du postfiltre de compensation d'écart
$\mathcal{C}$	Tableau 7	Répertoire (algébrique) fixe
L0	3.2.4	Répertoire du prédicteur à moyenne mobile
L1	3.2.4	Première étape du répertoire LSP
L2	3.2.4	Deuxième étape du répertoire LSP (partie inférieure)
L3	3.2.4	Deuxième étape du répertoire LSP (partie supérieure)
$\mathcal{GA}$	3.9	Répertoire de gain (première étape)
$\mathcal{GB}$	3.9	Répertoire de gain (deuxième étape)
w_{lag}	Equation (6)	Fenêtre du délai de corrélation
w_{lp}	Equation (3)	Fenêtre de l'analyse prédictive linéaire

TABLEAU 6/G.729

Glossaire des acronymes

Acronyme	Description
CELP	Prédiction linéaire avec excitation par code (*code-excited linear-prediction*)
CS-ACELP	Prédiction linéaire avec excitation par séquences codées à structure algébrique conjuguée (*conjugate-structure algebraic CELP*)
MA	Moyenne mobile (*moving average*)
MSB	Bit de poids fort (*most significant bit*)
MSE	Erreur quadratique (*mean-squared error*)
LAR	Ecart d'aire logarithmique (*log area ratio*)
LP	Prédiction linéaire (*linear prediction*)
LSP	Paire de raies spectrales (*line spectral pair*)
LSF	Fréquence de raie spectrale (*line spectral frequency*)
VQ	Quantification vectorielle (*vector quantization*)

3 Description fonctionnelle du codeur

Cet article décrit les différentes fonctions du codeur représenté par les blocs de la Figure 2. La Figure 4 montre en détail la circulation des signaux.

3.1 Prétraitement

Comme indiqué dans l'article 2, l'entrée dans le codeur de signaux vocaux est censée être un signal MIC de 16 bits. Deux fonctions de prétraitement sont appliquées avant le processus de codage:

1) la normalisation du signal;
2) son filtrage passe-haut.

La normalisation consiste à diviser par un facteur de 2 l'énergie d'entrée afin de diminuer la probabilité de dépassements de capacité dans une réalisation en virgule fixe. Le filtre passe-haut sert de précaution à l'encontre de composantes parasites à basse fréquence. On fait appel à un filtre du deuxième ordre avec section des pôles et section des zéros, dont la fréquence de coupure est de 140 Hz. On combine les deux opérations, de normalisation moitié et de filtrage passe-haut, en divisant par 2 les coefficients figurant au numérateur de ce filtre, dont l'équation résultante est donnée par la formule suivante:

$$H_{h1}(z) = \frac{0{,}46363718 - 0{,}92724705z^{-1} + 0{,}46363718z^{-2}}{1 - 1{,}9059465z^{-1} + 0{,}9114024z^{-2}} \tag{1}$$

Le signal d'entrée filtré par $H_{h1}(z)$ est dénommé $s(n)$. Il sera utilisé dans toutes les opérations ultérieures du codeur.

3.2 Analyse par prédiction linéaire et quantification

Les filtres d'analyse et de synthèse à court terme sont fondés sur des filtres de prédiction linéaire (LP) du 10e ordre. Le filtre de synthèse LP est défini comme suit:

$$\frac{1}{\hat{A}(z)} = \frac{1}{1 + \sum_{i=1}^{10} \hat{a}_i z^{-i}} \tag{2}$$

où les $\hat{a}_i$, $i = 1,...,10$, sont les coefficients (quantifiés) de prédiction linéaire (LP). On effectue une prédiction à court terme (ou analyse prédictive linéaire) à chaque trame vocale au moyen de la méthode d'autocorrélation dans une fenêtre asymétrique de 30 ms. Tous les 80 échantillons (soit 10 ms), on calcule les coefficients d'autocorrélation des données vocales fenêtrées puis on les convertit en coefficients LP par l'algorithme de Levinson. Ces coefficients LP sont ensuite transformés dans le domaine des paires de raies spectrales (LSP) en vue de leur quantification et de leur interpolation. Les filtres interpolés, quantifiés et non quantifiés, sont reconvertis en coefficients de filtre LP (afin de construire les filtres de synthèse et de pondération pour chaque sous-trame).

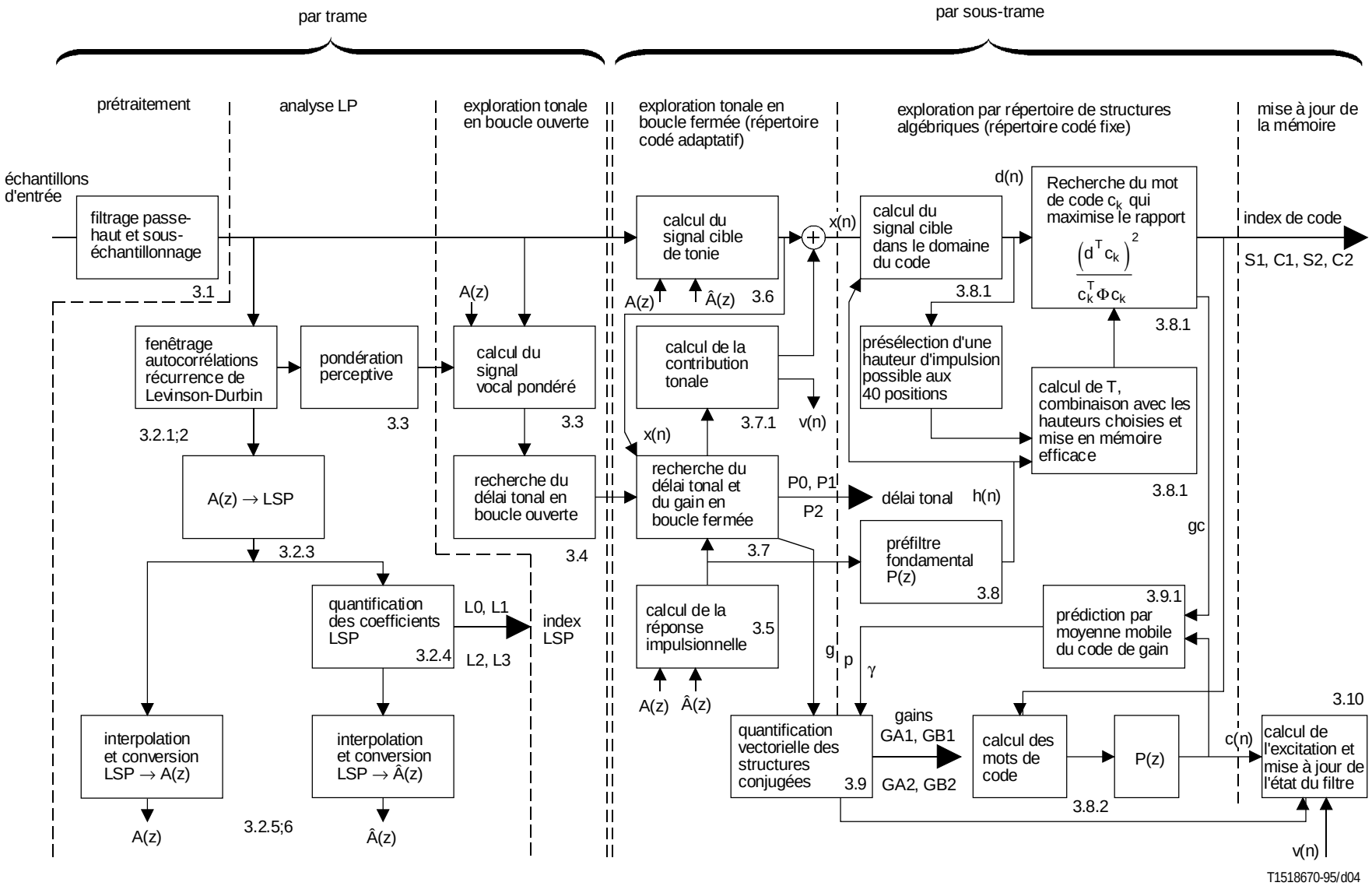

FIGURE 4/G.729

Circulation des signaux dans le codeur CS-ACELP

3.2.1 Fenêtrage et calcul des autocorrélations

La fenêtre d'analyse LP se compose de deux parties: la première est une demi-fenêtre de Hamming et la seconde un quart de période d'une fonction cosinus. Cette fenêtre est donnée par l'équation suivante:

$$w_{lp}(n) = \begin{cases} 0{,}54 - 0{,}46 \cos\left(\dfrac{2\pi n}{399}\right) & n = 0,...,199 \\ \cos\left(\dfrac{2\pi\,(n - 200)}{159}\right) & n = 200,...,239 \end{cases} \tag{3}$$

L'analyse LP comporte une exploration de 5 ms, c'est-à-dire qu'il faut 40 échantillons issus de la prochaine trame vocale. Cela se traduit par un délai algorithmique supplémentaire de 5 ms au niveau du codeur. La fenêtre d'analyse LP s'applique à 120 échantillons issus de trames vocales passées, à 80 échantillons issus de la trame vocale présente, et à 40 échantillons de la trame vocale future. La procédure de fenêtrage est décrite sur la Figure 5.

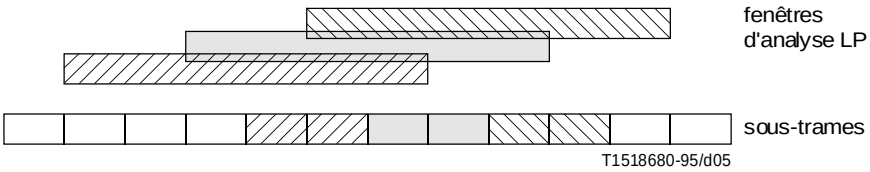

FIGURE 5/G.729

Procédure de fenêtrage en analyse LP

Les différents motifs d'ombrage indiquent les correspondances entre fenêtres d'excitation et d'analyse LP.

Le signal vocal fenêtré

$$s'(n) = w_{lp}(n)\, s(n) \qquad n = 0,...,239 \tag{4}$$

est utilisé pour calculer les coefficients d'autocorrélation:

$$r(k) = \sum_{n=k}^{239} s'(n)\, s'(n - k) \qquad k = 0,...,10 \tag{5}$$

Pour éviter des problèmes d'ordre arithmétique lors de signaux d'entrée de faible niveau, la valeur de $r(0)$ possède une borne inférieure $r(0) = 1{,}0$. Une extension de la largeur de bande est effectuée sur 60 Hz, par multiplication des coefficients d'autocorrélation comme suit:

$$w_{lag}(k) = \exp\left[-\frac{1}{2}\left(\frac{2\pi f_0 k}{f_s}\right)^2\right] \qquad k = 1,...,10 \tag{6}$$

où f_0 = 60 Hz est l'extension de largeur de bande et où f_s = 8000 Hz est la fréquence d'échantillonnage. En outre, le coefficient $r(0)$ est multiplié par un facteur de correction du bruit blanc, de 1,0001, ce qui est équivalent à l'introduction d'une borne inférieure de bruit à –40 dB. Les coefficients d'autocorrélation ainsi modifiés sont donnés par les relations suivantes:

$$\begin{aligned} r'(0) &= 1{,}0001\, r(0) \\ r'(k) &= w_{lag}(k)\, r(k) \qquad k = 1,...,10 \end{aligned} \tag{7}$$

3.2.2 Algorithme de Levinson-Durbin

Les coefficients d'autocorrélation modifiés, $r'(k)$, sont utilisés pour obtenir les coefficients de filtre LP, a_i, $i = 1,....,10$, après résolution de l'ensemble d'équations suivant:

$$\sum_{i=1}^{10} a_i r'(|i - k|) = -r'(k) \qquad k = 1,...,10 \tag{8}$$

On résout l'ensemble d'équations (8) en utilisant l'algorithme de Levinson-Durbin, qui fait appel à la récurrence suivante:

$E^{[0]} = r'(0)$

pour $i = 1$ *à* 10

$a_0^{[i-1]} = 1$

$k_i = -\left[\sum_{j=0}^{i-1} a_j^{[i-1]} r'(i-j)\right] / E^{[i-1]}$

$a_i^{[i]} = k_i$

pour $j = 1$ *à* $i - 1$

$a_j^{[i]} = a_j^{[i-1]} + k_i a_{i-j}^{[i-1]}$

fin

$E^{[i]} = \left(1 - k_i^2\right) E^{[i-1]}$

fin

La solution finale est donnée sous la forme $a_j = a_j^{[10]}$, $j = 0,...,10$, avec $a_0 = 1{,}0$.

3.2.3 Conversion des coefficients de prédiction linéaire (LP) en coefficients de paires de raies spectrales (LSP)

Les coefficients du filtre LP, a_i, $i = 0,...,10$, sont convertis en coefficients de paires de raies spectrales (LSP) aux fins de la quantification et de l'interpolation. Pour un filtre LP du 10e ordre, les coefficients LSP sont définis comme étant les racines des polynômes sommateurs et différentiateurs suivants:

$$F_1'(z) = A(z) + z^{-11}A(z^{-1}) \tag{9}$$

et

$$F_2'(z) = A(z) - z^{-11}A(z^{-1}) \tag{10}$$

respectivement. Le polynôme $F'_1(z)$ est symétrique, tandis que le polynôme $F'_2(z)$ est antisymétrique (conjugué). On peut démontrer que toutes les racines de ces polynômes se trouvent sur le cercle unité et qu'elles alternent l'une avec l'autre. Le polynôme $F'_1(z)$ possède une racine $z = -1$ ($\omega = \pi$) et le polynôme $F'_2(z)$ possède une racine $z = +1$ ($\omega = 0$). On élimine ces deux racines en définissant les deux nouveaux polynômes suivants:

$$F_1(z) = F'_1(z) \ / \ (1 + z^{-1}) \tag{11}$$

et

$$F_2(z) = F'_2(z) \ / \ (1 - z^{-1}) \tag{12}$$

Chacun de ces polynômes possède 5 racines conjuguées sur le cercle unité ($e_i^{\pm j\omega}$) et on peut les écrire comme suit:

$$F_1(z) = \prod_{i = 1, 3, ..., 9} (1 - 2q_i z^{-1} + z^{-2}) \tag{13}$$

et

$$F_2(z) = \prod_{i = 2, 4, ..., 10} (1 - 2q_i z^{-1} + z^{-2}) \tag{14}$$

où $q_i = \cos(\omega_i)$. Les coefficients ω_i sont les fréquences des raies spectrales (LSF) et satisfont à la relation d'ordre $0 < \omega_i < \omega_2 < ... < \omega_{10} < \pi$. Les coefficients q_i sont désignés comme étant les coefficients LSP dans le domaine cosinusoïdal.

Etant donné que les deux polynômes $F_1(z)$ et $F_2(z)$ sont symétriques, il suffit de calculer les 5 premiers coefficients de chaque polynôme, au moyen des relations de récurrence suivantes:

$$\begin{aligned} f_1(i + 1) &= a_{i+1} + a_{10-i} - f_1(i) \qquad i = 0,...,4 \\ f_2(i + 1) &= a_{i+1} - a_{10-i} + f_2(i) \qquad i = 0,...,4 \end{aligned} \tag{15}$$

où $f_1(0) = f_2(0) = 1{,}0$. Les coefficients LSP sont trouvés par évaluation des polynômes $F_1(z)$ et $F_2(z)$ à 60 points équidistants entre 0 et π puis en vérifiant les changements de signe. Chaque changement de signe correspond à l'existence d'une racine et l'intervalle du changement de signe est alors divisé par 4 afin d'affiner la recherche de la racine. On se sert des polynômes de Tchebycheff pour évaluer les polynômes $F_1(z)$ et $F_2(z)$. Dans cette méthode, on trouve directement les racines dans le domaine cosinusoïdal. Le polynôme $F_1(z)$ ou $F_2(z)$, évalué à la valeur $z = e^{j\omega}$, peut s'écrire comme suit:

$$F(\omega) = 2e^{-j5\omega} C(x) \tag{16}$$

avec

$$C(x) = T_5(x) + f(1)T_4(x) + f(2)T_3(x) + f(3)T_2(x) + f(4)T_1(x) + f(5)/2 \tag{17}$$

où le terme $T_m(x) = \cos(m\omega)$ est le polynôme de Tchebycheff du nième ordre et où les $f(i)$, $i = 1,...5$, sont les coefficients du polynôme $F_1(z)$ ou $F_2(z)$, calculés par l'équation (15). Le polynôme $C(x)$ est évalué à une certaine valeur de $x = \cos(\omega)$ au moyen de la relation de récurrence suivante:

pour k = *de* 4 *à* 1

$$b_k = 2xb_{k+1} - b_{k+2} + f(5 - k)$$

fin

$$C(x) = xb_1 - b_2 + f(5)/2$$

avec les valeurs initiales $b_5 = 1$ et $b_6 = 0$.

3.2.4 Quantification des coefficients LSP

Les coefficients des paires de raies spectrales, q_i, sont quantifiés par application de la représentation des fréquences LSF, ω_i, dans le domaine fréquentiel normalisé [0, π]; c'est-à-dire:

$$\omega_i = \arccos(q_i) \qquad i = 1,...,10 \tag{18}$$

On utilise une prédiction par moyenne mobile périodique du 4e ordre pour prédire les coefficients de fréquence LSF de la trame courante. La différence entre les coefficients calculés et les coefficients prédits est quantifiée au moyen d'un quantificateur vectoriel à deux étapes. La première étape est une quantification vectorielle à 10 dimensions qui utilise le répertoire L1 avec 128 entrées (sur 7 bits). La deuxième étape est une quantification vectorielle sur 10 bits, qui a été mise en œuvre sous forme dédoublée avec deux répertoires à 5 dimensions, L2 et L3 contenant 32 entrées (de 5 bits) chacun.

Pour expliquer le processus de quantification, il convient de décrire d'abord le processus de décodage. Chaque coefficient s'obtient en additionnant 2 séquences de répertoire:

$$\hat{l}_i = \begin{cases} \mathrm{L1}_i(L1) + \mathrm{L2}_i(L2) & i = 1,...,5 \\ \mathrm{L1}_i(L1) + \mathrm{L3}_{i-5}(L3) & i = 6,...,10 \end{cases} \tag{19}$$

où $L1$, $L2$, $L3$ sont les index de répertoire. Pour éviter des résonances soudaines dans le filtre de synthèse LP, les coefficients $\hat{l}_i$ sont disposés de manière que des coefficients adjacents aient une distance minimale de J. La routine de réarrangement est indiquée ci-dessous:

pour $i = 2,...,10$

  si $(\hat{l}_{i-1} > \hat{l}_i - J)$

    $\hat{l}_{i-1} = (\hat{l}_i + \hat{l}_{i-1} - J)/2$

    $\hat{l}_i = (\hat{l}_i + \hat{l}_{i-1} + J)/2$

  fin

fin

Ce processus de réarrangement est effectué deux fois. D'abord avec une valeur $J = 0{,}0012$, puis avec une valeur $J = 0{,}0006$. Après ce processus de réarrangement, les coefficients de fréquence LSF quantifiés, $\hat{\omega}_i^{(m)}$ pour la trame actuelle m sont obtenus à partir de la somme pondérée des précédents coefficients issus du quantificateur, $\hat{l}_i^{(m-k)}$, et du coefficient venant de sortir du quantificateur, $\hat{l}_i^{(m)}$:

$$\hat{\omega}_i^{(m)} = \left(1 - \sum_{k=1}^{4} \hat{p}_{i,k}\right)\hat{l}_i^{(m)} + \sum_{k=1}^{4} \hat{p}_{i,k}\,\hat{l}_i^{(m-k)} \qquad i = 1,...,10 \tag{20}$$

où les $\hat{p}_{i,k}$ sont les coefficients du prédicteur à moyenne mobile périodique. Une position binaire distincte, $L0$, détermine quel est le prédicteur à moyenne mobile à utiliser. Au démarrage, les valeurs initiales des $\hat{l}_i^{(k)}$ sont données par la relation $\hat{l}_i = i\pi/11$ pour tous les $k < 0$.

Après calcul des coefficients quantifiés $\hat{\omega}_i$, on contrôle la stabilité du filtre correspondant, comme suit:

1) classement des coefficients $\hat{\omega}_i$ par valeurs croissantes;

2) si $\hat{\omega}_i < 0{,}005$, alors $\hat{\omega}_i = 0{,}005$;

3) si $\hat{\omega}_{i+1} - \hat{\omega}_i = 0{,}0391$, alors $\hat{\omega}_{i+1} = \hat{\omega}_i + 0{,}0391$, $\quad i = 1,...,9$;

4) si $\hat{\omega}_{10} > 3{,}135$, alors $\hat{\omega}_{10} = 3{,}135$.

La procédure de codage des paramètres LSF peut être décrite comme suit. Pour chacun des deux prédicteurs à moyenne mobile, il faut trouver la meilleure approximation des coefficients LSF actuels. Cette meilleure approximation est définie comme étant celle qui minimise l'erreur quadratique pondérée suivante:

$$E_{lsf} = \sum_{i=1}^{10} w_i(\omega_i - \hat{\omega}_i)^2 \tag{21}$$

Les pondérateurs w_i sont rendus adaptatifs en fonction des coefficients de fréquence LSF non quantifiés:

$$w_i = \begin{cases} 1{,}0 & si\ \omega_2 - 0{,}04\pi - 1 > 0 \\ 10\,(\omega_2 - 0{,}04\pi - 1)^2 + 1 & sinon \end{cases}$$

$$w_i\ 2 \le i \le 9 = \begin{cases} 1{,}0 & si\ \omega_{i+1} - \omega_{i-1} - 1 > 0 \\ 10\,(\omega_{i+1} - \omega_{i-1} - 1)^2 + 1 & sinon \end{cases} \tag{22}$$

$$w_{10} = \begin{cases} 1{,}0 & si\ -\omega_9 + 0{,}92\pi - 1 > 0 \\ 10\,(-\omega_9 + 0{,}92\pi - 1)^2 + 1 & sinon \end{cases}$$

En outre, les pondérateurs w_5 et w_6 sont chacun multipliés par 1,2.

Le vecteur à quantifier pour la trame m actuelle est obtenu par les coefficients suivants:

$$l_i = \left[\omega_i^{(m)} - \sum_{k=1}^{4} \hat{p}_{i,k}\ \hat{l}_i^{(m-k)}\right] / \left(1 - \sum_{k=1}^{4} \hat{p}_{i,k}\right) \qquad i = 1,...,10 \tag{23}$$

Le premier répertoire, L1, est exploré et on choisit l'entrée $L1$ qui minimise l'erreur quadratique (non pondérée). On effectue ensuite une exploration du deuxième répertoire, L2, qui définit la partie inférieure de la deuxième étape. Pour chaque séquence possible, on reconstruit le vecteur partiel $\hat{\omega}_i$, $i = 1,...,5$, au moyen de l'équation (20). Ce vecteur partiel est repositionné de façon à assurer une distance minimale de 0,0012. Au moyen de l'équation (21), on calcule ensuite l'erreur MSE pondérée et on sélectionne le vecteur $L2$ qui minimise l'erreur quadratique. La partie supérieure de la seconde étape est explorée dans le répertoire L3 au moyen du vecteur $L1$ sélectionné lors de la première étape et du vecteur $L2$ qui correspond à la partie inférieure de la seconde étape. On applique de nouveau la procédure de réarrangement pour assurer une distance minimale de 0,0012 entre les vecteurs. Le vecteur $L3$ qui minimise l'erreur MSE pondérée. Les vecteurs résultants $\hat{l}_i$, $i = 1,...,10$ sont redisposés de manière à assurer une distance minimale de 0,0006 entre eux. Ce processus est repris pour chacun des deux prédicteurs à moyenne mobile contenus chacun dans un répertoire L0. On sélectionne alors le prédicteur à moyenne mobile (défini par le bit $L0$) qui minimise l'erreur quadratique pondérée. Comme expliqué au début du présent paragraphe, les vecteurs résultants $\hat{l}_i$ sont redisposés deux fois et un contrôle de stabilité est effectué pour obtenir les coefficients pondérés de fréquence LSF, $\hat{\omega}_i$.

3.2.5 Interpolation des coefficients LSP

Les coefficients LP (quantifiés et non quantifiés) sont utilisés pour la deuxième sous-trame. Pour la première sous-trame, les coefficients LP (quantifiés et non quantifiés) sont obtenus par interpolation linéaire des paramètres correspondants dans les sous-trames adjacentes. Cette interpolation est appliquée aux coefficients de paires LSP dans le domaine cosinusoïdal. Soit $q_i^{(current)}$ les coefficients LSP calculés pour la trame de 10 ms actuelle et $q_i^{(previous)}$ les coefficients LSP calculés dans la trame précédente de 10 ms. Les coefficients LSP (non quantifiés) qui ont été interpolés dans chacune des 2 sous-trames sont donnés par les relations suivantes:

$$\begin{aligned} &Sous\text{-}trame\ 1:\ q_i^{(1)} = 0{,}5q_i^{(previous)} + 0{,}5q_i^{(current)} && i = 1,...,10 \\ &Sous\text{-}trame\ 2:\ q_i^{(2)} = q_i^{(current)} && i = 1,...,10 \end{aligned} \tag{24}$$

On applique la même procédure pour l'interpolation des coefficients LSP quantifiés, en remplaçant dans l'équation (24) les q_i par les $\hat{q}_i$.

3.2.6 Conversion des coefficients LSP en coefficients LP

Une fois quantifiés et interpolés, les coefficients des paires LSP sont reconvertis en coefficients LP, a_i. Cette conversion est effectuée comme suit. On détermine les coefficients des polynômes $F_1(z)$ et $F_2(z)$ par expansion des équations (13) et (14), sur la base des coefficients LSP quantifiés et interpolés. A partir des coefficients q_i, on calcule les coefficients $f_1(i)$, $i = 1,...,5$, par la relation récurrente suivante:

$$
\begin{aligned}
&\textit{pour } i = 1 \textit{ à } 5\\
&\qquad f_1(i) = -2q_{2i-1} f_1(i-1) + 2f_1(i-2)\\
&\qquad \textit{pour } j = \textit{de } i-1 \textit{ à } 1\\
&\qquad\qquad f_1^{[i]}(j) = f_1^{[i-1]}(j) - 2q_{2i-1} f_1^{[i-1]}(j-1) + f_1^{[i-1]}(j-2)\\
&\qquad \textit{fin}\\
&\textit{fin}
\end{aligned}
$$

avec comme valeurs initiales $f_1(0) = 1$ et $f_1(-1) = 0$. Les coefficients $f_2(i)$ sont calculés de la même manière, avec remplacement des q_{2i-1} par q_{2i}.

Une fois les coefficients $f_1(i)$ et $f_2(i)$ calculés, on multiplie les polynômes $F_1(z)$ et $F_2(z)$ par, respectivement, $1 + z^{-1}$ et $1 - z^{-1}$, pour obtenir les polynômes $F'_1(z)$ et $F'_2(z)$, qui donnent les coefficients suivants:

$$
\begin{aligned}
f'_1(i) &= f_1(i) + f_1(i-1) \qquad i = 1,...,5\\
f'_2(i) &= f_2(i) - f_2(i-1) \qquad i = 1,...,5
\end{aligned}
\tag{25}
$$

Finalement, on retrouve les coefficients LP à partir des coefficients $f'_1(i)$ et $f'_2(i)$ comme suit:

$$
a_i = \begin{cases} 0{,}5 f'_1(i) + 0{,}5 f'_2(i) & i = 1,...,5\\ 0{,}5 f'_1(11-i) - 0{,}5 f'_2(11-i) & i = 6,...,10 \end{cases}
\tag{26}
$$

Cette équation est directement issue de la relation $A(z) = (F'_1(z) + F'_2(z))/2$, parce que les polynômes $F'_1(z)$ et $F'_2(z)$ sont, respectivement, symétrique et antisymétrique (conjugués).

3.3 Pondération perceptive

Le filtre de pondération perceptive utilise les coefficients non quantifiés a_i du filtre de prédiction linéaire, comme suit:

$$
W(z) = \frac{A(z/\gamma_1)}{A(z/\gamma_2)} = \frac{1 + \sum_{i=1}^{10} \gamma_1^i a_i z^{-i}}{1 + \sum_{i=1}^{10} \gamma_2^i a_i z^{-i}}
\tag{27}
$$

Les valeurs des termes γ_1 et γ_2 déterminent la réponse en fréquence du filtre $W(z)$. En ajustant correctement la valeur de ces variables, il est possible de rendre la pondération plus efficace. Pour cela, on fait dépendre γ_1 et γ_2 de la forme spectrale du signal d'entrée. Cette adaptation est effectuée à chaque trame de 10 ms mais on fait appel à une procédure d'interpolation pour chaque première sous-trame afin de lisser cette adaptation. La forme spectrale est obtenue au moyen d'un filtre de prédiction linéaire du 2e ordre, construit en tant que sous-produit de l'algorithme de récurrence de Levinson-Durbin (voir 3.2.2). Les coefficients de réflexion k_i sont convertis en coefficients LAR (écarts d'aire logarithmique), o_i, selon l'équation suivante:

$$
o_i = \log \frac{(1{,}0 + k_i)}{(1{,}0 - k_i)} \qquad i = 1, 2
\tag{28}
$$

Les coefficients d'écart LAR correspondant à la trame de 10 ms actuelle sont utilisés pour la deuxième sous-trame. Les coefficients d'écart LAR de la première sous-trame sont obtenus par interpolation linéaire avec les paramètres d'écart LAR issus de la trame précédente. Les coefficients d'écart LAR interpolés dans chacune des deux sous-trames sont calculés comme suit:

$$\begin{aligned} &\textit{Sous-trame } 1\text{:}\ o_i^{(1)} = 0{,}5\, o_i^{(previous)} + 0{,}5\, o_i^{(current)} && i = 1, 2 \\ &\textit{Sous-trame } 2\text{:}\ o_i^{(2)} = o_i^{(current)} && i = 1, 2 \end{aligned} \tag{29}$$

L'enveloppe spectrale est caractérisée comme étant soit plate (*flat* = 1) soit inclinée (*tilted* = 0). Pour chaque sous-trame, on calcule cette caractéristique en appliquant une fonction seuil aux coefficients d'écart LAR. Pour éviter des modifications brusques, on fait appel à une formule d'hystérésis tenant compte de la valeur du terme *flat* dans la sous-trame précédente, $m - 1$:

$$flat^{(m)} = \begin{cases} 0 & si\ \ o_1^{(m)} < -1{,}74\ et\ o_2^{(m)} > 0{,}65\ et\ flat^{(m-1)} = 1 \\ 1 & si\ \left(o_1^{(m)} > -1{,}52\ ou\ o_2^{(m)} < 0{,}43\right)\ et\ flat^{(m-1)} = 0 \\ flat^{(m-1)} & sinon \end{cases} \tag{30}$$

Si le spectre interpolé pour une sous-trame est classifié comme étant plat ($flat^{(m)} = 1$), les pondérateurs sont réglés comme suit: $\gamma_1 = 0{,}94$ et $\gamma_2 = 0{,}6$. Si le spectre est classifié comme étant incliné ($flat^{(m)} = 0$), la valeur de γ_1 est réglée à 0,98. Celle de γ_2 est adaptée à l'intensité des résonances dans le filtre de synthèse LP mais est bornée entre 0,4 et 0,7. En présence d'une forte résonance, la valeur de γ_2 est réglée plus près de la borne supérieure. On effectue cette adaptation sur la base d'un critère tenant compte de la distance minimale entre 2 coefficients successifs de paire LSP pour la sous-trame actuelle. La distance minimale est donnée par l'équation suivante:

$$d_{min} = min\,[\omega_{i+1} - \omega_i] \qquad i = 1,...,9 \tag{31}$$

La valeur de γ_2 est calculée au moyen de la relation linéaire ci-après:

$$\gamma_2 = -6{,}0\, d_{min} + 1{,}0 \qquad \text{entre les limites } 0{,}4 \le \gamma_2 \le 0{,}7 \tag{32}$$

Le signal vocal pondéré dans une sous-trame est donné par:

$$sw(n) = s(n) + \sum_{i=1}^{10} a_i \gamma_1^i s(n - i) - \sum_{i=1}^{10} a_i \gamma_2^i sw(n - i) \qquad n = 0,...,39 \tag{33}$$

Le signal vocal pondéré $sw(n)$ est ensuite utilisé pour trouver une estimation du délai tonal dans la trame vocale.

3.4 Analyse tonale en boucle ouverte

Afin de diminuer la complexité de la recherche du meilleur délai par répertoire codé adaptatif, l'étendue d'exploration est limitée de part et d'autre d'une valeur possible de délai tonal, T_{op}, obtenue par analyse tonale en boucle ouverte. Cette analyse est faite à chaque trame (10 ms). L'estimation de tonie en boucle ouverte fait appel au signal vocal pondéré $sw(n)$ de l'équation (33), comme suit. Dans une première étape, on recherche 3 valeurs maximales de la corrélation suivante:

$$R(k) = \sum_{n=0}^{79} sw(n)\, sw(n - k) \tag{34}$$

dans les trois étendues suivantes:

$$i = 1: \ 80,...,143$$

$$i = 2: \ 40,...,79$$

$$i = 3: \ 20,...,39$$

Les valeurs maximales retenues, $R(t_i)$, $i = 1,...,3$, sont normalisées selon l'équation suivante:

$$R'(t_i) = \frac{R(t_i)}{\sqrt{\sum_n sw^2(n - t_i)}} \qquad i = 1,...,3 \tag{35}$$

On choisit la meilleure valeur entre les trois coefficients de corrélation normalisés en recherchant les meilleurs délais dans les valeurs de l'étendue inférieure. Pour cela, on pondère les corrélations normalisées qui correspondent aux délais les plus longs. On détermine le meilleur délai tonal en boucle ouverte, T_{op}, comme suit:

$$T_{op} = t_1$$

$$R'(T_{op}) = R'(t_1)$$

$$si\ R'(t_2) \geq 0{,}85R'(T_{op})$$

$$\qquad R'(T_{op}) = R'(t_2)$$

$$\qquad T_{op} = t_2$$

$$fin$$

$$si\ R'(t_3) \geq 0{,}85R'(T_{op})$$

$$\qquad R'(T_{op}) = R'(t_3)$$

$$\qquad T_{op} = t_3$$

$$fin$$

Cette procédure, consistant à subdiviser l'étendue temporelle en 3 sections et à rechercher les plus faibles valeurs de délai, permet d'éviter de choisir des multiples de la hauteur tonale.

3.5 Calcul de la réponse impulsionnelle

La réponse impulsionnelle $h(n)$ du filtre de synthèse pondérée $W(z)/\hat{A}(z)$ est nécessaire pour l'exploration des répertoires adaptatifs et fixes. On calcule, pour chaque sous-trame, la réponse impulsionnelle $h(n)$ en filtrant un signal composé des coefficients du filtre $A(z/\gamma_1)$ complété par une section de zéros en passant par les deux filtres $1/\hat{A}(z)$ et $1/A(z/\gamma_2)$.

3.6 Calcul du signal cible

Le signal cible $x(n)$ pour l'exploration du répertoire codé adaptatif est habituellement calculé par soustraction de la réponse à entrée nulle du filtre de synthèse pondérée $W(z)/\hat{A}(z) = A(z/\gamma_1)/[\hat{A}(z)A(z/\gamma_2)]$ à partir du signal vocal pondéré $sw(n)$ de l'équation (33), sous-trame par sous-trame.

Une procédure équivalente pour calculer le signal cible, utilisée dans la présente Recommandation, consiste à filtrer le signal résiduel de prédiction linéaire en le faisant passer par la combinaison du filtre de synthèse $1/\hat{A}(z)$ et du filtre de pondération $A(z/\gamma_1)/A(z/\gamma_2)$. Après détermination de l'excitation pour la sous-trame, les états initiaux de ces filtres sont mis à jour par filtrage de la différence entre signal résiduel et signal d'excitation. La mise à jour de la mémoire de ces filtres est expliquée au 3.10.

Le signal résiduel $r(n)$, qui est nécessaire pour trouver le vecteur cible, est également utilisé lors de l'exploration du répertoire codé adaptatif pour aller au-delà de la mémoire tampon contenant l'excitation déjà enregistrée. Cela simplifie la procédure d'exploration du répertoire codé de séquences adaptatives pour les délais inférieurs à la longueur d'une sous-trame de 40 échantillons, comme expliqué au paragraphe suivant. Le signal résiduel de prédiction linéaire est donné par l'équation suivante:

$$r(n) = s(n) + \sum_{i=1}^{10} \hat{a}_i s(n - i) \qquad n = 0,...,39 \tag{36}$$

3.7 Exploration du répertoire codé de séquences adaptatives

Les paramètres du répertoire codé de séquences adaptatives (ou paramètres tonaux) sont le délai tonal et le gain tonal. Dans la méthode de filtrage tonal par répertoire codé adaptatif, le signal d'excitation est répété pendant des intervalles inférieurs à la longueur de la sous-trame. Dans l'étape d'exploration du répertoire, l'excitation est complétée par le signal résiduel de prédiction linéaire afin de simplifier la recherche en boucle fermée. L'exploration du répertoire codé adaptatif est effectuée à chaque sous-trame (de 5 ms). Dans la première sous-trame, un délai tonal fractionnaire, T_1, est utilisé avec une résolution de 1/3 dans l'étendue [19 1/3, 84 2/3] et avec seulement des entiers dans l'étendue [85, 143]. Pour la deuxième sous-trame, un délai tonal fractionnaire T_2 avec une résolution de 1/3 est toujours utilisé dans l'étendue [$int(T_1)$ – 5 2/3, $int(T_1)$ + 4 2/3], où $int(T_1)$ est le module entier du délai tonal fractionnaire T_1 de la première sous-trame. Cette étendue est adaptée aux cas où le délai T_1 chevauche une des bornes de son étendue.

Pour chaque sous-trame, le délai optimal est déterminé par analyse en boucle fermée qui minimise l'erreur quadratique pondérée. Dans la première sous-trame, le délai T_1 est déterminé par exploration d'une petite étendue (6 échantillons) des valeurs de délai de part et d'autre du délai tonal en boucle ouverte, T_{op} (voir 3.4). Les bornes de l'étendue d'exploration, t_{min} et t_{max} sont définies comme suit:

$$t_{min} = T_{op} - 3$$

$$si\ t_{min} < 20\ alors\ t_{min} = 20$$

$$t_{max} = t_{min} + 6$$

$$si\ t_{max} > 143\ alors$$

$$\quad t_{max} = 143$$

$$\quad t_{min} = t_{max} - 6$$

$$fin$$

Pour la deuxième sous-trame, l'analyse tonale en boucle fermée est effectuée de part et d'autre du délai tonal sélectionné dans la première sous-trame, afin de trouver le délai optimal, T_2. L'étendue d'exploration est bornée par t_{min} – 2/3 et t_{max} + 2/3, où t_{min} et t_{max} sont des limites déduites de T_1 comme suit:

$$t_{min} = int(T_1) - 5$$

$$si\ t_{min} < 20\ alors\ t_{min} = 20$$

$$t_{max} = t_{min} + 9$$

$$si\ t_{max} > 143\ alors$$

$$\quad t_{max} = 143$$

$$\quad t_{min} = t_{max} - 9$$

$$fin$$

La recherche du délai tonal en boucle fermée minimise l'erreur quadratique pondérée entre le signal vocal original et le signal vocal reconstitué. Pour cela, on recherche la valeur maximale du terme suivant:

$$R(k) = \frac{\sum_{n=0}^{39} x(n)\, y_k(n)}{\sqrt{\sum_{n=0}^{39} y_k(n)\, y_k(n)}} \tag{37}$$

où $x(n)$ est le signal cible et où $y_k(n)$ est la dernière excitation filtrée au délai k [excitation précédente convoluée avec le signal $h(n)$]. On notera que l'étendue d'exploration est limitée de part et d'autre d'une valeur présélectionnée, qui est le délai tonal en boucle ouverte, T_{op}, pour la première sous-trame, et qui est T_1 pour la deuxième sous-trame.

La convolution de signaux $y_k(n)$ est calculée pour le délai t_{min}. Elle est ensuite mise à jour par la relation récurrente suivante pour les autres valeurs entières du délai dans l'étendue d'exploration $k = t_{min} + 1, \ldots. t_{max}$:

$$y_k(n) = y_{k-1}(n-1) + u(-k)\, h(n) \qquad n = 39,\ldots,0 \tag{38}$$

où $u(n)$, $n = -143,\ldots,39$, est le tampon d'excitation et où $y_{k-1}(-1) = 0$. On notera que, dans l'étape d'exploration, les échantillons $u(n)$, $n = 0,\ldots,39$ ne sont pas connus mais sont nécessaires pour les délais tonaux inférieurs à 40. Pour simplifier la recherche, le signal résiduel de prédiction linéaire est copié selon les échantillons $u(n)$ afin de rendre la relation de l'équation (38) valide pour tous les délais.

Pour la détermination des délais T_2 et T_1, si la valeur entière du délai optimal en boucle fermée est inférieure à 85, il faut tester les valeurs fractionnaires de part et d'autre de cette valeur entière. La recherche des valeurs tonales fractionnaires est effectuée par interpolation des résultats de la corrélation normalisée de l'équation (37) et recherche de sa valeur maximale. Cette interpolation s'effectue au moyen d'un filtre à réponse impulsionnelle finie (FIR) b_{12}, fondé sur une fonction sin $(x)/x$ (coupure à la fréquence de Nyquist dans une fenêtre de Hamming, le registre sin $(x)/x$ étant tronqué à $\pm$ 11 et complété de zéros jusqu'à $\pm$ 12 [$b_{12}(12) = 0$]. Ce filtre a sa fréquence de coupure fixée à 3600 Hz (–3 dB) dans le domaine suréchantillonné. Les valeurs interpolées du terme $R(k)$ sont obtenues par la formule d'interpolation suivante pour les fractions –2/3, –1/3, 0, 1/3 et 2/3:

$$R(k)_t = \sum_{i=0}^{3} R(k-i)\, b_{12}(t+3i) + \sum_{i=0}^{3} R(k+1+i)\, b_{12}(3-t+3i) \qquad t = 0, 1, 2 \tag{39}$$

où $t = 0, 1, 2$ correspond, respectivement, aux fractions 0, 1/3 et 2/3. On notera qu'il est nécessaire de calculer les termes de corrélation dans l'équation (37) en utilisant une étendue de $t_{min} - 4$ à $t_{max} + 4$, pour permettre d'effectuer l'interpolation correcte.

3.7.1 Construction du vecteur de répertoire codé adaptatif

Une fois que le délai tonal a été déterminé, on calcule le vecteur de répertoire codé adaptatif $v(n)$ en interpolant le précédent signal d'excitation $u(n)$ pour la valeur entière k du délai tonal indiqué et pour la fraction t indiquée:

$$v(n) = \sum_{i=0}^{9} u(n-k+i)\, b_{30}(t+3i) + \sum_{i=0}^{9} u(n-k+1+i)\, b_{30}(3-t+3i) \qquad n = 0,\ldots,39 \qquad t = 0, 1, 2 \tag{40}$$

Le filtre d'interpolation, b_{30}, est fondé sur une fonction sin $(x)/x$ (coupure à la fréquence de Nyquist) dans une fenêtre de Hamming, le registre sin $(x)/x$ étant tronqué à $\pm$ 29 et complété de zéros jusqu'à $\pm$ 30 [$b_{30}(30) = 0$]. Ce filtre a sa fréquence de coupure fixée à 3600 Hz (–3 dB) dans le domaine suréchantillonné.

3.7.2 Calcul du mot de code pour les délais de répertoire codé adaptatif

Le délai tonal T_1 est codé sur 8 bits dans la première sous-trame et le délai relatif est codé sur 5 bits dans la deuxième sous-trame. Un délai tonal fractionnaire, T, est représenté par son module entier $int(T)$ et par une partie fractionnaire $frac/3$, $frac = -1, 0, 1$. L'index tonal $P1$ est donc codé comme suit:

$$P1 = \begin{cases} 3(int(T_1) - 19) + frac - 1 & si\ T_1 = [19,...,85],\ frac = [-1, 0, 1] \\ (int(T_1) - 85) + 197 & si\ T_1 = [86,...,143],\ frac = 0 \end{cases} \tag{41}$$

La valeur du délai tonal T_2 est codée par rapport à celle de T_1. En utilisant la même interprétation que ci-dessus, on représente le délai fractionnaire T_2 par son module entier $int(T_2)$ et par une partie fractionnaire $frac/3$, $frac = -1, 0, 1$. L'index tonal $P2$ est donc codé comme suit:

$$P2 = 3(int(T_2) - t_{min}) + frac + 2 \tag{42}$$

où t_{min} est déduit de T_1 comme dans 3.7.

Pour rendre le codeur plus résistant aux erreurs aléatoires sur les bits, on calcule un bit de parité $P0$ d'après l'index de délai tonal $P1$ de la première sous-trame. Ce bit de parité est construit par une opération logique XOR (ou exclusif) sur les 6 bits les plus forts de l'index $P1$. Dans le décodeur, ce bit de parité est recalculé et si la valeur ainsi déterminée ne correspond pas à la valeur émise, on applique une procédure de masquage d'erreur.

3.7.3 Calcul du gain par répertoire codé adaptatif

Une fois que l'on a déterminé le délai tonal par répertoire codé adaptatif, on calcule le gain par répertoire codé adaptatif au moyen de l'équation suivante:

$$g_p = \frac{\sum_{n=0}^{39} x(n)\, y(n)}{\sum_{n=0}^{39} y(n)\, y(n)} \qquad \text{entre les limites } 0 \le g_p \le 1{,}2 \tag{43}$$

où $x(n)$ est le signal cible et $y(n)$ le code vectoriel de répertoire adaptatif filtré [réponse à l'état zéro du filtre $W(z)/\hat{A}(z)$ au signal $v(n)$]. Ce vecteur est obtenu par convolution des signaux $v(n)$ et $h(n)$ comme suit:

$$y(n) = \sum_{i=0}^{n} v(i)h(n - i) \qquad n = 0,...,39 \tag{44}$$

3.8 Structure et exploration du répertoire codé de séquences fixes

Le répertoire codé de séquences fixes est fondé sur une structure algébrique utilisant un modèle de permutation entrelacée d'impulsions (ISPP) (*interleaved single-pulse permutation*) de Dirac. Dans ce répertoire codé, chaque vecteur contient 4 impulsions non nulles, dont chacune peut avoir l'amplitude +1 ou –1, avec les positions indiquées dans le Tableau 7.

TABLEAU 7/G.729

Structure du répertoire codé fixe $\mathcal{C}$

Impulsion	Signe	Positions
i_0	s_0: ± 1	m_0: 0, 5, 10, 15, 20, 25, 30, 35
i_1	s_1: ± 1	m_1: 1, 6, 11, 16, 21, 26, 31, 36
i_2	s_2: ± 1	m_2: 2, 7, 12, 17, 22, 27, 32, 37
i_3	s_3: ± 1	m_3: 3, 8, 13, 18, 23, 28, 33, 38 4, 9, 14, 19, 24, 29, 34, 39

On construit le vecteur de répertoire codé fixe $c(n)$ en prenant un vecteur zéro de dimension 40 et en mettant les 4 impulsions unités (de Dirac) aux positions trouvées, multipliées avec leur signe correspondant:

$$c(n) = s_0\delta(n - m_0) + s_1\delta(n - m_1) + s_2\delta(n - m_2) + s_3\delta(n - m_3) \quad n = 0,...,39 \tag{45}$$

où $\delta(0)$ est une impulsion unité. Le répertoire codé fixe possède la caractéristique particulière que le code vectoriel choisi dans le répertoire passe par un préfiltre adaptatif $P(z)$ qui renforce les composantes harmoniques pour améliorer la qualité du signal reconstitué. On utilise à cette fin le filtre suivant:

$$P(z) = 1 / (1 - \beta z^{-T}) \tag{46}$$

où T est la composante entière du délai tonal pour la sous-trame actuelle et où β est un gain tonal. La valeur de β est rendue adaptative par utilisation de la valeur quantifiée du gain de répertoire adaptatif issu de la sous-trame précédente, soit:

$$\beta = \hat{g}_p^{(m-1)} \qquad \text{entre les bornes } 0{,}2 \le \beta \le 0{,}8 \tag{47}$$

Pour les délais inférieurs à 40, le signal de répertoire $c(n)$ de l'équation (45) est modifié comme suit:

$$c(n) = \begin{cases} c(n) & n = 0,...,T - 1 \\ c(n) + \beta c(n - T) & n = T,...,39 \end{cases} \tag{48}$$

Cette modification est incorporée dans l'exploration du répertoire codé de séquences fixes par modification de la réponse impulsionnelle $h(n)$ conformément à l'équation suivante:

$$h(n) = \begin{cases} h(n) & n = 0,...,T - 1 \\ h(n) + \beta h(n - T) & n = T,...,39 \end{cases} \tag{49}$$

3.8.1 Procédure d'exploration du répertoire codé de séquences fixes

On explore le répertoire codé fixe en minimisant l'erreur quadratique mesurée entre le signal vocal d'entrée pondéré $sw(n)$ de l'équation (33) et le signal vocal pondéré reconstitué. Le signal cible utilisé lors de l'exploration tonale en boucle fermée est mis à jour par soustraction de la contribution du répertoire codé adaptatif, c'est-à-dire:

$$x'(n) = x(n) - g_p y(n) \qquad n = 0,...,39 \tag{50}$$

où $y(n)$ est le vecteur vectoriel filtré du répertoire codé adaptatif selon l'équation (44) et où g_p est le gain par répertoire codé adaptatif selon l'équation (43).

La matrice **H** est définie comme étant la matrice convolutionnelle triangulaire inférieure de Toeplitz avec une diagonale principale $h(0)$ et des sous-diagonales $h(1),...,h(39)$. La matrice $\Phi = \mathbf{H}^t\mathbf{H}$ contient les corrélations de la réponse $h(n)$ et les éléments de cette matrice symétrique sont donnés par l'équation suivante:

$$\phi(i, j) = \sum_{n=j}^{39} h(n - i)h(n - j) \qquad i = 0,...,39 \quad j = i,...,39 \tag{51}$$

On obtient le signal de corrélation $d(n)$ à partir du signal cible $x'(n)$ et de la réponse impulsionnelle $h(n)$ comme suit:

$$d(n) = \sum_{i=n}^{39} x'(i)h(i - n) \qquad n = 0,...,39 \tag{52}$$

Si c_k est le k^e code vectoriel du répertoire codé fixe, on explore ce répertoire en maximisant le terme:

$$\frac{C_k^2}{E_k} = \frac{\left(\sum_{n=0}^{39} d(n)c_k(n)\right)^2}{c_k^t \Phi c_k} \tag{53}$$

où t indique une opération de transposition.

Le signal $d(n)$ et la matrice Φ sont calculés avant l'exploration du répertoire. On notera que l'on ne calcule que les éléments réellement nécessaires et qu'une procédure de mémorisation efficace a été mise au point pour accélérer le processus d'exploration.

La structure algébrique du répertoire codé $\mathcal{C}$ permet une procédure d'exploration rapide car le code vectoriel c_k du répertoire ne contient que quatre impulsions non nulles. Au numérateur de l'équation (53), la corrélation pour un vecteur c_k donné est calculée comme suit:

$$C = \sum_{i=0}^{3} s_i d(m_i) \tag{54}$$

où m_i est la position de la $i$ème impulsion et s_i son amplitude. L'énergie correspondant au dénominateur de l'équation (53) est donnée par l'équation suivante:

$$E = \sum_{i=0}^{3} \phi(m_i, m_i) + 2 \sum_{i=0}^{2} \sum_{j=i+1}^{3} s_i s_j \phi(m_i, m_j) \tag{55}$$

Pour simplifier la procédure d'exploration, on prédétermine les amplitudes des impulsions en quantifiant le signal $d(n)$. Pour cela, on aligne l'amplitude d'une impulsion située à une certaine position sur le signe du signal $d(n)$ correspondant à cette position. Avant l'exploration du répertoire codé, les étapes suivantes sont suivies. Tout d'abord, le signal $d(n)$ est décomposé en deux parties: sa valeur absolue $|d(n)|$ et son signe, $sign[d(n)]$. Ensuite, on modifie la matrice Φ en y introduisant les informations de signe, c'est-à-dire:

$$\phi'(i, j) = sign\,[d(i)]\; sign\,[d(j)]\phi(i, j) \qquad i = 0,...,39 \quad j = i + 1,...,39 \tag{56}$$

Les éléments situés sur la diagonale principale de la matrice Φ sont normalisés afin d'éliminer le facteur 2 introduit dans l'équation (55):

$$\phi'(i, i) = 0{,}5\phi'(i, i) \qquad i = 0,...,39 \tag{57}$$

La corrélation de l'équation (54) est alors donnée par :

$$C = |d(m_0)| + |d(m_1)| + |d(m_2)| + |d(m_3)| \tag{58}$$

et l'énergie définie dans l'équation (55) est donnée par:

$$\begin{aligned} E/2 = {} & \phi'(m_0, m_0) \\ & + \phi'(m_1, m_1) + \phi'(m_0, m_1) \\ & + \phi'(m_2, m_2) + \phi'(m_0, m_2) + \phi'(m_1, m_2) \\ & + \phi'(m_3, m_3) + \phi'(m_0, m_3) + \phi'(m_1, m_3) + \phi'(m_2, m_3) \end{aligned} \tag{59}$$

Une méthode de recherche focalisée est utilisée afin de simplifier encore la procédure de recherche. Dans cette méthode, on teste un seuil précalculé avant d'entamer la dernière boucle, celle-ci n'étant lancée que si ce seuil est dépassé. Le nombre maximal de fois où la boucle peut être lancée est fixe, de sorte que l'on n'explore qu'un faible pourcentage du répertoire codé. Ce seuil est calculé sur la base du coefficient de corrélation C. Avant l'exploration du répertoire codé, on détermine la corrélation absolue maximale, max_3, et la corrélation moyenne, av_3, dues à la contribution des trois premières impulsions. Le seuil est donné par la somme suivante:

$$thr_3 = av_3 + K_3(max_3 - av_3) \tag{60}$$

La quatrième boucle n'est lancée que si la corrélation absolue (due à trois impulsions) dépasse le seuil thr_3, avec $0 \le K_3 < 1$. La valeur du facteur K_3, fixée ici à 0,4, règle le pourcentage d'exploration du répertoire codé. On notera que cela aboutit à une durée d'exploration variable. Pour limiter encore plus la recherche, le nombre de lancements de la dernière boucle (pour les 2 sous-trames) ne peut pas dépasser une certaine valeur maximale, qui est ici fixée à 180 (pour 1 sous-trame, le pire cas est de 90 lancements en moyenne).

3.8.2 Calcul des mots de code du répertoire codé de séquences fixes

Les positions des impulsions i_0, i_1 et i_2 sont codées sur 3 bits chacune, alors que la position de l'impulsion i_3 est codée sur 4 bits. Chaque amplitude d'impulsion est codée sur 1 bit, ce qui donne un total de 17 bits pour les 4 impulsions. Si l'on définit $s = 1$ si le signe est positif et $s = 0$ s'il est négatif, le mot de code du signe est donné par:

$$S = s_0 + 2s_1 + 4s_2 + 8s_3 \tag{61}$$

et le mot de code du répertoire codé fixe est obtenu comme suit:

$$C = (m_0/5) + 8(m_1/5) + 64(m_2/5) + 512(2(m_3/5) + jx) \tag{62}$$

où $jx = 0$ si $m_3 = 3,8,...,38$, et où $jx = 1$ si $m_3 = 4,9,...,39$.

3.9 Quantification des gains

Le gain par répertoire codé adaptatif (gain tonal) et le gain par répertoire codé fixe sont quantifiés en vecteurs sur 7 bits. L'exploration du répertoire codé de gain consiste à minimiser l'erreur quadratique pondérée entre le signal vocal original et le signal vocal reconstitué, comme suit:

$$E = \mathbf{x}^t\mathbf{x} + g_p^2\,\mathbf{y}^t\mathbf{y} + g_c^2\,\mathbf{z}^t\mathbf{z} - 2g_p\mathbf{x}^t\mathbf{y} - 2g_c\mathbf{x}^t\mathbf{z} + 2g_pg_c\mathbf{y}^t\mathbf{z} \tag{63}$$

où x est le vecteur cible (voir 3.6), y est le code vectoriel filtré du répertoire adaptatif selon l'équation (44), et z le code vectoriel du répertoire fixe convolué comme suit avec la réponse $h(n)$:

$$z(n) = \sum_{i=0}^{n} c(i)h(n - i) \qquad n = 0,...,39 \tag{64}$$

3.9.1 Prédiction des gains

On peut exprimer comme suit le gain par répertoire codé fixe, g_c:

$$g_c = \gamma g'_c \tag{65}$$

où g'_c est un gain prédit qui est fondé sur les énergies des séquences précédentes du répertoire codé fixe et où γ est un facteur de correction.

L'énergie moyenne de la contribution du répertoire codé fixe est donnée par:

$$E = 10 \log \left(\frac{1}{40} \sum_{n=0}^{39} c(n)^2 \right) \tag{66}$$

Après normalisation du vecteur $c(n)$ avec le gain de répertoire fixe g_c, l'énergie du répertoire fixe normalisé est donnée par l'expression 20 log $g_c + E$. Soit $E^{(m)}$ l'énergie (en décibels) après suppression de la moyenne de la contribution (normalisée) du répertoire fixe à la sous-trame m, donnée par:

$$E^{(m)} = 20 \log g_c + E - \bar{E} \tag{67}$$

où $\bar{E}$ = 30 dB est l'énergie moyenne de l'excitation par répertoire codé fixe. Le gain g_c peut être exprimé en fonction des énergies $E^{(m)}$, E et $\bar{E}$ par:

$$g_c = 10^{(E^{(m)} + \bar{E} - E)/20} \tag{68}$$

On détermine le gain prédit g'_c en prédisant l'énergie logarithmique de la contribution actuelle du répertoire codé fixe, d'après l'énergie logarithmique des précédentes contributions de ce répertoire. On effectue comme suit la prédiction par moyenne mobile du 4e ordre. L'énergie prédite est donnée par:

$$\tilde{E}^{(m)} = \sum_{i=1}^{4} b_i \hat{U}^{(m-i)} \tag{69}$$

où $[b_1\ b_2\ b_3\ b_4]$ = [0,68 0,58 0,34 0,19] sont les coefficients de prédiction par moyenne mobile et où $\hat{U}^{(m)}$ est la version quantifiée de l'erreur de prédiction $U^{(m)}$ à la sous-trame m, cette erreur étant définie comme suit:

$$U^{(m)} = E^{(m)} - \tilde{E}^{(m)} \tag{70}$$

On trouve le gain prédit, g'_c, en remplaçant $E^{(m)}$ par sa valeur prédite dans l'équation (68):

$$g'_c = 10^{(\tilde{E}^{(m)} + \bar{E} - E)/20} \tag{71}$$

Le facteur de correction γ est associé comme suit à l'erreur sur la prédiction de gain:

$$U^{(m)} = E^{(m)} - \tilde{E}^{(m)} = 20 \log(\gamma) \tag{72}$$

3.9.2 Exploration du répertoire pour quantifier les gains

Le gain par répertoire codé adaptatif, g_p, et le facteur γ sont quantifiés vectoriellement au moyen d'un répertoire codé à structure conjuguée en deux étapes. La première étape est formée d'un répertoire codé à deux dimensions sur 3 bits, $\mathcal{GA}$, et la deuxième étape est formée d'un répertoire codé à deux dimensions sur 4 bits, $\mathcal{GB}$. Le premier élément de chaque répertoire représente le gain par répertoire adaptatif quantifié, $\hat{g}_p$, et le deuxième élément représente le facteur de correction $\hat{\gamma}$ du gain par répertoire adaptatif quantifié. Si l'on a les index GA et GB pour, respectivement, les répertoires $\mathcal{GA}$ et $\mathcal{GB}$, le gain par répertoire codé adaptatif quantifié est donné par:

$$\hat{g}_p = \mathcal{GA}_1(GA) + \mathcal{GB}_1(GB) \tag{73}$$

et le gain par répertoire codé fixe quantifié est donné par:

$$\hat{g}_c = g'_c\hat{\gamma} = g'_c(\mathcal{GA}_2(GA) + \mathcal{GB}_2(GB)) \tag{74}$$

Cette structure conjuguée simplifie l'exploration du répertoire codé car elle permet d'appliquer un procédé de présélection. Le gain tonal optimal, g_p, et le gain par répertoire fixe, g_c, sont déduits de l'équation (63) et sont utilisés pour la présélection. Le répertoire codé $\mathcal{GA}$ contient 8 entrées dans lesquelles le second élément (correspondant à g_c) possède en général des valeurs plus grandes que le premier élément (qui correspond à g_p). Cette dissymétrie permet d'effectuer une présélection au moyen de la valeur g_c. Au cours de ce processus de présélection, on choisit une grappe de 4 vecteurs dont le deuxième élément est proche de g_p. De même, le répertoire codé $\mathcal{GB}$ contient 16 entrées tendant à se rapprocher du premier élément (qui correspond à g_p). On choisit une grappe de 8 vecteurs dont le premier élément est proche de g_p. Dans chaque répertoire codé, on choisit donc 50% des vecteurs les plus proches de la valeur optimale. Cette sélection est suivie d'une exploration complète des $4 \times 8 = 32$ autres possibilités restantes, de manière que la combinaison des deux index minimise l'erreur quadratique pondérée de l'équation (63).

3.9.3 Calcul des mots de code pour le quantificateur de gain

Les mots de code *GA* et *GB* pour le quantificateur de gain sont obtenus à partir des index correspondant au meilleur choix. Pour réduire l'influence des erreurs au niveau des éléments binaires, les index de répertoire sont associés à une structure particulière.

3.10 Mise à jour de la mémoire

Une mise à jour des états des filtres de synthèse et de pondération est nécessaire pour calculer le signal cible dans la sous-trame suivante. Une fois les deux gains quantifiés, on obtient le signal d'excitation, $u(n)$, dans la sous-trame présente, au moyen de la relation suivante:

$$u(n) = \hat{g}_p v(n) + \hat{g}_c c(n) \qquad n = 0,...,39 \tag{75}$$

où $\hat{g}_p$ et $\hat{g}_c$ sont, respectivement, le gain quantifié par répertoire adaptatif et le gain quantifié par répertoire fixe, où $v(n)$ est le code vectoriel du répertoire adaptatif (après interpolation de l'excitation antérieure), et où $c(n)$ est le code vectoriel du répertoire fixe après renforcement des composantes harmoniques. On peut mettre à jour les états des filtres en faisant passer le signal différentiel $r(n) - u(n)$ (entre signal résiduel et signal d'excitation) par les filtres $1/\hat{A}(z)$ et $A(z/\gamma_1)/A(z/\gamma_2)$ pour la sous-trame de 40 échantillons, puis en sauvegardant les états de ces filtres. Cela nécessitera 3 opérations sur les filtres. Une méthode plus simple, n'exigeant qu'une seule opération sur filtre, est la suivante. Le signal vocal reconstitué localement, $\hat{s}(n)$ est calculé par filtrage du signal d'excitation par $1/\hat{A}(z)$. La sortie du filtre, pour le signal d'entrée $r(n) - u(n)$ est équivalente à $e(n) = s(n) - \hat{s}(n)$. Les états du filtre de synthèse $1/\hat{A}(z)$ seront donc indiqués par le signal $e(n)$, $n = 30,...,39$. On peut effectuer la mise à jour des états du filtre $A(z/\gamma_1)/A(z/\gamma_2)$ en injectant le signal d'erreur $e(n)$ dans ce filtre afin d'obtenir l'erreur à pondération perceptive, $ew(n)$. Ce dernier signal peut également être trouvé par l'opération suivante:

$$ew(n) = x(n) - \hat{g}_p y(n) - \hat{g}_c z(n) \tag{76}$$

Etant donné que les signaux $x(n)$, $y(n)$ et $z(n)$ sont disponibles, on met à jour les états du filtre de pondération en calculant l'erreur $ew(n)$ comme dans l'équation (76) pour $n = 30,...,39$, ce qui économise deux opérations sur les filtres.

4 Description fonctionnelle du décodeur

Le principe du décodeur a été décrit dans l'article 2 (Figure 3). On décode d'abord les paramètres (coefficients LP, code vectoriel du répertoire adaptatif, code vectoriel du répertoire fixe, et gains). Les paramètres transmis sont énumérés dans le Tableau 8. Ces paramètres, une fois décodés, servent à calculer le signal vocal reconstitué, comme décrit dans 4.1. Ce signal reconstitué est amélioré par une opération de post-traitement qui met en œuvre un postfiltre, un filtre passe-haut et un suréchantillonnage (voir 4.2). Le paragraphe 4.4 décrit la procédure de masquage d'erreur qui est utilisée lorsqu'une erreur de parité s'est produite ou lorsque le fanion d'effacement de trame a été activé. La Figure 6 montre sous forme schématique détaillée le flux de signaux du décodeur.

TABLEAU 8/G.729

Description des index paramétriques transmis – L'ordonnancement du flux binaire correspond à l'ordre du tableau – Pour chaque paramètre, le bit de poids fort (MSB) est émis en premier

Symbole	Description	Bits
$L0$	Prédicteur à moyenne mobile périodique pour le quantificateur de paires LSP	1
$L1$	Vecteur de première étape du quantificateur de paires LSP	7
$L2$	Vecteur inférieur de seconde étape du quantificateur LSP	5
$L3$	Vecteur supérieur de seconde étape du quantificateur LSP	5
$P1$	Première sous-trame pour le délai tonal	8
$P0$	Bit de parité pour le délai tonal	1
$C1$	Première sous-trame de répertoire codé fixe	13
$S1$	Signes des impulsions de répertoire fixe dans la 1ère sous-trame	4
$GA1$	Première sous-trame du répertoire de gain (étape 1)	3
$GB1$	Première sous-trame du répertoire de gain (étape 2)	4
$P2$	Deuxième sous-trame du délai tonal	5
$C2$	Deuxième sous-trame du répertoire codé fixe	13
$S2$	Signes des impulsions de répertoire fixe dans la 2e sous-trame	4
$GA2$	Deuxième sous-trame du répertoire de gain (étape 1)	3
$GB2$	Deuxième sous-trame du répertoire de gain (étape 2)	4

4.1 Procédure de décodage des paramètres

Le processus de décodage s'effectue dans l'ordre ci-après.

4.1.1 Décodage des paramètres du filtre d'analyse par prédiction linéaire

Les indices $L0$, $L1$, $L2$ et $L3$ reçus du quantificateur de coefficients LSP servent à reconstruire les coefficients LSP quantifiés au moyen de la procédure décrite au 3.2.4. La procédure d'interpolation décrite au 3.2.5 est utilisée pour obtenir 2 ensembles de coefficients LSP interpolés (correspondant à 2 sous-trames). Pour chaque sous-trame, les coefficients LSP interpolés sont convertis en coefficients de filtre LP, a_i, qui sont utilisés pour synthétiser le signal vocal reconstitué dans la sous-trame.

Les étapes suivantes sont répétées pour chaque sous-trame:

1) décodage du code vectoriel de répertoire adaptatif;
2) décodage du code vectoriel de répertoire fixe;
3) décodage des gains par répertoire adaptatif et par répertoire fixe;
4) calcul du signal vocal reconstitué.

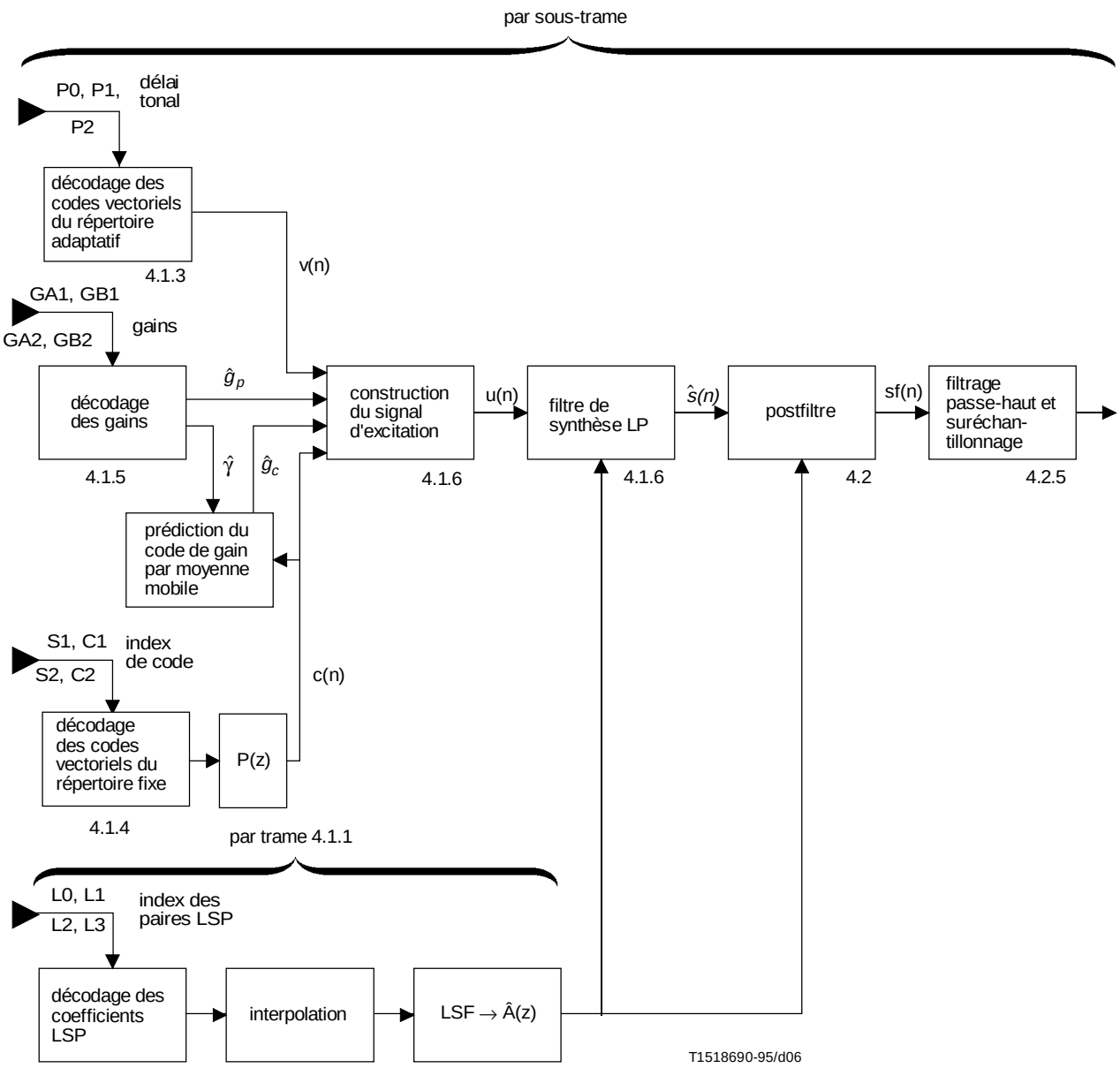

FIGURE 6/G.729

Circulation des signaux dans le codeur CS-ACELP

4.1.2 Calcul du bit de parité

Avant la reconstruction du signal d'excitation, le bit de parité est recalculé à partir de l'index $P1$ (voir 3.7.2) pointant sur le délai codé par répertoire adaptatif. Si ce bit n'est pas identique au bit de parité transmis, $P0$, il est probable que des erreurs binaires se soient produites au cours de la transmission.

Si une erreur de parité se produit sur l'index $P1$, la valeur de délai T_1 est réglée sur la partie entière de la valeur de délai T_2 de la trame précédente. La valeur du délai T_2 est calculée par la procédure décrite au 4.1.3, au moyen de cette nouvelle valeur de T_1.

4.1.3 Décodage du vecteur de répertoire codé adaptatif

Si aucune erreur de parité ne s'est produite, l'index $P1$ qui a été reçu au sujet du répertoire codé adaptatif est utilisé pour trouver les parties entière et fractionnaire du délai tonal T_1. La partie entière $int(T_1)$ et la partie fractionnaire $frac$ du délai T_1 s'obtiennent à partir de $P1$ comme suit:

$$
\begin{aligned}
&si\ P1 < 197\\
&\qquad int(T_1) = (P1 + 2)/3 + 19\\
&\qquad frac = P1 - 3\,int(T_1) + 58\\
&ou\ bien\\
&\qquad int(T_1) = P1 - 112\\
&\qquad frac = 0\\
&fin
\end{aligned}
$$

Les parties entière et fractionnaire du délai T_2 s'obtiennent à partir de $P2$ et de t_{min}, où t_{min} est déduit de T_1 comme suit:

$$
\begin{aligned}
&t_{min} = int(T_1) - 5\\
&si\ t_{min} < 20\ alors\ t_{min} = 20\\
&t_{max} = t_{min} + 9\\
&si\ t_{max} > 143\ alors\\
&\qquad t_{max} = 143\\
&\qquad t_{min} = t_{max} - 9\\
&fin
\end{aligned}
$$

On décode ensuite le délai T_2 en utilisant:

$$
\begin{aligned}
int(T_2) &= (P2 + 2)/3 - 1 + t_{min}\\
frac &= P2 - 2 - 3\,((P2 + 2)/3 - 1)
\end{aligned}
$$

Le code vectoriel de répertoire adaptatif $v(n)$ est obtenu par interpolation de l'excitation antérieure $u(n)$ (au délai tonal) au moyen de l'équation (40).

4.1.4 Décodage du vecteur de répertoire codé fixe

L'index C reçu, pointant sur le répertoire fixe, est utilisé pour extraire les positions des impulsions d'excitation, dont les signes sont déduits de S par inversion du processus décrit dans 3.8.2. Une fois les positions des impulsions et leurs signes décodés, on construit le code vectoriel du répertoire fixe, $c(n)$, au moyen de l'équation (45). Si la partie entière du délai tonal T est inférieure à la longueur de la sous-trame (40), le code $c(n)$ est modifié selon l'équation (48).

4.1.5 Décodage des gains par répertoire codé adaptatif et par répertoire codé fixe

L'index reçu du répertoire de gain pointe sur le gain $\hat{g}_p$ par répertoire codé adaptatif ainsi que sur le facteur de correction du gain par répertoire codé fixe, $\hat{\gamma}$. Cette procédure est décrite en détail au 3.9. Le gain estimé par répertoire codé fixe, g'_c, est trouvé au moyen de l'équation (71). Le code vectoriel du répertoire fixe est obtenu à partir du produit du facteur de correction de gain quantifié et de ce gain prédit [équation (74)]. Le gain par répertoire codé adaptatif est reconstitué au moyen de l'équation (73).

4.1.6 Calcul des signaux vocaux reconstruits

Le signal d'excitation $u(n)$ [voir l'équation (75)] est injecté dans le filtre de synthèse LP. Le signal vocal reconstitué pour la sous-trame est donné par:

$$\hat{s}(n) = u(n) - \sum_{i=1}^{10} \hat{a}_i \hat{s}(n - i) \qquad n = 0,...,39 \tag{77}$$

où les $\hat{a}_i$ sont les coefficients de filtre LP interpolés pour la sous-trame actuelle. Le signal vocal reconstitué $\hat{s}(z)$ est ensuite envoyé dans le postprocesseur décrit dans le paragraphe suivant.

4.2 Post-traitement

Le post-traitement se compose de trois fonctions: le postfiltrage adaptatif, le filtrage passe-haut et le suréchantillonnage du signal. Le postfiltre adaptatif est le montage en cascade de trois filtres: un postfiltre à long terme $H_p(z)$, un postfiltre à court terme $H_f(z)$ et un filtre de compensation des écarts de niveau à court terme, $H_t(z)$, suivis d'une procédure de commande du gain adaptatif. Les coefficients du postfiltre sont mis à jour toutes les sous-trames de 5 ms. Le processus de postfiltrage est organisé comme suit. Tout d'abord, le signal vocal reconstitué $\hat{s}(n)$ subit un filtrage inverse dans $\hat{A}(z/\gamma_n)$ afin de produire le signal résiduel $\hat{r}(n)$. Ce signal est utilisé pour calculer le délai T et le gain g_l du postfiltre à long terme $H_p(z)$. Le signal $\hat{r}(n)$ est ensuite filtré par le postfiltre à long terme $H_p(z)$ et par le filtre de synthèse $1/[g_f\hat{A}(z/\gamma_d)]$. Finalement, le signal de sortie du filtre de synthèse $1/[g_f\hat{A}(z/\gamma_d)]$ est envoyé dans le filtre de compensation des écarts de niveau $H_t(z)$, afin de produire le signal vocal reconstitué $sf(n)$. La commande de gain adaptatif est ensuite appliquée au signal $sf(n)$ pour correspondre à l'énergie du signal $\hat{s}(n)$. Le signal résultant $sf'(n)$ subit un filtrage passe-haut et une normalisation par suréchantillonnage afin de produire le signal de sortie du décodeur.

4.2.1 Postfiltre à long terme

Le postfiltre à long terme est donné par l'équation suivante:

$$H_p(z) = \frac{1}{1 + \gamma_p g_l}(1 + \gamma_p g_l z^{-T}) \tag{78}$$

où T est le délai tonal et g_l le coefficient de gain. On notera que le coefficient g_l est limité par 1 et qu'il est mis à zéro si le gain de prédiction à long terme est inférieur à 3 dB. Le facteur γ_p commande l'intensité du postfiltrage à long terme et a la valeur $\gamma_p = 0{,}5$. Le délai et le gain à long terme sont calculés à partir du signal résiduel $\hat{r}(n)$ obtenu par filtrage du signal vocal $\hat{s}(n)$ par le filtre $\hat{A}(z/\gamma_n)$, qui est le numérateur du postfiltre à court terme (voir 4.2.2):

$$\hat{r}(n) = \hat{s}(n) + \sum_{i=1}^{10} \gamma_n^i \hat{a}_i \hat{s}(n - i) \tag{79}$$

Le délai à long terme se calcule au moyen d'une procédure en deux étapes. La première consiste à sélectionner le meilleur entier T_0 dans la gamme $[int(T_1) - 1,\ int(T_1) + 1]$, où le terme $int(T_1)$ est la partie entière du délai tonal T_1 transmis dans la première sous-trame. Le meilleur nombre entier correspondant au délai est celui qui maximise la corrélation:

$$R(k) = \sum_{n=0}^{39} \hat{r}(n)\hat{r}(n - k) \tag{80}$$

La deuxième étape consiste à choisir le meilleur délai fractionnaire T de résolution 1/8 autour de T_0. On effectue ce choix en trouvant le délai qui maximise la corrélation pseudo-normalisée suivante:

$$R'(k) = \frac{\sum_{n=0}^{39} \hat{r}(n)\hat{r}_k(n)}{\sqrt{\sum_{n=0}^{39} \hat{r}_k(n)\hat{r}_k(n)}} \tag{81}$$

où $\hat{r}_k(n)$ est le signal résiduel au délai k. Une fois le délai optimal T trouvé, la corrélation $R'(T)$ correspondante est normalisée avec la racine carrée de l'énergie du signal $\hat{r}(n)$. Le carré de cette corrélation normalisée est utilisé pour déterminer s'il y a lieu d'inactiver le postfiltre à long terme, ce que l'on fait en réglant $g_l = 0$ si

$$\frac{R'(T)^2}{\sum_{n=0}^{39} \hat{r}(n)\hat{r}(n)} < 0{,}5 \tag{82}$$

Sinon, la valeur de g_l est calculée à partir de:

$$g_l = \frac{\sum_{n=0}^{39} \hat{r}(n)\hat{r}_k(n)}{\sum_{n=0}^{39} \hat{r}_k(n)\hat{r}_k(n)} \qquad \text{entre les limites } 0 \le g_l \le 1{,}0 \tag{83}$$

La valeur non entière du signal retardé $\hat{r}_k(n)$ est d'abord calculée au moyen d'un filtre d'interpolation de longueur 33. Après la sélection de T, le signal $\hat{r}_k(n)$ est recalculé au moyen d'un filtre d'interpolation plus long (129). Le nouveau signal ne remplace le précédent que si le filtre long augmente la valeur de $R'(T)$.

4.2.2 Postfiltre à court terme

Le postfiltre à court terme est donné par l'équation suivante:

$$H_f(z) = \frac{1}{g_f} \frac{\hat{A}(z/\gamma_n)}{\hat{A}(z/\gamma_d)} = \frac{1}{g_f} \frac{1 + \sum_{i=1}^{10} \gamma_n^i \hat{a}_i z^{-i}}{1 + \sum_{i=1}^{10} \gamma_d^i \hat{a}_i z^{-i}} \tag{84}$$

où $\hat{A}(z)$ est le filtre inverse LP des valeurs quantifiées reçues (l'analyse de prédiction linéaire n'est pas effectuée dans le décodeur) et où les facteurs γ_n et γ_d déterminent l'intensité du postfiltrage à court terme et sont réglés aux valeurs suivantes: $\gamma_n = 0{,}55$ et $\gamma_d = 0{,}7$. Le terme de gain g_f est calculé sur la réponse impulsionnelle tronquée $h_f(n)$ du filtre $\hat{A}(z/\gamma_n)/\hat{A}(z/\gamma_d)$ et est donné par:

$$g_f = \sum_{n=0}^{19} |h_f(n)| \tag{85}$$

4.2.3 Compensation des écarts de niveau à court terme

Le filtre $H_t(z)$ compense les écarts de niveau dans le postfiltre à court terme $H_f(z)$ et est donné par:

$$H_t(z) = \frac{1}{g_t} (1 + \gamma_t k_1' z^{-1}) \tag{86}$$

où $\gamma_t k_1'$ est un facteur d'écart, k_1' étant le premier coefficient de réflexion calculé à partir de $H_f(n)$ avec:

$$k_1' = -\frac{r_h(1)}{r_h(0)} \qquad r_h(i) = \sum_{j=0}^{19-i} h_f(j)\, h_f(j + i) \tag{87}$$

Le terme de gain $g_t = 1 - |\gamma_t k_1'|$ compense l'effet décroissant de g_f dans le filtre $H_f(z)$. En outre, on a démontré que le filtre combiné $H_f(z)H_t(z)$ n'avait généralement pas de gain. On utilise deux valeurs de γ_t selon le signe de k_1'. Si k_1' a une valeur négative, $\gamma_t = 0{,}9$ et si k_1' a une valeur positive, $\gamma_t = 0{,}2$.

4.2.4 Commande de gain adaptatif

La commande de gain adaptatif est utilisée pour compenser les différences de gain entre le signal vocal reconstruit $\hat{s}(n)$ et le signal postfiltré $sf(n)$. Le facteur de normalisation par le gain, G, est calculé comme suit pour la sous-trame actuelle:

$$G = \frac{\sum_{n=0}^{39} |\hat{s}(n)|}{\sum_{n=0}^{39} |sf(n)|} \tag{88}$$

Le signal normalisé en gain et postfiltré, $sf'(n)$, est donné par:

$$sf'(n) = g^{(n)} sf(n) \qquad n = 0,...,39 \tag{89}$$

où le terme $g^{(n)}$ est mis à jour échantillon par échantillon et est donné par:

$$g^{(n)} = 0{,}85\, g^{(n-1)} + 0{,}15\, G \qquad n = 0,...,39 \tag{90}$$

La valeur initiale de $g^{(-1)} = 1{,}0$ est utilisée. Puis, pour chaque nouvelle sous-trame, la valeur $g^{(-1)}$ est rendue égale à $g^{(39)}$ de la sous-trame précédente.

4.2.5 Filtrage passe-haut et suréchantillonnage

On applique un filtre passe-haut ayant une fréquence de coupure de 100 Hz au signal vocal postfiltré et reconstitué, $sf'(n)$. Ce filtre est donné par:

$$H_{h2}(z) = \frac{0{,}93980581 - 1{,}8795834z^{-1} + 0{,}93980581z^{-2}}{1 - 1{,}9330735z^{-1} + 0{,}93589199z^{-2}} \tag{91}$$

Le signal filtré est multiplié par un facteur 2 afin de rétablir le niveau du signal d'entrée.

4.3 Initialisation du codeur et du décodeur

Il y a lieu d'initialiser à 0 toutes les variables statiques du codeur et du décodeur, sauf celles qui sont énumérées dans le Tableau 9.

TABLEAU 9/G.729

Description des paramètres avec initialisation différente de zéro

Variable	Référence	Valeur initiale
β	3.8	0,8
$g^{(-1)}$	4.2.4	1,0
$\hat{l}_i$	3.2.4	$i\pi/11$
q_i	3.2.4	$\arccos(i\pi/11)$
$\hat{U}^{(k)}$	3.9.1	–14

4.4 Masquage des trames effacées

Une procédure de masquage des erreurs a été incorporée dans le décodeur afin de réduire la dégradation dans le signal vocal reconstitué en raison d'effacements de trame dans le flux binaire. Ce processus de masquage des erreurs est fonctionnel lorsque la trame des paramètres du codeur (correspondant à une trame de 10 ms) a été identifiée comme étant effacée. Le mécanisme de détection des effacements de trame n'est pas défini dans la présente Recommandation et dépend de l'application.

La stratégie de masquage consiste à reconstruire la trame actuelle sur la base d'informations déjà reçues. Cette méthode remplace le signal d'excitation manquant par un signal de caractéristiques similaires, tout en diminuant progressivement son énergie. Pour cela, on utilise un classificateur d'éléments voisés utilisant le gain de prédiction à long terme, qui est calculé dans le cadre de l'analyse par postfiltre à long terme. Celui-ci (voir 4.2.1) trouve le prédicteur à long terme pour lequel le gain de prédiction est supérieur à 3 dB. Pour cela, on fixe un seuil de 0,5 pour le carré de la corrélation normalisée [équation (82)]. Pour le processus de masquage d'erreur, une trame de 10 ms est déclarée «périodique» si au moins une sous-trame de 5 ms possède un gain de prédiction à long terme supérieur à 3 dB. Sinon, la trame est déclarée «apériodique». Une trame effacée hérite sa classe de la trame vocale (reconstituée) précédente. On notera que la classification des éléments voisés est mise à jour en permanence sur la base de ce signal vocal reconstitué.

Les étapes précises à suivre pour masquer une trame effacée sont les suivantes:

1) répétition des paramètres du filtre de synthèse;

2) affaiblissement de gains de répertoire codé adaptatif et de répertoire codé fixe;

3) affaiblissement de l'énergie mémorisée par le prédicteur de gain;

4) production de l'excitation de remplacement.

4.4.1 Répétition de paramètres du filtre de synthèse

Le filtre de synthèse pour une trame effacée utilise les paramètres de prédiction linéaire de la dernière trame non effacée. Le registre du prédicteur à moyenne mobile des coefficients LSF contient les valeurs des mots de code $\hat{l}_i$. Etant donné que le mot de code n'est pas disponible pour la trame actuelle *m*, il est calculé à partir des paramètres LSF répétés $\hat{\omega}_i$ et du registre de prédicteur selon la formule suivante:

$$\hat{l}_i = \left[\hat{\omega}_i^{(m)} - \sum_{k=1}^{4} \hat{p}_{i,k} \hat{l}_i^{(m-k)} \right] / \left(1 - \sum_{k=1}^{4} \hat{p}_{i,k} \right) \qquad i = 1,...,10 \tag{92}$$

où les coefficients du prédicteur à moyenne mobile, $\hat{p}_{i,k}$ sont ceux de la dernière bonne trame reçue.

4.4.2 Affaiblissement de gains de répertoire codé adaptatif et de répertoire codé fixe

Le gain de répertoire codé fixe est fondé sur une version affaiblie du précédent gain de répertoire codé fixe. Il est donné par:

$$g_c^{(m)} = 0{,}98 g_c^{(m-1)} \tag{93}$$

où *m* est l'index de sous-trame. Le gain de répertoire codé adaptatif est fondé sur une version affaiblie du précédent gain de répertoire codé adaptatif. Il est donné par:

$$g_p^{(m)} = 0{,}9 g_p^{(m-1)} \qquad \text{avec la limite } g_p^{(m)} < 0{,}9 \tag{94}$$

4.4.3 Affaiblissement de l'énergie mémorisée par le prédicteur de gain

Comme décrit au 3.9, le prédicteur de gain utilise l'énergie des codes vectoriels de répertoire fixe qui ont été précédemment sélectionnés, $c(n)$. Afin d'éviter des effets transitoires dans le décodeur, la mémoire du prédicteur de gain est rafraîchie dès que des trames normales sont reçues, au moyen d'une version affaiblie de l'énergie de répertoire codé. La valeur de $\hat{U}^{(m)}$ pour la sous-trame actuelle *m* est alignée sur la valeur moyenne quantifiée de l'erreur sur la prédiction de gain, affaiblie de 4 dB, comme suit:

$$\hat{U}^{(m)} = \left(0{,}25 \sum_{i=1}^{4} \hat{U}^{(m-i)} \right) - 4{,}0 \text{ avec la limite } \hat{U}^{(m)} \geq -14 \tag{95}$$

4.4.4 Production de l'excitation de remplacement

L'excitation utilisée dépend de la classification de périodicité. Si la dernière trame reconstituée a été classifiée comme étant périodique, la trame actuelle est également considérée comme périodique. Dans ce cas, seul le répertoire codé adaptatif est utilisé et la contribution du répertoire codé fixe est mise à zéro. Le délai tonal est fondé sur la partie entière du délai tonal contenu dans la trame précédente. Ce délai est répété pour chaque trame successive. Afin d'éviter une périodicité excessive, le délai est augmenté de 1 à chaque sous-trame successive mais jusqu'à une limite de 143. Le gain de répertoire codé adaptatif est fondé sur une valeur affaiblie selon l'équation (94).

Si la dernière trame reconstituée avait été classifiée comme étant apériodique, la trame actuelle est considérée également comme apériodique et la contribution du répertoire codé adaptatif est mise à zéro. La contribution du répertoire codé fixe est construite par sélection aléatoire d'un index de répertoire et d'un index de signe. Le générateur de séquences aléatoires utilise la fonction suivante:

$$germe = 31821\, germe + 13849 \qquad (96)$$

avec la valeur initiale 21845 comme germe. L'index de répertoire fixe est extrait des 13 éléments binaires de poids faible du nombre aléatoire suivant. Le signe de répertoire fixe est extrait des 4 éléments binaires de poids faible du nombre aléatoire suivant. Le gain de répertoire fixe est affaibli conformément à l'équation (93).

5 Description au bit près du codeur CS-ACELP

L'UIT-T peut fournir le code C de l'ANSI qui simule le codeur CS-ACELP en arithmétique à virgule fixe et mots de 16 bits. Les paragraphes suivants résument l'utilisation de ce code de simulation et la façon dont le logiciel est organisé.

5.1 Utilisation du logiciel de simulation

Le code C se compose de deux programmes: **coder.c**, qui simule le codeur, et **decoder.c**, qui simule le décodeur. Le codeur est mis en œuvre comme suit:

coder inputfile bitstreamfile

Le fichier d'entrée (**inputfile**) et le fichier de sortie (**outputfile**) sont des fichiers de données échantillonnées contenant des signaux MIC à mots de 16 bits. Le décodeur est mis en œuvre comme suit:

decoder bitstreamfile outputfile

Le tableau de mise en correspondance du train binaire codé figure dans le logiciel de simulation.

5.2 Organisation du logiciel de simulation

En simulation par code C de l'ANSI à virgule fixe, on n'utilise que deux types de données à virgule fixe, comme indiqué dans le Tableau 10. Pour faciliter la mise en œuvre du code de simulation, les index de boucle, les valeurs booléennes et les fanions utilisent le type **Flag**, qui sera un mot de 16 bits ou de 32 bits selon la plate-forme cible.

TABLEAU 10/G.729

Types de données utilisés en simulation par code C de l'ANSI

Type	Valeur maximale	Valeur minimale	Description
Word16	0x7fff	0x8000	Mot de 16 bits en complément à 2 signé
Word32	0x7fffffffL	0x80000000L	Mot de 32 bits en complément à 2 signé

Tous les calculs sont effectués au moyen d'un ensemble prédéfini d'opérateurs de base. La description de ces opérateurs est donnée dans le Tableau 11. Les tables utilisées par le codeur simulé sont résumées dans le Tableau 12. Ces programmes principaux utilisent une bibliothèque de routines qui sont résumées dans les Tableaux 13, 14 et 15.

TABLEAU 11/G.729

Opérations de base utilisées en simulation par code C de l'ANSI

Opération	Description
Word16 sature(Word32 L_var1)	Limite à 16 bits (mots courts)
Word16 add(Word16 var1, Word16 var2)	Addition courte
Word16 sub(Word16 var1, Word16 var2)	Soustraction courte
Word16 abs_s(Word16 var1)	Valeur absolue courte
Word16 sh1(Word16 var1, Word16 var2)	Décalage à gauche sur mot court
Word16 shr(Word16 var1, Word16 var2)	Décalage à droite sur mot court
Word16 mult(Word16 var1, Word16 var2)	Multiplication courte
Word32 L_mult(Word16 var1, Word16 var2)	Multiplication longue
Word16 negate(Word16 var1)	Négation de mot court
Word16 extract_h(Word32 L_var1)	Extraction de partie haute
Word16 extract_1(Word32 L_var1)	Extraction de partie basse
Word16 round(Word32 L_var1)	Arrondissage
Word32 L_mac(Word32 L_var3, Word16 var1, Word16 var2)	Multiplication et cumul
Word32 L_msu(Word32 L_var3, Word16 var1, Word16 var2)	Multiplication et soustraction
Word32 L_add(Word32 L_var1, Word32 L_var2)	Addition longue
Word32 L_sub(Word32 L_var1, Word32 L_var2)	Soustraction longue
Word32 L_negate(Word32 L_var1)	Négation de mot long
Word16 mult_r(Word16 var1, Word16 var2)	Multiplication avec arrondissage
Word32 L_sh1(Word32 L_var1, Word16 var2)	Décalage à gauche sur mot long
Word32 L_shr(Word32 L_var1, Word16 var2)	Décalage à droite sur mot long
Word16 shr_r(Word16 var1, Word16 var2)	Décalage à droite avec arrondissage
Word16 mac_r(Word32 L_var3, Word16 var1, Word16 var2)	Multiplication et cumul avec arrondissage
Word16 msu_r(Word32 L_var3, Word16 var1, Word16 var2)	Multiplication et soustraction avec arrondissage
Word32 L_deposit_h(Word16 var1)	Application d'un mot de 16 bits (var1) dans la partie haute d'un mot long
Word32 L_deposit_l(Word16 var1)	Application d'un mot de 16 bits (var1) dans la partie basse d'un mot long
Word32 L_shr_r(Word32 L_var1, Word16 var2)	Décalage à droite sur un mot long avec arrondissage
Word32 L_abs(Word32 L_var1)	Valeur absolue longue
Word16 norm_s(Word16 var1)	Normalisation de mot court
Word16 div_s(Word16 var1, Word16 var2)	Division courte
Word16 norm_1(Word32 L_var1)	Normalisation de mot long

TABLEAU 12/G.729

Résumé des tables utilisées dans tab. ld8.c

Nom de la table	Longueur	Description
tab_hup_s	28	Suréchantillonneur pour postfiltre
tab_hup_1	112	Suréchantillonneur pour postfiltre
inter_3	13	Filtre FIR pour interpoler la corrélation
inter_3	31	Filtre FIR pour interpoler l'excitation antérieure
lspcb1	128×10	Quantificateur de coefficients LSP (première étape)
lspcb2	32×10	Quantificateur de coefficients LSP (deuxième étape)
fg	$2 \times 4 \times 10$	Prédicteurs à moyenne mobile en quantification vectorielle des coefficients de paires LSP
fg_sum	2×10	Table pour quantification vectorielle des coefficients de paires LSP
fg_sum_inv	2×10	Table pour quantification vectorielle des coefficients de paires LSP
gbk1	8×2	Répertoire GA en quantification vectorielle de gain
gbk2	16×2	Répertoire GB en quantification vectorielle de gain
map1	8	Table utilisée pour quantifier vectoriellement le gain
imap1	8	Table utilisée pour quantifier vectoriellement le gain
map2	16	Table utilisée pour quantifier vectoriellement le gain
ima21	16	Table utilisée pour quantifier vectoriellement le gain
window	240	Fenêtre d'analyse par prédiction linéaire
lag_h	10	Fenêtre d'analyse du délai pour l'expansion de bande passante (partie haute)
lag_1	10	Fenêtre d'analyse du délai pour l'expansion de bande passante (partie basse)
grid	61	Points de grille pour la conversion des coefficients LP en coefficients LSP
tabsqr	49	Table de calcul des racines carrées inverses
tablog	33	Table de calcul des logarithmes de base 2
table	65	Table de conversion des fréquences LSF en paires LSP et vice versa
slope	64	Abaque de conversion des paires LSP en fréquences LSF
tabpow	33	Table de calcul des puissances de 2

TABLEAU 13/G.729

Résumé des routines propres au codeur

Nom de fichier	Description
acelp_co.c	Exploration de répertoire codé fixe
cod_ld8k.c	Routine de codage
lpc.c	Analyse par prédiction linéaire
pitch.c	Recherche de la tonie
pre_proc.c	Prétraitement (filtrage passe-haut et normalisation)
pwf.c	Calcul des coefficients de pondération perceptive
qua_gain.c	Quantificateur de gain
qua_1sp.c	Quantificateur des coefficients LSP

TABLEAU 14/G.729

Résumé des routines propres au décodeur

Nom de fichier	Description
de_acelp.c	Décodage du répertoire à structure algébrique
dec_gain.c	Décodage des gains
dec_lag3.c	Décodage du répertoire codé adaptatif
dec_ld8k.c	Routine de décodage
lspdec.c	Routine de décodage des coefficients LSP
post_pro.c	Post-traitement (filtrage passe-haut et normalisation)
pst.c	Routines de postfiltrage

TABLEAU 15/G.729

Résumé des routines générales

Nom de fichier	Description
basicop2.c	Opérateurs de base
oper_32b.c	Extension des opérateurs de base
bits.c	Routines de manipulation des bits
dspfunc.c	Fonctions mathématiques
filter.c	Fonctions de filtrage
gainpred.c	Prédicteur de gain
lpcfunc.c	Routines diverses se rapportant au filtre de prédiction linéaire
lspgetq.c	Quantificateur de coefficients LSP
p_parity.c	Calcul de la parité tonale
pred_lt3.c	Construction du répertoire adaptatif
util.c	Fonctions utilitaires