



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

G.723.1

(03/96)

**ASPECTS GÉNÉRAUX DES SYSTÈMES
DE TRANSMISSION NUMÉRIQUES**

**CODEUR DE SIGNAUX VOCAUX
À DOUBLE DÉBIT POUR COMMUNICATIONS
MULTIMÉDIAS ACHÉMINÉES À 5,3 kbit/s
ET À 6,3 kbit/s**

Recommandation UIT-T G.723.1

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, 1^{er}-12 mars 1993).

La Recommandation UIT-T G.723.1, 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

		<i>Page</i>
1	Introduction	1
	1.1 Domaine d'application.....	1
	1.2 Débits binaires	1
	1.3 Signaux d'entrée possibles	1
	1.4 Délai.....	1
	1.5 Description du codeur vocal	1
2	Principes de codage.....	2
	2.1 Description générale	2
	2.2 Trameur.....	2
	2.3 Filtre passe-haut.....	3
	2.4 Analyse en codage prédictif linéaire (LPC).....	4
	2.5 Quantificateur des coefficients de paires de raies spectrales (LSP).....	4
	2.6 Décodeur des coefficients LSP	5
	2.7 Interpolation des coefficients LSP	6
	2.8 Filtre de pondération perceptive des formants	6
	2.9 Estimation de la tonie	7
	2.10 Traitement des sous-trames.....	7
	2.11 Mise en forme du bruit harmonique	7
	2.12 Calculateur des impulsions de réponse	8
	2.13 Réponse à entrée nulle et soustraction des suroscillations.....	9
	2.14 Prédicteur de tonie	9
	2.15 Excitation au débit supérieur (MP-MLQ).....	9
	2.16 Excitation au débit inférieur (ACELP)	11
	2.17 Décodeur d'impulsions d'excitation	13
	2.18 Décodage des informations tonales	14
	2.19 Mise à jour de la mémoire	14
	2.20 Affectation des bits	15
	2.21 Initialisation du codeur	16
3	Principes de décodage	17
	3.1 Description générale	17
	3.2 Décodeur des coefficients LSP	17
	3.3 Interpolateur des coefficients LSP	17
	3.4 Décodage des informations tonales	18
	3.5 Décodeur des impulsions d'excitation.....	18
	3.6 Postfiltre fondamental.....	18
	3.7 Filtre de synthèse en codage LPC.....	20
	3.8 Postfiltre des formants	20
	3.9 Module de normalisation du gain	21
	3.10 Traitement par interpolation des trames.....	21
	3.11 Initialisation du décodeur	22
4	Condensation des flux binaires	22
5	Code C de l'ANSI.....	22
6	Glossaire.....	22

Résumé

La présente Recommandation spécifie, dans le cadre de la série de Recommandations de type général H.324, une représentation codée qui peut servir à comprimer la parole ou une autre composante d'un signal audio pour des services multimédias à très faible débit binaire. Ce codeur possède deux débits binaires associés: 5,3 kbit/s et 6,3 kbit/s. Le débit supérieur correspond à une meilleure qualité. Le débit inférieur assure un niveau de qualité satisfaisant et offre aux concepteurs de systèmes un degré supplémentaire de flexibilité. Ces deux débits sont des caractéristiques obligatoires du codeur et du décodeur. Il est possible de passer de l'un à l'autre à chaque limite de trame. Il est également possible de choisir le fonctionnement en débit variable au moyen d'une émission discontinue avec remplissage de bruit pendant les intervalles sans paroles.

Ce codeur a été optimisé afin de représenter les paroles avec une qualité élevée aux débits indiqués ci-dessus, au prix d'un faible degré de complexité. Il code des signaux de parole ou d'autres signaux audio pour les transformer en trames utilisant un algorithme de prédiction linéaire par analyse-synthèse. Le signal d'excitation est, pour le codeur à débit supérieur, conforme à l'algorithme de quantification par impulsions multiples selon le critère du maximum de vraisemblance (MP-MLQ) (*multipulse maximum likelihood quantization*) et, pour le codeur à débit inférieur, conforme à l'algorithme de prédiction linéaire à excitation par séquences codées à structure algébrique (ACELP) (*algebraic-code-excited-linear-prediction*). La durée des trames est de 30 ms, plus une exploration supplémentaire de 7,5 ms, soit un délai algorithmique total de 37,5 ms. Tous les délais additionnels apportés par ce codeur se composent des délais de traitement dans la réalisation, des délais de transmission dans la liaison de communication et des délais de stockage temporaire (tampons) pour le protocole de multiplexage.

La description donnée dans la présente Recommandation 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 et a priorité sur les développements mathématiques qui y sont contenus, 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.

CODEUR DE SIGNAUX VOCAUX À DOUBLE DÉBIT POUR COMMUNICATIONS MULTIMÉDIAS ACHÉMINÉES À 5,3 kbit/s ET À 6,3 kbit/s

(Genève, 1996)

1 Introduction

1.1 Domaine d'application

La présente Recommandation spécifie une représentation codée qui peut être utilisée pour comprimer la composante vocale ou un autre signal audio de services multimédias à très faible débit binaire. Lors de la conception de ce codeur, la principale application étudiée a été la visiophonie à très faible débit binaire, dans le cadre de la série des normes du type H.324.

1.2 Débits binaires

Ce codeur possède deux débits binaires associés: 5,3 kbit/s et 6,3 kbit/s. Le débit supérieur correspond à une meilleure qualité. Le débit inférieur assure un niveau de qualité satisfaisant et offre aux concepteurs de systèmes un degré supplémentaire de flexibilité. Ces deux débits sont des caractéristiques obligatoires du codeur et du décodeur. Il est possible de passer de l'un à l'autre à chaque limite de trame (toutes les 30 ms). Il est également possible de choisir le fonctionnement en débit variable au moyen d'une émission discontinue avec remplissage de bruit pendant les intervalles sans paroles.

1.3 Signaux d'entrée possibles

Ce codeur a été optimisé afin de représenter les paroles avec une qualité élevée aux débits indiqués ci-dessus, au prix d'un faible degré de complexité. Les signaux acheminant des fréquences musicales et autres signaux audio ne sont pas restitués aussi fidèlement que les signaux vocaux mais ils peuvent être comprimés et décomprimés au moyen de ce codeur.

1.4 Délai

Ce codeur transforme les signaux de parole ou d'autres signaux audio en trames de 30 ms plus une période d'exploration de 7,5 ms, soit un délai algorithmique total de 37,5 ms. Tous les délais additionnels apportés par la réalisation et par le fonctionnement de ce codeur se composent des durées suivantes:

- i) délais de traitement des données dans le codeur et dans le décodeur;
- ii) délais de transmission dans la liaison de communication;
- iv) délais additionnels dus au stockage temporaire (tampons) pour le protocole de multiplexage.

1.5 Description du codeur vocal

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 et reflète ce mode de description en virgule fixe au bit près. Les développements mathématiques du codeur et du décodeur, figurant respectivement dans les articles 2 et 3, 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 en code C de l'algorithme figurant dans l'article 5 a donc priorité sur les descriptions mathématiques des articles 2 et 3, 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 Principes de codage

2.1 Description générale

Ce codeur est conçu de façon à fonctionner avec un signal numérique que l'on obtient en filtrant d'abord la bande téléphonique (Recommandation G.712) de l'entrée analogique, puis en l'échantillonnant à 8000 Hz puis en le convertissant linéairement en mots MIC de 16 éléments binaires pour l'entrée du codeur. Il convient d'effectuer la conversion inverse vers le mode analogique par des moyens similaires à la sortie du décodeur. D'autres caractéristiques d'entrée/sortie, comme celles qui sont spécifiées par la Recommandation G.711 pour les données MIC à 64 kbit/s, seront converties linéairement en mots MIC de 16 bits avant codage, ou seront reconverties dans leur format approprié après décodage linéaire des mots MIC de 16 bits. Le flux binaire allant du codeur au décodeur est défini dans le cadre de la présente Recommandation.

Le codeur est fondé sur les principes du codage prédictif linéaire (LPC) par analyse et synthèse, en vue de minimiser un signal d'erreur pondéré par une courbe de perception. Le codeur travaille sur des blocs (trames) contenant chacun 240 échantillons, ce qui correspond à une longueur de trame de 30 ms à 8 kHz. Chaque bloc passe d'abord par un filtre passe-haut afin d'éliminer la composante aperiodique. Il est ensuite subdivisé en quatre sous-trames de 60 échantillons chacune. Au moyen du signal d'entrée non traité, on calcule pour chaque sous-trame un filtre de codage prédictif linéaire (LPC) (*linear prediction coder*) du 10^{ème} ordre. Le filtre LPC pour la dernière sous-trame est calculé au moyen d'un quantificateur des vecteurs prédictifs dédoublés (PSVQ) (*predictive split vector quantizer*). Les coefficients de codage LPC non quantifiés servent à construire le filtre de pondération perceptive à court terme qui est utilisé pour filtrer la trame complète afin de donner le signal vocal pondéré par la perception.

Le signal vocal ainsi pondéré sert à calculer, toutes les deux sous-trames (120 échantillons), la période fondamentale en boucle ouverte, L_{OL} . Cette estimation de la hauteur tonale est effectuée sur des blocs de 120 échantillons. La période fondamentale (inverse de la fréquence fondamentale) est recherchée dans l'étendue de 18 à 142 échantillons.

A partir de ce point, le signal vocal est traité à raison de 60 échantillons par sous-trame.

Au moyen de la période fondamentale estimée, on construit un filtre de mise en forme du bruit harmonique. La combinaison du filtre de synthèse à codage LPC, du filtre de pondération perceptive des formants, et du filtre de mise en forme du bruit harmonique sert à créer une réponse impulsionnelle qui est utilisée pour d'autres calculs.

La réponse impulsionnelle et l'estimation de la période fondamentale, L_{OL} , permettent de calculer un prédicteur de tonie en boucle ouverte. On fait appel à un prédicteur de tonie du cinquième ordre. La période fondamentale est calculée sous la forme d'une petite valeur différentielle de part et d'autre de l'estimation L_{OL} . La contribution du prédicteur de tonie est ensuite soustraite du vecteur cible initial. La période fondamentale et la valeur différentielle sont toutes deux transmises au décodeur.

Finalement, la composante aperiodique de l'excitation est approchée. Pour le débit supérieur, on fait appel à l'excitation par quantification d'impulsions multiples selon le critère du maximum de vraisemblance (MP-MLQ). Pour le débit inférieur, on fait appel à l'excitation par séquences codées à structure algébrique (ACELP).

Le schéma fonctionnel du codeur est représenté sur la Figure 1.

2.2 Trameur

Fichier: LBCCODEC.C	Procédure: main()	Lit les trames d'entrée de 240 échantillons
Fichier: CODER.C	Procédure: Coder()	Effectue la subdivision en sous-trames

Le codeur traite les signaux vocaux en insérant des échantillons vocaux consécutifs, $y[n]$, dans des trames de 240 échantillons, $s[n]$. Chaque trame est subdivisée en deux moitiés de 120 échantillons afin de calculer l'estimation de la période fondamentale. Chacune de ces parties est de nouveau divisée par deux, de sorte que chaque trame est finalement subdivisée en quatre sous-trames de 60 échantillons chacune.

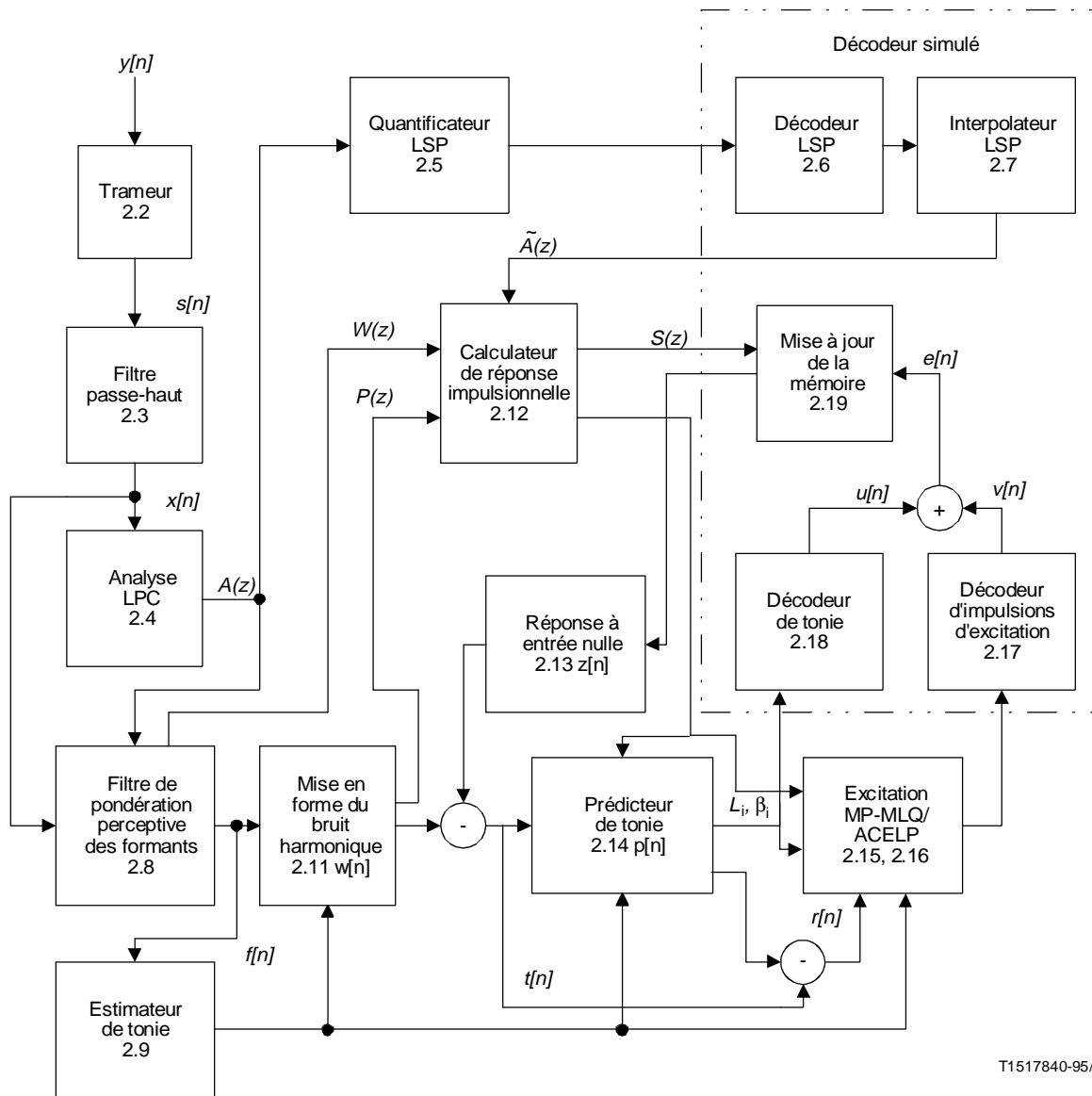
2.3 Filtre passe-haut

Fichier: UTIL_LBC.C	Procédure: Rem_Dc()	Effectue le filtrage passe-haut
---------------------	---------------------	---------------------------------

Ce bloc élimine l'élément aperiodique des signaux vocaux d'entrée, $s[n]$. La fonction de transfert de ce filtre est la suivante:

$$H(z) = \frac{1 - z^{-1}}{1 - \frac{127}{128} z^{-1}} \quad (1)$$

La sortie de ce filtre est $x[n]_{n=0..239}$.



T1517840-95/d01

FIGURE 1/G.723.1

Organigramme du codeur de signaux vocaux
(avec indication du numéro de paragraphe correspondant à chaque bloc)

2.4 Analyse en codage prédictif linéaire (LPC)

Fichier: LPC.C	Procédure: Comp_Lpc()	Effectue le calcul des coefficients LPC
Fichier: LPC.C	Procédure: Durbin()	Récurrence de Levinson-Durbin

L'analyse en codage LPC est effectuée comme suit sur le signal $x[n]$. On fait appel à une analyse prédictive linéaire (LP) (*linear predictive*) du dixième ordre. Une fenêtre de 180 échantillons est centrée sur chaque sous-trame. Une fenêtre de Hamming est appliquée à ces échantillons. Onze coefficients d'autocorrélation (réflexion) sont calculés à partir du signal ainsi fenêtré. Un facteur de correction pour le bruit blanc de (1025/1024) est appliqué selon la formule $R[0] = R[0](1 + 1/1024)$. Les 10 autres coefficients d'autocorrélation sont multipliés par la table des coefficients binomiaux de la fenêtre. (Les valeurs de cette table sont données, comme les autres, en code C.) Les coefficients de codage prédictif linéaire (LPC) (*linear predictive coefficients*) sont calculés par la récurrence de Levinson-Durbin conventionnelle. Pour chaque trame d'entrée, quatre séries de coefficients LPC sont calculés, une par sous-trame. Ces séries de codage LPC servent à construire le filtre de pondération perceptive à court terme. Le filtre de synthèse en codage LPC est défini comme suit:

$$A_i(z) = \frac{1}{10 - \sum_{j=1}^{10} a_{ij}z^{-j}}, \quad 0 \leq i \leq 3 \quad (2)$$

où i est l'index de la sous-trame, défini comme étant compris entre 0 et 3.

2.5 Quantificateur des coefficients de paires de raies spectrales (LSP)

Fichier: LSP.C	Procédure: AtoLsp()	Convertit les coefficients LPC en coefficients LSP
Fichier: LSP.C	Procédure: LspQnt()	Quantification du vecteur LSP
Fichier: LSP.C	Procédure: Lsp_Svq()	Quantification des sous-vecteurs LSP

On effectue d'abord une petite expansion de bande additionnelle (7,5 Hz). Puis le filtre prédictif linéaire résultant $A_3(z)$ est quantifié comme suit au moyen d'un quantificateur de vecteurs prédictifs dédoublés (PSVQ).

- 1) Les coefficients LP, $\{a_j\}_{j=1..10}$, sont convertis en coefficients LSP, $\{p'_j\}_{j=1..10}$ par recherche sur le cercle unité et interpolation des passages à zéro.
- 2) La composante apériodique à long terme, p_{DC} , est soustraite des coefficients LSP, p' , et un nouveau vecteur LSP sans composante apériodique, p , est obtenu.
- 3) Un prédicteur fixe (non adaptatif) du premier ordre, $b = (12/32)$, est appliqué au vecteur LSP déjà décodé \tilde{p}_{n-1} , ce qui donne le vecteur LSP prédit sans composante apériodique, \bar{p}_n , ainsi que le vecteur d'erreur LSP résiduelle, e_n , au temps (à la trame) n .

$$\mathbf{p}_n^T = [p_{1,n} p_{2,n} \dots p_{10,n}] \quad (3.1)$$

$$\bar{\mathbf{p}}_n^T = [\bar{p}_{1,n} \bar{p}_{2,n} \dots \bar{p}_{10,n}] \quad (3.2)$$

$$\bar{p}_n = b[\bar{p}_{n-1} - p_{DC}] \quad (3.3)$$

$$e_n = p_n - \bar{p}_n \quad (3.4)$$

- 4) Le vecteur des coefficients LSP non quantifiés, p'_n , le vecteur des coefficients LSP quantifiés, \tilde{p}_n , et le vecteur des erreurs LSP résiduelles, e_n , sont subdivisés en 3 sous-vecteurs de dimensions 3, 3 et 4 respectivement. Chaque $m^{\text{ième}}$ sous-vecteur est quantifié vectoriellement au moyen d'un répertoire de séquences codées sur 8 éléments binaires. L'index, l , de l'entrée appropriée dans ce répertoire pour sous-vecteurs est choisi de façon à minimiser le critère d'erreur $E_{l,m}$.

$$p'_m{}^T = [p'_{1+3m} \ p'_{2+3m} \ \dots \ p'_{K_m+3m}], \quad K_m = \begin{cases} 3, & m=0 \\ 3, & m=1 \\ 4, & m=2 \end{cases} \quad (4.1)$$

$$\tilde{p}_{l,m}{}^T = [\tilde{p}_{1,l,m} \ \tilde{p}_{2,l,m} \ \dots \ \tilde{p}_{K_m,l,m}], \quad \begin{matrix} 0 \leq m \leq 2 \\ 1 \leq l \leq 256 \end{matrix} \quad (4.2)$$

$$p' = p + p_{DC} \quad (4.3)$$

$$\tilde{p}_{l,m} = \bar{p}_m + p_{DC_m} + e_{l,m}, \quad \begin{matrix} 0 \leq m \leq 2 \\ 1 \leq l \leq 256 \end{matrix} \quad (4.4)$$

$$E_{l,m} = (p'_m - \tilde{p}_{l,m})^T W_m (p'_m - \tilde{p}_{l,m}), \quad \begin{matrix} 0 \leq m \leq 2 \\ 1 \leq l \leq 256 \end{matrix} \quad (4.5)$$

où $e_{l,m}$ est la $l^{\text{ième}}$ entrée du $m^{\text{ième}}$ répertoire de codes LSP résiduels dédoublés et où W_n est une matrice de pondération par filtrage diagonal, déterminée d'après le vecteur p' des coefficients LSP non quantifiés, les pondérations étant définies comme suit:

$$\begin{aligned} w_{j,j} &= \frac{1}{\min \{p'_j - p'_{j-1}, p'_{j+1} - p'_j\}}, \quad 2 \leq j \leq 9 \\ w_{1,1} &= \frac{1}{p'_2 - p'_1} \\ w_{10,10} &= \frac{1}{p'_{10} - p'_9} \end{aligned} \quad (5)$$

- 5) Les index ainsi sélectionnés sont transmis au canal.

2.6 Décodeur des coefficients LSP

Fichier: LSP.C	Procédure: Lsp_Inq()	Quantification inverse des coefficients LSP
----------------	----------------------	---

Le décodage des coefficients LSP est effectué comme suit:

- 1) tout d'abord, les trois sous-vecteurs, $\{e_{m,n}\}_{m=0,2}$, sont décodés afin de former un vecteur du dixième ordre, \tilde{e}_n ;
- 2) le vecteur prédit, \bar{p}_n , est ajouté au vecteur décodé, \tilde{e}_n , et au vecteur des composantes apériodiques, p_{DC} , afin de former le vecteur des coefficients LSP décodés, \tilde{p}_n ;

- 3) une vérification de stabilité est effectuée sur le vecteur des coefficients LSP décodés, $\tilde{\mathbf{p}}_n$, afin de s'assurer que ce vecteur respecte la condition d'ordre suivante:

$$\tilde{p}_{j+1,n} - \tilde{p}_{j,n} \geq \Delta_{\min}, 1 \leq j \leq 9 \quad (6)$$

Δ_{\min} est égal à 31,25 Hz. Si cette condition de stabilité (6) n'est pas vérifiée pour \tilde{p}_i et pour \tilde{p}_{i+1} , on modifie comme suit \tilde{p}_j et \tilde{p}_{j+1} :

$$\tilde{p}_{avg} = (\tilde{p}_j + \tilde{p}_{j+1}) / 2 \quad (7.1)$$

$$\tilde{p}_j = \tilde{p}_{avg} - \Delta_{\min} / 2 \quad (7.2)$$

$$\tilde{p}_{j+1} = \tilde{p}_{avg} + \Delta_{\min} / 2 \quad (7.3)$$

Cette modification est effectuée jusqu'à ce que la condition (6) soit vérifiée. Si, à l'issue de 10 itérations, cette condition de stabilité n'est toujours pas vérifiée, on fait appel au vecteur LSP précédent.

2.7 Interpolation des coefficients LSP

Fichier: LSP.C	Procédure: Lsp_Int()	Interpolateur de coefficients LSP
Fichier: LSP.C	Procédure: LsptoA()	Conversion des coefficients LSP en coefficients LPC

Une interpolation linéaire est effectuée entre le vecteur LSP décodé, $\tilde{\mathbf{p}}_n$, et les vecteur LSP précédent, $\tilde{\mathbf{p}}_{n-1}$, pour chaque sous-trame. Quatre vecteurs LSP interpolés, $\{\tilde{\mathbf{p}}_i\}_{i=0..3}$, sont convertis en vecteurs LPC, $\{\tilde{\mathbf{a}}_i\}_{i=0..3}$.

$$\tilde{\mathbf{p}}_{ni} = \begin{cases} 0,75\tilde{\mathbf{p}}_{n-1} + 0,25\tilde{\mathbf{p}}_n, & i = 0 \\ 0,5\tilde{\mathbf{p}}_{n-1} + 0,5\tilde{\mathbf{p}}_n, & i = 1 \\ 0,25\tilde{\mathbf{p}}_{n-1} + 0,75\tilde{\mathbf{p}}_n, & i = 2 \\ \tilde{\mathbf{p}}_n, & i = 3 \end{cases} \quad (8)$$

$$\tilde{\mathbf{a}}_i^T = [\tilde{a}_{i1}\tilde{a}_{i2} \dots \tilde{a}_{i10}]^T, 0 \leq i \leq 3 \quad (9)$$

Le filtre de synthèse des coefficients LPC quantifiés, $\tilde{A}_i(z)$, est utilisé pour restituer le signal vocal décodé. Il est défini comme suit:

$$\tilde{A}_i(z) = \frac{1}{1 - \sum_{j=1}^{10} \tilde{a}_{ij} z^{-j}}, 0 \leq i \leq 3 \quad (10)$$

2.8 Filtre de pondération perceptive des formants

Fichier: LPC.C	Procédure: Wght_Lpc()	Calcule les coefficients du filtre de pondération perceptive
Fichier: LPC.C	Procédure: Error_Wght()	Applique le filtre de pondération perceptive

Un filtre de pondération perceptuelle des formants est construit pour chaque sous-trame, au moyen des coefficients LPC non quantifiés $\{a_{ij}\}_{j=1,\dots,10}$. Ce filtre a la fonction de transfert suivante:

$$W_i(z) = \frac{1 - \sum_{j=1}^{10} a_{ij} z^{-j} \gamma_1^j}{1 - \sum_{j=1}^{10} a_{ij} z^{-j} \gamma_2^j}, \quad 0 \leq i \leq 3 \quad (11)$$

où $\gamma_1 = 0,9$ et $\gamma_2 = 0,5$. La trame du signal vocal d'entrée, $\{x[n]\}_{n=0..239}$, est ensuite subdivisée en quatre sous-trames dont chacune passe par le filtre $W_i(z)$, ce qui donne le signal vocal de sortie pondéré, $\{f[n]\}_{n=0..239}$.

2.9 Estimation de la tonie

Fichier: EXC_LBC.C	Procédure: Estim_Pitch()	Estimation de la tonie en boucle ouverte
--------------------	--------------------------	--

On calcule deux estimations de tonie pour chaque trame: l'une pour les deux premières sous-trames et une pour les deux dernières. L'estimation de la période fondamentale en boucle ouverte, L_{OL} , est calculée à partir du signal vocal pondéré perceptivement, $f[n]$. Pour déterminer la période fondamentale, on fait appel à une méthode de maximisation du critère d'intercorrélation, $C_{OL}(j)$, exprimé comme suit:

$$C_{OL}(j) = \frac{\left(\sum_{n=0}^{119} f[n] \cdot f[n-j] \right)^2}{\sum_{n=0}^{119} f[n-j] \cdot f[n-j]}, \quad 18 \leq j \leq 142 \quad (12)$$

L'index j qui maximise le critère d'intercorrélation, $C_{OL}(j)$, est choisi comme estimation de la période fondamentale en boucle ouverte pour les deux sous-trames appropriées. Lorsque l'on recherche le meilleur index, on donne une certaine préférence aux périodes fondamentales de plus faible grandeur, afin d'éviter de choisir des multiples de la tonie. Les valeurs maximales du critère $C_{OL}(j)$ sont recherchées en commençant par $j = 18$. Chaque valeur maximale de $C_{OL}(j)$ trouvée est comparée au meilleur maximum déjà obtenu, $C_{OL}(j')$. Si la différence entre les index j et j' est inférieure à 18 et si $C_{OL}(j) > C_{OL}(j')$, on choisit le nouveau maximum. Si la différence entre les index est supérieure ou égale à 18, on ne choisit le nouveau maximum que si $C_{OL}(j')$ est supérieur à $C_{OL}(j)$ de 1,25 dB.

2.10 Traitement des sous-trames

A partir de ce point, tous les blocs de calcul sont appliqués à raison d'une fois par sous-trame.

2.11 Mise en forme du bruit harmonique

Fichier: EXC_LBC.C	Procédure: Comp_Pw()	Calcule les coefficients du filtre de bruit harmonique
Fichier: EXC_LBC.C	Procédure: Filt_Pw()	Applique le filtre de bruit harmonique

Afin d'améliorer la qualité des signaux vocaux codés, on construit un filtre de mise en forme du bruit harmonique, exprimé comme suit:

$$P_i(z) = 1 - \beta z^{-L} \quad (13)$$

L'inertie optimale, L , de ce filtre est celle qui maximise le critère, $C_{PW}(j)$, lorsque l'on ne prend en compte que les valeurs positives de corrélation contenues dans le numérateur, $N(j)$, avant d'élever celui-ci au carré:

$$N(j) = \sum_{n=0}^{59} f[n] \cdot f[n-j] \quad (14.1)$$

$$C_{PW}(j) = \frac{(N(j))^2}{\sum_{n=0}^{59} f[n-j] \cdot f[n-j]}, \quad L_1 \leq j \leq L_2 \quad (14.2)$$

où $L_1 = L_{OL} - 3$ et où $L_2 = L_{OL} + 3$. La valeur maximale sera désignée par C_L . Le gain optimal du filtre, G_{opt} , est:

$$G_{opt} = \frac{\sum_{n=0}^{59} f[n] f[n-L]}{\sum_{n=0}^{59} f[n-L] f[n-L]} \quad (15)$$

Le gain G_{opt} est limité à l'étendue [0,1]. L'énergie, E , du signal vocal pondéré, $\{f[n]\}_{n=0..59}$, est donnée par:

$$E = \sum_{n=0}^{59} f^2[n] \quad (16)$$

Ensuite, le coefficient β du filtre de mise en forme du bruit harmonique, $P(z)$, est donné par la formule suivante:

$$\beta = \begin{cases} 0,3125 G_{opt}, & \text{si } -10 \log_{10} \left(1 - \frac{C_L}{E} \right) \geq 2,0 \\ 0,0, & \text{sinon} \end{cases} \quad (17)$$

Après calcul des coefficients du filtre de bruit harmonique, le signal vocal pondéré en fonction de la perception des formants, $f[n]$, est filtré par $P(z)$ afin d'obtenir le vecteur cible, $w[n]$, comme suit:

$$w[n] = f[n] - \beta f[n-L], \quad 0 \leq n \leq 59 \quad (18)$$

2.12 Calculateur des impulsions de réponse

Fichier: LPC.C	Procédure: Comp_Ir()	Calcul de la réponse impulsionnelle
----------------	----------------------	-------------------------------------

Pour l'analyse en boucle fermée, le filtre combiné suivant, $S_i(z)$, est utilisé:

$$S_i(z) = \tilde{A}_i(z) \cdot W_i(z) \cdot P_i(z), \quad 0 \leq i \leq 3 \quad (19)$$

où les composantes du filtre $S_i(z)$ sont définies dans les formules 10, 11 et 13. La réponse impulsionnelle de ce filtre est calculée et désignée par $\{h_i[n]\}_{n=0..59, i=0..3}$.

2.13 Réponse à entrée nulle et soustraction des suroscillations

Fichier: LPC.C	Procédure: Sub_Ring()	Effectue la soustraction des suroscillations
----------------	-----------------------	--

On obtient la réponse à entrée nulle du filtre combiné $S_i(z)$ en calculant la sortie de ce filtre lorsque le signal d'entrée ne contient que des échantillons de valeur nulle. La réponse à l'entrée nulle est désignée par $\{z[n]\}_{n=0..59}$. La soustraction des suroscillations consiste à soustraire la réponse à entrée nulle du vecteur vocal pondéré par le filtre de bruit harmonique, $\{w[n]\}_{n=0..59}$. Le vecteur résultant est désigné par $t[n] = w[n] - z[n]$.

2.14 Prédicteur de tonie

Fichier: EXC_LBC.C	Procédure: Find_Acbk()	Contribution du répertoire adaptatif. Appelle les procédures Get_Rez() et Decod_Acbk()
Fichier: EXC_LBC.C	Procédure: Get_Rez()	Recueille les résidus de la mémoire d'excitation
Fichier: EXC_LBC.C	Procédure: Decod_Acbk()	Décode la contribution du répertoire adaptatif

La contribution du prédicteur de tonie est traitée sous la forme d'une contribution de répertoire adaptatif conventionnel. Le prédicteur de tonie est du cinquième ordre (voir l'équation 41.2). Pour les sous-trames 0 et 2, le délai tonal en boucle fermée est choisi aux environs du délai tonal en boucle ouverte dans l'intervalle ± 1 et codé sur 7 éléments binaires. (On notera que le délai tonal en boucle ouverte n'est jamais transmis.) Pour les sous-trames 1 et 3, le délai tonal en boucle fermée est codé différemment sur 2 éléments binaires. Par rapport au délai de la sous-trame précédente, il ne peut différer que de $-1, 0, +1$ ou $+2$. Les valeurs quantifiées et décodées du délai tonal seront désormais désignées par L_i . Les gains du prédicteur de tonie sont quantifiés vectoriellement au moyen de deux répertoires à 85 ou 170 entrées pour le débit binaire supérieur et à 170 entrées pour le débit inférieur. Le répertoire à 170 entrées est le même pour les deux débits. Pour le débit supérieur, si le délai L_0 est inférieur à 58 pour les sous-trames 0 et 1 ou si le délai L_2 est inférieur à 58 pour les sous-trames 2 et 3, il faut utiliser le répertoire à 85 entrées pour la quantification du gain tonal. Celui-ci, si les conditions précédentes ne sont pas vérifiées, est quantifié au moyen du répertoire à 170 entrées. La contribution du prédicteur tonal, $\{p[n]\}_{n=0..59}$, est soustraite du vecteur cible $\{t[n]\}_{n=0..59}$, afin d'obtenir le signal résiduel $\{r[n]\}_{n=0..59}$.

$$r[n] = t[n] - p[n] \quad (20)$$

2.15 Excitation au débit supérieur (MP-MLQ)

Fichier: EXC_LBC.C	Procédure: Find_Fcbk()	Contribution du répertoire fixe (non adaptatif)
Fichier: EXC_LBC.C	Procédure: Find_Best()	Quantification du signal résiduel
Fichier: EXC_LBC.C	Procédure: Gen_Trn()	Production d'un train de fonctions de Dirac
Fichier: EXC_LBC.C	Procédure: Fcbk_Pack()	Codage combinatoire des positions impulsionnelles

Le signal résiduel $\{r[n]\}_{n=0..59}$, est transféré sous la forme d'un nouveau vecteur cible dans le bloc qui effectue la quantification MP-MLQ de ce vecteur. Le processus de quantification consiste à approcher le vecteur cible $r[n]$ au moyen de $r'[n]$:

$$r'[n] = \sum_{j=0}^n h[j] \cdot v[n-j], \quad 0 \leq n \leq 59 \quad (21)$$

où $v[n]$ est l'excitation du filtre combiné $S(z)$ par la réponse impulsionnelle $h[n]$, cette excitation étant définie comme suit:

$$v[n] = G \sum_{k=0}^{M-1} \alpha_k d[n-m_k], \quad 0 \leq n \leq 59 \quad (22)$$

où G est le facteur de gain, $\delta[n]$ est une fonction de Dirac, $\{\alpha_k\}_{k=0..M-1}$ et $\{m_k\}_{k=0..M-1}$ sont les signes (± 1) et les positions respectives des fonctions de Dirac, M étant le nombre d'impulsions, soit 6 pour les sous-trames paires et 5 pour les sous-trames impaires. Une condition est imposée aux positions des impulsions, car ces positions peuvent être soit toutes impaires soit toutes paires. Cela sera indiqué par un bit de grille. Le problème consiste donc à estimer les paramètres inconnus, G , $\{\alpha_k\}_{k=0..M-1}$ et $\{m_k\}_{k=0..M-1}$, qui minimisent la moyenne quadratique du signal d'erreur $err[n]$:

$$err[n] = r[n] - r'[n] = r[n] - G \sum_{k=0}^{M-1} \alpha_k h[n-m_k] \quad (23)$$

Les processus d'estimation et de quantification des paramètres sont fondés sur une méthode d'analyse-synthèse. Le paramètre G_{max} est estimé et quantifié comme suit. On calcule d'abord la fonction d'intercorrélation $d[j]$ entre la réponse impulsionnelle, $h[n]$ et le nouveau vecteur cible, $r[n]$:

$$d[j] = \sum_{n=j}^{59} r[n] \cdot h[n-j], \quad 0 \leq j \leq 59 \quad (24)$$

Le gain estimé est donné par:

$$G_{max} = \frac{\max \{ |d[j]| \}_{j=0..59}}{\sum_{n=0}^{59} h[n] \cdot h[n]} \quad (25)$$

Puis on quantifie le gain estimé G_{max} au moyen d'un quantificateur logarithmique. Ce quantificateur de gain scalaire est commun aux deux débits et se compose de 24 échelons de 3,2 dB chacun. Autour de cette valeur quantifiée, \tilde{G}_{max} , on choisit des valeurs de gain additionnelles dans l'étendue $[\tilde{G}_{max} - 3,2, \tilde{G}_{max} + 6,4]$. Pour chacune de ces valeurs de gain, les signes et les emplacements des impulsions sont optimisés séquentiellement. Cette procédure est répétée pour les deux grilles de positions, la paire et l'impair. Finalement, on choisit la combinaison de paramètres quantifiés qui produit la plus petite moyenne quadratique du signal d'erreur $err[n]$. On transmet la combinaison optimale des positions impulsionnelles et du gain. On fait appel à un codage combinatoire avec les éléments $\binom{30}{M}$ pour transmettre les positions des impulsions. Par ailleurs, grâce au fait que le nombre de mots de code contenus dans les répertoires n'est pas une puissance de deux, 3 éléments binaires supplémentaires sont récupérés par la combinaison des 4 éléments binaires de plus fort poids contenus dans les codes combinatoires pour les 4 sous-trames, ce qui permet de former un index de 13 bits. Le code C précise comment ces informations sont condensées en mots.

Afin d'améliorer la qualité des signaux vocaux contenant une courte période fondamentale, on utilise la procédure suivante. Si le délai L_0 est inférieur à 58 pour les sous-trames 0 et 1, ou si le délai L_2 est inférieur à 58 pour les sous-trames 2 et 3, on fait appel à un train de fonctions de Dirac utilisant pour chaque position m_k la période du délai tonal, L_0

ou L_2 , au lieu de faire appel à une seule fonction de Dirac comme dans la procédure de quantification précédente. Pour choisir entre un train de fonctions de Dirac ou une seule fonction de Dirac afin de représenter le signal résiduel, on se fonde sur le calcul de l'erreur quadratique moyenne. On choisit alors la configuration qui donne la plus petite erreur quadratique moyenne et on transmet les index de ses paramètres.

2.16 Excitation au débit inférieur (ACELP)

Fichier: EXC_LBC.C	Procédure: search_T0()	Excitation tonale synchrone
Fichier: EXC_LBC.C	Procédure: ACELP_LBC_code()	Calcul du vecteur innovant
Fichier: EXC_LBC.C	Procédure: Cor_h()	Corrélations de réponse impulsionnelle
Fichier: EXC_LBC.C	Procédure: Cor_h_X()	Corrélation du vecteur cible avec la réponse impulsionnelle
Fichier: EXC_LBC.C	Procédure: D4i64_LBC()	Recherche du répertoire de séquences algébriques
Fichier: EXC_LBC.C	Procédure: G_code()	Calcul du gain du vecteur innovant (actualisateur)

On utilise un répertoire de séquences algébriques codées sur 17 bits pour l'excitation par répertoire fixe, $v[n]$. Chaque code vectoriel («vecteurcode») fixe contient, au plus, quatre impulsions non nulles. Ces 4 impulsions peuvent avoir les signes et les positions indiqués dans le Tableau 1:

TABLEAU 1/G.723.1

Répertoire d'excitation en codage ACELP

Signe	Positions
± 1	0, 8, 16, 24, 32, 40, 48, 56
± 1	2, 10, 18, 26, 34, 42, 50, 58
± 1	4, 12, 20, 28, 36, 44, 52, (60)
± 1	6, 14, 22, 30, 38, 46, 54, (62)

Les positions de toutes les impulsions peuvent être simultanément décalées d'une unité (pour occuper des positions impaires), ce qui nécessite un bit supplémentaire. On notera que la dernière position de chacune des deux dernières impulsions s'inscrit à l'extérieur de la limite de sous-trame, c'est-à-dire que l'impulsion n'est pas présente.

Chaque position impulsionnelle est codée sur 3 éléments binaires et chaque signe d'impulsion est codé sur 1 bit. Cela donne un total de 16 éléments binaires pour les 4 impulsions. Par ailleurs, un bit supplémentaire est utilisé pour coder le décalage, ce qui donne un répertoire de séquences de 17 bits.

On explore ce répertoire en minimisant l'erreur quadratique moyenne entre le signal vocal pondéré, $r[n]$, et le signal vocal pondéré par le filtre de synthèse, donné par la formule:

$$E_{\xi} = \|\mathbf{r} - \mathbf{GH}\mathbf{v}_{\xi}\|^2 \quad (26)$$

où \mathbf{r} est le vecteur cible composé du signal vocal pondéré après soustraction de la réponse à entrée nulle du filtre de synthèse pondérée et de la contribution du prédicteur tonal; où G est le gain du répertoire; \mathbf{v}_{ξ} est le mot de code algébrique à l'index ξ ; et où \mathbf{H} est une matrice convolutionnelle triangulaire inférieure de Toeplitz avec une diagonale $h(0)$ et des diagonales secondaires $h(1), \dots, h(L-1)$, où $h(n)$ est la réponse impulsionnelle du filtre de synthèse pondérée $S_i(z)$.

On peut montrer que le mot de code optimal est celui qui maximise le terme

$$\tau_{\xi} = \frac{C_{\xi}^2}{\varepsilon_{\xi}} = \frac{(d^T v_{\xi})^2}{v_{\xi}^T \Phi v_{\xi}} \quad (27)$$

où $d = H^T r$ est la corrélation entre le signal vectoriel cible, $r[n]$, et la réponse impulsionnelle, $h[n]$, et où $\Phi = H^T H$ est la matrice de covariance de la réponse impulsionnelle. Le vecteur d et la matrice Φ sont calculés avant l'exploration du répertoire. Les éléments du vecteur d sont calculés comme suit:

$$d(j) = \sum_{n=j}^{59} r[n] \cdot h[n-j], \quad 0 \leq j \leq 59 \quad (28)$$

et les éléments de la matrice symétrique $\Phi(i, j)$ sont calculés par la formule suivante:

$$\Phi(i, j) = \sum_{n=j}^{59} h[n-i] \cdot h[n-j], \quad \begin{array}{l} j \geq i \\ 0 \leq i \leq 59 \end{array} \quad (29)$$

NOTE – Seuls les éléments réellement nécessaires sont calculés et un système efficace de mise en mémoire a été conçu afin d'accélérer la procédure d'exploration du répertoire.

La structure algébrique du répertoire autorise des procédures d'exploration très rapides car le vecteur d'excitation v_{ξ} ne contient que 4 impulsions non nulles. L'exploration est effectuée en quatre boucles imbriquées, correspondant à chaque position d'impulsion, la contribution de chaque nouvelle impulsion étant ajoutée à chaque boucle d'itération. La corrélation C contenue dans l'équation (27) est donnée par la relation suivante:

$$C = \alpha_0 d[m_0] + \alpha_1 d[m_1] + \alpha_2 d[m_2] + \alpha_3 d[m_3] \quad (30)$$

où m_k est la position de la $k^{\text{ième}}$ impulsion et α_k est son signe (± 1). L'énergie pour les codes vectoriels des positions impulsionnelles paires, indiquée par ε dans l'équation (27) ci-dessus, est donnée par la relation suivante:

$$\begin{aligned} \varepsilon &= \Phi(m_0, m_0) \\ &+ \Phi(m_1, m_1) + 2\alpha_0\alpha_1\Phi(m_0, m_1) \\ &+ \Phi(m_2, m_2) + 2[\alpha_0\alpha_2\Phi(m_0, m_2) + \alpha_1\alpha_2\Phi(m_1, m_2)] \\ &+ \Phi(m_3, m_3) + 2[\alpha_0\alpha_3\Phi(m_0, m_3) + \alpha_1\alpha_3\Phi(m_1, m_3) + \alpha_2\alpha_3\Phi(m_2, m_3)] \end{aligned} \quad (31)$$

Pour les codes vectoriels des positions impulsionnelles impaires, l'énergie indiquée dans l'équation (27) est approchée par celle du code vectoriel de la position impulsionnelle paire équivalente, obtenue par décalage des positions impulsionnelles impaires de manière à les recaler sur l'échantillon immédiatement antérieur. Pour simplifier la procédure d'exploration, les fonctions $d[j]$ et $\Phi(m_1, m_2)$ sont modifiées. Cette simplification est effectuée comme suit (avant l'exploration du répertoire). Tout d'abord, le signal $s[j]$ est défini. Ensuite, le signal $d'[j]$ est construit.

$$\begin{aligned} s[2j] &= s[2j + 1] = \text{sign}(d[2j]) && \text{si } |d[2j]| > |d[2j + 1]| \\ s[2j] &= s[2j + 1] = \text{sign}(d[2j + 1]) && \text{sinon} \end{aligned} \quad (32)$$

et le signal d' est donné par $d'[j] = d[j]s[j]$. En deuxième lieu, la matrice Φ est modifiée par insertion de l'information contenue dans le signal, c'est-à-dire que $\Phi'(i, j) = s[i]s[j]\Phi(i, j)$. Le coefficient de corrélation indiqué dans l'équation (30) est alors donné par:

$$C = d'[m_0] + d'[m_1] + d'[m_2] + d'[m_3] \quad (33)$$

et l'énergie indiquée dans l'équation (31) est donnée par la relation suivante:

$$\begin{aligned}
 \varepsilon &= \Phi'(m_0, m_0) \\
 &+ \Phi'(m_1, m_1) + 2\Phi'(m_0, m_1) \\
 &+ \Phi'(m_2, m_2) + 2[\Phi'(m_0, m_2) + \Phi'(m_1, m_2)] \\
 &+ \Phi'(m_3, m_3) + 2[\Phi'(m_0, m_3) + \Phi'(m_1, m_3) + \Phi'(m_2, m_3)]
 \end{aligned} \tag{34}$$

Une méthode de recherche focalisée est utilisée afin de simplifier encore la procédure d'exploration du répertoire. 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. Ce seuil est calculé sur la base du coefficient de corrélation C . Avant l'exploration du répertoire, 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:

$$thr_3 = av_3 + (max_3 - av_3) / 2 \tag{35}$$

La quatrième boucle n'est lancée que si la corrélation absolue (due à trois impulsions) dépasse le seuil thr_3 . On notera que cela se traduit par une exploration de complexité variable. Pour limiter encore plus la recherche, le nombre de lancement de la dernière boucle (pour les 4 sous-trames) n'est pas autorisé à dépasser la valeur 600. (Dans le pire cas, ce nombre est de 150 fois en moyenne par sous-trame. Cela revient à n'explorer que 150×8 entrées du répertoire, sans tenir compte du résidu des trois premières boucles).

Une caractéristique spéciale du répertoire est que, pour des délais tonaux inférieurs à 60, on ajoute à la séquence codée une contribution tonale qui est fonction de l'index $PGInd_i$ du vecteur de gain apporté par le prédicteur de tonie à long terme (LTP) (*long term predictor*). En d'autres termes, après la détermination du code algébrique optimal, $v[n]$, ce code est modifié selon la relation $v[n] \leftarrow v[n] + b(PGInd_i)v[n - L_i - e(PGInd_i)]$ où L_i est la période fondamentale exprimée par un entier, les valeurs $\beta(PGInd_i)$ et $\varepsilon(PGInd_i)$ étant présentées sous forme de table. On notera qu'avant l'exploration du répertoire, il y a lieu de modifier de manière similaire la réponse impulsionnelle, si la période L_i a une valeur inférieure à 60.

La dernière étape, une fois la séquence codée $v[n]$ obtenue, consiste à quantifier le gain G . Pour cette opération, on suit la même méthode que pour l'excitation au débit supérieur, c'est-à-dire que l'on progresse pas à pas dans la table de quantification du gain et que l'on choisit l'index $MGInd_i$ qui minimise l'expression suivante: $|G - \tilde{G}_j|, 0 \leq j \leq 23$.

2.17 Décodeur d'impulsions d'excitation

Fichier: EXC_LBC.C	Procédure: Fcbk_Unpk()	Décodage des impulsions d'excitation par répertoire fixe
--------------------	------------------------	--

Le décodage des impulsions d'excitation s'effectue comme suit:

- 1) Tout d'abord, on détermine l'index de gain maximal, G_{max} , par la relation suivante:

$$MGInd_i = GInd_i - PGInd_i \cdot GSize \tag{36}$$

où $GSize = 24$ est la taille de la table des gains \tilde{G} et où $PGInd_i$ est obtenu selon 2.18.

- 2) On décode ensuite les positions des impulsions par décodage combinatoire $\binom{30}{M}$, où M est soit 6 ou 5, pour le débit supérieur. Au débit inférieur, on effectue un décodage direct des index de position.
- 3) Au moyen du bit de grille, on calcule la position impulsionnelle dans la grille (paire/impair).
- 4) Les signes des impulsions sont déduits des bits de signe.
- 5) Pour le codeur à débit supérieur, le décodage du bit indicateur de train d'impulsions n'est effectué que si la période $L_i < 58$.
- 6) Ensuite, le vecteur $v[n]$ est reconstitué au moyen des paramètres ainsi décodés.
- 7) Finalement, on additionne la contribution du prédicteur tonal, $u[n]$, et les contributions impulsionnelles, $v[n]$, pour obtenir le vecteur d'excitation $e[n]$.

2.18 Décodage des informations tonales

Fichier: EXC_LBC.C	Procédure: Get_Rez()	Récupération des résidus dans la mémoire d'excitation
Fichier: EXC_LBC.C	Procédure: Decod_Acbk()	Décodage de la contribution du répertoire adaptatif

Le décodage des informations tonales est effectué comme suit:

- 1) Tout d'abord, on décode le délai du prédicteur tonal pour les sous-trames paires:

$$L_i = PInd_i + 18, \quad i = 0,2 \quad (37)$$

- 2) Ensuite, on décode le délai du prédicteur tonal pour les sous-trames impaires:

$$L_i = L_{i-1} + \Delta_i, \quad i = 1,3 \quad (38)$$

où $\Delta_i \in \{-1,0,+1,+2\}$.

- 3) Le vecteur de gain du prédicteur tonal dans la $i^{\text{ème}}$ sous-trame est déduit de l'index de gain $GInd_i$. Pour le débit inférieur, cet index contient les informations sur le vecteur de gain du prédicteur tonal et sur l'index du gain de la séquence d'impulsions. Dans ce cas, l'index de gain tonal, $PGInd_i$, est calculé comme suit:

$$PGInd_i = \lfloor GInd_i / GSize \rfloor, \quad i = 0..3 \quad (39)$$

où $\lfloor x \rfloor$ indique le plus grand entier $\leq x$. Pour le débit supérieur, si la condition $L_i \geq 58$ est satisfaite, cet index est calculé de la même façon que dans la relation (39). Dans les cas précédents, $PGInd_i$ est un pointeur sur le répertoire de 170 entrées de codes vectoriels du gain. Si la condition n'est pas satisfaite, cet index est un pointeur sur le répertoire de 85 entrées de codes vectoriels du gain et contient une information additionnelle sur le bit indicateur de train d'impulsions. Dans ce cas, l'index de gain tonal est calculé comme suit:

$$PGInd_i = \lfloor GInd_i \&0x7FF / GSize \rfloor, \quad i = 0..3 \quad (40)$$

Le délai du prédicteur tonal et le vecteur de gain sont décodés à partir de ces index et utilisés pour extraire, comme décrit ci-dessous, la contribution tonale $u[n]$. On définit d'abord un signal $e'[n]$ comme suit:

$$\begin{aligned} e'[0] &= e[-L_i - 2] \\ e'[1] &= e[-L_i - 1] \\ e'[n] &= e[(n \bmod L_i) - L_i], \quad 2 \leq n \leq 63 \end{aligned} \quad (41.1)$$

où le terme «mod» désigne l'opérateur de base arithmétique. Ensuite,

$$u[n] = \sum_{j=0}^{j=4} \beta_{ij} e'[n + j], \quad 0 \leq n \leq 59 \quad (41.2)$$

2.19 Mise à jour de la mémoire

Fichier: LPC.C	Procédure: Upd_Rin()	Mise à jour de la mémoire
----------------	----------------------	---------------------------

La dernière tâche de la $i^{\text{ème}}$ sous-trame, avant le début du codage de la sous-trame suivante, consiste à mettre à jour les mémoires du filtre de synthèse $\tilde{A}_i(z)$, du filtre de pondération perceptuelle des formants $W_i(z)$, et du filtre de mise en forme du bruit harmonique $P_i(z)$. Pour effectuer cette mise à jour, on calcule la réponse complète du filtre combiné $S_i(z)$, en faisant passer par ce filtre la séquence d'excitation reconstituée. A la fin de ce filtrage d'excitation, la mémoire du filtre combiné est sauvegardée afin de pouvoir être utilisée pour le calcul de la réponse à entrée nulle au cours du codage du vecteur vocal suivant.

2.20 Affectation des bits

Fichier: UTIL_LBC.C	Procédure: Line_Pack()	Condensation du flux binaire
---------------------	------------------------	------------------------------

Ce paragraphe présente les tables d'affectation des bits aussi bien pour le débit supérieur que pour le débit inférieur. Les principales différences entre ces deux débits résident dans les positions des impulsions et dans le codage des amplitudes. Par ailleurs, au débit inférieur, on utilise toujours le répertoire à 170 entrées pour construire le vecteur de gain du prédicteur à long terme. Voir les Tableaux 2, 3 et 4.

TABLEAU 2/G.723.1

Affectation des bits dans l'algorithme de codage à 6,3 kbit/s

Paramètres codés	Sous-trame 0	Sous-trame 1	Sous-trame 2	Sous-trame 3	Total
Index de codage LPC					24
Délais du répertoire adaptatif	7	2	7	2	18
Combinaison de tous les gains	12	12	12	12	48
Positions des impulsions	20	18	20	18	73 (Note)
Signes des impulsions	6	5	6	5	22
Index de grille	1	1	1	1	4
Total:					189
NOTE – Grâce au fait que le nombre de mots de code contenus dans le répertoire n'est pas une puissance de deux, 3 éléments binaires supplémentaires sont récupérés par la combinaison des 4 éléments binaires de plus fort poids contenus dans l'index de chaque position d'impulsion, ce qui permet de former un seul mot de 13 bits.					

TABLEAU 3/G.723.1

Affectation des bits dans l'algorithme de codage à 5,3 kbit/s

Paramètres codés	Sous-trame 0	Sous-trame 1	Sous-trame 2	Sous-trame 3	Total
Index de codage LPC					24
Délais du répertoire adaptatif	7	2	7	2	18
Combinaison de tous les gains	12	12	12	12	48
Positions des impulsions	12	12	12	12	48
Signes des impulsions	4	4	4	4	16
Index de grille	1	1	1	1	4
Total					158

TABLEAU 4/G.723.1

Liste des paramètres transmis

Nom	Paramètres transmis	Nombre de bits au débit supérieur	Nombre de bits au débit inférieur
LPC	Index de quantification des vecteurs LSP	24	24
ACL0	Délai du répertoire adaptatif	7	7
ACL1	Délai du répertoire adaptatif différentiel	2	2
ACL2	Délai du répertoire adaptatif	7	7
ACL3	Délai du répertoire adaptatif différentiel	2	2
GAIN0	Combinaison des gains adaptatifs et fixes	12	12
GAIN1	Combinaison des gains adaptatifs et fixes	12	12
GAIN2	Combinaison des gains adaptatifs et fixes	12	12
GAIN3	Combinaison des gains adaptatifs et fixes	12	12
POS0	Index des positions impulsionnelles	20 (Note)	12
POS1	Index des positions des impulsions	18 (Note)	12
POS2	Index des positions des impulsions	20 (Note)	12
POS3	Index des positions des impulsions	18 (Note)	12
PSIG0	Index des signes des impulsions	6	4
PSIG1	Index des signes des impulsions	5	4
PSIG2	Index des signes des impulsions	6	4
PSIG3	Index des signes des impulsions	5	4
GRID0	Index de grille	1	1
GRID1	Index de grille	1	1
GRID2	Index de grille	1	1
GRID3	Index de grille	1	1

NOTE – Les 4 bits de plus fort poids de ces mots de code sont combinés pour former un index de 13 bits, MSBPOS.

2.21 Initialisation du codeur

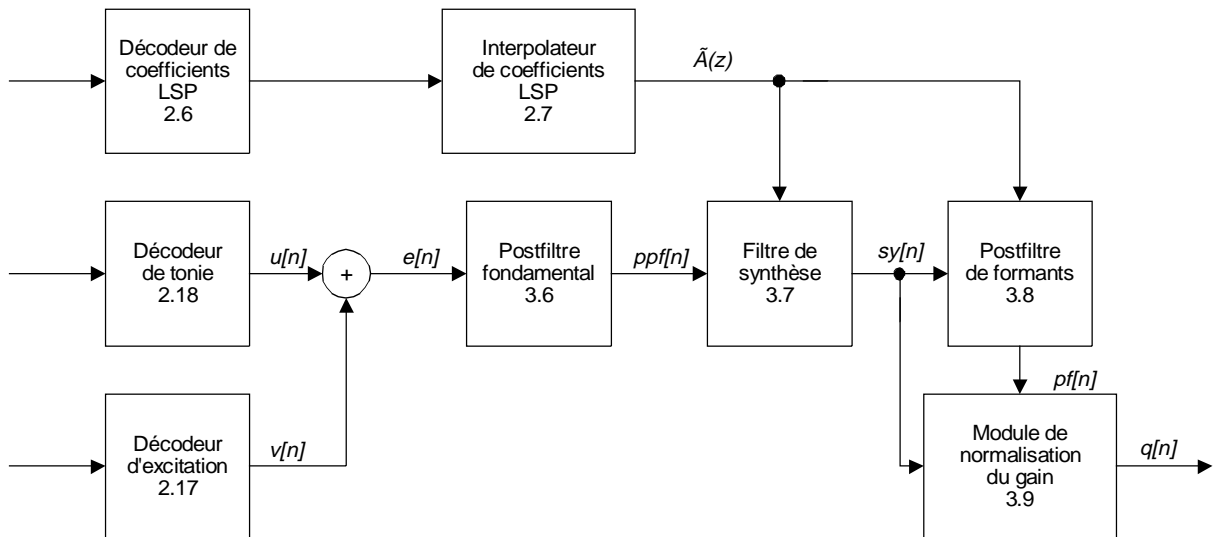
Fichier: CODER.C	Procédure: Init_Coder()	Initialisation du codeur
------------------	-------------------------	--------------------------

Il y a lieu d'initialiser à 0 toutes les variables statiques du codeur, à l'exception du vecteur LSP précédent, qu'il y a lieu d'initialiser au vecteur LSP aperiodique, p_{DC} .

3 Principes de décodage

3.1 Description générale

Comme le codeur, le décodeur fonctionne trame par trame. Il décode d'abord les index de codage LPC quantifiés, puis construit le filtre de synthèse des coefficients LPC. Pour chaque sous-trame, on décode aussi bien les séquences d'excitation par répertoire adaptatif que les séquences d'excitation par répertoire fixe, avant de les injecter dans le filtre de synthèse. Le postfiltre adaptatif se compose d'un postfiltre de formants et d'un postfiltre fondamental anticipée/différée. Le signal d'excitation envoyé dans le postfiltre fondamental, qui à son tour attaque le filtre de synthèse dont la sortie va vers le postfiltre de formants. Un module de normalisation du gain maintient l'énergie du signal au niveau d'entrée du postfiltre de formants.



T1517850-95/d02

FIGURE 2/G.723.1

**Organigramme du décodeur de signaux vocaux
(le numéro de paragraphe correspondant à chaque bloc est indiqué)**

3.2 Décodeur des coefficients LSP

Fichier: LSP.C	Procédure: Lsp_Inq()	Quantification inverse des coefficients LSP
----------------	----------------------	---

Ce bloc est le même qu'au 2.6.

3.3 Interpolateur des coefficients LSP

Fichier: LSP.C	Procédure: Lsp_Int()	Interpolateur de coefficients LSP
Fichier: LSP.C	Procédure: LsptoA()	Conversion des coefficients LSP en coefficients LPC

Ce bloc est le même qu'au 2.7.

3.4 Décodage des informations tonales

Fichier : EXC_LBC.C	Procédure : Get_Rez()	Récupération des résidus dans la mémoire d'excitation
Fichier : EXC_LBC.C	Procédure : Decod_Acbk()	Décodage de la contribution du répertoire adaptatif

Ce bloc est le même qu'au 2.18.

3.5 Décodeur des impulsions d'excitation

Fichier: EXC_LBC.C	Procédure: Fcbk_Unpk()	Décodage des impulsions d'excitation par répertoire fixe
--------------------	------------------------	--

Ce bloc est le même qu'au 2.17.

3.6 Postfiltre fondamental

Fichier: EXC_LBC.C	Procédure: Comp_Lpf()	Calcul des paramètres du postfiltre fondamental
Fichier: EXC_LBC.C	Procédure: Find_F()	Maximisation de l'intercorrélacion anticipée
Fichier: EXC_LBC.C	Procédure: Find_B()	Maximisation de l'intercorrélacion différée
Fichier: EXC_LBC.C	Procédure: Get_Ind()	Calcul du gain
Fichier: EXC_LBC.C	Procédure: Filt_Lpf()	Postfiltrage de la fondamentale

Le postfiltre fondamental sert à améliorer la qualité du signal synthétisé. Il importe de noter que le postfiltrage de la fondamentale est effectué pour chaque sous-trame. Pour l'appliquer, il est nécessaire que l'ensemble du signal d'excitation de trame $\{e[n]\}_n = 0..239$ soit construit et sauvegardé. On obtient l'amélioration de qualité vocale en augmentant le rapport signal sur bruit (SNR) à des fréquences multiples de la période fondamentale, de la façon suivante. Le signal postfiltré $\{ppf[n]\}_n = 0..59$ est déduit du signal d'excitation décodé $\{e[n]\}_n = 0..59$ conformément à l'expression suivante:

$$\begin{aligned} ppf[n] &= g_p \cdot \{e[n] + g_{tp} (w_f \cdot g_f \cdot e[n + M_f] + w_b \cdot g_b \cdot e[n - M_b])\} \\ &= g_p \cdot ppf'[n] \end{aligned} \quad (42)$$

où $e[n]$ est le signal d'excitation décodé. Le calcul des gains, g_p , g_f , g_b , et des délais, M_f , M_b , est fondé sur une analyse par intercorrélacion anticipée et différée. Les pondérations w_f , w_b peuvent avoir les valeurs suivantes: (0,0), (0,1) et (1,0). Les délais sont choisis par maximisation des intercorrélacions. L'intercorrélacion pour le délai tonal anticipé est donnée par:

$$C_f = \sum_{n=0}^{59} e[n]e[n + M_f], \quad M_1 \leq M_f \leq M_2 \quad (43.1)$$

et l'intercorrélacion pour le délai tonal différé est donnée par:

$$C_b = \sum_{n=0}^{59} e[n]e[n - M_b], \quad M_1 \leq M_b \leq M_2 \quad (43.2)$$

où $M_1 = L_i - 3$ et $M_2 = L_i + 3$ et où les valeurs $\{L_i\}_{i=0,2}$ sont les délais tonaux reçus pour la première et pour la troisième sous-trame. Le délai L_0 est utilisé pour les deux premières sous-frames et le délai L_2 est utilisé pour les deux dernières. On notera que si l'une des corrélations maximales est négative ou qu'il n'y ait aucune valeur d'échantillon $e[n + M_f]$ disponible pour un terme $n \in [0...59]$, la pondération et le délai correspondants sont mis à zéro. Cela donne quatre cas possibles: (0) les deux corrélations maximales sont négatives et aucune pondération du postfiltre fondamental n'a besoin d'être calculée; (1) seule la corrélation anticipée maximale est positive, si bien qu'on la choisit; (2) seule la corrélation différée maximale est positive, si bien qu'on la choisit; (3) les deux corrélations maximales sont positives et on choisit celle qui apporte la plus grande contribution. Cette procédure est décrite ci-dessous. Pour les cas (1), (2) et (3), on calcule les énergies de signal correspondantes (T_{en} , D_f et/ou D_b) pour le délai tonal optimal (M_f ou M_b), conformément aux équations (44.1), (44.2) et (44.3):

$$D_f = \sum_{n=0}^{59} e[n + M_f] e[n + M_f] \quad (44.1)$$

$$D_b = \sum_{n=0}^{59} e[n - M_b] e[n - M_b] \quad (44.2)$$

$$T_{en} = \sum_{n=0}^{59} e[n] e[n] \quad (44.3)$$

L'énergie tonale anticipée est donnée par l'équation suivante:

$$E_f = \sum_{n=0}^{59} (e[n] - g_f e[n + M_f])^2 \quad (45.1)$$

et l'énergie tonale différée est donnée par l'équation suivante:

$$E_b = \sum_{n=0}^{59} (e[n] - g_b e[n - M_b])^2 \quad (45.2)$$

L'énergie E est minimisée si l'on maximise le rapport $\frac{C^2}{D}$. Dans le cas (3), la sélection entre énergie anticipée et énergie différée est effectuée par sélection du plus grand rapport entre $\frac{C_f^2}{D_f}$ et $\frac{C_b^2}{D_b}$. Le gain de prédiction est égal à $-10 \log_{10} \left(1 - \frac{C^2}{DT_{en}}\right)$. Si ce gain est inférieur à 1,25 dB, la contribution est jugée négligeable et aucun postfiltre fondamental n'est utilisé. Si le postfiltrage de la fondamentale est effectué, le gain optimal est donné par le rapport:

$$g = \frac{C}{D} \quad (46)$$

Selon le débit binaire du codeur de signaux vocaux, le gain optimal est multiplié par un facteur de pondération, γ_{lp} , qui est égal à 0,1875 pour le débit supérieur et à 0,25 pour le débit inférieur. Finalement, le gain de normalisation, g_p , est calculé comme suit:

$$g_p = \sqrt{\frac{\sum_{n=0}^{59} e^2[n]}{\sum_{n=0}^{59} (ppf'[n])^2}} \quad (47)$$

Si le dénominateur de l'équation (47) est inférieur au numérateur, le gain est réglé à 1.

3.7 Filtre de synthèse en codage LPC

Fichier: LPC.C	Procédure: Synt()	Synthétise le signal vocal reconstruit
----------------	-------------------	--

Le filtre de synthèse LPC du 10^e ordre $\tilde{A}_l(z)$ est utilisé pour synthétiser le signal vocal à partir du résidu tonal postfiltré et décodé, $ppf[n]$, comme suit:

$$sy[n] = ppf[n] + \sum_{j=1}^{10} \tilde{a}_{ij} sy[n - j] \quad (48)$$

3.8 Postfiltre des formants

Fichier: LPC.C	Procédure: Spf()	Postfiltrage des formants
Fichier: UTIL_LBC.C	Procédure: Comp_En()	Calcul de l'énergie du signal synthétisé

On utilise un postfiltre à moyenne mobile autorégressive (ARMA) (*autoregressive moving average*) conventionnel. La fonction de transfert du postfiltre à court terme est donnée par les équations suivantes:

$$k = \frac{\sum_{n=1}^{59} sy[n]sy[n-1]}{\sum_{n=0}^{59} sy[n]sy[n]} \quad (49.1)$$

$$k_1 = \frac{3}{4}k_{old} + \frac{1}{4}k \quad (49.2)$$

$$F(z) = \frac{1 - \sum_{i=1}^{10} \tilde{a}_i \lambda_1^i z^{-i}}{1 - \sum_{i=1}^{10} \tilde{a}_i \lambda_2^i z^{-i}} (1 - 0,25k_1 z^{-1}) \quad (49.3)$$

où $\lambda_1 = 0,65$ et où $\lambda_2 = 0,75$, k est le premier coefficient d'autocorrélation (réflexion) qui est estimé à partir du signal vocal synthétisé $sy[n]$ et k_{1old} est la valeur de k_1 issue de la sous-trame précédente. Le signal postfiltré $pf[n]$ est obtenu en sortie du postfiltre de formants avec le signal synthétisé $sy[n]$ en entrée.

3.9 Module de normalisation du gain

Fichier: UTIL_LBC.C	Procédure: Scale()	Ajustement du gain du signal postfiltré
---------------------	--------------------	---

Ce module reçoit deux vecteurs d'entrée: le vecteur vocal synthétisé $\{sy[n]\}_{n=0..59}$ et le vecteur de sortie postfiltré $\{pf[n]\}_{n=0..59}$. On calcule d'abord, au moyen de l'équation ci-dessous, le rapport d'amplitude g_s :

$$g_s = \sqrt{\frac{\sum_{n=0}^{59} sy^2 [n]}{\sum_{n=0}^{59} pf^2 [n]}} \quad (50)$$

Si le dénominateur est égal à 0, le rapport g_s est mis à 1.

On obtient ensuite le vecteur de sortie $q[n]$ en normalisant le signal postfiltré $pf[n]$, puis on actualise le gain $g[n]$ au moyen des expressions suivantes, selon le cas:

$$g[n] = (1 - \alpha)g[n - 1] + \alpha g_s \quad (51)$$

$$q[n] = pf[n] \cdot g[n] \cdot (1 + \alpha) \quad (52)$$

où α est égal à 1/16.

3.10 Traitement par interpolation des trames

Fichier: EXC_LBC.C	Procédure: Comp_Info()	Calcule l'index d'interpolation
Fichier: EXC_LBC.C	Procédure: Regen()	Régénération de la trame actuelle

Ce codeur a été conçu pour résister aux effacements de trame indiqués. Une stratégie de masquage des erreurs par effacement de trames a été intégrée dans le décodeur. Il est toutefois nécessaire que cette stratégie soit déclenchée par une indication externe du fait que le flux binaire a été effacé pour la trame actuelle. Etant donné que le codeur a été conçu pour résister à des erreurs sporadiques, aucun mécanisme de correction n'est prévu pour les erreurs aléatoires sur les bits. Si un effacement de trame s'est produit, le décodeur passe du mode de décodage normal au mode de masquage d'effacement de trame. La procédure d'interpolation de trame est lancée indépendamment pour les coefficients LSP et pour le signal résiduel.

3.10.1 Interpolation des coefficients LSP

Le décodage des coefficients LSP en mode d'interpolation de trame se déroule comme suit:

- 1) le vecteur \tilde{e}_n est mis à zéro;
- 2) le vecteur prédit, \bar{p}_n , est ajouté au vecteur, \tilde{e}_n , et au vecteur aperiodique, p_{DC} , afin de former le vecteur LSP décodé, \tilde{p}_n . Un prédicteur non adaptatif différent est utilisé pour prédire le vecteur \bar{p}_n : $b_e = 23/32$.

A partir de là, le décodage des coefficients LSP se poursuit comme indiqué au 2.6, sauf que $\Delta_{min} = 62,5$ Hz au lieu de 31,25 Hz.

3.10.2 Interpolation des résidus

On effectue l'interpolation des résidus du signal de deux façons différentes, selon la dernière bonne trame trouvée avant la trame effacée. Cette trame est vérifiée par un classificateur d'éléments voisés et non voisés.

Ce classificateur est fondé sur une fonction de maximisation d'une intercorrélation. Les 120 derniers échantillons de la trame sont intercorrélés avec le délai $L_2 \pm 3$. L'index qui atteint la valeur maximale d'intercorrélacion est retenu comme éventuel index d'interpolation. Puis on teste le gain prédictif du meilleur vecteur. Si ce gain est supérieur à 0,58 dB, la trame est déclarée de type voisé, sinon de type non voisé.

Le classificateur revient à 0 dans le cas d'une trame non voisée et à la valeur tonale estimée dans le cas d'une trame voisée. Si la trame a été déclarée non voisée, on sauvegarde la moyenne des index de gain pour les sous-trames 2 et 3.

Si la trame actuelle était marquée comme effacée et que la trame précédente ait été classifiée comme non voisée, l'excitation de la trame actuelle est produite au moyen d'un générateur de nombres aléatoires uniformes, dont la sortie est normalisée au moyen de la valeur de gain précédemment calculée.

Dans le cas d'une trame voisée, la trame actuelle est régénérée par une excitation périodique dont la fondamentale est égale à la valeur fournie par le classificateur.

Si l'état d'effacement de trame se maintient pour les deux trames suivantes, le vecteur régénéré est affaibli de 2,5 dB supplémentaires dans chaque trame. Après 3 trames interpolées, la sortie est totalement supprimée.

3.11 Initialisation du décodeur

Fichier: DECOD.C	Procédure: Init_Decod()	Initialisation du décodeur
------------------	-------------------------	----------------------------

Il convient d'initialiser à 0 toutes les variables statiques du décodeur, à l'exception des suivantes:

- 1) le vecteur LSP précédent, qu'il convient d'initialiser au vecteur LSP aperiodique, p_{DC} ;
- 2) le gain de postfiltrage $g[-1]$, qu'il convient d'initialiser à 1.

4 Condensation des flux binaires

NOTE – Chaque bit des paramètres transmis est désigné par l'expression PAR(x)_By: dans laquelle PAR est le nom du paramètre et x désigne l'index de sous-trame si applicable et où y désigne la position binaire à partir de 0 (bit de plus faible poids) jusqu'au bit de poids fort. L'expression PAR $_x$ _By..PAR $_x$ _Bz désigne l'étendue des bits y à z qui sont transmis. Le bit non utilisé est désigné par UB (valeur = 0). L'expression RATEFLAG_B0 indique si l'on utilise le débit supérieur (0) ou inférieur (1) pour la trame actuelle. L'expression VADFLAG_B0 indique si la trame actuelle correspond à de la conversation active (0) ou à des échantillons non vocaux (1). La combinaison des expressions RATEFLAG et VADFLAG mises toutes les deux à 1 est réservée pour usage ultérieur. Les octets sont transmis dans l'ordre indiqué par les Tableaux 5 et 6. A l'intérieur de chaque octet, les bits sont disposés de la manière suivante: bit de poids fort à gauche et bit de plus faible poids à droite.

5 Code C de l'ANSI

Le code C de l'ANSI simule le codeur/décodeur à double débit en arithmétique à virgule fixe en mots de 16 bits. Il est disponible auprès de l'UIT-T. Le Tableau 7 énumère tous les fichiers composant ce code.

6 Glossaire

Le glossaire ci-dessous énumère les symboles mathématiques utilisés dans le texte, avec une brève description de ce qu'ils représentent. Voir le Tableau 8.

TABLEAU 5/G.723.1

Condensation en octets du flux binaire pour le codec à débit supérieur

Débit supérieur

Octets transmis	PARx By, ...
1	LPC_B5...LPC_B0, VADFLAG_B0, RATEFLAG_B0
2	LPC_B13...LPC_B6
3	LPC_B21...LPC_B14
4	ACL0_B5...ACL0_B0, LPC_B23, LPC_B22
5	ACL2_B4...ACL2_B0, ACL1_B1, ACL1_B0, ACL0_B6
6	GAIN0_B3...GAIN0_B0, ACL3_B1, ACL3_B0, ACL2_B6, ACL2_B5
7	GAIN0_B11...GAIN0_B4
8	GAIN1_B7...GAIN1_B0
9	GAIN2_B3...GAIN2_B0, GAIN1_B11...GAIN1_B8
10	GAIN2_B11...GAIN2_B4
11	GAIN3_B7...GAIN3_B0
12	GRID3_B0, GRID2_B0, GRID1_B0, GRID0_B0, GAIN3_B11...GAIN3_B8
13	MSBPOS_B6...MSBPOS_B0, UB
14	POS0_B1, POS0_B0, MSBPOS_B12...MSBPOS_B7
15	POS0_B9...POS0_B2
16	POS1_B2, POS1_B0, POS0_B15...POS0_B10
17	POS1_B10...POS1_B3
18	POS2_B3...POS2_B0, POS1_B13...POS1_B11
19	POS2_B11...POS2_B4
20	POS3_B3...POS3_B0, POS2_B15...POS2_B12
21	POS3_B11...POS3_B4
22	PSIG0_B5...PSIG0_B0, POS3_B13, POS3_B12
23	PSIG2_B2...PSIG2_B0, PSIG1_B4...PSIG1_B0
24	PSIG3_B4...PSIG3_B0, PSIG2_B5...PSIG2_B3

TABLEAU 6/G.723.1

Condensation en octets du flux binaire pour le codec à débit inférieur

Débit inférieur

Octets transmis	PAR _x B _y , ...
1	LPC_B5...LPC_B0, VADFLAG_B0, RATEFLAG_B0
2	LPC_B13...LPC_B6
3	LPC_B21...LPC_B14
4	ACL0_B5...ACL0_B0, LPC_B23, LPC_B22
5	ACL2_B4...ACL2_B0, ACL1_B1, ACL1_B0, ACL0_B6
6	GAIN0_B3...GAIN0_B0, ACL3_B1, ACL3_B0, ACL2_B6, ACL2_B5
7	GAIN0_B11...GAIN0_B4
8	GAIN1_B7...GAIN1_B0
9	GAIN2_B3...GAIN2_B0, GAIN1_B11...GAIN1_B8
10	GAIN2_B11...GAIN2_B4
11	GAIN3_B7...GAIN3_B0
12	GRID3_B0, GRID2_B0, GRID1_B0, GRID0_B0, GAIN3_B11...GAIN3_B8
13	POS0_B7...POS0_B0
14	POS1_B3...POS1_B0, POS0_B11...POS0_B8
15	POS1_B11...POS1_B4
16	POS2_B7...POS2_B0
17	POS3_B3...POS3_B0, POS2_B11...POS2_B8
18	POS3_B11...POS3_B4
19	PSIG1_B3...PSIG1_B0, PSIG0_B3...PSIG0_B0
20	PSIG3_B3...PSIG3_B0, PSIG2_B3...PSIG2_B0

TABLEAU 7/G.723.1

Liste des noms de fichiers du logiciel

Nom du fichier	Description
TYPEDEF.H	définition du type de données en fonction de la machine
CST_LBC.H	définition des constantes pour la Recommandation G.723
LBCCODEC.C	programme principal pour codecs vocaux G.723
LBCCODEC.H	fonctions prototypes
CODER.C	codeur de signaux vocaux G.723 pour les deux débits binaires
CODER.H	fonctions prototypes
DECOD.C	décodeur de signaux vocaux G.723 pour les deux débits binaires
DECOD.H	prototypes des deux fonctions
LPC.C	analyse prédictive linéaire
LPC.H	fonctions prototypes
LSP.C	fonctions relatives aux paires de raies spectrales (LSP), quantific.
LSP.H	fonctions prototypes
EXC_LBC.C	répertoires adaptatifs et fixes (MP-MLQ, ACELP)
EXC_LBC.H	fonctions prototypes
UTIL_LBC.C	fonctions diverses (filtre HPF, condensation, décondensation, contrôle d'entrées/sortie, etc.)
UTIL_LBC.H	fonctions prototypes
TAB_LBC.C	tables de constantes
TAB_LBC.H	déclaration externe des tables de constantes
BASOP.C	arithmétique en virgule fixe et opérateurs logiques
BASOP.H	fonctions prototypes

TABLEAU 8/G.723.1

Glossaire des symboles utilisés dans le texte

Symbole	Description
$y[j]$	échantillons vocaux d'entrée
$s[j]$	trame vocale d'entrée à 240 échantillons
$x[j]$	trame vocale filtrée en passe-haut
$R[j]$	fonction d'autocorrélation, $n = 0, 1, \dots, 10$
a_i	vecteur de coefficient LPC de la sous-trame i
\tilde{a}_i	vecteur de coefficient LPC quantifié de la sous-trame i
p'	vecteur de coefficient LSP non quantifié
p	vecteur de coefficient LSP sans composante aperiodique
P_{DC}	vecteur aperiodique à long terme des valeurs de coefficient LSP
\tilde{p}_n	vecteur de coefficient LSP décodé pour la trame n
\bar{p}_n	vecteur de coefficient LSP prédit sans composante aperiodique
e_n	vecteur d'erreur résiduelle sur coefficient LSP pour la trame n
\tilde{e}_n	valeur quantifiée du vecteur e_n
W_n	matrice de pondération en diagonale pour la quantification des coefficients LSP
γ_1, γ_2	poils pour le filtre de pondération perceptive, 0,9, 0,5
W_i	filtre de pondération perceptive des formants pour la sous-trame i
$f[n]$	signal vocal à la sortie du filtre de pondération perceptive des formants
L_{OL}	estimation de la période fondamentale en boucle ouverte
$C_{OL}(j)$	fonction de détermination du critère d'intercorrélation maximale pour l'estimation de la période fondamentale en boucle ouverte
$C_{PW}(j)$	fonction de détermination du critère d'intercorrélation maximale pour l'estimation de la période fondamentale à la sortie du filtre de mise en forme du bruit harmonique
β	gain du filtre de mise en forme du bruit harmonique
L	délai optimal pour le filtre de mise en forme du bruit harmonique
G_{opt}	gain optimal pour le filtre de mise en forme du bruit harmonique
E	énergie du signal vocal pondéré
P_i	filtre de mise en forme du bruit harmonique pour la sous-trame i
S_i	combinaison des filtres de mise en forme du bruit harmonique, de pondération des formants et de synthèse pour la sous-trame i
$h[n]$	réponse impulsionnelle du filtre combiné
$z[n]$	réponse à entrée nulle du filtre combiné
$w[n]$	signaux vocaux pondérés par le filtre de mise en forme du bruit harmonique
$t[n]$	vecteur cible
$p[n]$	vecteur de contribution du prédicteur de tonie
$r[n]$	vecteur de signal résiduel
$r'[n]$	vecteur d'excitation filtré
$v[n]$	vecteur d'excitation par répertoire fixe
M	nombre d'impulsions

TABLEAU 8/G.723.1 (fin)

Glossaire des symboles utilisés dans le texte

α_k	signe de l'impulsion k
m_k	position de l'impulsion k
$d[n]$	fonction d'intercorrélation des vecteurs $h[n]$ et $r[n]$
G_{\max}	gain estimé des impulsions en débit supérieur
L_i	délai tonal pour la sous-trame i
\mathbf{H}	matrice convolutionnelle triangulaire inférieure de Toeplitz avec diagonales $h[n]$
Φ	matrice de covariance formée par $\mathbf{H}^T \mathbf{H}$
max_3	corrélacion maximale pour les 3 premières impulsions au débit inférieur
av_3	corrélacion moyenne pour les 3 premières impulsions au débit inférieur
thr_3	seuil pour la corrélacion des 3 premières impulsions au débit inférieur
$MGIInd_i$	index de gain maximal de la sous-trame i
$GInd_i$	index de gain de la sous-trame i
$PIInd_i$	index de tonie de la sous-trame i
$PGInd_i$	index de délai tonal de la sous-trame i
$GSize$	longueur du répertoire de gain d'excitation, 24
\tilde{G}	gain quantifié
β_{ij}	vecteur de gain du prédicteur tonal
$u[n]$	vecteur d'excitation par répertoire adaptatif
$e[n]$	vecteur d'excitation en sortie du décodeur combiné
$ppf[n]$	signal d'excitation en sortie du postfiltre fondamental
$M_f M_b$	délais anticipés/différés optimaux pour le postfiltre fondamental
$ppf'[n]$	signal d'excitation non normalisé en gain à la sortie du postfiltre fondamental
γ_{lp}	facteur de pondération du gain, 0,1875 ou 0,25
g_p	gain de normalisation en sortie de postfiltre fondamental
$g_f g_b$	gains anticipés/différés optimaux pour le postfiltre fondamental
$C_f C_b$	intercorrélacions des délais anticipés/différés du signal d'excitation
$D_f D_b$	énergies correspondant aux délais anticipés/différés du signal d'excitation
$E_f E_b$	énergies tonales anticipées et différées pour le postfiltre fondamental
T_{en}	énergie du signal d'excitation
$sy[n]$	signal vocal en sortie du filtre de synthèse des coefficients LPC
$pf[n]$	signal en sortie du postfiltre de formants
$q[n]$	vecteur du signal vocal de sortie
$g[n]$	vecteur du signal de gain à la sortie du postfiltre de formants
k_1	coefficient d'autocorrélacion (réflexion) interpolé pour le filtre de compensation des écarts de niveau à court terme
λ_1, λ_2	poinds pour le postfiltre de formants, 0,65, 0,75
g_s	rapport d'amplitude des vecteurs sy et pf
α	valeur 0,0625 du module de normalisation du gain
b_e	prédicteur fixe pour l'interpolation des coefficients LSP en mode de masquage d'effacement de trame, 23/32