



UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

**UIT-T**

SECTOR DE NORMALIZACIÓN  
DE LAS TELECOMUNICACIONES  
DE LA UIT

**G.723.1**

(03/96)

**ASPECTOS GENERALES DE LOS  
SISTEMAS DE TRANSMISIÓN DIGITAL**

---

**CÓDEC DE VOZ DE DOBLE VELOCIDAD  
PARA LA TRANSMISIÓN  
EN COMUNICACIONES MULTIMEDIOS  
A 5,3 Y 6,3 kbit/s**

**Recomendación UIT-T G.723.1**

---

## PREFACIO

El UIT-T (Sector de Normalización de las Telecomunicaciones) es un órgano permanente de la Unión Internacional de Telecomunicaciones (UIT). Este órgano estudia los aspectos técnicos, de explotación y tarifarios y publica Recomendaciones sobre los mismos, con miras a la normalización de las telecomunicaciones en el plano mundial.

La Conferencia Mundial de Normalización de las Telecomunicaciones (CMNT), que se celebra cada cuatro años, establece los temas que han de estudiar las Comisiones de Estudio del UIT-T, que a su vez producen Recomendaciones sobre dichos temas.

La aprobación de Recomendaciones por los Miembros del UIT-T es el objeto del procedimiento establecido en la Resolución N.º 1 de la CMNT (Helsinki, 1 al 12 de marzo de 1993).

La Recomendación UIT-T G.723.1 ha sido preparada por la Comisión de Estudio 15 (1993-1996) del UIT-T y fue aprobada por el procedimiento de la Resolución N.º 1 de la CMNT el 19 de marzo de 1996.

---

### NOTA

En esta Recomendación, la expresión «Administración» se utiliza para designar, en forma abreviada, tanto una administración de telecomunicaciones como una empresa de explotación reconocida de telecomunicaciones.

© UIT 1996

Es propiedad. Ninguna parte de esta publicación puede reproducirse o utilizarse, de ninguna forma o por ningún medio, sea éste electrónico o mecánico, de fotocopia o de microfilm, sin previa autorización escrita por parte de la UIT.

# ÍNDICE

		<i>Página</i>
1	Introducción .....	1
	1.1 Alcance .....	1
	1.2 Velocidades binarias .....	1
	1.3 Señales de entrada posibles.....	1
	1.4 Retardo.....	1
	1.5 Descripción del códec de voz .....	1
2	Principios del codificador.....	2
	2.1 Descripción general .....	2
	2.2 Formador de trama.....	2
	2.3 Filtro paso alto .....	3
	2.4 Análisis LPC.....	4
	2.5 Cuantificador LSP .....	4
	2.6 Decodificador LSP.....	5
	2.7 Interpolación LSP .....	6
	2.8 Filtro de ponderación perceptual formante .....	6
	2.9 Estimación del tono .....	7
	2.10 Procesamiento de subtrama .....	7
	2.11 Conformación del ruido armónico.....	7
	2.12 Calculador de la respuesta de impulso.....	8
	2.13 Respuesta de entrada cero y substracción de timbre.....	9
	2.14 Predictor del tono.....	9
	2.15 Excitación de velocidad alta (MP-MLQ).....	9
	2.16 Excitación de velocidad baja (ACELP) .....	11
	2.17 Decodificador de excitación .....	13
	2.18 Decodificación de la información de tono .....	14
	2.19 Actualización de la memoria .....	14
	2.20 Asignación de bits.....	15
	2.21 Inicialización del códec .....	16
3	Principios del decodificador.....	17
	3.1 Descripción general .....	17
	3.2 Decodificador LSP.....	17
	3.3 Interpolador LSP.....	17
	3.4 Decodificación de la información de tono .....	18
	3.5 Decodificador de excitación .....	18
	3.6 Postfiltro de tono.....	18
	3.7 Filtro de síntesis LPC.....	20
	3.8 Postfiltro formante .....	20
	3.9 Unidad de escalamiento de ganancia .....	21
	3.10 Tratamiento de la interpolación de trama .....	21
	3.11 Inicialización del decodificador.....	22
4	Empaquetamiento del tren de bits .....	22
5	Código C de ANSI .....	22
6	Glosario.....	22

## Sumario

Esta Recomendación especifica una representación codificada que se puede utilizar para comprimir la voz u otra señal de audio componente de servicios multimedia a velocidad binaria muy baja como parte de la familia general de normas H.324. Este códec tiene dos velocidades binarias asociadas: 5,3 y 6,3 kbit/s. La velocidad más alta tiene mejor calidad. La velocidad más baja da una calidad buena y proporciona a los diseñadores de sistemas más flexibilidad. Ambas velocidades son una parte obligatoria del codificador y del decodificador. Es posible conmutar entre las dos velocidades en cualquier frontera de trama. También es posible una opción para el funcionamiento con velocidad variable utilizando transmisión discontinua y relleno de ruido durante los intervalos sin voz.

Este códec se optimizó para representar la voz con una calidad alta a las velocidades mencionadas mediante una complejidad limitada. Codifica la voz u otras señales audio en tramas mediante la codificación predictiva lineal de análisis por síntesis. La señal de excitación del códec de alta velocidad es la cuantificación multiimpulso de máxima verosimilitud (MP-MLQ, *multipulse maximum likelihood quantization*) y la del códec de velocidad baja es la predicción lineal excitada por tabla de códigos algebraicos (ACELP, *algebraic code excited linear prediction*). La duración de la trama es de 30 ms y hay un preanálisis adicional de 7,5 msec, lo que resulta en un retardo algorítmico total de 37,5 msec. Todos los retardos adicionales de este códec son retardos de procesamiento de la implementación, retardos de transmisión del enlace de comunicación y retardos de la memoria intermedia para el protocolo de multiplexación.

La descripción de esta Recomendación se efectúa en términos de operaciones matemáticas de coma fija y exactitud de bit. El código C de ANSI que se indica en la cláusula 5 es parte integrante de la misma y tendrá preferencia frente a las descripciones matemáticas de este texto en el caso de que aparezcan discrepancias. La UIT también puede proporcionar un conjunto no exhaustivo de secuencias de prueba que se pueden emplear junto con el código C.

## **CÓDEC DE VOZ DE DOBLE VELOCIDAD PARA LA TRANSMISIÓN EN COMUNICACIONES MULTIMEDIOS A 5,3 Y 6,3 kbit/s**

*(Ginebra, 1996)*

### **1 Introducción**

#### **1.1 Alcance**

Esta Recomendación especifica una representación de códec (codificador-decodificador) que se puede utilizar para comprimir la voz u otras señales audio componentes de servicios multimedia a velocidad binaria muy baja. Al diseñar este códec, la principal aplicación considerada fue la telefonía visual a velocidad binaria muy baja como parte de la familia general de normas H.324.

#### **1.2 Velocidades binarias**

Este códec tiene asociadas dos velocidades binarias. Se trata de 5,3 y 6,3 kbit/s. La velocidad más alta tiene mucho mejor calidad. La velocidad más baja da una buena calidad y proporciona a los diseñadores de sistema más flexibilidad. Ambas velocidades son una parte obligatoria del codificador y del decodificador. Se puede conmutar entre ambas velocidades en cualquier frontera de trama de 30 ms. También se puede utilizar el funcionamiento con velocidad variable mediante la transmisión discontinua y el relleno de ruido durante los intervalos sin voz.

#### **1.3 Señales de entrada posibles**

El códec se optimizó de forma que represente la voz con gran calidad a las velocidades mencionadas y con una complejidad restringida. La música y otras señales audio no se representan con la misma fidelidad que la voz, pero con este códec se pueden comprimir y descomprimir.

#### **1.4 Retardo**

Este códec codifica la voz u otras señales audio en tramas de 30 ms. Además, tiene un preanálisis de 7,5 ms, lo que resulta en un retardo algorítmico total de 37,5 ms. Todos los demás retardos en la implementación y el funcionamiento de este códec se deben a:

- i) el tiempo real del procesamiento de los datos en el codificador y el decodificador;
- ii) el tiempo de transmisión por el enlace de comunicaciones;
- iii) el retardo adicional de la memoria intermedia para el protocolo de multiplexación.

#### **1.5 Descripción del códec de voz**

La descripción del algoritmo de codificación de la voz en esta Recomendación se hace en términos de operaciones matemáticas de coma fija y exactitud de bits. El código C de ANSI que se indica en la cláusula 5, y que constituye una parte integrante de esta Recomendación, refleja este enfoque de descripción de coma fija y exactitud de bits. Las descripciones matemáticas del codificador y del decodificador, que aparecen en las cláusulas 2 y 3, respectivamente, pueden tener realizaciones diversas, que quizá conduzcan a una realización de códec que no cumple las disposiciones de esta Recomendación. Por consiguiente, la descripción del algoritmo del código C de la cláusula 5 tendrá precedencia sobre las descripciones matemáticas de las cláusulas 2 y 3 en el caso de que se aparezcan discrepancias. Puede solicitarse a la UIT una lista no exhaustiva de secuencias de prueba para utilizar junto con el código C.

## 2 Principios del codificador

### 2.1 Descripción general

Este códec está diseñado para el funcionamiento con una señal digital obtenida filtrando primero la entrada analógica con la anchura de banda de telefonía (Recomendación G.712), muestreándola luego a 8000 Hz y convirtiéndola a señal MIC lineal de 16 bit para su entrada en el codificador. Habrá que convertir la salida del decodificador a señal analógica mediante medios similares. Otras características entrada/salida, como las especificadas en la Recomendación G.711 para los datos MIC a 64 kbit/s, se convertirán a MIC lineal a 16 bit antes de la codificación o de MIC lineal a 16 bit al formato apropiado después de la decodificación. En esta Recomendación se define el tren de bits que va del codificador hasta el decodificador.

El códec se basa en los principios de la codificación de predicción lineal análisis por síntesis, y trata de hacer mínima una señal de error ponderada perceptualmente. El codificador funciona con bloques (tramas) de 240 muestras cada uno. Ello equivale a 30 ms a una velocidad de muestreo de 8 kHz. Cada bloque se pasa primero por un filtro paso alto para suprimir la componente continua, y luego se divide en cuatro subtramas de 60 muestras cada una. Para cada subtrama, se calcula un filtro de códec de predicción lineal (LPC, *linear prediction coder*) de décimo orden utilizando la señal de entrada no procesada. El filtro LPC para la última subtrama se cuantifica con un cuantificador vectorial de división predictiva (PSVQ, *predictive split vector quantizer*). Los coeficientes LPC no cuantificados se utilizan para construir el filtro de ponderación perceptual de corto plazo, que se utiliza para filtrar toda la trama y obtener la señal de voz ponderada perceptualmente.

Para cada dos subtramas (120 muestras), se calcula el periodo de tono en bucle abierto  $L_{OL}$  mediante la señal vocal ponderada. Esta estimación del tono se realiza con bloques de 120 muestras. El periodo de tono se busca en la gama de 18 a 142 muestras.

A partir de ese punto, la voz se procesa a 60 muestras por subtramas.

Utilizando el periodo de tono estimado calculado anteriormente, se construye un filtro de conformación de ruido armónico. La combinación del filtro de síntesis LPC, el filtro de ponderación perceptual formante y el filtro de conformación del ruido armónico se utiliza para crear una respuesta de impulso. La respuesta del impulso se utiliza para los cálculos posteriores.

Con la estimación del periodo de tono,  $L_{OL}$ , y la respuesta de impulso, se calcula un predictor de tono en bucle cerrado. Se utiliza un predictor de tono de quinto orden. El periodo de tono se calcula como un valor diferencial pequeño respecto de la estimación de tono en bucle abierto. La contribución del predictor de tono se resta del vector objetivo inicial. Tanto el periodo de tono como el valor diferencial se transmiten al decodificador.

Por último, se aproxima la componente no periódica de la excitación. Para la velocidad alta, se utiliza la excitación del tipo cuantificación multiimpulso de máxima verosimilitud (MP-MLQ), y para la velocidad baja, una predicción lineal con excitación por tabla de códigos algebraicos (ACELP).

El diagrama de bloques del codificador se muestra en la Figura 1.

### 2.2 Formador de trama

File : LBCCODEC.C	Procedure : main()	Lee tramas de entrada de 240 muestras
File : CODER.C	Procedure : Coder ()	Ejecuta la división en subtramas

El códec procesa la voz mediante el almacenamiento en memoria intermedia de muestras vocales consecutivas,  $y[n]$ , en tramas de 240 muestras,  $s[n]$ . Para el cálculo de la estimación del tono, cada trama se divide en dos partes de 120 muestras. Cada parte se divide a su vez en dos, de manera que cada trama queda dividida en cuatro subtramas de 60 muestras cada una.

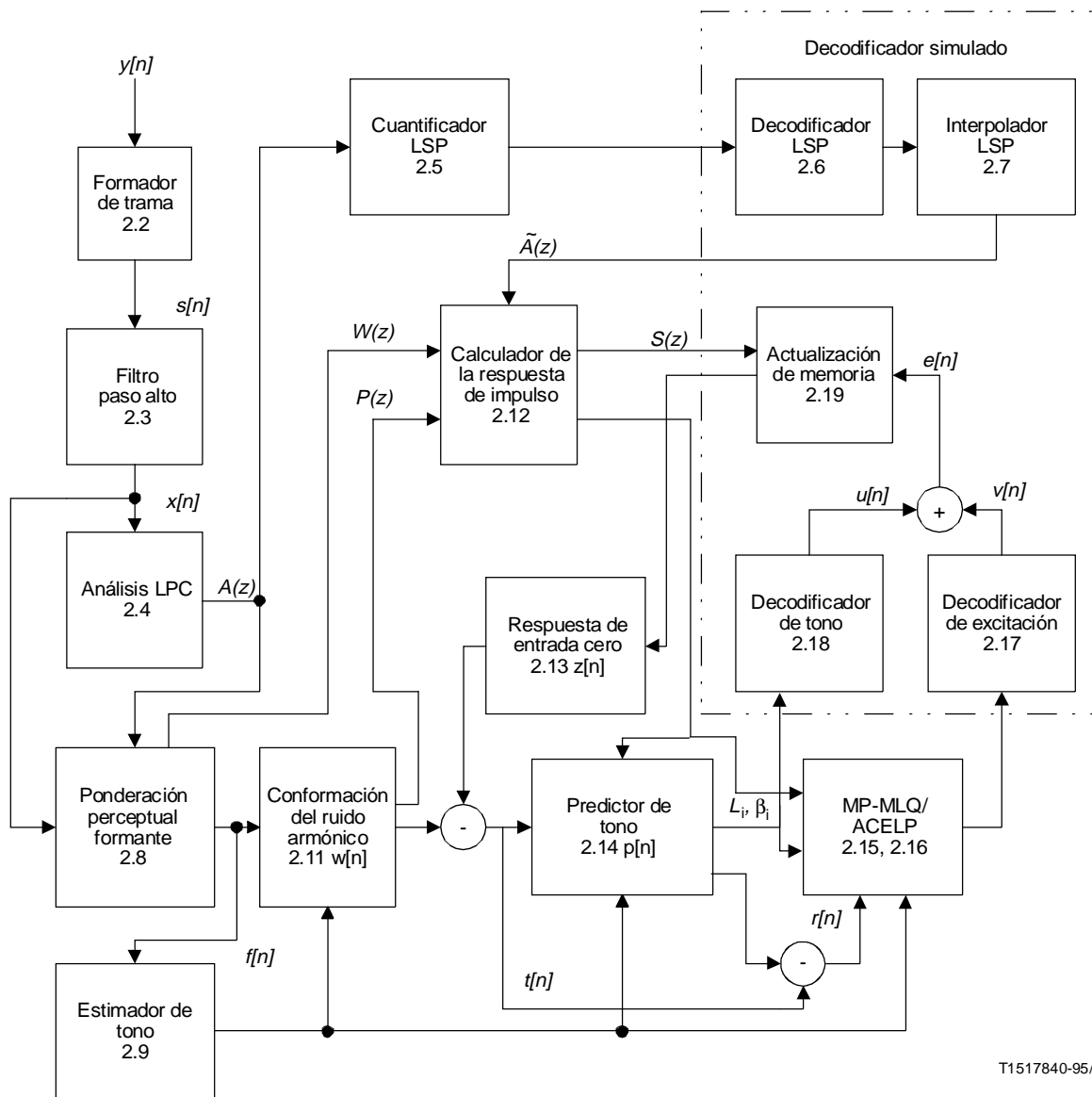
### 2.3 Filtro paso alto

File : UTIL_LBC.C	Procedure : Rem_Dc()	Aplica el filtro paso alto
-------------------	----------------------	----------------------------

En este bloque se suprime la componente continua de la señal de entrada de voz  $s[n]$ . La función de transferencia del filtro es:

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

La salida de este filtro es:  $x[n]_{n=0..239}$ .



T1517840-95/d01

FIGURA 1/G.723.1

Diagrama de bloques del códec de voz – En cada bloque se indica el número de la subcláusula correspondiente

## 2.4 Análisis LPC

File : LPC.C	Procedure : Comp_Lpc()	Realiza el cálculo de los coeficientes LPC
File : LPC.C	Procedure : Durbin()	Recursión de Levinson-Durbin

El análisis LPC se realiza sobre la señal  $x[n]$  de la manera siguiente. Se aplica el análisis predictivo lineal (LP, *linear predictive*) de décimo orden. En cada subtrama, se centra una ventana de 180 muestras. A esas muestras se les aplica una ventana de Hamming. Se calculan 11 coeficientes de autocorrelación a partir de la señal ventanizada. Se aplica un factor de corrección de ruido blanco de (1025/1024) con la fórmula  $R[0] = R[0](1 + 1/1024)$ . Los otros 10 coeficientes de autocorrelación se multiplican por la tabla binomial de coeficientes de ventana. (Los valores de esta tabla y de todas las demás aparecen en el código C.) Se calculan los coeficientes de predicción lineal (LPC, *linear predictive coefficients*) mediante la fórmula de recursión de Levinson-Durbin. Para cada trama de entrada, se calculan cuatro conjuntos LPC, uno por subtrama. A continuación, se utilizan estos conjuntos LPC para construir el filtro de ponderación perceptual de corto plazo. El filtro de síntesis LPC se define como:

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

donde  $i$  es el índice de la subtrama y está comprendido entre 0 y 3.

## 2.5 Cuantificador LSP

File : LSP.C	Procedure : AtoLsp()	Convierte los coeficientes LPC a LSP
File : LSP.C	Procedure : LspQnt()	Cuantificación del vector LSP
File : LSP.C	Procedure : Lsp_Svq()	Cuantificación de los subvectores LSP

Primero, se efectúa una pequeña expansión adicional de la anchura de banda (7,5 Hz). Después, la señal resultante del filtro LP  $A_3(z)$  se cuantifica mediante un cuantificador vectorial de división predictiva. La cuantificación se realiza de la manera siguiente:

- 1) Los coeficientes LP,  $\{a_j\}_{j=1..10}$ , se convierten en coeficientes LSP,  $\{p'_j\}_{j=1..10}$ , buscando a lo largo del círculo unidad e interpolando para los cruzamientos con cero.
- 2) La componente continua de largo plazo,  $p_{DC}$ , se suprime de los coeficientes LSP,  $p'$ , y se obtiene un nuevo vector LSP sin componente continua,  $p$ .
- 3) Se aplica un predictor fijo de primer orden,  $b = (12/32)$  al vector LSP previamente decodificado,  $\bar{p}_{n-1}$ , para obtener el vector LSP predicho sin componente continua,  $\bar{p}_n$ , y el vector de error LSP residual,  $e_n$  en el instante (trama)  $n$ .

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

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



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

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

- 4) El vector LSP no cuantificado,  $p'_n$ , el vector LSP cuantificado,  $\tilde{p}_n$ , y el vector de error LSP residual,  $e_n$ , se dividen en 3 subvectores con dimensión 3, 3 y 4, respectivamente. Se ejecuta una cuantificación vectorial de cada subvector  $m$ -ésimo mediante una tabla de códigos de 8 bits. Se selecciona el índice,  $l$ , de la entrada de la tabla de códigos de subvector apropiada, que hace mínimo el criterio de error  $E_{l,m}$

$$p'_m = [p'_{1+3m} \ p'_{2+3m} \ \dots \ p'_{K_m+3m}], \quad K_m = \begin{cases} 3, & m=0 \\ 3, & m=1 \\ 4, & m=2 \end{cases} \quad (4.1)$$

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

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

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

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

donde  $e_{l,m}$  es la entrada  $l$ -ésima de la tabla de códigos LSP residual dividida  $m$ -ésima, y  $W_n$  es una matriz de ponderación en diagonal, determinada a partir del vector  $p'$  de coeficientes LSP no cuantificados, con ponderaciones definidas por:

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

- 5) Los índices seleccionados se transmiten al canal.

## 2.6 Decodificador LSP

File : LSP.C	Procedure : Lsp_Inq()	Invertir la cuantificación de LSP
--------------	-----------------------	-----------------------------------

La decodificación de los coeficientes LSP se realiza de la manera siguiente:

- 1) Primero, se decodifican los tres subvectores  $\{e_{m,n}\}_{m=0..2}$ , para formar un vector de décimo orden  $\tilde{e}_n$ .
- 2) El vector predicho  $\bar{p}_n$  se suma al vector decodificado,  $\tilde{e}_n$ , y al vector de componente continua,  $p_{DC}$ , para formar el vector LSP decodificado,  $\tilde{p}_n$ .

- 3) El vector LSP decodificado se somete a una prueba de estabilidad,  $\tilde{\mathbf{p}}_n$ , para asegurarse de que está ordenado conforme a la siguiente condición:

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

donde  $\Delta_{\text{mín}}$  es igual a 31,25 Hz. Si esta prueba de estabilidad (6) falla para  $\tilde{p}_i$  y  $\tilde{p}_{i+1}$ , entonces  $\tilde{p}_j$  y  $\tilde{p}_{j+1}$  se modifican de la manera siguiente:

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

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

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

Se realiza esta modificación hasta que se cumpla la condición (6). Si después de 10 iteraciones la condición de estabilidad no se cumple, se utiliza el vector LSP precedente.

## 2.7 Interpolación LSP

File : LSP.C	Procedure : Lsp_Int()	Interpolador LSP
File : LSP.C	Procedure : LsptoA()	Convierte los coeficientes LSP a LPC

Se realiza la interpolación lineal entre el vector LSP decodificado,  $\tilde{\mathbf{p}}_n$ , y el vector LSP precedente,  $\tilde{\mathbf{p}}_{n-1}$ , para cada subtrama. Cuatro vectores LSP interpolados,  $\{\tilde{\mathbf{p}}_i\}_{i=0,3}$ , se convierten a vectores LPC,  $\{\tilde{\mathbf{a}}_i\}_{i=0,3}$ .

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

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

Se utiliza el filtro de síntesis LPC cuantificado,  $\tilde{A}_i(z)$ , para generar la señal de voz decodificada y se define como:

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

## 2.8 Filtro de ponderación perceptual formante

File : LPC.C	Procedure : Wght_Lpc()	Calcula los coeficientes del filtro perceptual
File : LPC.C	Procedure : Error_Wght()	Aplica el filtro de ponderación perceptual

Para cada subtrama, se construye un filtro de ponderación perceptual formante utilizando los coeficientes LPC no cuantificados  $\{a_{ij}\}_{j=1,\dots,10}$ . El filtro tiene una función de transferencia:

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

donde  $\gamma_1 = 0,9$  y  $\gamma_2 = 0,5$ . A continuación, la trama de voz de entrada,  $\{x[n]\}_{n=0..239}$ , se divide en cuatro subtramas, y cada subtrama se filtra con el filtro  $W_i(z)$ , con lo que se obtiene la señal de voz de salida,  $\{f[n]\}_{n=0..239}$ .

## 2.9 Estimación del tono

File : EXC_LBC.C	Procedure : Estim_Pitch()	Estimación del tono en bucle abierto
------------------	---------------------------	--------------------------------------

Se hacen dos estimaciones del tono por trama, una para las dos primeras subtramas, y otra para las dos últimas. Se calcula la estimación del periodo del tono en bucle abierto,  $L_{OL}$ , mediante la señal vocal ponderada perceptualmente  $f[n]$ . Se emplea un criterio de correlación cruzada,  $C_{OL}(j)$ , como método de maximización para determinar el periodo del tono aplicando la expresión siguiente:

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

El índice  $j$  que hace máxima la correlación cruzada,  $C_{OL}(j)$ , se selecciona como la estimación del tono en bucle abierto para las dos subtramas apropiadas. En la búsqueda del mejor índice, se da una cierta preferencia a los periodos de tono más pequeños para evitar la elección de los múltiplos del tono fundamental. Se buscan los máximos de  $C_{OL}(j)$  empezando por  $j = 18$ . Para cada máximo  $C_{OL}(j)$  encontrado, se compara su valor con el mejor máximo anterior encontrado,  $C_{OL}(j')$ . Si la diferencia entre los índices  $j$  y  $j'$  es inferior a 18 y  $C_{OL}(j) > C_{OL}(j')$ , se selecciona el nuevo máximo. Si la diferencia entre los índices es mayor o igual a 18, el nuevo máximo se selecciona sólo si  $C_{OL}(j)$  es mayor que  $C_{OL}(j)$  en 1,25 dB.

## 2.10 Procesamiento de subtrama

A partir de este punto, se ejecutan todos los bloques de cálculo una vez por subtrama.

## 2.11 Conformación del ruido armónico

File : EXC_LBC.C	Procedure : Comp_Pw()	Calcula los coeficientes del filtro del ruido armónico
File : EXC_LBC.C	Procedure : Filt_Pw()	Aplica el filtro del ruido armónico

Para mejorar la calidad de la voz codificada, se construye un filtro de conformación del ruido armónico. El filtro es:

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

El retardo «lag» óptimo,  $L$ , para este filtro es el retardo que maximiza el criterio,  $C_{PW}(j)$ , mientras que para el numerador,  $N(j)$ , se consideran sólo valores de correlación positivos antes de elevar al cuadrado:

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

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

donde  $L_1 = L_{OL}-3$  y  $L_2 = L_{OL}+3$ . El valor máximo se definirá como ( $C_L$ ). La ganancia del filtro óptima, ( $G_{opt}$ ) es

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

$G_{opt}$ , está limitada a la gama [0,1]. La energía,  $E$ , de la señal de voz ponderada  $\{f[n]\}_{n=0..59}$  viene dada por

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

Entonces, el coeficiente  $\beta$  del filtro de conformación del ruido armónico,  $P(z)$ , viene dado por:

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

Después de calcular los coeficientes del filtro del ruido armónico, se filtra la voz perceptualmente ponderada,  $f[n]$ , con  $P(z)$ , para obtener el vector objetivo,  $w[n]$ .

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

## 2.12 Calculador de la respuesta de impulso

File : PC.C	Procedure : Comp_Ir()	Cálculo de la respuesta de impulso
-------------	-----------------------	------------------------------------

En el análisis de bucle cerrado, se utiliza el filtro combinado,  $S_i(z)$ , siguiente:

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

donde las componentes  $S_i(z)$  se definen en las fórmulas 10, 11 y 13. Se calcula la respuesta de impulso de este filtro, que se denominará  $\{h_i[n]\}_{n=0..59, i=0..3}$ .

### 2.13 Respuesta de entrada cero y substracción de timbre

File : LPC.C	Procedure : Sub_Ring ()	Ejecuta la substracción de timbre
--------------	-------------------------	-----------------------------------

La respuesta de entrada cero del filtro combinado,  $S_i(z)$ , se obtiene calculando la salida de ese filtro cuando la señal de entrada contiene muestras con todos los valores cero. La respuesta de entrada cero se denota  $\{z[n]\}_{n=0..59}$ . La substracción de timbre se efectúa restando la respuesta de entrada cero del vector de voz con ponderación de armónicos,  $\{w[n]\}_{n=0..59}$ . El vector resultante se define como  $t[n] = w[n] - z[n]$ .

### 2.14 Predictor del tono

File : EXC_LBC.C	Procedure : Find_Acbk()	Contribución de la tabla de códigos adaptativos. Llama a Get_Rez() y Decod_Acbk()
File : EXC_LBC.C	Procedure : Get_Rez()	Obtiene el residuo de la memoria intermedia de excitación
File : EXC_LBC.C	Procedure : Decod_Acbk()	Decodifica la contribución de la tabla de códigos adaptativos

La contribución de predicción del tono se trata como una contribución de tabla de códigos adaptativos convencional. El predictor de tono es de quinto orden, (véase la ecuación 41.2). Para las subtramas 0 y 2, se selecciona el retardo del tono en bucle cerrado a partir del retardo del tono en bucle abierto en la gama  $\pm 1$  y se codifica con 7 bits. (Nótese que el retardo del tono en bucle abierto no se transmite jamás.) Para las subtramas 1 y 3, el retardo del tono en bucle cerrado se codifica diferencialmente usando 2 bits y puede diferir del retardo de las subtramas anteriores sólo en  $-1, 0, +1$  o  $+2$ . Los valores del retardo del tono decodificados y cuantificados se denominarán  $L_i$  a partir de este punto. Las ganancias de predictor de tono se cuantifican vectorialmente mediante dos tablas de códigos con 85 ó 170 entradas para la velocidad alta y 170 entradas para la velocidad baja. La tabla de códigos de 170 entradas es la misma para ambas velocidades. En el caso de la velocidad alta, si  $L_0$  es inferior a 58 para las subtramas 0 y 1, o si  $L_2$  es inferior a 58 para las subtramas 2 y 3, se utiliza entonces la tabla de códigos de 85 entradas para la cuantificación de la ganancia del tono. En caso contrario, se cuantifica la ganancia del tono con la tabla de códigos de 170 entradas. La contribución del predictor de tono,  $\{p[n]\}_{n=0..59}$ , se resta del vector objetivo  $\{t[n]\}_{n=0..59}$ , para obtener la señal residual  $\{r[n]\}_{n=0..59}$ .

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

### 2.15 Excitación de velocidad alta (MP-MLQ)

File : EXC_LBC.C	Procedure : Find_Fcbk()	Contribución de tabla de códigos fijada
File : EXC_LBC.C	Procedure : Find_Best()	Cuantificación de la señal residual
File : EXC_LBC.C	Procedure : Gen_Trn()	Genera un tren de funciones Dirac
File : EXC_LBC.C	Procedure : Fcbk_Pack()	Codificación combinatoria de posiciones de impulso

Se transfiere la señal residual  $\{r[n]\}_{n=0..59}$ , como un nuevo vector objetivo al bloque MP-MLQ. Este bloque cuantifica este vector. El proceso de cuantificación consiste en aproximar el vector objetivo  $r[n]$  a  $r'[n]$ :

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

donde  $v[n]$  es la excitación para el filtro combinado  $S(z)$  con respuesta de impulso  $h[n]$  y definida como:

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

donde  $G$  es el factor ganancia,  $\delta[n]$  es una función Dirac,  $\{\alpha_k\}_{k=0..M-1}$  y  $\{m_k\}_{k=0..M-1}$  son los signos ( $\pm 1$ ) y las posiciones de las funciones Dirac, respectivamente, y  $M$  es el número de impulsos, siendo 6 para cada subtrama par y 5 para cada subtrama impar. Hay una limitación en las posiciones de impulso. Las posiciones pueden ser todas impares o todas pares. Esto se indicará mediante un bit de retícula. Entonces, el problema consiste en estimar los parámetros desconocidos,  $G$ ,  $\{\alpha_k\}_{k=0..M-1}$ , y  $\{m_k\}_{k=0..M-1}$ , que minimizan la media cuadrática de la señal de error  $err[n]$ :

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

Los procesos de cuantificación y estimación de los parámetros se basan en un método de análisis por síntesis. El parámetro  $G_{m\acute{a}x}$  se estima y cuantifica de la siguiente manera. Primero, se calcula la función de correlación cruzada  $d[j]$  entre la respuesta de impulso,  $h[n]$ , y el nuevo vector objetivo  $r[n]$ :

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

La ganancia estimada viene dada por:

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

A continuación, se cuantifica la ganancia estimada  $G_{m\acute{a}x}$  mediante un cuantificador logarítmico. Este cuantificador de ganancia escalar es común a ambas velocidades, y consiste en 24 pasos, de 3,2 dB cada uno. Alrededor de este valor cuantificado  $\tilde{G}_{m\acute{a}x}$ , se seleccionan otros valores de ganancia dentro de la gama  $[\tilde{G}_{m\acute{a}x} - 3,2, \tilde{G}_{m\acute{a}x} + 6,4]$ . Para cada uno de estos valores de ganancia, se optimizan secuencialmente los signos y las posiciones de los impulsos. Este procedimiento se repite para las retículas pares e impares. Por último, se elige la combinación de parámetros cuantificados que arroja la media cuadrática mínima de  $err[n]$ . Se transmite la combinación óptima de posiciones de impulso y ganancias. Se emplea la codificación combinatoria  $\binom{30}{M}$  para transmitir las posiciones de impulso. Además, usando el hecho de que el número de palabras de código de las tablas de códigos fijados no es una potencia de 2, se ahorran 3 bits adicionales combinando los 4 bits más significativos de los códigos combinatorios para las 4 subtramas y formando así un índice de 13 bits. El código C proporciona los detalles acerca de cómo empaquetar esta información.

Para mejorar la calidad de la voz con un periodo de tono corto, se utiliza el siguiente procedimiento adicional. Si  $L_0$  es menor que 58 para las subtramas 0 y 1 o si  $L_2$  es menor que 58 para las subtramas 2 y 3, se utiliza un tren de funciones Dirac con el periodo del índice de tono,  $L_0$  o  $L_2$ , para cada posición  $m_k$  en vez de una única función Dirac del procedimiento de

cuantificación descrito más arriba. La decisión de utilizar varias funciones Dirac o una sola para representar la señal residual se basa en el cálculo del error cuadrático medio. Se elige la configuración que arroja el error cuadrático medio más bajo y se transmiten sus índices de parámetro.

## 2.16 Excitación de velocidad baja (ACELP)

File : EXC_LBC.C	Procedure : search_T0()	Excitación de tono síncrona
File : EXC_LBC.C	Procedure : ACELP_LBC_code()	Calcula el vector innovador
File : EXC_LBC.C	Procedure : Cor_h()	Correlaciona la respuesta de impulso
File : EXC_LBC.C	Procedure : Cor_h_X()	Correlaciona el vector objetivo con la respuesta de impulso
File : EXC_LBC.C	Procedure : D4i64_LBC()	Búsqueda de tabla de códigos algebraicos
File : EXC_LBC.C	Procedure : G_code()	Calcula la ganancia del vector de innovación

Se utiliza una tabla de códigos algebraicos de 17 bits para la excitación de la tabla de códigos fijada  $v[n]$ . Cada vector de código fijado contiene, como máximo, cuatro impulsos no cero. Estos 4 impulsos pueden tomar los signos y posiciones que se ofrecen en el Cuadro 1:

CUADRO 1/G.723.1

**Tabla de códigos de excitación ACELP**

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

Las posiciones de todos los impulsos se pueden desplazar simultáneamente en una unidad (para ocupar posiciones impares), lo que necesita un bit extra. Nótese que la última posición de cada uno de los dos últimos impulsos cae fuera de la frontera de subtrama, lo que significa que el impulso no está presente.

Cada posición de impulso se codifica con 3 bits y cada signo de impulso se codifica en 1 bit. Esto arroja un total de 16 bits para los 4 impulsos. Además, se utiliza un bit extra para codificar el desplazamiento, lo que resulta en una tabla de códigos de 17 bits.

Se hace una búsqueda en la tabla de códigos para hacer mínimo el error cuadrático medio entre la señal de voz ponderada,  $r[n]$ , y la señal de voz de síntesis ponderada dada por:

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

donde  $\mathbf{r}$  es el vector objetivo que consiste en la señal de voz ponderada después de restarle la respuesta de entrada cero del filtro de síntesis ponderado y la contribución de tono,  $G$  es la ganancia de tabla de códigos;  $\mathbf{v}_{\xi}$  es la palabra de código algebraica en índice  $\xi$ ; y  $\mathbf{H}$  es una matriz convolucional de Toeplitz triangular inferior con diagonal  $h(0)$  y diagonales inferiores  $h(1), \dots, h(L-1)$ , siendo  $h(n)$  las respuestas de impulso del filtro de síntesis ponderado  $S_i(z)$ .

Puede demostrarse que la palabra de código óptima es una que maximiza el término:

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

donde  $d = H^T r$  es la correlación entre la señal de vector objetivo,  $r[n]$ , y la respuesta de impulso,  $h(n)$ , y  $\Phi = H^T H$  es la matriz de covarianza de la respuesta de impulso. El vector  $d$  y la matriz  $\Phi$  se calculan antes de la búsqueda de la tabla de códigos. Los elementos del vector  $d$  se calculan con:

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

y los elementos de la matriz simétrica  $\Phi(i, j)$  se calculan mediante

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

NOTA – Sólo se calculan los elementos realmente necesarios y se ha diseñado suficiente memoria de almacenamiento para que el procedimiento de búsqueda sea más rápido.

La estructura algebraica de la tabla de códigos posibilita procedimientos de búsqueda muy rápidos, pues el vector de excitación  $v_{\xi}$  contiene sólo 4 impulsos no cero. La búsqueda se realiza en 4 bucles anidados, que corresponden a cada posición de impulso, donde a cada bucle se añade la contribución de un nuevo impulso. La correlación de la ecuación (27) viene dada por:

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

donde  $m_k$  es la posición del impulso  $k$ -ésimo, y  $\alpha_k$  es su signo ( $\pm 1$ ). La energía de los vectores de código de posición de los impulsos pares de la ecuación (27) viene dada por:

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

Para los vectores de código de posición de los impulsos impares, la energía de la ecuación (27) se aproxima mediante la energía del vector de código de posición de impulso par equivalente obtenida desplazando los impulsos de posición impar a una muestra anterior en el tiempo. Para simplificar el procedimiento de búsqueda, se modifican las funciones  $d[j]$  y  $\Phi(m_1, m_2)$ . La simplificación se lleva a cabo como se indica a continuación (antes de la búsqueda de la tabla de códigos). Primero, se define la señal  $s[j]$ , y luego se construye la señal  $d'[j]$ .

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

y la señal  $d'$  viene dada por  $d'[j] = d[j]s[j]$ . Segundo, se modifica la matriz  $\Phi$  por la inclusión de la información de señal; es decir,  $\Phi'(i, j) = s[i]s[j]\Phi(i, j)$ . La correlación de la ecuación (30) viene dada ahora por:

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



y la energía de la ecuación (31) viene dada por:

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

Se utiliza una técnica de búsqueda enfocada a simplificar aún más el procedimiento de búsqueda. En esta técnica, se prueba un umbral precalculado antes de entrar el último bucle, y se entra el bucle sólo si este umbral es excedido. Se fija el número máximo de veces que el bucle puede ser entrado de manera que sólo se hace la búsqueda en un porcentaje bajo de la tabla de códigos. Se calcula el umbral sobre la base de la correlación  $C$ . La correlación absoluta máxima y la correlación promedio,  $máx_3$  y  $av_3$ , debidas a la contribución de los primeros tres impulsos se encuentran antes de la búsqueda de la tabla de códigos. El umbral viene dado por

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

Se entra el cuarto bucle sólo si la correlación absoluta (debida a tres impulsos) rebasa  $thr_3$ . Nótese que esto resulta en una búsqueda de complejidad variable. Para controlar aún más la búsqueda, el número de veces que se entra el último bucle (para las 4 subtramas) no se permite que exceda de 600. (El promedio para el caso más desfavorable por subtrama es de 150 veces. Esto puede contemplarse como la búsqueda de sólo  $150 \times 8$  entradas de la tabla de códigos ignorando la tara de los primeros tres bucles).

Una característica especial de la tabla de códigos es que, para los retardos de tono inferiores a 60, se añade al código una contribución de tono que depende del índice  $PGInd_i$  del vector ganancia del predictor de tono LTP. O sea que, después de determinar el código algebraico óptimo  $v[n]$ , se modifica por  $v[n] \leftarrow v[n] + b(PGInd_i)v[n - L_i - e(PGInd_i)]$ , se tabulan los valores  $\beta(PGInd_i)$  y  $\varepsilon(PGInd_i)$ , siendo  $L_i$  la parte entera del periodo de tono. Obsérvese que, antes de la búsqueda de la tabla de códigos, habrá que modificar la respuesta de impulso de manera similar si  $L_i < 60$ .

El último paso, después de obtener la secuencia  $v[n]$ , es cuantificar la ganancia  $G$ . La ganancia se cuantifica de la misma manera que el proceso de excitación de velocidad alta. Esto se efectúa explorando la tabla de cuantificación de ganancias y seleccionando el índice  $MGInd_i$ , que minimiza la siguiente expresión:  $|G - \tilde{G}_j|$ ,  $0 \leq j \leq 23$ .

## 2.17 Decodificador de excitación

File : EXC_LBC.C	Procedure : Fcbk_Unpk()	Decodificar la excitación de tabla de códigos fijos
------------------	-------------------------	---

La decodificación de los impulsos se realiza así:

- 1) Primero, obtener el índice de ganancia máxima  $G_{máx}$  mediante:

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

donde  $GSize = 24$  es el tamaño de la tabla  $\tilde{G}$  y  $PGInd_i$  se obtiene en 2.18.

- 2) Se decodifican las posiciones de los impulsos utilizando la decodificación combinatoria  $\binom{30}{M}$ , donde  $M$  es 6 ó 5 en el caso de la velocidad alta. Para la velocidad baja, se ejecuta una decodificación directa de los índices de posición.
- 3) La posición de retícula (par/impar) se deriva del bit de retícula.
- 4) Los signos de los impulsos se derivan de los bits de signo.
- 5) Para el códec de velocidad alta, se ejecuta la decodificación del bit del tren impulsos sólo si  $L_i < 58$ .
- 6) Entonces, se reconstruye el vector  $v[n]$  mediante los parámetros decodificados.
- 7) Por último, se suman la contribución de tono,  $u[n]$ , y las contribuciones de impulso,  $v[n]$ , para formar el vector excitación  $e[n]$ .

## 2.18 Decodificación de la información de tono

File : EXC_LBC.C	Procedure : Get_Rez()	Obtiene los residuos de la memoria intermedia de excitación
File : EXC_LBC.C	Procedure : Decod_Acbk()	Decodifica la contribución de tabla de códigos adaptativos

La decodificación de la información de tono se realiza como se describe a continuación:

- 1) Primero, se decodifica el retardo del predictor de tono para las subtramas pares:

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

- 2) Se decodifica el retardo del predictor de tono de las subtramas impares como sigue:

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

donde:  $\Delta_i \in \{-1,0,+1,+2\}$ .

- 3) El vector ganancia del predictor de tono en la subtrama  $i$ -ésima se deriva del índice de ganancia  $GInd_i$ . En el caso de la velocidad baja, este índice contiene la información sobre el vector ganancia del predictor de tono y el índice de la ganancia de la secuencia de impulsos. En este caso, el índice de ganancia de tono  $PGInd_i$  se deriva como sigue:

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

donde  $\lfloor x \rfloor$  indica el entero más grande  $\leq x$ . En caso de velocidad alta, si se cumple la condición  $L_i \geq 58$  y, este índice se deriva de la misma manera que se describe en (39). En los casos mencionados más arriba,  $PGInd_i$  es una tabla de códigos de vector de ganancia de 170 entradas. En los demás casos, este índice es un puntero de tabla de códigos de vector de ganancia de 85 entradas y contiene información adicional acerca del bit de trenes de impulsos. En este caso, el índice de ganancia de timbre se deriva como sigue:

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

El retardo del predictor de tono y el vector de ganancia se decodifican a partir de estos índices y se utilizan para la extracción de la contribución de tono  $u[n]$  que se describe a continuación. Primero, se define una señal  $e'[n]$  por

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

donde  $mod$  es la operación modulus. Entonces,

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

## 2.19 Actualización de la memoria

File : LPC.C	Procedure : Upd_Ring ( )	Actualización de la memoria
--------------	--------------------------	-----------------------------

La última tarea de la subtrama *i-ésima* antes de codificar la subtrama siguiente es actualizar las memorias del filtro de síntesis  $\tilde{A}_i(z)$ , el filtro de ponderación perceptual formante  $W_i(z)$ , y el filtro de conformación de ruido armónico  $P_i(z)$ . Para ello, se calcula la respuesta completa del filtro combinado  $S_i(z)$ , pasando la secuencia de excitación reconstruida a través de este filtro. Al final del filtrado de excitación, se guarda la memoria del filtro combinado, que se utilizará para calcular la respuesta de entrada nula durante la codificación del siguiente vector de voz.

## 2.20 Asignación de bits

File : UTIL_LBC.C	Procedure : Line_Pack ( )	Empaquetamiento de tren de bits
-------------------	---------------------------	---------------------------------

Esta subcláusula presenta las tablas de asignación de bits para las velocidades binarias alta y baja. Las diferencias principales entre las dos velocidades radican en la codificación de las posiciones y las amplitudes de los impulsos. Además, en el caso de la velocidad baja se utiliza siempre la tabla de códigos de 170 entradas para el vector de ganancia del predictor de largo plazo.

CUADRO 2/G.723.1

### Asignación de bits en el algoritmo de codificación de 6,3 kbit/s

Parámetros codificados	Subtrama 0	Subtrama 1	Subtrama 2	Subtrama 3	Total
Índices LPC					24
Retardos de tabla de códigos adaptativos	7	2	7	2	18
Todas las ganancias combinadas	12	12	12	12	48
Posiciones de impulso	20	18	20	18	73 (Nota)
Signos de impulso	6	5	6	5	22
Índice de retícula	1	1	1	1	4
<b>TOTAL:</b>					<b>189</b>
NOTA – Utilizando el hecho de que el número de palabras de código de la tabla de códigos fijados no es una potencia de 2, se ahorran 3 bits adicionales combinando los 4 MSB (bits más significativos) de cada índice de posición de impulso en una única palabra de 13 bits.					

CUADRO 3/G.723.1

### Asignación de bits en el algoritmo de codificación de 5,3 kbit/s

Parámetros codificados	Subtrama 0	Subtrama 1	Subtrama 2	Subtrama 3	Total
Índices LPC					24
Retardos de tabla de códigos adaptativos	7	2	7	2	18
Todas las ganancias combinadas	12	12	12	12	48
Posiciones de impulso	12	12	12	12	48
Signos de impulso	4	4	4	4	16
Índice de retícula	1	1	1	1	4
<b>TOTAL:</b>					<b>158</b>

CUADRO 4/G.723.1

**Lista de parámetros transmitidos**

Nombre	Parámetros transmitidos	N.º de bits en velocidad alta	N.º de bits en velocidad baja
LPC	Índice LSP VQ	24	24
ACL0	Retardo de tabla de códigos adaptativos	7	7
ACL1	Retardo de tabla de códigos adaptativos diferencial	2	2
ACL2	Retardo de tabla de códigos adaptativos	7	7
ACL3	Retardo de tabla de códigos adaptativos diferencial	2	2
GAIN0	Combinación de ganancias adaptativa y fija	12	12
GAIN1	Combinación de ganancias adaptativa y fija	12	12
GAIN2	Combinación de ganancias adaptativa y fija	12	12
GAIN3	Combinación de ganancias adaptativa y fija	12	12
POS0	Índice de posiciones de impulso	20 (Nota)	12
POS1	Índice de posiciones de impulso	18 (Nota)	12
POS2	Índice de posiciones de impulso	20 (Nota)	12
POS3	Índice de posiciones de impulso	18 (Nota)	12
PSIG0	Índice de signo de impulso	6	4
PSIG1	Índice de signo de impulso	5	4
PSIG2	Índice de signo de impulso	6	4
PSIG3	Índice de signo de impulso	5	4
GRID0	Índice de retícula	1	1
GRID1	Índice de retícula	1	1
GRID2	Índice de retícula	1	1
GRID3	Índice de retícula	1	1
NOTA – Los cuatro MSB (bits más significativos) de estas palabras de código se combinan para formar un índice de 13 bits: MSBPOS.			

**2.21 Inicialización del códec**

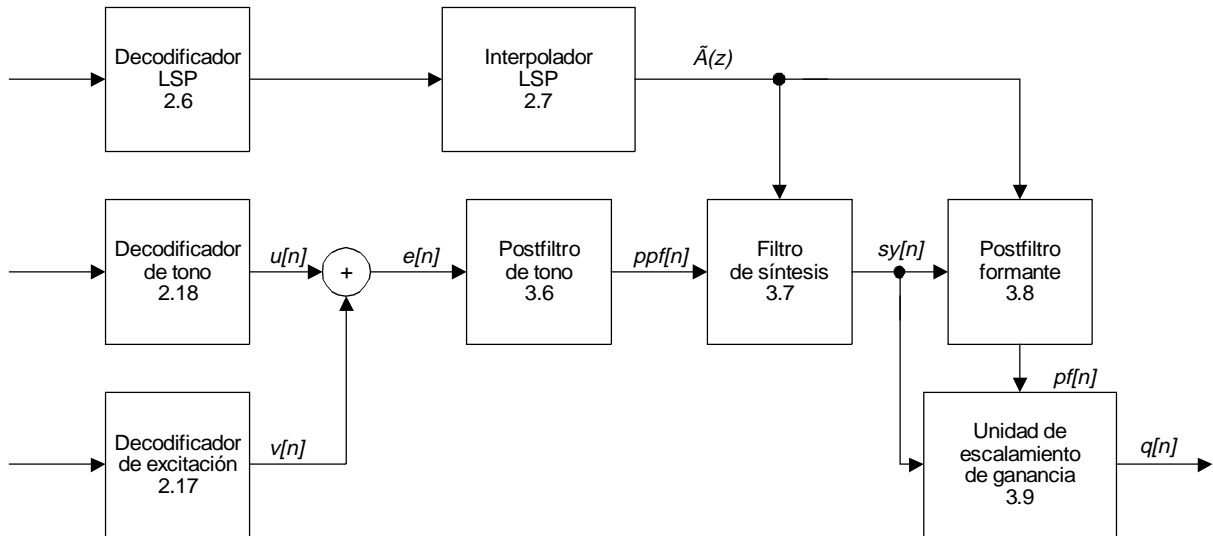
File : CODER.C	Procedure : Init_Coder()	Inicialización del códec
----------------	--------------------------	--------------------------

Todas las variables del códec estático se inicializarán con 0, a excepción del vector LSP precedente, que se inicializará con el vector DC LSP,  $p_{DC}$ .

### 3 Principios del decodificador

#### 3.1 Descripción general

La operación del decodificador también se ejecuta trama por trama. Primero, se decodifican los índices LPC cuantificados, después el decodificador construye el filtro de síntesis LPC. Para cada subtrama, se decodifican la excitación de tabla de códigos adaptativos y la excitación de tabla de códigos fijos y se entran en el filtro de síntesis. El postfiltro adaptativo consiste en un postfiltro de tono formante y un postfiltro de tono hacia adelante y atrás. La señal de excitación se inyecta en el postfiltro de tono que, a su vez, ataca al filtro de síntesis, cuya salida es la entrada al postfiltro formante. Una unidad de escalamiento de la ganancia mantiene la energía en el nivel de entrada del postfiltro formante.



T1517850-95/d02

FIGURA 2/G.723.1

**Diagrama de bloques del decodificador de voz – En cada bloque se indica el número de referencia correspondiente**

#### 3.2 Decodificador LSP

File : LSP.C	Procedure : Lsp_Inq()	Invertir la cuantificación de LSP
--------------	-----------------------	-----------------------------------

Este bloque es el mismo que el de 2.6.

#### 3.3 Interpolador LSP

File : LSP.C	Procedure : Lsp_Int()	Interpolador LSP
File : LSP.C	Procedure : LsptoA()	Convierte coeficientes LSP a coeficientes LPC

Este bloque es el mismo que el de 2.7.

### 3.4 Decodificación de la información de tono

File : EXC_LBC.C	Procedure : Get_Rez()	Obtiene residuales de la memoria intermedia de excitación
File : EXC_LBC.C	Procedure : Decod_Acbk()	Decodifica la contribución de tabla de códigos adaptativos

Este bloque es el mismo que el de 2.18.

### 3.5 Decodificador de excitación

File : EXC_LBC.C	Procedure : Fcbk_Unpk()	Decodifica la excitación de la tabla de códigos fijos
------------------	-------------------------	---

Este bloque es el mismo que el de 2.17.

### 3.6 Postfiltro de tono

File : EXC_LBC.C	Procedure : Comp_Lpf()	Calcula los parámetros del postfiltro de tono
File : EXC_LBC.C	Procedure : Find_F()	Maximización de la correlación cruzada hacia adelante
File : EXC_LBC.C	Procedure : Find_B()	Maximización de la correlación cruzada hacia atrás
File : EXC_LBC.C	Procedure : Get_Ind()	Cálculo de la ganancia
File : EXC_LBC.C	Procedure : Filt_Lpf()	Postfiltrado del tono

Se utiliza un postfiltro de timbre para mejorar la calidad de la señal sintetizada. Es importante observar que el postfiltrado de timbre se ejecuta para cada subtrama, y que para realizarlo, se necesita que toda la señal de excitación de trama  $\{e[n]\}_{n=0..239}$  sea generada y guardada. La mejora de la calidad se obtiene incrementando la relación señal/ruido en el múltiplo del periodo del tono fundamental. Esto se realiza de la manera siguiente. Se obtiene la señal postfiltrada  $\{ppf[n]\}_{n=0..59}$  de la señal de excitación decodificada  $\{e[n]\}_{n=0..59}$  como aparece en la expresión siguiente:

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

donde  $e[n]$  es la señal de excitación decodificada. El cálculo de las ganancias,  $g_p$ ,  $g_f$ ,  $g_b$ , y de los retardos,  $M_f$ ,  $M_b$ , se basa en un análisis de correlación cruzada hacia adelante y atrás. Los pesos  $w_f$ ,  $w_b$  pueden tener los siguientes valores: (0,0), (0,1), y (1,0). Los retardos se seleccionan maximizando las correlaciones cruzadas. La correlación cruzada del retardo de tono hacia adelante viene dada por:

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

y la correlación cruzada del retardo de tono hacia atrás viene dada por:

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

donde  $M_1 = L_i - 3$  y  $M_2 = L_i + 3$  y  $\{L_i\}_{i=0,2}$  son los retardos de tono recibidos para la primera y tercera subtramas.  $L_0$  se utiliza para las dos primeras subtramas y  $L_2$  para las dos últimas. Nótese que si una de las correlaciones máximas es negativa, o si para algún  $n \in [0..59]$  no hay ningún valor de muestra  $e[n + M_f]$  disponible, entonces el peso y el retardo correspondientes se fijan a 0. Esto da lugar a cuatro casos posibles: (0) ambas máximas son negativas y no hay que calcular las ponderaciones del postfiltro de tono, (1) sólo el máximo hacia adelante es positivo, por lo que se lo selecciona, (2) sólo el máximo hacia atrás es positivo, por lo que se lo selecciona, o (3) ambas máximas son positivas y la que aporta la contribución más grande se selecciona. Este procedimiento se describe a continuación. Para los casos (1), (2) y (3) se calculan las energías de señal correspondientes ( $T_{en}$ ,  $D_f$  y/o  $D_b$ ) para el retardo de tono óptimo ( $M_f$  o  $M_b$ ) según las ecuaciones (44.1), (44.2) y (44.3):

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

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

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

La energía hacia adelante viene dada por:

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

y la energía hacia atrás viene dada por:

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

$E$  se hace mínimo efectuando la maximización de  $\frac{C^2}{D}$ . En el caso (3), la elección entre la energía hacia adelante y hacia atrás se efectúa seleccionando la más grande de  $\frac{C_f^2}{D_f}$  y  $\frac{C_b^2}{D_b}$ . La ganancia de predicción es igual a  $-10 \log_{10} \left(1 - \frac{C^2}{DT_{en}}\right)$ . Si esta ganancia es menor que 1,25 dB, entonces se entiende que la contribución es despreciable y no se utiliza el postfiltro de tono. Cuando se utiliza el postfiltro de tono, la ganancia óptima viene dada por:

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

De acuerdo con la velocidad binaria del códec de voz, se multiplica la ganancia óptima por un factor de ponderación,  $\gamma_{ltp}$ , que es 0,1875 para la velocidad alta y 0,25 para la baja. Por último, se calcula la ganancia por escalamiento,  $g_p$ , como:

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

Si el denominador de la ecuación (47) es menor que el numerador, la ganancia se fija a 1.

### 3.7 Filtro de síntesis LPC

File : LPC.C	Procedure : Synt()	Sintetiza la voz reconstruida
--------------	--------------------	-------------------------------

El filtro de síntesis LPC de décimo orden  $\tilde{A}_i(z)$  se utiliza para sintetizar la señal vocal  $sy[n]$  procedente del residuo postfiltrado de tono decodificado  $ppf[n]$ .

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

### 3.8 Postfiltro formante

File : LPC.C	Procedure : Spf()	Postfiltrado formante
File : UTIL_LBC.C	Procedure : Comp_En()	Calcula la energía de la señal sintetizada

Se usa un postfiltro ARMA convencional. La función de transferencia del postfiltro de corto plazo viene dada por las ecuaciones siguientes:

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

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

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



donde  $\lambda_1 = 0,65$  y  $\lambda_2 = 0,75$ ,  $k$  es el primer coeficiente de autocorrelación que se calcula a partir de la voz sintetizada  $sy[n]$ , y  $k_{1old}$  es el valor de  $k_1$  de la subtrama precedente. Se obtiene la señal posfiltrada  $pf[n]$  como la salida del posfiltro formante con la señal de entrada  $sy[n]$ .

### 3.9 Unidad de escalamiento de ganancia

File : UTIL_LBC.C	Procedure : Scale()	Ajuste de la ganancia de la señal posfiltrada
-------------------	---------------------	---

Esta unidad recibe dos vectores de entrada, el vector de voz sintetizada  $\{sy[n]\}_{n=0..59}$  y el vector de salida posfiltrada  $\{pf[n]\}_{n=0..59}$ . Primero se calcula la relación de amplitudes  $g_s$  mediante:

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

Si el denominador es igual a 0,  $g_s$  se fija a 1.

A continuación, se obtiene el vector de salida  $q[n]$  normalizando la señal posfiltrada  $pf[n]$  y se actualiza la ganancia  $g[n]$  mediante las expresiones siguientes respectivamente:

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

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

donde  $\alpha$  es igual a 1/16.

### 3.10 Tratamiento de la interpolación de trama

File : EXC_LBC.C	Procedure : Comp_Info()	Calcula el índice de interpolación
File : EXC_LBC.C	Procedure : Regen()	Regeneración de la trama actual

Se ha diseñado este códec de forma que sea robusto para los borrados de trama indicados. En el decodificador se incluye una estrategia de ocultación de error para los borrados de trama. Ahora bien, esta estrategia debe ser activada mediante una indicación externa de que el tren de bits de la trama actual ha sido borrado. Puesto que el códec se ha diseñado para errores de ráfaga, no se suministra ningún mecanismo de corrección para los errores de bit aleatorios. Cuando se produce un borrado de trama, el decodificador pasa del modo de decodificación normal al modo de ocultación del borrado de trama. El procedimiento de interpolación de trama se efectúa independientemente para los coeficientes LSP y la señal residual.

#### 3.10.1 Interpolación LSP

La decodificación de los coeficientes LSP en el modo de interpolación de trama se ejecuta de la manera siguiente:

- 1) El vector  $\tilde{e}_n$  se fija a cero.
- 2) El vector predicho,  $\bar{p}_n$ , se suma al vector,  $\tilde{e}_n$ , y al vector DC,  $p_{DC}$ , para formar el vector LSP decodificado,  $\bar{p}_n$ . Para generar  $\bar{p}_n$  se utiliza un predictor fijo diferente:  $b_e = 23/32$ .

A partir de este momento, la decodificación de LSP continúa como en 2.6, salvo que  $\Delta_{\min} = 62,5$  Hz en vez de 31,25 Hz.

### 3.10.2 Interpolación residual

La interpolación residual se efectúa de dos maneras diferentes, dependiendo de la última trama buena precedente anterior a la trama borrada. Se comprueba la trama con un clasificador de tramas con y sin voz.

El clasificador se basa en una función de maximización de la correlación cruzada. Se efectúa una correlación cruzada de las últimas 120 muestras de la trama con  $L_2 \pm 3$ . El índice que alcanza el valor de correlación máximo se elige como índice de interpolación candidato. Después, se prueba la ganancia de predicción del mejor vector. Si esta ganancia es mayor que 0,58 dB, se declara que la trama tiene voz; en caso contrario, se declara que la trama no tiene voz.

El clasificador vuelve a 0 en el caso sin voz, y al valor de tono estimado en el caso con voz. Si se ha declarado que la trama no tiene voz, se guarda el promedio de los índices de ganancia de las subtramas 2 y 3.

Cuando la trama actual se señaló como borrada, y la trama precedente se clasificó como sin voz, se genera la excitación de trama actual mediante un generador de números aleatorios uniforme. Se normaliza la salida del generador de números aleatorios con el valor de ganancia calculado previamente.

En el caso con voz, la trama actual se regenera con excitación periódica de un periodo igual al valor proporcionado por el clasificador.

Si el estado de borrado de trama continúa durante las dos tramas siguientes, se atenúa el vector regenerado en 2,5 dB adicionales para cada trama. Después de 3 tramas interpoladas, se anula completamente la salida.

### 3.11 Inicialización del decodificador

File : DECOD.C	Procedure : Init_Decod()	Inicialización del decodificador
----------------	--------------------------	----------------------------------

Todas las variables del decodificador estático se inicializarán a 0, con la excepción de:

- 1) El vector LSP precedente, que se inicializará con el vector DC LSP,  $p_{DC}$ .
- 2) La ganancia postfiltro  $g[-1]$ , que se inicializará a 1.

## 4 Empaquetamiento del tren de bits

NOTA – Cada bit de los parámetros transmitidos se denomina  $PAR(x)_By$ : donde PAR es el nombre del parámetro y  $x$  indica el índice de subtrama si es pertinente, e  $y$  es la posición de bit comenzando por 0 (LSB) hasta el MSB. La expresión  $PARx\_By..PARx\_Bz$  representa la gama de bits transmitidos desde el bit  $y$  hasta el bit  $z$ . El bit no utilizado se denomina UB (valor = 0). RATEFLAG\_B0 indica si se utiliza la velocidad alta (0) o la velocidad baja (1) en la trama actual. VADFLAG\_B0 indica si la trama actual tiene voz (0) o no tiene voz (1). La combinación de RATEFLAG y VADFLAG, ambas puestas a 1, se reserva para uso ulterior. Los octetos se transmiten en el orden que aparecen en los Cuadros 5 y 6. Dentro de cada octeto, los bits aparecen con el MSB a la izquierda y el LSB a la derecha.

## 5 Código C de ANSI

El código C de ANSI que simula el codificador/decodificador (códec) de dos velocidades con una aritmética de coma fija de 16 bits puede solicitarse al UIT-T. El Cuadro 7 proporciona una lista de todos los ficheros incluidos en este código.

## 6 Glosario

Este glosario contiene los símbolos matemáticos utilizados en el texto y una breve descripción de lo que ellos representan. Véase el Cuadro 8.

CUADRO 5/G.723.1

**Empaquetamiento de los bits del octeto en el códec de alta velocidad binaria**

Velocidad alta

Octetos transmitidos	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

CUADRO 6/G.723.1

**Empaquetamientos de los bits del octeto en el códec de baja velocidad binaria**

Velocidad baja

Octetos transmitidos	PAR <sub>x</sub> B <sub>y</sub> , ...
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

CUADRO 7/G.723.1

**Lista de los nombres de ficheros del soporte lógico**

Nombre de fichero	Descripción
TYPDEF.H	La definición de los tipos de datos depende de la máquina
CST_LBC.H	Definición de las constantes para G.723
LBCCODEC.C	Programa principal para los códecs de voz G.723
LBCCODEC.H	Prototipos de funciones
CODER.C	Codificador de voz G.723 para las dos velocidades binarias
CODER.H	Prototipos de funciones
DECOD.C	Decodificador de voz G.723 para las dos velocidades binarias
DECOD.H	Prototipos de funciones
LPC.C	Análisis predictivo lineal
LPC.H	Prototipos de funciones
LSP.C	Funciones de pares espectrales de línea, cuantificador
LSP.H	Prototipos de funciones
EXC_LBC.C	Excitación adaptativa y fija (MP-MLQ, ACELP)
EXC_LBC.H	Prototipos de funciones
UTIL_LBC.C	Funciones varias (HPF, empaquetamiento, desempaquetamiento, I/O ...)
UTIL_LBC.H	Prototipos de funciones
TAB_LBC.C	Tablas de constantes
TAB_LBC.H	Declaración externa para tablas de constantes
BASOP.C	Aritmética de coma fija y operación lógica
BASOP.H	Prototipos de funciones

## Glosario de símbolos del texto

Símbolo	Descripción
$y[j]$	Muestras de la señal vocal de entrada
$s[j]$	Trama de 240 muestras de la señal vocal de entrada
$x[j]$	Trama de voz tratada con filtro paso alto
$R[j]$	Función de autocorrelación, $n = 0, 1, \dots, 10$
$a_i$	Vector de coeficientes LPC de la subtrama $i$
$\tilde{a}_i$	Vector de coeficientes LPC cuantificado de la subtrama $i$
$p'$	Vector LSP no cuantificado
$p$	Vector LSP eliminada la componente continua
$P_{DC}$	Vector de componente continua a largo plazo de valores LSP
$\tilde{p}_n$	Vector LSP decodificado para la trama $n$
$\bar{p}_n$	Vector LSP predicho eliminada la componente continua
$e_n$	Vector de error LSP residual para la trama $n$
$\tilde{e}_n$	Valor cuantificado de $e_n$
$W_n$	Matriz de ponderación diagonal para la cuantificación LSP
$\gamma_1, \gamma_2$	Pesos para el filtro de ponderación perceptual, 0,9, 0,5
$W_i$	Filtro de ponderación perceptual formante para la subtrama $i$
$f[n]$	Señal vocal ponderada perceptualmente formante
$L_{OL}$	Periodo de tono en bucle abierto estimado
$C_{OL}(j)$	Función criterio de correlación cruzada para la estimación del tono en bucle abierto
$C_{PW}(j)$	Función criterio de correlación cruzada para la estimación del tono con configuración de ruido armónico
$\beta$	Ganancia de filtro de conformación de ruido armónico
$L$	Retardo óptimo para el filtro de ruido armónico
$G_{opt}$	Ganancia óptima para el filtro de conformación de ruido armónico
$E$	Energía de la señal vocal ponderada
$P_i$	Filtro de conformación de ruido armónico para la subtrama $i$
$S_i$	Combinación de los filtros de síntesis, ponderación formante y ruido armónico para la subtrama $i$
$h[n]$	Respuesta de impulso del filtro combinado
$z[n]$	Respuesta de entrada cero del filtro combinado
$w[n]$	Señal vocal con ponderación de ruido armónico
$t[n]$	Vector objetivo
$p[n]$	Vector de contribución del predictor de tono
$r[n]$	Vector de señal residual
$r'[n]$	Vector de excitación filtrado
$v[n]$	Vector de excitación de tabla de códigos fijos
$M$	Número de impulsos

CUADRO 8/G.723.1 (fin)

Glosario de símbolos del texto

Símbolo	Descripción
$\alpha_k$	Signo del impulso k
$m_k$	Posición del impulso k
$d[n]$	Función de correlación cruzada de $h[n]$ y $r[n]$
$G_{m\acute{a}x}$	Ganancia estimada de los impulsos para velocidad alta
$L_i$	Retardo de tono para la subtrama $i$
$\mathbf{H}$	Matriz convolucional Toeplitz triangular inferior con diagonales $h[n]$
$\Phi$	Matriz de covarianza formada por $\mathbf{H}^T\mathbf{H}$
$m\acute{a}x_3$	Correlación máxima de los primeros 3 impulsos para velocidad baja
$av_3$	Correlación media de los primeros 3 impulsos para velocidad baja
$thr_3$	Umbral para la correlación de los primeros 3 impulsos para velocidad baja
$MGInd_i$	Índice de ganancia máxima de la subtrama $i$
$GInd_i$	Índice de ganancia de la subtrama $i$
$PInd_i$	Índice de tono de la subtrama $i$
$PGInd_i$	Índice de retardo de tono de la subtrama $i$
$GSize$	Tamaño de tabla de códigos de ganancia de excitación, 24
$\tilde{G}$	Ganancia cuantificada
$\beta_{ij}$	Vector de ganancia de predictor de tono
$u[n]$	Vector de excitación de tabla de códigos adaptativos
$e[n]$	Vector de excitación combinado decodificado
$ppf[n]$	Señal de excitación postfiltrada de tono
$M_f, M_b$	Retardos óptimos de postfiltro de tono hacia adelante y hacia atrás
$ppf'[n]$	Señal de excitación postfiltrada de tono no normalizada
$\gamma_{lp}$	Factor de ponderación de ganancia, 0,1875 ó 0,25
$g_p$	Ganancia con escalamiento de postfiltro de tono
$g_f, g_b$	Ganancias óptimas hacia adelante y hacia atrás para postfiltro de tono
$C_f, C_b$	Correlaciones cruzadas de excitación hacia adelante y hacia atrás
$D_f, D_b$	Energías de excitación hacia adelante y hacia atrás
$E_f, E_b$	Energías hacia adelante y hacia atrás para postfiltro de tono
$T_{en}$	Energía de señal de excitación
$sy[n]$	Señal vocal sintetizada LPC
$pf[n]$	Señal postfiltrada formante
$q[n]$	Señal vocal de salida
$g[n]$	Señal de ganancia de postfiltro formante
$k_1$	Coefficiente de reflexión interpolado para filtro de compensación de pendiente
$\lambda_1, \lambda_2$	Pesos para postfiltro formante, 0,65, 0,75
$g_s$	Relación de amplitud de los vectores $sy$ y $pf$
$\alpha$	0,0625 en unidad de escalamiento de ganancia
$b_e$	Predictor fijado para la interpolación LSP durante la ocultación de borrado de trama, 23/32