



UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

UIT-T

SECTOR DE NORMALIZACIÓN
DE LAS TELECOMUNICACIONES
DE LA UIT

G.728

Anexo I
(05/99)

SERIE G: SISTEMAS Y MEDIOS DE TRANSMISIÓN,
SISTEMAS Y REDES DIGITALES

Sistemas de transmisión digital – Equipos terminales –
Codificación de señales analógicas mediante métodos
diferentes de la MIC

Codificación de señales vocales a 16 kbit/s
utilizando predicción lineal con excitación por
código de bajo retardo

**Anexo I: Ocultamiento de la pérdida de tramas o de
paquetes para el decodificador LD-CELP**

Recomendación UIT-T G.728 – Anexo I

(Anteriormente Recomendación del CCITT)

RECOMENDACIONES UIT-T DE LA SERIE G
SISTEMAS Y MEDIOS DE TRANSMISIÓN, SISTEMAS Y REDES DIGITALES

CONEXIONES Y CIRCUITOS TELEFÓNICOS INTERNACIONALES	G.100–G.199
SISTEMAS INTERNACIONALES ANALÓGICOS DE PORTADORAS	
CARACTERÍSTICAS GENERALES COMUNES A TODOS LOS SISTEMAS ANALÓGICOS DE PORTADORAS	G.200–G.299
CARACTERÍSTICAS INDIVIDUALES DE LOS SISTEMAS TELEFÓNICOS INTERNACIONALES DE PORTADORAS EN LÍNEAS METÁLICAS	G.300–G.399
CARACTERÍSTICAS GENERALES DE LOS SISTEMAS TELEFÓNICOS INTERNACIONALES EN RADIOENLACES O POR SATÉLITE E INTERCONEXIÓN CON LOS SISTEMAS EN LÍNEAS METÁLICAS	G.400–G.449
COORDINACIÓN DE LA RADIOTELEFONÍA Y LA TELEFONÍA EN LÍNEA	G.450–G.499
EQUIPOS DE PRUEBAS	
CARACTERÍSTICAS DE LOS MEDIOS DE TRANSMISIÓN	G.600–G.699
SISTEMAS DE TRANSMISIÓN DIGITAL	
EQUIPOS TERMINALES	G.700–G.799
Generalidades	G.700–G.709
Codificación de señales analógicas mediante modulación por impulsos codificados (MIC)	G.710–G.719
Codificación de señales analógicas mediante métodos diferentes de la MIC	G.720–G.729
Características principales de los equipos múltiplex primarios	G.730–G.739
Características principales de los equipos múltiplex de segundo orden	G.740–G.749
Características principales de los equipos múltiplex de orden superior	G.750–G.759
Características principales de los transcodificadores y de los equipos de multiplicación de circuitos digitales	G.760–G.769
Características de operación, administración y mantenimiento de los equipos de transmisión	G.770–G.779
Características principales de los equipos múltiplex de la jerarquía digital síncrona	G.780–G.789
Otros equipos terminales	G.790–G.799
REDES DIGITALES	G.800–G.899
SECCIONES DIGITALES Y SISTEMAS DIGITALES DE LÍNEA	G.900–G.999

Para más información, véase la Lista de Recomendaciones del UIT-T.

RECOMENDACIÓN UIT-T G.728

CODIFICACIÓN DE SEÑALES VOCALES A 16 kbit/s UTILIZANDO PREDICCIÓN LINEAL CON EXCITACIÓN POR CÓDIGO DE BAJO RETARDO

ANEXO I

Ocultamiento de la pérdida de tramas o de paquetes para el decodificador LD-CELP

Resumen

Este anexo define una ampliación para el funcionamiento robusto del algoritmo G.728 a 9,6, 12,8 y 16 kbit/s en casos de borrado de trama o de pérdida de paquetes en entornos de comunicaciones móviles o de paquetes.

Orígenes

El anexo I a la Recomendación UIT-T G.728 ha sido preparado por la Comisión de Estudio 16 (1997-2000) del UIT-T y fue aprobado por el procedimiento de la Resolución N.º 1 de la CMNT el 27 de mayo de 1999.

PREFACIO

La UIT (Unión Internacional de Telecomunicaciones) es el organismo especializado de las Naciones Unidas en el campo de las telecomunicaciones. El UIT-T (Sector de Normalización de las Telecomunicaciones de la UIT) es un órgano permanente de la 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.

En ciertos sectores de la tecnología de la información que corresponden a la esfera de competencia del UIT-T, se preparan las normas necesarias en colaboración con la ISO y la CEI.

NOTA

En esta Recomendación, la expresión *empresa de explotación reconocida (EER)* designa a toda persona, compañía, empresa u organización gubernamental que explote un servicio de correspondencia pública. Los términos *Administración*, *EER* y *correspondencia pública* están definidos en la *Constitución de la UIT (Ginebra, 1992)*.

PROPIEDAD INTELECTUAL

La UIT señala a la atención la posibilidad de que la utilización o aplicación de la presente Recomendación suponga el empleo de un derecho de propiedad intelectual reivindicado. La UIT no adopta ninguna posición en cuanto a la demostración, validez o aplicabilidad de los derechos de propiedad intelectual reivindicados, ya sea por los miembros de la UIT o por terceros ajenos al proceso de elaboración de Recomendaciones.

En la fecha de aprobación de la presente Recomendación, la UIT ha recibido notificación de propiedad intelectual, protegida por patente, que puede ser necesaria para aplicar esta Recomendación. Sin embargo, debe señalarse a los usuarios que puede que esta información no se encuentre totalmente actualizada al respecto, por lo que se les insta encarecidamente a consultar la base de datos sobre patentes de la TSB.

© UIT 2000

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

Página

Anexo I – Ocultamiento de la pérdida de tramas o de paquetes para el decodificador		1
LD-CELP		1
I.1	Ámbito	1
I.2	Referencias normativas	1
I.3	Introducción	1
I.4	Principios de funcionamiento	2
I.4.1	Extrapolación de la excitación	2
I.4.2	Filtro LPC	3
I.4.3	Adaptación hacia atrás del LPC y de los predictores de ganancia	4
I.4.4	Postfiltro	5
I.4.5	Limitación del crecimiento de la ganancia después de un borrado de trama	6
I.5	Seudocódigo para el ocultamiento del borrado de trama	6
I.5.1	Nuevo bucle de programa principal para el decodificador	6
I.5.2	Bloque 31SF – Fijación de banderas y factor de escalado para el borrado de tramas	8
I.5.3	Bloque 31FE – Extrapolación de la señal de excitación	9
I.5.4	Bloque 31E – Actualización de la señal de excitación	11
I.5.5	Bloque 43FE – Módulo de ventanización híbrida	11
I.5.6	Bloque 47AF – Limitador de ganancia logarítmica después del borrado de trama	12
I.5.7	Bloque 49FE – Módulo de ventana híbrida para el filtro de síntesis	13
I.5.8	Bloque 51FE – Módulo de ampliación de la anchura de banda para borrado de tramas	16
I.5.9	Bloque 97FE – Actualización de GSTATE durante el borrado de tramas	17
I.5.10	Bloque 98AF – Limitador de ganancia logarítmica después del borrado de trama	18
I.6	Parámetros y variables adicionales del codificador	18
Apéndice I.I – Valores utilizados para el escalado de ETPAST durante el borrado de tramas		19

Recomendación G.728

CODIFICACIÓN DE SEÑALES VOCALES A 16 kbit/s UTILIZANDO PREDICCIÓN LINEAL CON EXCITACIÓN POR CÓDIGO DE BAJO RETARDO

ANEXO I

Ocultamiento de la pérdida de tramas o de paquetes para el decodificador LD-CELP

(Ginebra, 1999)

I.1 **Ámbito**

Este anexo se ocupa de la forma de enmascarar la pérdida de información del flujo binario como consecuencia de la pérdida de trama o de paquete en el canal de comunicación. En funcionamiento normal, el decodificador se comporta de forma idéntica con la Recomendación G.728 (parte principal de la Recomendación), con la Recomendación G.728 anexo G a 16 kbit/s o con la Recomendación G.728 anexo H a 12,8 o 9,6 kbit/s. Los cambios propuestos en este anexo sólo implican modificar el decodificador durante los intervalos de tiempo en los que el flujo binario no está disponible. Se presupone que de forma externa se informa al decodificador de la pérdida del flujo binario. Este anexo no es esencial para el funcionamiento normal conforme con la Recomendación G.728.

I.2 **Referencias normativas**

- Recomendación CCITT G.728 (1992), *Codificación de señales vocales a 16 kbit/s utilizando predicción lineal con excitación por código de bajo retardo.*
- Recomendación UIT-T G.728 anexo G (1994), *Especificación de coma fija a 16 kbit/s.*
- Recomendación UIT-T G.728 anexo H (1997), *Operaciones de la LD-CELP a velocidad binaria variable, especialmente para equipos digitales de multiplicación de circuitos a velocidades inferiores a 16 kbit/s.*

I.3 **Introducción**

Este anexo describe las modificaciones que debe sufrir el decodificador G.728 para que pueda manejar el borrado de tramas en el flujo binario recibido. Las modificaciones incluyen la extrapolación de la excitación, la ampliación de la anchura de banda para el filtro LPC, la continuación de parte de la adaptación hacia atrás utilizando la excitación extrapolada, la continuación del postfiltrado y del adaptador del postfiltro utilizando la excitación extrapolada y la limitación de la velocidad a la que aumenta la ganancia adaptable hacia atrás después de los borrados de trama. Dichos cambios del decodificador mantienen la compatibilidad del flujo binario con la Recomendación G.728, siendo despreciable la complejidad adicional que se introduce.

A fin de mantener la compatibilidad del flujo binario con la Recomendación G.728, las modificaciones sólo afectan al decodificador. Sólo tienen lugar durante el borrado de tramas y las primeras tramas "correctas" después del borrado de tramas. El sistema propuesto funciona exactamente igual a lo especificado en la Recomendación G.728 durante las tramas "correctas" restantes.

Este anexo supone que el lector está familiarizado con la Recomendación G.728. No se discute el algoritmo G.728 y sólo se describen las modificaciones al mismo. Para evitar confusiones, en lo sucesivo la palabra *trama* se utiliza para hacer referencia al tamaño del bloque asociado al borrado

de trama (típicamente un múltiplo de 2,5 ms). El término *ciclo de adaptación* se utiliza para hacer referencia al bloque de 2,5 ms preciso para la adaptación de los parámetros de filtro G.728.

En base a las aportaciones del UIT-R (anteriormente CCIR), mediante un canal de borrado de trama puede determinarse si una trama de bits recibida es totalmente correcta (sin errores) o errónea (la tasa de errores de la trama se aproxima al 50%). También se supone que se ha utilizado un código de detección de errores de modo que el receptor determina de forma fiable qué tramas son correctas y cuales erróneas, permitiendo que dicha información esté disponible para el decodificador de señales vocales G.728.

Este anexo se estructura como sigue. La subcláusula I.4, se titula "Principios de funcionamiento", contiene subcláusulas que describen los cambios en el funcionamiento del decodificador durante el ocultamiento del borrado de tramas. La subcláusula I.5, "Seudocódigo para el ocultamiento del borrado de trama" contiene todo el seudocódigo necesario para las modificaciones del decodificador. Para que la descripción sea completa, el seudocódigo se describe para las especificaciones de coma flotante y para las de coma fija. Por último, la subcláusula I.6 contiene cuadros con los parámetros y variables adicionales del codificador que se han añadido para las operaciones de ocultamiento del borrado de trama.

I.4 Principios de funcionamiento

Las modificaciones a la Recomendación G.728 para tramas borradas incluyen la extrapolación de la excitación y los coeficientes del filtro LPC, la continuación de algunas operaciones de adaptación hacia atrás para el LPC y predictores de ganancia y la limitación de la tasa de crecimiento de la ganancia después del borrado de trama. Todo ello se describe de forma separada en cada una de las subcláusulas siguientes.

I.4.1 Extrapolación de la excitación

En primer lugar, hágase referencia a la figura 3/G.728. No se extrapola la salida del código cifrado VQ (bloque 29) de excitación. En lugar de ello, se extrapola la salida de la unidad de escalado de ganancia (bloque 31), o *excitación de ganancia escalada*, que se identifica mediante el ordenamiento ET() en el seudocódigo de 5.14/G.728. En el decodificador original de G.728 sólo se mantienen en memoria 5 muestras de ET(). Para poder extrapolar ET(), es necesario mantener un ordenamiento adicional ETPAST() que almacena los vectores ET() actuales y anteriores.

A continuación, hágase referencia al seudocódigo de 5.14/G.728. En el decodificador G.728, el módulo de extracción de tono (bloque 82) proporciona un periodo de tono KP. El calculador de la derivación del predictor de tono (bloque 83) calcula el peso de la derivación óptima de un predictor de tono con una única derivación para la señal vocal decodificada. Dicho peso de derivación PTAP se utiliza como indicador grosero de si el ciclo de adaptación actual corresponde o no a voz. Si PTAP es mayor que PPFTH, siendo PPFTH = 0,6, se considera que el ciclo de adaptación actual corresponde a voz, aplicándose el postfiltro de largo plazo; en cualquier otro caso, se inhibe el postfiltro de largo plazo. Dado que en el decodificador G.728 original se dispone del periodo del tono y del indicador de vocalización, éstos pueden utilizarse directamente en la extrapolación de excitación sin tener que calcularlos de nuevo.

Al comienzo de la primera trama borrada de un borrado, se utiliza el valor del indicador de vocalización PTAP que se obtuvo en el último ciclo de adaptación de la última trama correcta a fin de determinar si la señal vocal corresponde a voz (vocalizada) o no (no vocalizada). Durante el borrado, se extrapola la excitación de forma diferente para tramas vocalizadas y no vocalizadas. Para realizar la extrapolación, se utiliza un umbral de vocalización, $VTH = PPFTH/1,4$, que es menor que PPFTH.

Si $PTAP > VTH$, la última trama se trata como si fuera vocalizada. La extrapolación de ET() se realiza repitiendo de forma periódica un versión reducida de las últimas muestras KP del ordenamiento ETPAST(). Este funcionamiento puede explicarse también como sigue: Considérese

una ventana rectangular de 5 muestras que tiene valores de muestra unitarios y que se encuentra ubicada en la posición que corresponde al vector actual de ETPAST(). Desplácese entonces dicha ventana rectangular hacia atrás en el tiempo KP muestras, extraíganse las 5 muestras correspondientes de ETPAST(), y redúzcanse multiplicándolas por un factor de escalado. El resultado es el vector ET() extrapolado buscado. Este vector extrapolado ET() se copia en la posición adecuada del ordenamiento ETPAST() para actualizar el vector actual ETPAST(). Este proceso se repite para todos los vectores de la trama borrada. Durante las tramas borradas, este funcionamiento produce una secuencia ET() con formas periódicas pero con magnitudes decrecientes. El periodo es KP muestras. Dicha extrapolación periódica del ordenamiento ETPAST() prosigue hasta que se recibe una trama correcta. El factor de escalado, FESCALE, es 0,8 ms durante los primeros 20 ms de borrado y disminuye 0,2 ms por cada 10 ms adicionales. Los factores de escalado se almacenan en VOICEDFEGAIN(). Si el borrado dura más de 50 ms, FESCALE se pone a 0 para evitar que en la señal vocal de salida se introduzcan perturbaciones largas, antinaturales y periódicas.

Si $PTAP \leq VTH$, el último ciclo de adaptación se considera "no vocalizado". En un sentido estricto, es cualquier cosa salvo "vocalizado", ya que podría incluir silencios, transiciones, etc. En este caso, no se realiza la extrapolación periódica debido a que no se desea introducir una periodicidad artificial que no se encuentra en la señal vocal original. La periodicidad se evita repitiendo de forma aleatoria 5 muestras consecutivas de las últimas 140 muestras de ETPAST(). Para cada vector de una trama errónea, se utiliza un generador de números aleatorios para generar un número entero aleatorio comprendido entre 5 y 140. Este número aleatorio determina cuantas muestras debe retroceder en el tiempo la ventana rectangular deslizante a fin de extraer las 5 muestras correspondientes de ETPAST() que corresponden a la ET() extrapolada. Por supuesto, si así se desea, dicha secuencia de números aleatorios puede precalcularse y almacenarse.

Para asegurar que el nivel de la señal generada durante un borrado de trama no vocalizada se parezca a su historia reciente, se hace corresponder el tamaño o magnitud promedio de cada vector de excitación extrapolado con la magnitud de la excitación promedio que se produjo durante los 5 ms (40 muestras) anteriores al borrado. Ello se realiza tal como se indica a continuación: Al comienzo de la primera trama borrada, y si se determina que la señal vocal es no vocalizada, se calcula la magnitud promedio de los últimos 8 vectores de ETPAST(). Sea AVMAG el valor promedio resultante. Cuando se extrapola cada vector de excitación se calcula, en primer lugar, la magnitud (*MAG, magnitude*) del vector [que está constituido por 5 muestras consecutivas seleccionadas de forma aleatoria de entre las últimas 140 muestras de ETPAST()]. A continuación, se divide AVMAG entre MAG para obtener un factor de escalado. Dicho factor de escalado se utiliza para multiplicar cada una de las cinco muestras del vector de excitación extrapolado actual. Esta operación garantiza que cada vector de excitación extrapolado y de ganancia escalada tenga un valor medio AVMAG. A fin de evitar perturbaciones causadas por borrados muy largos, AVGMAG se reduce un 20% por cada 10 ms de duración adicionales del borrado que exceda de 20 ms. Los factores de atenuación se almacenan en el ordenamiento UNVOICEDFEGAIN().

I.4.2 Filtro LPC

Durante las tramas borradas, se extrapola la excitación de ganancia escalada ET() tal como se ha descrito anteriormente, pero sigue siendo necesario un filtro de síntesis de LPC para producir la señal vocal de salida. Se utiliza una versión "suavizada" del último conjunto correcto de coeficientes LPC de la última trama correcta. Dicho "suavizado" se consigue mediante la ampliación de la anchura de banda, tal como se hace en el módulo de ampliación de la anchura de banda del bloque 51 (5.6/G.728), excepto en que el factor de ampliación de anchura de banda FAC (5.1/G.728) es 0,97 en lugar de $253/256 \approx 0,9883$. Durante la primera trama borrada que sigue a una trama correcta, la operación de ampliación de anchura de banda se realiza en el tercer vector del primer ciclo de adaptación, precisamente en el momento en que los coeficientes LPC deberían normalmente actualizarse, que no sería el caso para el borrado de trama.

Sea a_i el i -ésimo coeficiente LPC utilizado en el último ciclo de adaptación de la última trama correcta. En ese caso, el nuevo conjunto de coeficientes LPC suavizados se obtiene como $a'_i = (0,97)^i a_i$, $i = 1, 2, \dots, 50$. Este nuevo conjunto de $\{a'_i\}$ se utiliza durante los primeros 10 ms del borrado (si éste dura todo ese tiempo), incluso aunque se puedan producir ulteriores ciclos de adaptación en la misma trama. Si el borrado dura más de 10 ms, este conjunto de $\{a'_i\}$ sufre una ampliación de anchura de banda adicional por un factor de 0,97. En otras palabras, si el borrado dura más de 10 ms, los coeficientes LPC se actualizan de nuevo en el tercer vector del quinto ciclo de adaptación y el coeficiente LPC i -ésimo es $a''_i = (0,97)^i a'_i = (0,97)^{2i} a_i$. De nuevo, este conjunto de $\{a''_i\}$ se utiliza durante los siguientes 10 ms de borrado. Igualmente, si el borrado dura ($k * 10$ ms), los coeficientes LPC son versiones de anchura de banda ampliada del último conjunto correcto de coeficientes LPC en el que el factor de ampliación es $(0,97)^k$. Cuanto más lejos nos encontremos de la última trama correcta, mayor será la ampliación de anchura de banda que se aplique, continuando así hasta que se recibe la siguiente trama correcta. A partir de ese momento, el proceso comienza de nuevo con la siguiente trama errónea a partir de un factor de ampliación de anchura de banda de 0,97.

En la primera trama de un borrado se hace algo especial para conseguir los coeficientes LPC más recientes para la ampliación de la anchura de banda. Si la ampliación de la anchura de banda se realiza directamente sobre el ordenamiento A() que se produce en la última trama correcta, dicho ordenamiento A() corresponde a la función de autocorrelación calculada en base a la señal vocal sintetizada *antes* de la última trama correcta. Por lo tanto, ésta queda obsoleta durante la primera trama borrada. Para obtener una anchura de banda del ordenamiento A() de coeficientes del predictor LPC más reciente, se realizan la ventana híbrida y la recursión de Durbin para el análisis LPC de orden 50, tal como es habitual en la primera trama borrada. Se amplía entonces la anchura de banda del ordenamiento A() resultante en un factor 0,97. Los coeficientes LPC resultantes están más actualizados debido a que se basan en la señal vocal sintetizada disponible hasta el último vector de la última trama correcta. Si el borrado dura más de 10 ms, cesa el tratamiento especial y la ampliación de anchura de banda se basa en el último ordenamiento A() que reside en memoria, cuya anchura de banda había sido anteriormente ampliada tal como se ha descrito en el párrafo anterior.

Nótese que durante el borrado de tramas no es necesaria dicha ampliación de anchura de banda del predictor de ganancia. Dado que ya se ha extrapolado el ET() de excitación de ganancia escalada, dicho ET() extrapolado se inyecta directamente al filtro de síntesis LPC para producir la señal vocal sintetizada. Por lo tanto, durante las tramas borradas, no es necesario producir ganancia de excitación adaptada hacia atrás ni actualizar los coeficientes del predictor de ganancia.

1.4.3 Adaptación hacia atrás del LPC y de los predictores de ganancia

En el transcurso de tramas borradas, no se necesitan las salidas del adaptador de filtro de síntesis hacia atrás ni del adaptador de ganancia vectorial hacia atrás (bloques 33 y 30 de la figura 3/G.728). No obstante, es deseable continuar las operaciones vitales del proceso de adaptación hacia atrás en ambos bloques a fin de reducir las discontinuidades en los estados internos cuando se reciba la siguiente trama correcta. Por "operaciones vitales" se entiende aquellas partes de las operaciones de adaptación hacia atrás que afectan a la salida de los bloques 30 y 33 de tramas posteriores.

Hágase ahora referencia al bloque 30 (adaptador de ganancia vectorial hacia atrás), que es idéntico al bloque 20 (5.7/G.728 y figura 6/G.728). En el caso del bloque 30 (adaptador de ganancia vectorial hacia atrás), las operaciones vitales incluyen los bloques 67, 39, 40, 41, 42 (que conjuntamente calculan la ganancia logarítmica con supresión de desplazamiento) y partes del bloque 43 (módulo de ventana híbrida) y del bloque 46 (predictor lineal de ganancia logarítmica). Los bloques 67 y 39 a 42 utilizan el ET() extrapolado para calcular la ganancia logarítmica con supresión de desplazamiento, cuyo valor se utiliza para actualizar GTMP(), que constituye la entrada al bloque 43, y GSTATE(), que es la memoria filtro del predictor lineal de ganancia logarítmica. En el interior del bloque 43 (módulo de ventana híbrida), es necesario utilizar GTMP() para actualizar la memoria

intermedia SBLG() e igualmente es necesario actualizar REXPLG(), que es el componente recursivo de la función de autocorrelación. Las operaciones antes mencionadas mantienen los estados internos que afectarán a las futuras salidas del bloque 30. Durante el borrado de tramas, no es necesario realizar ninguna otra operación, tal como la recursión Levinson-Durbin (bloque 44), la ampliación de la anchura de banda (bloque 45), el calculador logarítmico inverso (bloque 48), etc., pues en todo caso se descartan los resultados de las mismas.

Nótese que dado que los coeficientes del predictor de ganancia GP() no se actualizan durante un borrado de trama, el primer vector después del borrado de trama utilizará el GP() anterior calculado en la última trama correcta previa al borrado de trama. Ello se debe a que el ordenamiento GP() no se actualiza hasta el segundo vector de un ciclo de adaptación G.728. No se pretende que el ordenamiento GP() esté más actualizado durante el primer vector ya que ello complicaría de forma innecesaria el control de flujo del algoritmo. Asimismo, el recorte de ganancia que se describe en I.4.5 siguiente protege contra una posible variación significativa de la ganancia durante este primer vector. Por lo tanto, no es probable que la utilización de un predictor de ganancia ligeramente obsoleto para un único vector produzca una variación perceptible de la calidad de la señal de voz.

A continuación, hágase ahora referencia al bloque 33 (adaptador del filtro de síntesis hacia atrás del decodificador de 5.14/G.728), que es idéntico al bloque 23 (5.6/G.728 y figura 5/G.728). Para el bloque 33, las operaciones vitales incluyen lo siguiente:

- 1) la actualización de la memoria intermedia SB() en el módulo de ventanización híbrida (bloque 49) utilizando la señal vocal sintetizada (que se obtiene pasando el ET() extrapolado a través de la versión de anchura de banda ampliada del último filtro LPC correcto); y
- 2) el cálculo de REXP() de forma normal utilizando SB().

Durante las tramas borradas no es preciso realizar la mayoría de las restantes partes del bloque 33. Ello se debe a que los coeficientes LPC no se actualizan utilizando los resultados de los bloques 50 y 51, sino mediante una versión de anchura de banda ampliada de los coeficientes LPC de la última trama correcta. No obstante, durante el postfiltro debe realizarse una parte adicional del bloque 49 y una pequeña parte del bloque 50. En I.4.4 se describe esto último.

Tal como se ha explicado, en el transcurso del borrado de tramas no es necesario realizar muchas de las operaciones de cálculo intensivo de los bloques 30 y 33. La capacidad computacional (o tiempo de proceso del DSP) que se ahorra puede utilizarse para realizar la extrapolación de excitación y la ampliación de anchura de banda del último filtro LPC correcto, actividades ambas que precisan poca capacidad computacional. Éste es el motivo por el que las modificaciones al decodificador G.728 no aumentan la complejidad computacional en su conjunto.

I.4.4 Postfiltro

En aras de la simplicidad del control de flujo y del programa, la operación del postfiltro (5.14/G.728) y del adaptador del postfiltro (bloques 34 y 35 de la figura 3/G.728) es la misma en tramas correctas y en tramas borradas. Ello tiene algunas implicaciones.

En primer lugar, durante las tramas borradas, el adaptador de postfiltro aún necesita los coeficientes del predictor LPC de décimo orden y el primer coeficiente de reflexión del adaptador de filtro de síntesis hacia atrás (bloque 33 de la figura 3/G.728). Por lo tanto, en el bloque 49 (módulo de ventanización híbrida, figura 5/G.728), es aún necesario calcular los valores de RTMP(1) a RTMP(11), no siendo, sin embargo, obligado calcular de RTMP(12) a RTMP(51). Igualmente, en el bloque 50 es necesario realizar la recursión de Durbin desde el orden 1 al orden 10, aunque no sea necesario continuar desde el orden 11 al orden 50. Nótese que durante las tramas borradas los coeficientes LPC de décimo orden y el primer coeficiente de reflexión se obtienen a partir de la señal vocal sintetizada que, en última instancia, está controlada por el ET() de excitación extrapolada.

En segundo lugar, durante las tramas borradas el postfiltro actualiza todos sus coeficientes en base a la señal vocal sintetizada de las tramas borradas y utiliza los coeficientes actualizados para filtrar dicha señal vocal sintetizada a fin de producir la señal vocal de salida final. No se pretende "congelar" los coeficientes del postfiltro o utilizar una versión de anchura de banda ampliada del mismo en la última trama correcta, tal como se hizo para el filtro LPC. Intencionadamente, se deja que el postfiltro "flote" con la voz sintetizada a partir de la excitación extrapolada. Ello garantiza que los picos y los valles espectrales de la respuesta en frecuencia del postfiltro concuerden en todo momento con los de la señal vocal sintetizada. Se impide así la potencial cancelación o de énfasis de los picos espectrales de la señal vocal sintetizada por efecto de la falta de concordancia entre éstos y los valles espectrales de la respuesta en frecuencia del postfiltro.

Si bien durante un borrado el postfiltro continua la actualización del periodo de tono KP y de la derivación del predictor de tono PTAP, mientras dura el borrado no se utilizan valores actualizados para modificar la decisión VOICED, vocalizada/no vocalizada, ni el periodo de tono almacenado FEDELAY. Estos valores se fijan una sola vez al comienzo del borrado y no se modifican durante el mismo.

I.4.5 Limitación del crecimiento de la ganancia después de un borrado de trama

Se ha observado que después de un borrado muy prolongado de trama (de hasta 100 ms), o cuando un borrado de trama es precedido por una región de la señal vocal de baja amplitud, seguida de una región de señal de gran amplitud, la señal vocal decodificada después del borrado de trama tiene ocasionalmente un pico notable ("pop") o sobreganancia. Ello se debe a que la memoria intermedia de la ganancia logarítmica SBG() de la ventana híbrida contiene niveles de ganancia muy bajos al final del borrado de trama y dichos niveles bajos de ganancia son seguidos de niveles de ganancia mucho más altos en la primeras tramas correctas que se reciben después del borrado de trama. De esta forma, el análisis LPC genera un predictor de ganancia que intenta adaptarse rápidamente al salto abrupto del nivel de ganancia. Por lo tanto, la ganancia predicha (es decir, adaptada hacia atrás), crece demasiado rápidamente y en demasía, provocando la mencionada sobreganancia y el "pop" resultante.

Este problema se evita limitando la velocidad de crecimiento de la ganancia adaptada hacia atrás durante la primeras tramas correctas recibidas después de un borrado de trama. Después de un borrado de trama, el crecimiento de la ganancia se limita a un máximo de FEGAINMAX = +2 dB por cada vector de 5 muestras. Si el aumento de la ganancia logarítmica predicha es superior a +2 dB/vector, ésta se recorta a un nivel 2 dB por encima de la última ganancia logarítmica. La duración de este "recorte de ganancia" depende de la duración del borrado de trama. Si el borrado de trama es de AFTERFEMAX = 16 (40 ms) o menos, la duración del recorte de ganancia es la del borrado de trama. Si el borrado de trama dura más de 40 ms, el recorte de ganancia se limita a los primeros 40 ms después del borrado. Las dos constantes FEGAINMAX y AFTERINMAX son parámetros ajustables del codificador.

I.5 Seudocódigo para el ocultamiento del borrado de trama

En la subcláusula siguiente se presenta elseudocódigo del programa principal del decodificador. Sólo se muestra la secuencia de ejecución de bloque y no se describe ningún detalle de bajo nivel de paso de parámetros. En las subcláusulas siguientes se presentan losseudocódigos para las nuevas rutinas.

I.5.1 Nuevo bucle de programa principal para el decodificador

La única modificación en el bucle del programa principal para el decodificador, tanto para coma fija como para coma flotante, radica exclusivamente en el adaptador de ganancia vectorial hacia atrás, que se ha modificado en la especificación de coma fija. Debido a que dichos cambios en el bucle principal son relativamente pequeños, en lo que sigue sólo se presenta una versión delseudocódigo. Las construcciones del preprocesador C `#ifdef`, `#else` y `#endif` sirven para determinar qué segmentos

pertencen a la especificación de coma fija y cuales pertenecen a la de coma flotante. El pseudocódigo entre un `#ifdef FIXPT` y el siguiente `#else` es para la definición de coma fija, mientras que el pseudocódigo entre dicho `#else` y el siguiente `#endif` es para la definición de coma flotante.

```

Initialize all decoder variables to their initial values.
ILLCOND=.FALSE.
ILLCONDG=.FALSE.
FERROR=.FALSE. | TRUE if current frame is erased
FESIZE=4 | Number of 2.5 msec adaptation cycles in a frame
          | erasure(FE)
          | e.g.: 1 for 2.5 msec FEs, 4 for 10 msec FEs
          | FEs may be any multiple of 2.5 msec. Set by user.
ICOUNT=4 | Vector counter, 1,2,3,4,1,2,..., .625 ms each
ADCOUNT=FESIZE | Adaptation cycle counter, 2.5 msec each
                | Counts from 1,2,...,FESIZE,1,2,...,FESIZE,...;
FECOUNT=0 | Number of consecutively erased 2.5 msec adaptation
          | cycles
AFTERFE=0 | Number of 2.5 msec adaptations cycles after a FE to
          | limit the gain. Counts down.
OGAINDB=-32 | Last predicted gain in dB

VEC_LOOP:
  If ICOUNT = 4, do the next 14 lines
    ICOUNT = 0 | reset vector counter
              | If the last adaptation cycle was not erased, and the
              | gain was clamped decrease the number of adaptation cycles
              | left to clamp the gain.
    If FERROR = .FALSE. and AFTERFE > 0, do AFTERFE = AFTERFE - 1
    | Check if the next frame is erased
    If ADCOUNT = FESIZE, do the next 7 lines
      ADCOUNT = 0 | reset adaptation cycle counter
      read FERROR | read in new ferror
      | At the first good frame after an erasure set AFTERFE
      | so the gain will be clamped for the next few good frames.
      If FERROR = .FALSE. and FECOUNT > 0, do the next 4 lines
        AFTERFE = AFTERFE + FECOUNT | add the length of the FE to AFTERFE
        If AFTERFE > AFTERFEMAX, do the next line
          AFTERFE = AFTERFEMAX | clamp gain for 40 msec max
          FECOUNT = 0 | reset FECOUNT
      If FERROR = .TRUE., do the next 2 lines
        FECOUNT = FECOUNT + 1 | Update FECOUNT
        | At the start of a FE, and every 10 msec thereafter,
        | call block31SF to initialize the FE flags and update the
        | excitation FE gains. e.g. when FECOUNT is equal to 1, 5, 9, 13, ...
        If (FECOUNT & 3) = 1, do block31SF
          ADCOUNT = ADCOUNT + 1 | update adaptation cycle counter
          ICOUNT = ICOUNT + 1 | update vector counter

    Get ICHAN of the current vector from the input buffer.
    Obtain the shape index IS and gain index IG from ICHAN
    | check whether to update LPC coeff.
    If ICOUNT = 3, do the next 2 lines
      If FERROR = .FALSE. and ILLCOND = .FALSE., do block 51.
      If FERROR = .TRUE. and (FECOUNT & 3) = 1, do block 51FE
    If ICOUNT = 2 and ILLCONDG=.FALSE. and FERROR = .FALSE., do block 45.

    do block 46 | GSTATE(1:9) shifted down 1 position

#ifdef FIXPT
  If FERROR = .FALSE., do the next 3 lines
    do blocks 98AF, 99 and 48 | get backward-adapted gain
    OGAINDB = output of block 98AF | for limiting gain growth after FE
    do blocks 29 and 31 | scale selected excitation codevector
#else
  If FERROR = .FALSE., do the next 3 lines
    do blocks 47AF and 48 | get backward-adapted gain
    OGAINDB = output of block 47AF | for limiting gain growth after FE
    do blocks 29 and 31 | scale selected excitation codevector

```

```

#endif
  If FERROR = .TRUE., do the next line
    do block 31FE | get extrapolated excitation

  do block 31E | update ETPAST()
  do block 32 | get ST()
  If ICOUNT = 1, do block 85 | update short-term postfilter coeff.
  do block 81 | 10th-order LPC inverse filtering
  If ICOUNT = 3, do the next 3 lines
    do block 82 | pitch period extraction
    do block 83 | compute pitch predictor tap
    do block 84 | update long-term postfilter coeff.
  do block 71 | long-term postfilter
  do block 72 | short-term postfilter
  do blocks 73 and 74 | calculate sums of absolute values
  do block 75 | ratio of sums of absolute values
  do block 76 | low-pass filter of scaling factor
  do block 77 | gain control of postfilter output

#ifdef FIXPT
  If FERROR = .FALSE., do the next 2 lines
    do blocks 93, 94, 96, and 97 | update log-gain
    GSTATE(1) = output of block 97 | update gain predictor memory
#else
  If FERROR = .FALSE., do the next 2 lines
    do blocks 39, 40 and 42 | update log-gain
    GSTATE(1) = output of block 42 | update gain predictor memory
#endif

  If FERROR = .TRUE., do the next 3 lines
    do block 97FE | compute log gain of extrapolated ET()
    OGAINDB = output of 97FE | save for limiting gain growth after FE
    GSTATE(1) = output of block 97FE | then update gain predictor memory

  I=(ICOUNT-1)*IDIM | I=starting address of STTMP()
  copy ST(1:5) to STTMP(I+1:I+5) | update STTMP()
  NLSSTTMP(ICOUNT)=NLSST | update NLSSTTMP()
  | end of once-per-vector processing
  | start once-per-cycle processing

  If ICOUNT = 4, do the next 6 lines
    do block 49FE | output ill-condition flag = ILLCOND
    do block 50, order 1 to 10 | output pred coef= ATMP() with NLSATMP
    | output ill-condition flag = ILLCOND
    NLSAPF=NLSATMP | save the 10th-order predictor for
    copy ATMP(2:11) to APF(2:11) | postfilter use later
    If FERROR = .FALSE., do the next line
      continue block 50, | continue to finish block 50
      order 11 to 50 | output pred coef= ATMP() with
      | NLSATMP output ill-condition flag = ILLCOND

  If ICOUNT = 1, do the next 7 lines
    GTMP(1)=GSTATE(4) | update GTMP() in one shot
    GTMP(2)=GSTATE(3)
    GTMP(3)=GSTATE(2)
    GTMP(4)=GSTATE(1)
    do block 43FE | output ill-condition flag = ILLCOND
    If FERROR = .FALSE., do the next line
      do block 44 | output pred. coeff. = GPTMP()
      | output ill-condition flag = ILLCOND
      | end of once-per-frame processing

Go to VEC_LOOP

```

1.5.2 Bloque 31SF – Fijación de banderas y factor de escalado para el borrado de tramas

Este bloque es nuevo. Se le llama al comienzo de un borrado de trama y a partir de entonces cada 10 ms durante un borrado. Al comienzo de un borrado fija la bandera vocalizado/no vocalizado en función del valor de PTAP. Si la última trama correcta era vocalizada, guarda el periodo de tono KP en FEDELAY para extrapolación periódica. Si era no vocalizada, calcula la suma de las magnitudes

de las últimas 40 muestras de ETPAST() de la última trama correcta y divide el resultado por ocho. Estos valores se calculan una sola vez al comienzo de un borrado y se utilizan durante todo el mismo. Si el borrado dura más de 10 ms, se llama de nuevo al bloque 31SF (cada 10 ms) y se atenúa el factor de escalado de la extrapolación de excitación, FESCALE, en función de la longitud del borrado y de si la trama era vocalizada o no vocalizada; poniendo FESCALE a "0" para detener las perturbaciones de "beeping" para voz vocalizada con duraciones de 60 ms o superiores. Si el borrado es no vocalizado, FESCALE se pone a 0 en 70 ms. En primer lugar se presenta el código de coma flotante:

```

VOICEDFEGAIN(0:4) = .8, .8, .6, .4, .2
UNVOICEDFEGAIN(0:5) = 1., 1., .8, .6, .4, .2

N10MSEC = FECOUNT >> 2
If N10MSEC == 0, do the following indented lines
  If PTAP > VTH, do the next 2 indented lines
    FEDELAY = KP
    VOICED = .TRUE.
  Otherwise, do the following indented lines
    VOICED = .FALSE.
    AVMAG = 0.
    For I = -39, -38, ..., 0, do the next line
      AVMAG = AVMAG + |ETPAST(I)|      | sum of ETPAST() magnitudes
      AVMAG = 0.125*AVMAG              | divide by 8
If VOICED = .TRUE., do the following indented lines
  If N10MSEC < 5, do FESCALE = VOICEDFEGAIN(N10MSEC)
  Otherwise, do FESCALE = 0.
Otherwise, do the following indented lines      | not voiced
  If N10MSEC < 6, do FESCALE = AVMAG * UNVOICEDFEGAIN(N10MSEC)
  Otherwise, do FESCALE = 0.

```

El código de coma flotante es el siguiente.

```

| Constants below are in Q15
VOICEDFEGAIN(0:4) = 26214, 26214, 19661, 13107, 6554
UNVOICEDFEGAIN(0:5) = 32767, 32767, 26214, 19661, 13107, 6554

N10MSEC = FECOUNT >> 2
If N10MSEC == 0, do the following indented lines
  If PTAP > VTH, do the next 2 indented lines
    FEDELAY = KP
    VOICED = .TRUE.
  Otherwise, do the following indented lines
    VOICED = .FALSE.
    AA0 = 0
    For I = -39, -38, ..., 0, do the next line
      AA0 = AA0 + |ETPAST(I)|      | sum of 40 ETPAST magnitudes
      Call VSCALE(AA0,1,1,30,AA0,NLS) | normalize numerator
      AVMAG = RND(AA0)
      NLSAVMAG = NLS - 16 + 3      | "+3" is for "divided by 8"
If VOICED = .TRUE., do the following indented lines
  If N10MSEC < 5, do FESCALE = VOICEDFEGAIN(N10MSEC)
  Otherwise, do FESCALE = 0
Otherwise, do the following indented lines      | not voiced
  If N10MSEC < 6, do the following indented lines
    FESCALE = UNVOICEDFEGAIN(N10MSEC) | scaling factor in Q15
    AA0 = FESCALE * AVMAG              | scale AVMAG by FESCALE
    Call VSCALE(AA0,1,1,30,AA0,NLSFESCALE) | normalize FESCALE
    FESCALE = RND(AA0)                 | save result in FESCALE
    NLSFESCALE = NLSFESCALE + NLSAVMAG - 1 | NLS for FESCALE
                                          | -1 is from (Q15 - 16)
  Otherwise, do FESCALE = 0

```

1.5.3 Bloque 31FE – Extrapolación de la señal de excitación

Este bloque es nuevo y sólo se utiliza durante tramas borradas. Extrapola la señal de excitación de dos formas posibles dependiendo de si la última trama correcta se consideró vocalizada o no. Para una señal de voz vocalizada, la señal se extrapola periódicamente con un valor decreciente. El factor

de escalado FESCALE comienza en 0,8 y disminuye cada 10 ms durante el borrado en el bloque 31SF.

En cualquier otro caso (no vocalizado, silencio, etc.) se utiliza un segmento aleatorio a partir de las 140 muestras anteriores. La magnitud de este segmento aleatorio se modifica de manera que tenga la misma magnitud promedio que las 40 muestras anteriores. Para calcular las magnitudes medias del segmento aleatorio de 5 muestras de las anteriores 40 muestras, debe dividirse la suma de las magnitudes por 5 y 40 respectivamente. No obstante, sólo es necesario disponer de la relación entre las dos magnitudes medias. Por lo tanto, en la práctica se calcula MAG como la suma de las magnitudes del segmento aleatorio de 5 muestras y AVMAG como la suma de las magnitudes de las 40 muestras anteriores dividido por 8. La relación entre los dos números resultantes es la relación correcta. El valor de AVMAG se ha calculado en el bloque 31SF durante el primer ciclo de adaptación de la primera trama borrada. AVMAG se atenúa en el bloque 31SF en base a la duración del borrado y se ubica en FESCALE. Nótese que debido a que la interoperabilidad con el codificador G.728 no es relevante durante el borrado de trama, no es necesario disponer de una secuencia predefinida de números aleatorios. Asimismo, los números no tienen porqué ser realmente aleatorios. Puede utilizarse cualquier secuencia de números enteros comprendidos entre 5 y 140, en tanto que dicha secuencia no sea periódica o cuasiperiódica. Por lo tanto, en lo que sigue no se incluye ningún seudocódigo para el generador de números aleatorios.

En primer lugar se presenta el código de coma flotante.

```

                                | Voiced case
If VOICED = .TRUE., do the next 2 indented lines
  For I = 1, 2, ..., IDIM, do the next line
    ET(I) = FESCALE * ETPAST(I-FEDELAY) | KP is pitch period
                                        | from 82

                                | Unvoiced case
If VOICED = .FALSE., do the following indented lines
  set FEDELAY = a random number between 5 and 140
  MAG = 0.
  For I = 1, 2, ..., IDIM, do the next 2 lines
    ET(I) = ETPAST(I-FEDELAY)
    MAG = MAG + |ET(I)|                | sum of ET() magnitudes
  If MAG = 0 do MAG = 1.                | avoid divide by zero
  RATIO = FESCALE / MAG

  For I = 1, 2, ..., IDIM, do the next line
    ET(I) = RATIO*ET(I)                | scale to have same magnitude

```

En el seudocódigo de coma fija, ET y ETPAST utilizan representaciones diferentes. Tal como se especifica en el anexo G/G.728, ET se representa en coma flotante de bloque de 15 bits con un cero inicial. Por otro lado, para simplificar el funcionamiento de este bloque, se utiliza el formato Q2 fijo a fin de representar el ordenamiento ETPAST largo. No tiene sentido utilizar coma flotante de bloque para ETPAST debido a que incluso el error numérico de la representación Q2 fija, es mucho menor que la diferencia entre la excitación extrapolada y la verdadera excitación en el codificador. En el caso extremadamente infrecuente de que ocurra un desbordamiento de ETPAST, el valor de ETPAST se recorta al nivel de saturación de +32767 ó -32768 para evitar el "reciclaje". El seudocódigo siguiente escribe en primer lugar el vector de excitación extrapolado (en Q2) en ETPAST(1) a ETPAST(5), convirtiendo entonces las 5 muestras de ETPAST extrapoladas en el ordenamiento ET de coma flotante de bloque de 15 bits.

```

If VOICED = .TRUE., do the next 4 lines      | Voiced case
  TEMP = FESCALE
  NLSTEMP = 15                               | FESCALE is Q15
  For I = 1, 2, ..., IDIM, do the next line
    ETPAST(I) = ETPAST(I-FEDELAY) | FEDELAY is pitch period

If VOICED = .FALSE., do the next 13 lines   | Unvoiced case
  set FEDELAY = a random number between 5 and 140

```



```

AA1 = 0
For I = 1, 2, ..., IDIM, do the next 2 lines
    ETPAST(I) = ETPAST(I-FEDELAY)
    AA1 = AA1 + |ETPAST(I)|           | sum of ETPAST(1:5) magnitudes
If AA1 = 0 or FESCALE = 0, do the next 2 lines
    TEMP = 0
    NLSTEMP = 15
Otherwise, do the next 4 lines
    Call VSCALE(AA1,1,1,30,AA1,NLSDEN) | normalize
                                       | denominator
    DEN = RND(AA1)
    NLSDEN = NLSDEN - 16
    Call DIVIDE(FESCALE,NLSFESCALE,DEN,NLSDEN,TEMP,NLSTEMP)

For I = 1, 2, ..., IDIM, do the next 5 lines
    AA0 = TEMP * ETPAST(I)           | AA0 in Q(2+NLSTEMP) format
    AA0 = AA0 >> NLSTEMP             | shift AA0 to make Q2
    if AA0 > 32767, set AA0 = 32767  | clip if necessary
    if AA0 < -32768, set AA0 = -32768
    ETPAST(I) = AA0                 | scaled ETPAST in Q2
Call VSCALE(ETPAST,IDIM,IDIM,13,ET,NLS) | convert to ET in 15-bit
                                       | BFL
NLSET = NLS + 2                     | update NLSET

```

I.5.4 Bloque 31E – Actualización de la señal de excitación

Este bloque actualiza la memoria intermedia de la señal de excitación. Debe completarse con independencia de si en ese momento se está ocultando un borrado de trama, de tal forma que la memoria intermedia siempre esté disponible cuando se necesite. En primer lugar se presenta el código de coma flotante.

```

For I = -KPMAX+1, -KPMAX+2, ..., -IDIM, do the next line
    ETPAST(I) = ETPAST(I+IDIM)       | shift ETPAST by IDIM samples
For I = 1, 2, ..., IDIM, do the next line
    ETPAST(I-IDIM) = ET(I)          | copy ET to ETPAST

```

En el seudocódigo de coma fija, ETPAST se almacena en formato Q2 y ET se almacena en coma flotante de bloque de 15 bits con NLSET de formato Q. Por lo tanto, es preciso una conversión de formato cuando se copia ET a ETPAST. El código de coma fija es el siguiente.

```

For I = -KPMAX+1, -KPMAX+2, ..., -IDIM, do the next line
    ETPAST(I) = ETPAST(I+IDIM)       | shift Q2 ETPAST by 5 samples
NRS = NLSET - 2                       | # of right shifts to make Q2
For I = 1, 2, ..., IDIM, do the next 6 lines
    AA0 = ET(I)                       | get 15-bit mantissa of ET
    if NRS > 0, AA0 >> NRS             | perform the shifting
    if NRS < 0, AA0 << -NRS
    if AA0 > 32767, set AA0 = 32767    | clip if necessary
    if AA0 < -32768, set AA0 = -32768
    ETPAST(I-IDIM) = AA0              | ETPAST is now Q2

```

I.5.5 Bloque 43FE – Módulo de ventanización híbrida

En esta subcláusula se presentan los seudocódigos de coma flotante y de coma fija del bloque 43FE. Este código se ha modificado para el caso de ocultamiento de error de trama y sustituye al bloque 43 original descrito en 5.7/G.728. La única modificación en el código de coma flotante sólo implica una línea adicional de seudocódigo que omite el cálculo de la función de autocorrelación final en caso de borrado de trama. En primer lugar se presenta el código de coma flotante.

```

N1=LPCLG+NUPDATE                       | compute some constants (can be
N2=LPCLG+NONRLG                         | precomputed and stored in memory)
N3=LPCLG+NUPDATE+NONRLG

For N=1,2,...,N2, do the next line
    SBLG(N)=SBLG(N+NUPDATE)             | shift the old signal buffer;

```

```

For N=1,2,...,NUPDATE, do the next line
  SBLG(N2+N)=GTMP(N)      | shift in the new signal;
                          | SBLG(N3) is the newest sample
K=1
For N=N3,N3-1,...,3,2,1, do the next 2 lines
  WS(N)=SBLG(N)*WNRLG(K)  | multiply the window function
  K=K+1

For I=1,2,...,LPCLG+1, do the next 4 lines
  TMP=0.
  For N=LPCLG+1,LPCLG+2,...,N1, do the next line
    TMP=TMP+WS(N)*WS(N+1-I)
  REXPLG(I)=(3/4)*REXPLG(I)+TMP  | update the recursive
                                  | component

If FERROR = .TRUE., skip all the lines below
  For I=1,2,...,LPCLG+1, do the next 3 lines
    R(I)=REXPLG(I)
    For N=N1+1,N1+2,...,N3, do the next line
      R(I)=R(I)+WS(N)*WS(N+1-I) | add the non-recursive
                                  | component

  R(1)=R(1)*WNCF          | white noise correction

```

Se presenta a continuación la versión de coma fija del mismo módulo. La única modificación en el código de coma fija tiene lugar en el módulo HWMCOREFE que sustituye a HWMCORE. Este seudocódigo de coma fija sustituye al código de coma fija original del bloque 43 de G.3.14/G.728. La subrutina HWMCOREFE puede encontrarse en el bloque 49FE en I.5.7.

```

N1=LPCLG+NUPDATE (=10+4)      | compute some constants (can be
N2=LPCLG+NONRLG (=10+20)     | precomputed and stored in memory)
N3=LPCLG+NUPDATE+NONRLG (=10+4+20)

For N=1,2,...,N2, do the next line
  SBLG(N)=SBLG(N+NUPDATE)     | shift the old signal buffer
For N=1,2,...,NUPDATE, do the next line
  SBLG(N2+N)=GTMP(N)         | SBLG(N3) is the newest sample
                              | all SBLG are Q9 and represented
                              | in 16 bits precision
Call FINDNLS(SBLG,N3,N3,14,NLS) | find the amount of left shifts
NLSTMP=NLS-1                 | needed in the next loop for 2
                              | bits of headroom later

K=1
For N=34,33,...,1, do the next 5 lines
  P=SBLG(N)*WNRLG(K)         | WNRLG is Q15
  If NLSTMP = -1, set AA0=P >> 1
  If NLSTMP > -1, set AA0=P << NLSTMP
  WS(N)=RND(AA0)             | WS(N) is 14 bits or less
  K=K+1

NLSATTLG = 14
Call HWMCOREFE(LPCLG,N1,N3,NLSATTLG,WS,NLSTMP,REXPLG,NLSREXPLG,R,ILLCOND)

```

I.5.6 Bloque 47AF – Limitador de ganancia logarítmica después del borrado de trama

Este nuevo bloque debe reemplazar el bloque 47 de 5.7/G.728. A continuación se presenta el seudocódigo de coma flotante.

```

If GAIN < 0., set GAIN = 0.   | Correspond to linear gain 1.
If GAIN > 60., set GAIN = 60. | Correspond to linear gain 1000.
TMP = GAIN - OGAINDB
If AFTERFE > 0 and TMP > FEGAINMAX, do the next line
  GAIN = OGAINDB + FEGAINMAX

```

I.5.7 Bloque 49FE – Módulo de ventana híbrida para el filtro de síntesis

Este es el bloque del módulo de ventana híbrida para el filtro de síntesis modificado en el caso de borrado de tramas. Este bloque sustituye al bloque 49 descrito en 5.6/G.728. En primer lugar se presenta el pseudocódigo de coma flotante del módulo de ventanización híbrida.

```

N1=LPC+NFRSZ           | compute some constants (can be
N2=LPC+NONR           | precomputed and stored in memory)
N3=LPC+NFRSZ+NONR

For N=1,2,...,N2, do the next line
  SB(N)=SB(N+NFRSZ)    | shift the old signal buffer;
For N=1,2,...,NFRSZ, do the next line
  SB(N2+N)=STTMP(N)   | shift in the new signal;
                        | SB(N3) is the newest sample

K=1
For N=N3,N3-1,...,3,2,1, do the next 2 lines
  WS(N)=SB(N)*WNR(K)  | multiply the window function
  K=K+1

For I=1,2,...,LPC+1, do the next 4 lines
  TMP=0.
  For N=LPC+1,LPC+2,...,N1, do the next line
    TMP=TMP+WS(N)*WS(N+1-I)
  REXP(I)=(3/4)*REXP(I)+TMP | update the recursive component

If FERROR = .TRUE., do the next 4 lines
  For I=1,2,...,11, do the next 3 lines
    RTMP(I)=REXP(I)
    For N=N1+1,N1+2,...,N3, do the next line
      RTMP(I)=RTMP(I)+WS(N)*WS(N+1-I) | add non-recursive
                                      | component

If FERROR = .FALSE., do the next 4 lines
  For I=1,2,...,LPC+1, do the next 3 lines
    RTMP(I)=REXP(I)
    For N=N1+1,N1+2,...,N3, do the next line
      RTMP(I)=RTMP(I)+WS(N)*WS(N+1-I) | add non-recursive
                                      | component

RTMP(1)=RTMP(1)*WNCF | white noise correction

```

El pseudocódigo de coma fija del módulo de ventanización híbrida (bloque 49FE) se modifica de forma similar. En esta ocasión, el cambio se encuentra en HWMCOREFE. Este pseudocódigo de coma fija reemplaza al código de coma fija original del bloque 49 de G.3.17/G.728.

```

N1=LPC+NFRSZ(=70)     | compute some constants (can be
N2=LPC+NONR(=85)     | precomputed and stored in memory)
N3=LPC+NFRSZ+NONR(=105)
N4=N3/IDIM(=21)
N5=NFRSZ/IDIM(=4)
N6=N4-N5(=17)

For N=1,2,...,N2, do the next line
  SB(N)=SB(N+NFRSZ)   | shift old part of buffer SB
For N=1,2,...,N6, do the next line
  NLSSB(N)=NLSSB(N+N5) | shift old NLSSB
For N=1,2,...,NFRSZ, do the next line
  SB(N2+N)=STTMP(N)  | shift in new part of SB
For N=1,2,...,N5, do the next line
  NLSSB(N6+N)=NLSSB(N) | shift in new NLSSB

| Now find the minimum NLSSB - this determines NLSTMP
NLSTMP=Min{NLSSB(1),NLSSB(2),...,NLSSB(N4)}

K=1                    | now multiply SB by the
N=N3                   | hybrid window function
For J=1,2,...,N4, do the next 8 lines
  NRSH=NLSSB(J)-NLSTMP-1 | -1 to compensate for Q15 mult

```

```

For M=1,2,...,IDIM, do the next 5 lines
  P=SB(K)*WNR(N)      | WNR is Q15 multiplication
  If NRSH = -1, set AA0=P << 1
  If NRSH > -1, set AA0=P >> NRSH
  WS(K)=RND(AA0)      | round upper word & store in WS
  N=N-1
K=K+1

```

NLSATT50 = 14

Call HWMCOREFE(LPC,N1,N3,NLSATT50,WS,NLSTMP,REXP,NLSREXP,RTMP,ILLCOND)

El código de coma fija de HWMCOREFE se ha modificado de forma que en el caso de borrado de trama, sólo se calculan los valores de RTMP(1), ... , RTMP(11), aunque se actualizan REXP(1), ... , REXP(51).

```

SUBROUTINE HWMCOREFE(LPO,N1,N3,NLSATT,WS,NLSTMP,
  RREC,NLSRREC,R,ILLCOND)
NLSAA0=2*NLSTMP
AA0=0
For N=LPO+1,...,N1, do the next 2 lines | compute recursive part of RREC(1)
  P=WS(N)*WS(N) | WS has 2 bits of headroom
  AA0=AA0+P | AA0 will have 5 bits of headroom
  | for energy calculation
  | Case 1: NLSRREC > NLSAA0
If NLSRREC > NLSAA0, do the next 22 lines
  AA0=AA0 >> 1
  IR=NLSRREC-NLSAA0+1
  AA1=RREC(1) << NLSATT | this can be done by multiplication
  AA1=-AA1+(RREC(1) << 16) | scale RREC by attenuation factor
  AA1=AA1 >> IR | align AA0 & AA1
  AA0=AA0+AA1
  Call VSCALE(AA0,1,1,30,AA0,NLSRE) | Find NLS for RREC
  RREC(1)=RND(AA0) | upper 16 bits of AA1 saved
  NLSRREC=NLSAA0-1+NLSRE
  For I=1,2,...,LPO, do the next 11 lines
    AA0=0 | compute recursive part of RREC(I+1)
    For N=LPO+1,...,N1, do the next 2 lines
      P=WS(N)*WS(N-I)
      AA0=AA0+P
      AA0=AA0 >> 1
      AA1=RREC(I+1) << NLSATT | scale RREC by 3/4 or 1/2
      AA1=-AA1+(RREC(I+1) << 16) |
      AA1=AA1 >> IR
      AA0=AA0+AA1
      AA0=AA0 << NLSRE
      RREC(I+1)=RND(AA0) | upper 16 bits of AA0 saved
  Go to FIN_RECUR
  | Case 2: NLSRREC = NLSAA0
If NLSRREC = NLSAA0, do the next 21 lines
  AA1=RREC(1) << NLSATT | scale RREC by 3/4 or 1/2
  AA1=-AA1+(RREC(1) << 16) |
  AA0=AA0 >> 1
  AA1=AA1 >> 1
  AA0=AA0+AA1
  Call VSCALE(AA0,1,1,30,AA0,NLSRE) | Find NLS for RREC
  RREC(1)=RND(AA0) | upper 16 bits of AA1 saved
  NLSRREC=NLSRREC-1+NLSRE
  For I=1,2,...,LPO, do the next 11 lines
    AA0=0 | compute recursive part of RREC(I+1)
    For N=LPO+1,...,N1, do the next 2 lines
      P=WS(N)*WS(N-I)
      AA0=AA0+P
      AA0=AA0 >> 1
      AA1=RREC(I+1) << NLSATT | scale RREC by 3/4 or 1/2
      AA1=-AA1+(RREC(I+1) << 16) |
      AA1=AA1 >> 1
      AA0=AA0+AA1
      AA0=AA0 << NLSRE

```

```

        RREC(I+1)=RND(AA0)          | upper 16 bits of AA0 saved
Go to FIN_RECUR

| Case 3: NLSRREC < NLSAA0
If NLSRREC < NLSAA0, do the next 21 lines
IR=NLSAA0-NLSRREC+1
AA0=AA0 >> IR
AA1=RREC(1) << NLSATT          | scale RREC by 3/4 or 1/2
AA1=-AA1+(RREC(1) << 16)      |
AA1=AA1 >> 1
AA0=AA0+AA1
Call VSCALE(AA0,1,1,30,AA0,NLSRE)
RREC(1)=RND(AA0)              | upper 16 bits of AA1 saved
NLSRREC=NLSRREC+1+NLSRE
For I=1,2,...,LPO, do the next 11 lines
    AA0=0                      | compute recursive part of RREC(I+1)
    For N=LPO+1,...,N1, do the next 2 lines
        P=WS(N)*WS(N-I)
        AA0=AA0+P
    AA0=AA0 >> IR
    AA1=RREC(I+1) << NLSATT      | scale RREC by 3/4 or 1/2
    AA1=-AA1+(RREC(I+1) << 16) |
    AA1=AA1 >> 1
    AA0=AA0+AA1
    AA0=AA0 << NLSRE
    RREC(I+1)=RND(AA0)          | upper 16 bits of AA0 saved

FIN_RECUR:                    | when you reach this point the
                              | recursive component has been computed

AA0=0                          | compute non-recursive part of R(1)
For N=N1+1,...,N3, do the next 2 lines
    P=WS(N)*WS(N)
    AA0=AA0+P

LPFE = LPO
If FERROR .=. TRUE, set LPFE = 10

| Case 1: NLSRREC > NLSAA0
If NLSRREC > NLSAA0, do the next 21 lines
IR=NLSRREC-NLSAA0+1
AA1=RREC(1) << 16
AA1=AA1 >> IR
AA0=AA0 >> 1
AA1=AA0+AA1
AA0=AA1 >> 8                    | apply white noise correction factor
AA1=AA1+AA0
Call VSCALE(AA1,1,1,30,AA1,NLSRR)
R(1)=RND(AA1)                  | upper 16 bits of AA1 saved
For I=1,2,...,LPFE, do the next 10 lines
    AA0=0                      | compute non-recursive part of R(I+1)
    For N=N1+1,...,N3, do the next 2 lines
        P=WS(N)*WS(N-I)
        AA0=AA0+P
    AA0=AA0 >> 1
    AA1=RREC(I+1) << 16
    AA1=AA1 >> IR
    AA1=AA0+AA1
    AA1=AA1 << NLSRR
    R(I+1)=RND(AA1)            | save upper 16 bits
Go to END

| Case 2: NLSRREC = NLSAA0
If NLSRREC = NLSAA0, do the next 18 lines
AA0=AA0 >> 1
AA1=RREC(1) << 15                | this can be done by multiplication
AA1=AA0+AA1
AA0=AA1 >> 8                    | apply white noise correction factor
AA1=AA1+AA0
Call VSCALE(AA1,1,1,30,AA1,NLSRR)
R(1)=RND(AA1)                  | upper 16 bits of AA1 saved
For I=1,2,...,LPFE, do the next 9 lines

```

```

AA0=0 | compute non-recursive part of R(I+1)
For N=N1+1,...,N3, do the next 2 lines
  P=WS(N)*WS(N-I)
  AA0=AA0+P
AA0=AA0 >> 1
AA1=RREC(I+1) << 15
AA1=AA0+AA1
AA1=AA1 << NLSRR
R(I+1)=RND(AA1) | save upper 16 bits
Go to END

| Case 3: NLSRREC < NLSAA0
If NLSRREC < NLSAA0, do the next 18 lines
IR=NLSAA0-NLSRREC+1
AA0=AA0 >> IR
AA1=RREC(1) << 15 | this can be done by multiplication
AA1=AA0+AA1
AA0=AA1 >> 8 | apply white noise correction factor
AA1=AA1+AA0
Call VSCALE(AA1,1,1,30,AA1,NLSRR)
R(1)=RND(AA1) | upper 16 bits of AA1 saved
For I=1,2,...,LPFE, do the next 9 lines
  AA0=0 | compute non-recursive part of R(I+1)
  For N=N1+1,...,N3, do the next 2 lines
    P=WS(N)*WS(N-I)
    AA0=AA0+P
  AA0=AA0 >> IR
  AA1=RREC(I+1) << 15
  AA1=AA0+AA1
  AA1=AA1 << NLSRR
  R(I+1)=RND(AA1) | save upper 16 bits

END: | one last job, check for ill-conditioning
ILLCOND=.FALSE.
If AA1 = 0, set ILLCOND=.TRUE. | AA1 still contains 32 bit R(LPFE+1)

```

1.5.8 Bloque 51FE – Módulo de ampliación de la anchura de banda para borrado de tramas

Se trata del seudocódigo de coma flotante para el bloque 51FE, el módulo de ampliación de anchura de banda para el filtro de síntesis LPC para condiciones de borrado de trama. Este módulo es similar al bloque 51 de 5.6/G.728, excepto en que difieren los coeficientes de ampliación de anchura de banda. Igualmente, los valores de entrada son los del ordenamiento A(I).

```

If FECOUNT = 1 and ILLCOND = .FALSE., do the next 2 lines
  For I=2,3,...,LPC+1, do the next line
    A(I)=FACVFE(I) * ATMP(I) | scale coeff.
Otherwise, do the next 2 lines
  For I=2,3,...,LPC+1, do the next line
    A(I)=FACVFE(I) * A(I) | scale coeff.

```

Los cuadros para FACVFE se presentan en formato Q14. Los valores del ordenamiento A de entrada siempre están en formato Q14. Los valores finales de A siempre pueden representarse en formato Q14. Por lo tanto, después de la multiplicación es necesario un desplazamiento. A continuación se describe el seudocódigo de coma fija del bloque 51.

```

If FECOUNT = 1 and ILLCOND = .FALSE., do the next 4 lines
  For I=2,3,4,...,LPC+1, do the next 3 lines
    AA0 = FACVFE(I) * ATMP(I) | AA0 is Q28
    AA0 = AA0 << 2 | make AA0 Q30
    A(I) = RND(AA0) | round to high word and get Q14
Otherwise, do the next 4 lines
  For I=2,3,4,...,LPC+1, do the next 3 lines
    AA0 = FACVFE(I) * A(I) | AA0 is Q28
    AA0 = AA0 << 2 | make AA0 Q30
    A(I) = RND(AA0) | round to high word and get Q14

```

I.5.9 Bloque 97FE – Actualización de GSTATE durante el borrado de tramas

En primer lugar, se presenta la versión de coma flotante de este nuevo bloque. Se trata del mismo código que los bloques 67, 39, 40 y 42 que se describen en 5.7/G.728.

```

ETRMS = 0.
For K = 1,...,IDIM, do the next line | compute energy
    ETRMS = ETRMS + ET(K)*ET(K) | of excitation
ETRMS = ETRMS * DIMINV
If ETRMS < 1., set ETRMS = 1. | check lower limit
ETRMS = 10.* log10 (ETRMS) | compute dB value

GSTATE(1) = ETRMS - GOFF | subtract dB gain offset

```

La versión de coma fija de este nuevo bloque realiza las operaciones equivalentes. La función log₁₀ fue eliminada de la especificación original de G.728 de coma fija. Ahora debe introducirse de nuevo para realizar los cálculos anteriores. Debido a que no se pretende una concordancia plena con el resultado de la especificación de coma flotante, no es necesario conseguir la máxima precisión en el resultado de log₁₀. A continuación se describe el código de coma fija. Asimismo, debido a que no es necesaria la compatibilidad de bit exacta para el borrado de tramas, no se ha incluido una descripción exacta a nivel de bit de la función LIN2DB en las implementaciones de coma fija. Dicho LIN2DB convierte la representación escalar de coma flotante de ETRMS al valor en dB representado en formato Q9. Puede utilizarse cualquier implementación de coma fija de LIN2DB con una precisión razonable. A continuación se incluye un ejemplo de dicho código de coma fija de LIN2DB.

```

AA0 = 0
For K = 1,...,IDIM, do the next line | compute energy of excitation
    AA0 = AA0 + ET(K)*ET(K) | ET is Q2, so AA0 is Q4

Call VSCALE(AA0,1,1,30,AA0,NLS) | scale AA0
ETRMS = AA0 >> 16 | take high word of AA0
NLSETRMS = (4 + NLS) - 16 | NLS of ETRMS
AA0 = ETRMS * DIMINV | DIMINV = 0.2 in Q16
Call VSCALE(AA0,1,1,30,AA0,NLS) | scale AA0
ETRMS = AA0 >> 16 | take high word of AA0
NLSETRMS = NLSETRMS + NLS | NLS of ETRMS
If NLSETRMS > 14, do the next 2 lines | if ETRMS is Q15 or
| smaller (<1)
    ETRMS = 16384 | set ETRMS to 1 in Q16
    NLSETRMS = 14
Call LIN2DB(ETRMS,NLSETRMS,GSTATE(1)) | convert linear to dB (fixed
| Q9)
| gain offset subtraction is done
| inside LIN2DB

```

A continuación se enumeran las constantes que son necesarias en el código de coma fija de LIN2DB.

c(0)	=	-3 400	=	-0,4150374993 en Q13
c(1)	=	15 758	=	1,9235933879 en Q13
c(2)	=	-10 505	=	-1,2823955919 en Q13
c(3)	=	9 338	=	1,1399071928 en Q13
c(4)	=	-9 338	=	-1,1399071928 en Q13
c(5)	=	9 961	=	1,2159010057 en Q13
FAC1	=	24 660	=	3,0102999566 en Q13
GOFF	=	-16 384	=	-32 en Q9
THRQTR	=	24 576	=	0,75 en Q15

Se presenta a continuación el código de coma fija de LIN2DB. Esta función calcula una ampliación de la serie de Taylor truncada de la función \log_2 alrededor de 0,75. Asume que la entrada X es una mantisa positiva de 16 bits, con NLSX como el formato Q. Realiza el escalado necesario para obtener el valor en dB y resta el desplazamiento de ganancia GOFF del número de 32 bits resultante previamente antes de realizar la conversión a la salida en formato Q fijo Q9.

```

LIN2DB(X, NLSX, RET)
AA0 = x           | normalized .5 <= x <= 1, Q15
AA0 = AA0 - THRQTR | subtract .75, now -.25 <= x .25, Q15
AA0 = AA0 << 1    | convert to Q16
AA1 = 0
For I=5,4...,1 do the next 3 lines
    AA1 = AA1 + C(I) | Q13
    P = AA1 * AA0    | Q13 * Q16 = Q29
    AA1 = P >> 16   | Q29 -> Q13
AA1 = AA1 + C(0)   | Q13
AA1 = AA1 >> 3     | convert to Q10
AA0 = 15 - NLSX    | exponent in Q0
AA0 = AA0 << 10    | convert exponent to Q10
AA1 = AA0 + AA1    | Add exp to log of mantissa, AA1 now log2(x)
AA1 = AA1 * FAC1   | Q10 * Q13 = Q23, convert to 10*log10(x)
AA1 = AA1 >> 14   | Q23 -> Q9
AA1 = AA1 - GOFF   | Subtract GOFF
RET = AA1          | return Q9 value

```

I.5.10 Bloque 98AF – Limitador de ganancia logarítmica después del borrado de trama

Este nuevo bloque sustituye al bloque 98 de G.3.16/G.728. Se presenta a continuación el pseudocódigo de coma fija del bloque 98AF.

```

If LOGGAIN > 14336, set LOGGAIN=14336
If LOGGAIN < -16384, set LOGGAIN=-16384
TMP = LOGGAIN - OGAINDB
If AFTERFE > 0 and TMP > FEGAINMAX, do the next line
    LOGGAIN = OGAINDB + FEGAINMAX

```

I.6 Parámetros y variables adicionales del codificador

Esta cláusula incluye un cuadro con las nuevas variables que han sido añadidas al decodificador G.728 para el ocultamiento del borrado de trama. En la columna de formato se enumera el formato de coma fija. En una implementación de coma flotante, todas las variables de formato Q fijo o de coma flotante de bloque pueden representarse en coma flotante.

Cuadro I.1/G.728 – Parámetros adicionales del codificador necesarios para el ocultamiento del borrado de trama

Nombre	Valor de coma flotante	Valor de coma fija	formato Q	Descripción
AFTERFEMAX	16	16	Q0	Tramas de un máximo de 2,5 ms para el recorte de ganancia
DIMINV	0,2	13 107	Q16	Inverso de la dimensión del vector
FEGAINMAX	2	1 024	Q9	Aumento máximo de ganancia DB después del borrado de trama
VTH	0,4285714	7 022	Q14	Umbral de vocalización para borrado de tramas
FESCALE	0,8	26 214	Q15	Factor de escalado ETPAST para borrado de tramas

**Cuadro I.2/G.728 – Variables de procesamiento internas de LD-CELP
adicionales necesarias para el ocultamiento del borrado de trama**

Nombre	Rango del índice del ordenamiento	Formato de coma fija	Descripción
MAG	1	a)	Tamaño medio del vector de excitación actual
ADCOUNT	1	Entero	Contador de ciclo de adaptación G.728 (2,5 ms)
AFTERFE	1	Entero	Contador del recorte de ganancia después del FE (2,5 ms)
AVMAG	1	SFL	Tamaño medio de los anteriores 8 vectores de excitación
ETPAST	-139 a IDIM	Q2	Excitación anterior (inicialmente cero)
ETRMS	1	SFL	Energía en dB del vector actual
FECOUNT	1	Entero	Longitud del actual borrado de trama (2,5 ms)
FEDELAY	1	Entero	Tono (KP) almacenado al comienzo del FE
FERROR	1	Lógico	Indicador de borrado de trama
FESCALE	1	Entero	Factor de escalado ETPAST para borrado de tramas
FESIZE	1	Entero	Número de ciclos de adaptación de 2,5 ms en un FE
NLSFESCALE	1	Entero	NLS para FESCALE
NLSAVMAG	1	Entero	NLS para AVGMAG
NLSETRMS	1	Entero	NLS para ETRMS
OGAINDB	1	Q9	Ganancia de predicción antigua en dB
VOICED	1	Lógico	Bandera de vocalizada/no vocalizada
a) MAG no se utiliza en el seudocódigo de coma fija.			

APÉNDICE I.I

Valores utilizados para el escalado de ETPAST durante el borrado de tramas

Los cuadros I.I.1 y I.I.2 proporcionan los valores enteros y en coma flotante utilizados para escalar ETPAST durante el borrado de tramas tanto para señales vocalizadas (VOICEDFEGAIN como no vocalizadas (UNVOICEDFEGAIN). En el caso de coma fija, los valores se encuentran en formato Q15.

Cuadro I.I.1/G.728 – Valores utilizados para escalar ETPAST con señales vocalizadas (VOICEDFEGAIN)

Índice de VOICEDFEGAIN	Valor de coma flotante	Valor de coma fija Q15
0	0,8	26 214
1	0,8	26 214
2	0,6	19 661
3	0,4	13 107
4	0,2	6 554

Cuadro I.I.2/G.728 – Valores utilizados para escalar ETPAST con señales no vocalizadas (UNVOICEDFEGAIN)

Índice de VOICEDFEGAIN	Valor de coma flotante	Valor de coma fija Q15
0	1,0	32 767
1	1,0	32 767
2	0,8	26 214
3	0,6	19 661
4	0,4	13 107
5	0,2	6 554

Valores utilizados para la ampliación de la anchura de banda del predictor LPC durante el borrado de tramas

El cuadro siguiente proporciona los valores enteros del ordenamiento de coeficientes de ampliación de anchura de banda FACVFE utilizados para suavizar el predictor LPC durante el borrado de tramas. Los valores se encuentran en formato Q14. Para obtener los valores equivalentes en coma flotante, divídanse los valores enteros por 16384. Este cuadro debe leerse de izquierda a derecha y de arriba a abajo, comenzando en FACVFE(1) y terminando en FACVFE(51).

16384	15892	15416	14953	14505
14069	13647	13238	12841	12456
12082	11719	11368	11027	10696
10375	10064	9762	9469	9185
8910	8642	8383	8131	7888
7651	7421	7199	6983	6773
6570	6373	6182	5996	5816
5642	5473	5309	5149	4995
4845	4700	4559	4422	4289
4161	4036	3915	3797	3683
3573				

SERIES DE RECOMENDACIONES DEL UIT-T

Serie A	Organización del trabajo del UIT-T
Serie B	Medios de expresión: definiciones, símbolos, clasificación
Serie C	Estadísticas generales de telecomunicaciones
Serie D	Principios generales de tarificación
Serie E	Explotación general de la red, servicio telefónico, explotación del servicio y factores humanos
Serie F	Servicios de telecomunicación no telefónicos
Serie G	Sistemas y medios de transmisión, sistemas y redes digitales
Serie H	Sistemas audiovisuales y multimedios
Serie I	Red digital de servicios integrados
Serie J	Transmisiones de señales radiofónicas, de televisión y de otras señales multimedios
Serie K	Protección contra las interferencias
Serie L	Construcción, instalación y protección de los cables y otros elementos de planta exterior
Serie M	RGT y mantenimiento de redes: sistemas de transmisión, circuitos telefónicos, telegrafía, facsímil y circuitos arrendados internacionales
Serie N	Mantenimiento: circuitos internacionales para transmisiones radiofónicas y de televisión
Serie O	Especificaciones de los aparatos de medida
Serie P	Calidad de transmisión telefónica, instalaciones telefónicas y redes locales
Serie Q	Conmutación y señalización
Serie R	Transmisión telegráfica
Serie S	Equipos terminales para servicios de telegrafía
Serie T	Terminales para servicios de telemática
Serie U	Conmutación telegráfica
Serie V	Comunicación de datos por la red telefónica
Serie X	Redes de datos y comunicación entre sistemas abiertos
Serie Y	Infraestructura mundial de la información y aspectos protocolo Internet
Serie Z	Lenguajes y aspectos generales de soporte lógico para sistemas de telecomunicación