

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 933 287**

51 Int. Cl.:

G10L 19/26 (2013.01)

G10L 19/02 (2013.01)

G10L 19/04 (2013.01)

G10L 19/028 (2013.01)

G10L 19/038 (2013.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **06.04.2017** **E 20168799 (3)**

97 Fecha y número de publicación de la concesión europea: **26.10.2022** **EP 3696813**

54 Título: **Codificador de audio para codificar una señal de audio, método para codificar una señal de audio y programa informático en consideración de una región espectral del pico detectada en una banda de frecuencia superior**

30 Prioridad:

12.04.2016 EP 16164951

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
03.02.2023

73 Titular/es:

**FRAUNHOFER-GESELLSCHAFT ZUR
FÖRDERUNG DER ANGEWANDTEN
FORSCHUNG E.V. (100.0%)
Hansastr. 27c
80686 München, DE**

72 Inventor/es:

**MULTRUS, MARKUS;
NEUKAM, CHRISTIAN;
SCHNELL, MARKUS y
SCHUBERT, BENJAMIN**

74 Agente/Representante:

ARIZTI ACHA, Monica

ES 2 933 287 T3

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín Europeo de Patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre Concesión de Patentes Europeas).

DESCRIPCIÓN

Codificador de audio para codificar una señal de audio, método para codificar una señal de audio y programa informático en consideración de una región espectral del pico detectada en una banda de frecuencia superior

5 La presente invención se refiere a la codificación de audio, y preferiblemente, a un método, aparato o programa informático para controlar la cuantificación de los coeficientes espectrales para TCX basada en MDCT en el códec EVS.

10 Un documento de referencia para el códec EVS es 3GPP TS 24.445 V13.1.0 (2016-03), proyecto de asociación de tercera generación; Memoria descriptiva Técnica de Servicios de Grupo y Aspectos del Sistema; Códec para servicios de voz mejorados (EVS); Descripción algorítmica detallada (versión 13).

15 Sin embargo, la presente invención es adicionalmente útil en otras versiones EVS, por ejemplo, definidas por otras versiones que la versión 13 y, adicionalmente, la presente invención es adicionalmente útil en todos los otros codificadores de audio diferentes de EVS que, sin embargo, se basan en un detector, un conformador y una etapa de cuantificador y codificador como se define, por ejemplo, en las reivindicaciones.

20 Adicionalmente, se debe tener en cuenta que todas las realizaciones definidas no solo por las reivindicaciones independientes, sino también definidas por las reivindicaciones dependientes, se pueden usar separadamente entre sí o juntas como se indica mediante las interdependencias de las reivindicaciones o como se discute más adelante bajo los ejemplos preferidos.

25 El códec EVS [1], como se especifica en 3GPP, es un códec híbrido moderno para contenido de habla y audio de banda estrecha (NB), banda ancha (WB), banda super ancha (SWB) o de banda completa (FB) que puede cambiar entre varios enfoques de codificación, basándose en la clasificación de la señal:

30 La figura 1 ilustra un procesamiento común y diferentes esquemas de codificación en EVS. Particularmente, una porción del procesamiento común del codificador en la figura 1 comprende un bloque de remuestreo de señales 101 y un bloque de análisis de señales 102. La señal de entrada de audio se introduce en una entrada de señal de audio 103 en la porción de procesamiento común y, en particular, en el bloque de remuestreo de señales. El bloque de remuestreo de señales 101 tiene adicionalmente una entrada de línea de comandos para recibir los parámetros de la línea de comandos. La salida de la etapa de procesamiento común se introduce en diferentes elementos como se puede ver en la figura 1. En particular, la figura 1 comprende un bloque de codificación basado en predicción lineal (codificación basada en LP) 110, un bloque de codificación de dominio de frecuencia 120 y un bloque de codificación de señal inactiva/CNG 130. Los bloques 110, 120, 130 están conectados a un multiplexor del flujo de bits 140. Además, se proporciona un conmutador 150 para la conmutación, que depende de una decisión del clasificador, la salida de la etapa de procesamiento común al bloque de codificación basado en LP 110, el bloque de codificación de dominio de frecuencia 120 o el bloque de codificación de señal inactiva/CNG (generación de ruido de confort). Además, el multiplexor de flujo de bits 140 recibe una información del clasificador, es decir, si una porción actual determinada de la entrada de señal de audio entra en el bloque 103 y es procesada por la porción del procesamiento común se codifica usando cualquiera de los bloques 110, 120, 130.

45 - La codificación basada en LP (basada en la predicción lineal), tal como la codificación CELP, se utiliza principalmente para contenido de habla o habla dominante y contenido de audio genérico con alta fluctuación temporal.

50 - La codificación de dominio de frecuencia se utiliza para todos los demás contenidos de audio genéricos, tal como música o ruido de fondo.

Para proporcionar la máxima calidad para las tasas de bits bajas y medias, se realiza una conmutación frecuente entre la codificación basada en LP y la codificación del dominio de frecuencia, basándose en el análisis de señal en un módulo de procesamiento común. Para ahorrar en complejidad, el códec se optimizó para reutilizar elementos de la etapa de análisis de señal también en módulos posteriores. Por ejemplo: el módulo de análisis de señal presenta una etapa de análisis LP. Los coeficientes de filtro LP (LPC) resultantes y la señal residual se utilizan en primer lugar para varias etapas de análisis de señal, tales como el detector de actividad de voz (VAD) o el clasificador de habla/música. En segundo lugar, el LPC es también una parte elemental del esquema de codificación basado en LP y el esquema de codificación de dominio de frecuencia. Para ahorrar en complejidad, el análisis LP se realiza a la velocidad de muestreo interna del codificador CELP (SR_{CELP}).

60 El codificador CELP funciona a una velocidad de muestreo interna de 12,8 o 16 kHz (SR_{CELP}) y, por tanto, puede representar señales de ancho de banda de audio de hasta 6,4 o 8 kHz directamente. Para el contenido de audio que excede este ancho de banda en WB, SWB o FB, el contenido de audio por encima de la representación de frecuencia de CELP está codificado por un mecanismo de extensión de ancho de banda.

El TCX basado en MDCT es un submodo de la codificación del dominio de frecuencia. Al igual que para el enfoque de codificación basado en LP, la formación de ruido en TCX se realiza mediante la aplicación de factores de ganancia calculados a partir de coeficientes de filtro LP cuantificados ponderados al espectro MDCT (lado decodificador). En el lado del codificador, los factores de ganancia inversa se aplican antes del bucle de velocidad. Esto se conoce posteriormente como aplicación de ganancias de conformación de LPC. TCX funciona en la velocidad de muestreo de entrada (SR_{inp}). Esto se aprovecha para codificar el espectro completo directamente en el dominio MDCT, sin extensión de ancho de banda adicional. La velocidad de muestreo de entrada SR_{inp} , en la que se realiza la transformación MDCT, puede ser mayor que la tasa de muestreo CELP SR_{CELP} , para la que se calculan los coeficientes LP. Por lo tanto, las ganancias de conformación de LPC solo se pueden calcular para la parte del espectro MDCT correspondiente al intervalo de frecuencias CELP (f_{CELP}). Para la parte restante del espectro (si existe), se utiliza la ganancia de conformación de la banda de frecuencia más alta.

La figura 2 ilustra en un alto nivel la aplicación de ganancias de conformación de LPC y para el TCX basado en MDCT. En particular, la figura 2 ilustra un principio de formación y codificación de ruido en el bloque de codificación de TCX o de dominio de frecuencia 120 de la figura 1 en el lado del codificador.

Particularmente, la figura 2 ilustra un diagrama de bloques esquemático de un codificador. La señal de entrada 103 se introduce en el bloque de remuestreo 201 con el fin de realizar un remuestreo de la señal a la velocidad de muestreo CELP SR_{CELP} , es decir, la velocidad de muestreo requerida por el bloque de codificación basado en LP 110 de la figura 1. Además, se proporciona un calculador 203 de LPC que calcula los parámetros de LPC en el bloque 205, se realiza una ponderación basada en LPC para que la señal sea procesada adicionalmente por el bloque de codificación basado en LP 110 en la figura 1, es decir, la señal residual LPC que se codifica usando el procesador ACELP.

Adicionalmente, se introduce la señal de entrada 103, sin ningún remuestreo, en un convertidor espectral de tiempo 207 que se ilustra a modo de ejemplo como una transformada MDCT. Además, en el bloque 209, los parámetros LPC calculados mediante el bloque 203 se aplican después de algunos cálculos. Particularmente, el bloque 209 recibe los parámetros LPC calculados a partir del bloque 203 a través de la línea 213 o alternativa o adicionalmente a partir del bloque 205 y luego deriva los factores de ponderación del MDCT o, en general, del dominio espectral para aplicar las ganancias de conformación de LPC inversas correspondientes. Entonces, en el bloque 211, se realiza una operación del cuantificador/codificador general que puede ser, por ejemplo, un bucle de velocidad que ajusta la ganancia global y, adicionalmente, realiza una cuantificación/codificación de coeficientes espectrales, preferiblemente usando la codificación aritmética como se ilustra en la memoria descriptiva del codificador EVS conocida para finalmente obtener el flujo de bits.

En contraste con el enfoque de codificación CELP, que combina un codificador central en SR_{CELP} y un mecanismo de extensión de ancho de banda a una mayor velocidad de muestreo, los enfoques de codificación basados en MDCT funcionan directamente sobre la velocidad de muestreo de entrada SR_{inp} y codifican el contenido del espectro completo en el dominio MDCT.

El TCX basado en MDCT codifica hasta 16 kHz de contenido de audio a bajas tasas de bits, tales como SWB de 9,6 o 13,2 kbit/s. Debido a que a tales tasas de bits bajas solo un pequeño subconjunto de los coeficientes espectrales puede ser codificado directamente por medio del codificador aritmético, los intervalos resultantes (regiones de valores cero) en el espectro están ocultos por dos mecanismos:

- Llenado de ruido, que inserta ruido aleatorio en el espectro decodificado. La energía del ruido es controlada por un factor de ganancia, que se transmite en el flujo de bits.
- Llenado de intervalo inteligente (IGF), que inserta porciones de la señal de partes de frecuencias más bajas del espectro. Las características de estas porciones de frecuencia insertadas son controladas por parámetros que se transmiten en el flujo de bits.

El llenado de ruido se utiliza para las porciones de frecuencia más baja hasta la frecuencia más alta, que puede ser controlada por el LPC transmitido (f_{CELP}). Por encima de esta frecuencia, se utiliza la herramienta IGF, que proporciona otros mecanismos para controlar el nivel de las porciones de frecuencia insertadas.

Existen dos mecanismos para la decisión sobre qué coeficientes espectrales sobreviven al procedimiento de codificación o cuáles serán reemplazados por llenado de ruido o IGF:

1) Bucle de velocidad

Después de la aplicación de las ganancias de conformación LPC inversa, se aplica un bucle de velocidad. Para ello, se estima una ganancia global. Posteriormente, los coeficientes espectrales se cuantifican, y los coeficientes espectrales cuantificados se codifican con el codificador aritmético. Basándose en la demanda

de bits real o estimada del codificador aritmético y el error de cuantificación, la ganancia global se incrementa o disminuye. Esto afecta a la precisión del cuantificador. Cuanto menor es la precisión, más coeficientes espectrales se cuantifican a cero. La aplicación de las ganancias de conformación LPC inversa utilizando un LPC ponderado antes del bucle de velocidad asegura que las líneas perceptualmente relevantes sobreviven con una probabilidad significativamente mayor que el contenido perceptivamente irrelevante.

2) Enmascaramiento tonal IGF

Por encima de f_{CELP} , donde no está disponible LPC, se utiliza un mecanismo diferente para identificar los componentes espectrales perceptualmente relevantes: la energía de forma lineal se compara con la energía promedio en la región IGF. Se mantienen líneas espectrales predominantes, que corresponden a porciones de señales perceptualmente relevantes, todas las otras líneas se ajustan a cero. El espectro de MDCT, que se preprocesó con el enmascaramiento tonal de IGF, se alimenta posteriormente en el bucle de velocidad.

El LPC ponderado sigue la envolvente espectral de la señal. Mediante la aplicación de las ganancias de conformación LPC inversa usando el LPC ponderado se realiza un blanqueamiento perceptual del espectro. Esto reduce significativamente la dinámica del espectro MDCT antes del bucle de codificación y, por tanto, también controla la distribución de bits entre los coeficientes espectrales MDCT en el bucle de codificación.

Como se explicó anteriormente, el LPC ponderado no está disponible para frecuencias superiores a f_{CELP} . Para estos coeficientes MDCT, se aplica la ganancia de conformación de la banda de frecuencia más alta por debajo de f_{CELP} . Esto funciona bien en los casos donde la ganancia de conformación de la banda de frecuencias más alta por debajo de f_{CELP} corresponde aproximadamente a la energía de los coeficientes por encima de f_{CELP} , lo que a menudo se debe a la inclinación espectral y que se puede observar en la mayoría de las señales de audio. Por lo tanto, este procedimiento es ventajoso, ya que no es necesario calcular o transmitir la información de conformación para la banda superior.

Sin embargo, en caso de que existan componentes espectrales fuertes por encima de f_{CELP} y la ganancia de conformación de la banda de frecuencia más alta por debajo de f_{CELP} sea muy baja, esto da como resultado un desajuste. Este desajuste impacta fuertemente en el funcionamiento o en el bucle de velocidad, que se centra en los coeficientes espectrales que tienen la amplitud más alta. A tasas de bits bajas, esto reducirá a cero los componentes de señal restantes, especialmente en la banda baja, y producirá una calidad perceptualmente mala.

Las figuras 3–6 ilustran el problema. La figura 3 muestra el espectro MDCT absoluto antes de la aplicación de las ganancias de conformación LPC inversa, la figura 4 las ganancias de conformación LPC correspondientes. Hay picos fuertes por encima de f_{CELP} visibles, que están en el mismo orden de magnitud que los picos más altos debajo de f_{CELP} . Los componentes espectrales por encima de f_{CELP} son el resultado del preprocesamiento usando el enmascaramiento tonal de IGF. La figura 5 muestra el espectro de MDCT absoluto después de aplicar las ganancias de LPC inversa, aún antes de la cuantificación. Ahora los picos por encima de f_{CELP} exceden significativamente los picos por debajo de f_{CELP} , con el efecto de que el bucle de velocidad se centrará principalmente en estos picos. La figura 6 muestra el resultado del bucle de velocidad a bajas tasas de bits: todos los componentes espectrales excepto los picos por encima de f_{CELP} se cuantificaron a 0. Esto produce un resultado perceptualmente muy pobre después del proceso de decodificación completo, ya que las porciones de señal psicoacústicamente muy relevantes en las frecuencias bajas faltan por completo.

La figura 3 ilustra un espectro MDCT de una trama crítica antes de la aplicación de ganancias de conformación LPC inversa.

La figura 4 ilustra las ganancias de conformación de LPC tal como se aplican. En el lado del codificador, el espectro se multiplica con la ganancia inversa. El último valor de ganancia se utiliza para todos los coeficientes MDCT por encima de f_{CELP} . La figura 4 indica f_{CELP} en el borde derecho.

La figura 5 ilustra un espectro MDCT de una trama crítica de la aplicación de ganancias de conformación LPC inversa. Los altos picos por encima de f_{CELP} son claramente visibles.

La figura 6 ilustra un espectro MDCT de una trama crítica después de la cuantificación. El espectro mostrado incluye la aplicación de la ganancia global, pero sin las ganancias de conformación LPC. Se puede observar que todos los coeficientes espectrales excepto el pico por encima de f_{CELP} se cuantifican a 0.

El documento EP 2 980 794 A1 da a conocer un codificador de audio que comprende: un primer procesador de codificación para codificar una primera porción de la señal de audio en un dominio de la frecuencia, en el que el primer procesador de codificación comprende: un convertidor de frecuencia y de tiempo para convertir la primera porción de la señal de audio en una representación de un dominio de la frecuencia que tiene líneas espectrales hasta una frecuencia máxima de la primera porción de la señal de audio; un analizador para analizar la representación en el

dominio de la frecuencia hasta la frecuencia máxima para determinar primeras porciones espectrales para codificar con una primera resolución espectral y segundas regiones espectrales para codificar con una segunda resolución espectral, siendo la segunda resolución espectral inferior a la primera resolución espectral; un codificador espectral para codificar las primeras porciones espectrales con la primera resolución espectral y para codificar las segundas porciones espectrales con la segunda resolución espectral; un segundo procesador de codificación para codificar una segunda porción de la señal de audio diferente en el dominio del tiempo; un controlador configurado para analizar la señal de audio y para determinar qué porción de la señal de audio es la primera porción de la señal de audio codificada en el dominio de la frecuencia y qué porción de la señal de audio es la segunda porción de la señal de audio codificada en el dominio del tiempo; y un conformador de señales codificadas para formar una señal de audio codificada que comprende una primera porción de la señal codificada para la primera porción de la señal de audio y una segunda porción de la señal codificada para la segunda porción de la señal de audio.

Un objeto de la presente invención es proporcionar un concepto de codificación de audio mejorado.

Este objeto se obtiene mediante un codificador de audio según la reivindicación 1, un método para codificar una señal de audio según la reivindicación 14 o un programa informático según la reivindicación 15.

La presente invención se basa en el hallazgo de que tales problemas de la técnica anterior se pueden resolver mediante el preprocesamiento de la señal de audio para codificar dependiendo de una característica específica de la etapa de cuantificador y codificador incluida en el codificador de audio. Con este fin, se detecta una región espectral del pico de una banda de frecuencia superior de la señal de audio. Luego, se usa un conformador para conformar la banda de frecuencia más baja usando la información de conformación para la banda más baja y para conformar la banda de frecuencia superior usando al menos una porción de la información de conformación para la banda más baja. En particular, el conformador se configura adicionalmente para atenuar los valores espectrales en una región espectral del pico detectada, es decir, en una región espectral del pico detectado por el detector de la banda de frecuencia superior de la señal de audio. Luego, la banda de frecuencia más baja conformada y la banda de frecuencia superior atenuada se cuantifican y codifican por entropía.

Debido al hecho de que la banda de frecuencia superior se ha atenuado selectivamente, es decir, dentro de la región espectral de pico detectada, esta región espectral de pico detectada ya no puede dominar completamente el comportamiento de la etapa de cuantificador y codificador.

En cambio, debido al hecho de que se ha formado una atenuación en la banda de frecuencia superior de la señal de audio, se mejora la calidad perceptual global del resultado de la operación de codificación. Particularmente a bajas tasas de bits, en las que una tasa de bits relativamente baja es un objetivo principal de la etapa de cuantificador y codificador, los picos espectrales altos en la banda de frecuencia superior pueden consumir todos los bits requeridos por la etapa de cuantificador y codificador, ya que el codificador puede ser guiado por las porciones de alta frecuencia superior y, por lo tanto, pueden usar la mayoría de los bits disponibles en estas porciones. Esto genera automáticamente una situación en la que los bits para los intervalos de frecuencia más bajos perceptualmente más importantes ya no están disponibles. De este modo, tal procedimiento puede producir una señal que solo tiene porciones de alta frecuencia codificadas, mientras que las porciones de frecuencia más baja no están codificadas en absoluto o solo están codificadas de manera muy gruesa. Sin embargo, se ha encontrado que tal procedimiento es menos agradable perceptualmente en comparación con una situación, donde se detecta tal situación problemática con regiones espectrales altas predominantes y se atenúan los picos en el intervalo de frecuencias más altas antes de realizar el procedimiento del codificador que comprende una etapa de cuantificador y codificador de entropía.

Preferiblemente, la región espectral de pico se detecta en la banda de frecuencia superior de un espectro MDCT. Sin embargo, también se pueden usar otros convertidores espectrales–temporales tales como un banco de filtros, un banco de filtros QMF, un DFT, un FFT o cualquier otra conversión tiempo–frecuencia.

Además, la presente invención es útil porque, para la banda de frecuencia superior, no se requiere calcular la información de conformación. En cambio, se usa una información de conformación originalmente calculada para la banda de frecuencia más baja para conformar la banda de frecuencia superior. En consecuencia, la presente invención proporciona un codificador muy eficiente desde el punto de vista computacional ya que también se puede usar una información de conformación de banda baja para conformar la banda alta, debido a que los problemas que pueden resultar de tal situación, es decir, altos valores espectrales en la banda de frecuencia superior, se pueden resolver mediante la atenuación adicional aplicada adicionalmente por el conformador además de la conformación sencilla típicamente basada en la envolvente espectral de la señal de banda baja que, por ejemplo, se puede caracterizar mediante unos parámetros de LPC para la señal de banda baja. Pero la envolvente espectral también se puede representar por cualquier otra medida correspondiente que sea utilizable para realizar una conformación en el dominio espectral.

La etapa de cuantificador y codificador realiza una operación de cuantificación y codificación sobre la señal conformada, es decir, sobre la señal de banda baja conformada y sobre la señal de banda alta conformada, pero la

señal de banda alta conformada ha recibido adicionalmente la atenuación adicional.

Aunque la atenuación de la banda alta en la región espectral de pico detectada es una operación de preprocesamiento que no puede ser recuperada por el decodificador, el resultado del decodificador es sin embargo más agradable en comparación con una situación en la que no se aplica la atenuación adicional, debido a que la atenuación produce el hecho de que los bits permanecen para la banda de frecuencia más baja perceptualmente más importante. Por lo tanto, en situaciones problemáticas en las que una región espectral elevada con picos puede dominar el resultado de codificación entero, la presente invención proporciona una atenuación adicional de dichos picos de modo que, al final, el codificador "ve" una señal que tiene porciones de alta frecuencia atenuada y, por lo tanto, la señal codificada todavía tiene información de baja frecuencia útil y perceptualmente agradable. El "sacrificio" con respecto a la banda espectral alta no es o casi no es perceptible por los oyentes, ya que los oyentes, en general, no tienen una imagen clara del contenido de alta frecuencia de una señal, pero tienen, con una probabilidad mucho mayor, una expectativa con respecto al contenido de frecuencia baja. En otras palabras, una señal que tiene un contenido de baja frecuencia de nivel muy bajo pero un contenido de frecuencia de alto nivel significativo es una señal que se percibe típicamente como no natural.

Las realizaciones preferidas de la invención comprenden un analizador de predicción lineal para derivar los coeficientes de predicción lineal durante un marco de tiempo y estos coeficientes de predicción lineal representan la información de conformación o la información de conformación se deriva de estos coeficientes de predicción lineal.

En una realización adicional, varios factores de conformación se calculan para varias sub-bandas de la banda de frecuencia más baja, y para la ponderación en la banda de frecuencia mayor, se usa el factor de conformación calculado para la sub-banda más alta de la banda de baja frecuencia.

En una realización adicional, el detector determina una región espectral del pico en la banda de frecuencia superior cuando al menos un grupo de condiciones es válido, donde el grupo de condiciones comprende al menos una condición de amplitud de banda de baja frecuencia, una condición de distancia del pico y una condición de amplitud del pico. Aún más preferiblemente, una región espectral del pico solo se detecta cuando dos condiciones son verdaderas al mismo tiempo y aún más preferiblemente, una región espectral del pico solo se detecta cuando las tres condiciones son verdaderas.

En una realización adicional, el detector determina varios valores usados para examinar las condiciones ya sea antes o después de la operación de conformación con o sin la atenuación adicional.

En una realización, el conformador atenúa adicionalmente los valores espectrales usando un factor de atenuación, donde este factor de atenuación se deriva de una amplitud espectral máxima en la banda de frecuencia más baja multiplicada por un número predeterminado que es mayor o igual a 1 y dividida por la amplitud espectral máxima en la banda de frecuencia superior.

Además, la forma específica, en cuanto a cómo se aplica la atenuación adicional, se puede realizar de varias maneras diferentes. Una forma es que el conformador realice en primer lugar la información de ponderación usando al menos una porción de la información de conformación para la banda de frecuencia más baja con el fin de conformar los valores espectrales en la región espectral de pico detectada. A continuación, se realiza una operación de ponderación posterior usando la información de atenuación.

Un procedimiento alternativo es aplicar primero una operación de ponderación usando la información de atenuación y luego realizar una ponderación posterior usando una información de ponderación que responde a por lo menos la porción de la información de conformación para la banda de frecuencias más baja. Otra alternativa es aplicar una única información de ponderación usando una información de ponderación combinada que se deriva de la atenuación por una parte y la porción de la información de conformación para la banda de frecuencia más baja por otra parte.

En una situación donde la ponderación se realiza usando una multiplicación, la información de atenuación es un factor de atenuación y la información de conformación es un factor de conformación y la información de ponderación combinada real es un factor de ponderación, es decir, un factor de ponderación único para la información de ponderación única, donde este factor de ponderación único se deriva por multiplicación de la información de atenuación y la información de conformación para la banda más baja. En consecuencia, es evidente que el conformador se puede implementar de muchas maneras diferentes, pero no obstante, el resultado es una conformación de la banda de alta frecuencia usando información de conformación de la banda más baja y una atenuación adicional.

En una realización, la etapa de cuantificador y codificador comprende un procesador del bucle de velocidad para estimar una característica del cuantificador de modo que se obtiene la tasa de bits predeterminada de una señal de audio codificada por entropía. En una realización, esta característica del cuantificador es una ganancia global, es decir, un valor de ganancia aplicado al intervalo de frecuencia entero, es decir, aplicado a todos los valores espectrales que

se van a cuantificar y codificar. Cuando parece que la tasa de bits requerida es menor que una tasa de bits obtenida usando una cierta ganancia global, entonces la ganancia global se incrementa y se determina si la tasa de bits real está ahora en línea con el requisito, es decir, ahora es menor o igual a la tasa de bits requerida. Este procedimiento se realiza cuando la ganancia global se utiliza en el codificador antes de la cuantificación de tal manera que los valores espectrales se dividen por la ganancia global. Sin embargo, cuando la ganancia global se utiliza de manera diferente, es decir, multiplicando los valores espectrales por la ganancia global antes de realizar la cuantificación, entonces la ganancia global disminuye cuando una tasa de bits real es demasiado alta o la ganancia global puede aumentar cuando la tasa de bits real es más baja que la admisible.

Sin embargo, se pueden usar otras características de etapa del codificador también en una condición de bucle de velocidad determinada. Una forma puede ser, por ejemplo, una ganancia selectiva de frecuencia. Un procedimiento adicional puede ser ajustar el ancho de banda de la señal de audio que depende de la tasa de bits requerida. Generalmente, se puede influir en diferentes características de cuantificador de manera que, al final, se obtiene una tasa de bits que está en línea con la tasa de bits requerida (típicamente baja).

Preferiblemente, este procedimiento es particularmente adecuado para combinarse con el procesamiento de llenado de intervalos inteligente (procesamiento de IGF). En este procedimiento, se aplica un procesador de enmascaramiento tonal para determinar, en la banda de frecuencias superior, un primer grupo de valores espectrales para cuantificar y codificar por entropía y un segundo grupo de valores espectrales para codificar paramétricamente mediante el procedimiento de llenado de intervalos. El procesador de enmascaramiento tonal ajusta el segundo grupo de valores espectrales de valores 0, de manera que estos valores no consumen muchos bits en la etapa de cuantificador/codificador. Por otra parte, parece que típicamente los valores pertenecientes al primer grupo de valores espectrales que han de ser cuantificados y codificados por entropía son los valores de la región espectral de pico que, bajo ciertas circunstancias, se pueden detectar y atenuar adicionalmente en caso de una situación problemática para la etapa cuantificador/codificador. Por lo tanto, la combinación de un procesador de enmascaramiento tonal dentro de una estructura de llenado de intervalos inteligente con la atenuación adicional de las regiones espectrales de pico detectadas da como resultado un procedimiento del codificador eficiente que es, además, compatible con el retorno y, no obstante, produce una calidad perceptual buena incluso a tasas de bits muy bajas.

Las realizaciones son ventajosas respecto a soluciones potenciales para tratar este problema que incluyen métodos para extender el intervalo de frecuencias de LPC u otros medios para ajustar mejor las ganancias aplicadas a frecuencias por encima de f_{CELP} a los coeficientes espectrales MDCT reales. Este procedimiento, sin embargo, destruye la compatibilidad de retorno, cuando un códec ya está implementado en el mercado, y los métodos descritos previamente pueden romper la interoperabilidad con las implementaciones existentes.

Posteriormente, se ilustran las realizaciones preferidas de la presente invención con respecto a los dibujos adjuntos, en los que:

la figura 1 ilustra un procesamiento común y diferentes esquemas de codificación en EVS;

la figura 2 ilustra un principio de formación de ruido y codificación en TCX del lado del codificador;

la figura 3 ilustra un espectro de MDCT de una trama crítica antes de la aplicación de ganancias de conformación de LP inversas;

la figura 4 ilustra la situación de la figura 3, pero con las ganancias de conformación de LP aplicadas;

la figura 5 ilustra un espectro de MDCT de una trama crítica después de la aplicación de las ganancias de conformación de LP inversas, donde los picos altos por encima de f_{CELP} son claramente visibles;

la figura 6 ilustra un espectro de MDCT de una trama crítica después de la cuantificación que solo tiene información de paso alto y que no tiene ninguna información de paso bajo;

la figura 7 ilustra un espectro de MDCT de una trama crítica después de la aplicación de las ganancias de conformación de LP inversas y el preprocesamiento del lado del codificador de la invención;

la figura 8 ilustra una realización preferida un codificador de audio para codificar una señal de audio;

la figura 9 ilustra la situación para el cálculo de la información de conformación diferente para bandas de frecuencia diferentes y el uso de la información de conformación la banda más baja para la banda más alta;

la figura 10 ilustra una realización preferida de un codificador de audio;

la figura 11 ilustra un diagrama de flujo para ilustrar la funcionalidad del detector para detectar la región espectral

del pico;

- la figura 12 ilustra una implementación preferida de la implementación de la condición de amplitud de banda
baja;
- la figura 13 ilustra una realización preferida de la implementación de la condición de distancia del pico;
- la figura 14 ilustra una implementación preferida de la implementación de la condición de amplitud del pico;
- la figura 15a ilustra una implementación preferida de la etapa de cuantificador y codificador;
- la figura 15b ilustra un diagrama de flujo para ilustrar la operación de la etapa de cuantificador y codificador como un procesador del bucle de velocidad;
- la figura 16 ilustra un procedimiento de determinación para determinar el factor de atenuación en una realización preferida; y
- la figura 17 ilustra una implementación preferida para aplicar la información de conformación de la banda baja a la banda de frecuencia superior y la atenuación adicional de los valores espectrales conformados en dos etapas posteriores.

La figura 8 ilustra una realización preferida de un codificador de audio para codificar una señal de audio 403 que tiene una banda de frecuencia más baja y una banda de frecuencia superior. El codificador de audio comprende un detector 802 para detectar una región espectral del pico en la banda de frecuencia superior de la señal de audio 103. Además, el codificador de audio comprende un conformador 804 para conformar la banda de frecuencia más baja usando la información de conformación para la banda más baja y para conformar la banda de frecuencia superior usando al menos una porción de la información de conformación para la banda de frecuencia más baja. Además, el conformador se configura para atenuar adicionalmente valores espectrales en la región espectral del pico detectada en la banda de frecuencia superior.

En consecuencia, el conformador 804 realiza una clase de “conformación única” en la banda baja usando la información de conformación para la banda baja. Además, el conformador realiza adicionalmente una clase de una conformación “única” en la banda alta usando la información de conformación para la banda baja y normalmente, la banda baja de frecuencia más alta. Esta conformación “única” se realiza en algunas realizaciones en la banda alta donde no se ha detectado la región espectral del pico por el detector 802. Además, para la región espectral del pico dentro de la banda alta, una clase de una conformación “doble” se realiza, es decir, la información de conformación de la banda baja se aplica a la región espectral del pico y, adicionalmente, se aplica la atenuación adicional a la región espectral del pico.

El resultado del conformador 804 es una señal conformada 805. La señal conformada es una banda de frecuencia más baja conformada y una banda de frecuencia superior conformada, donde la banda de frecuencia superior conformada comprende la región espectral del pico. Esta señal conformada 805 se envía a una etapa de cuantificador y codificador 806 para cuantificar la banda de frecuencia más baja conformada y la banda de frecuencia superior conformada que incluye la región espectral del pico y para codificar por entropía los valores espectrales cuantificados de la banda de frecuencia más baja conformada y la banda de frecuencia superior conformada que comprende la región espectral del pico de nuevo para obtener la señal de audio codificada 814.

Preferiblemente, el codificador de audio comprende un analizador de codificación de predicción lineal 808 para derivar coeficientes de predicción lineal durante un marco de tiempo de la señal de audio mediante el análisis de un bloque de muestras de audio en el marco de tiempo. Preferiblemente, estas muestras de audio son de banda limitada para la banda de frecuencia más baja.

Adicionalmente, el conformador 804 se configura para conformar la banda de frecuencia más baja usando los coeficientes de predicción lineal como la información de conformación que se ilustra en 812 en la figura 8. Adicionalmente, el conformador 804 está configurado para usar al menos la porción de los coeficientes de predicción lineal derivados del bloque de muestras de audio de banda limitada para la banda de frecuencia más baja para conformar la banda de frecuencia superior en el marco de tiempo de la señal de audio.

Como se ilustra en la figura 9, la banda de frecuencia más baja preferiblemente está subdividida en una pluralidad de sub-bandas tales como, por ejemplo, cuatro sub-bandas SB1, SB2, SB3 y SB4. Además, tal como se ilustra esquemáticamente, el ancho de la sub-banda aumenta desde las sub-bandas inferiores hasta las superiores, es decir, la sub-banda SB4 es más amplia en frecuencia que la sub-banda SB1. En otras realizaciones, sin embargo, se pueden usar bandas que tienen un ancho de banda igual.

Las sub-bandas SB1 a SB4 se extienden hasta la frecuencia límite que es, por ejemplo, f_{CELP} . En consecuencia, todas las sub-bandas por debajo de la frecuencia límite f_{CELP} constituyen la banda más baja y el contenido de frecuencia por encima de la frecuencia límite constituye la banda más alta.

5 En particular, el analizador LPC 808 de la figura 8 normalmente calcula la información de conformación para cada sub-banda individualmente. En consecuencia, el analizador LPC 808 preferiblemente calcula cuatro clases diferentes de la información de sub-banda para las cuatro sub-bandas SB1 a SB4 de modo que cada sub-banda tiene su información de conformación asociada.

10 Además, la conformación se aplica mediante el conformador 804 para cada sub-banda SB1 a SB4 usando la información de conformación calculada para exactamente esta sub-banda y, de manera importante, también se realiza una conformación para la banda más alta, pero la información de conformación para la banda más alta no se calcula debido al hecho de que el analizador de predicción lineal que calcula la información de conformación recibe una banda de señal limitada de banda limitada a la banda de frecuencia más baja. No obstante, a fin de realizar una conformación para la banda de frecuencia mayor, la información de conformación para la sub-banda SB4 se usa para conformar la banda más alta. En consecuencia, el conformador 804 se configura para ponderar los coeficientes espectrales de la banda de frecuencia superior usando un factor de conformación calculado para una sub-banda más alta de la banda de frecuencia más baja. La sub-banda más alta correspondiente a SB4 en la figura 9 tiene una frecuencia central más alta entre todas las frecuencias centrales de las sub-bandas de la banda de frecuencia más baja.

20 La figura 11 ilustra un diagrama de flujo preferido para explicar la funcionalidad del detector 802. En particular, el detector 802 se configura para determinar una región espectral del pico en la banda de frecuencia superior, cuando al menos uno de un grupo de condiciones es válido, donde el grupo de condiciones comprende una condición de amplitud de banda baja 1102, una condición de distancia del pico 1104 y una condición de amplitud del pico 1106.

25 Preferiblemente, las diferentes condiciones se aplican exactamente en el orden ilustrado en la figura 11. En otras palabras, la condición de amplitud de banda baja 1102 se calcula antes de la condición de distancia del pico 1104, y la condición de distancia del pico se calcula antes de la condición de amplitud del pico 1106. En una situación en la que las tres condiciones deben ser verdaderas para detectar la región espectral de pico, se obtiene un detector eficiente desde el punto de vista computacional mediante la aplicación del procesamiento secuencial en la figura 11, donde, tan pronto como una cierta condición no es verdadera, es decir, es falsa, se detiene el proceso de detección para un cierto marco de tiempo y se determina que no se requiere una atenuación de una región espectral de pico en este marco de tiempo. Por lo tanto, cuando ya está determinado, durante un cierto marco de tiempo, que la condición de amplitud de banda baja 1102 no se cumple, es decir, es falsa, entonces el control procede a la decisión de que una atenuación de una región espectral de pico en este marco de tiempo no es necesaria y el procedimiento continúa sin ninguna atenuación adicional. Sin embargo, cuando el controlador determina para la condición 1102 que la misma es verdadera, se determina la segunda condición 1104. Esta configuración de distancia de pico se determina una vez más antes de la amplitud de pico 1106 de manera que el control determina que no se realiza la atenuación de la región espectral de pico, cuando la condición 1104 produce un resultado falso. Solo cuando la condición de distancia del pico 1104 tiene un resultado verdadero, se determina la tercera condición de amplitud de pico 1106.

40 En otras realizaciones, se pueden determinar más o menos condiciones, y se puede realizar una determinación secuencial o paralela, aunque la determinación secuencial ilustrada a modo de ejemplo en la figura 11 es preferible para ahorrar recursos computacionales que son particularmente valiosos en aplicaciones móviles que están alimentadas por baterías.

Las figuras 12, 13, 14 proporcionan realizaciones preferidas para las condiciones 1102, 1104 y 1106.

50 En la condición de amplitud de banda baja, se determina una amplitud espectral máxima en la banda más baja como se ilustra en el bloque 1202. Este valor es max_low . Además, en el bloque 1204, se determina una amplitud espectral máxima en la banda superior que se indica como max_high .

55 En el bloque 1206, los valores determinados de los bloques 1232 y 1234 se procesan preferiblemente juntos con un número predeterminado c_1 a fin de obtener el resultado falso o verdadero de condición 1102. Preferiblemente, las condiciones en los bloques 1202 y 1204 se realizan antes de la conformación con la información de conformación de la banda más baja, es decir, antes del procedimiento realizado por el conformador espectral 804 o, con respecto a la figura 10, 804a.

60 Con respecto al número predeterminado c_1 de la figura 12 usado en el bloque 1206, se prefiere un valor de 16, pero se ha comprobado que valores entre 4 y 30 también son útiles.

La figura 13 ilustra una realización preferida de la condición de distancia del pico. En el bloque 1302, se determina una primera amplitud espectral máxima en la banda más baja que se indica como max_low .

Además, se determina una primera distancia espectral como se ilustra en el bloque 1304. Esta primera distancia espectral se indica como `dist_low`. En particular, la primera distancia espectral es una distancia de la primera amplitud espectral máxima determinada por el bloque 1302 a partir de una frecuencia límite entre una frecuencia central de la banda de frecuencia más baja y una frecuencia central de la banda de frecuencia superior. Preferiblemente, la frecuencia límite es `f_celp`, pero esta frecuencia puede tener cualquier otro valor como se describió antes.

Además, el bloque 1306 determina una segunda amplitud espectral máxima en la banda superior que se llama `max_high`. Además, una segunda distancia espectral 1308 se determina e indica como `dist_high`. La segunda distancia espectral de la segunda amplitud espectral máxima de la frecuencia límite preferiblemente se determina una vez más con `f_celp` espectral como la frecuencia límite.

Además, en el bloque 1310, se determina si la condición de distancia del pico es verdadera, cuando la primera amplitud espectral máxima ponderada por la primera distancia espectral y ponderada por un número predeterminado que es mayor de 1 es mayor que la segunda amplitud espectral máxima ponderada por la segunda distancia espectral.

Preferiblemente, un número predeterminado c_2 es igual a 4 en la realización más preferida. Se ha probado que los valores entre 1,5 y 8 son útiles.

Preferiblemente, la determinación en el bloque 1302 y 1306 se realiza después de la conformación con la información de conformación de la banda más baja, es decir, posterior al bloque, pero, obviamente, antes del bloque 804b de la figura 10.

La figura 14 ilustra una implementación preferida de la condición de amplitud del pico. En particular, el bloque 402 determina una primera amplitud espectral máxima en la banda más baja y el bloque 1404 determina una segunda amplitud espectral máxima en la banda superior donde el resultado del bloque 1402 se indica como `max_low2` y el resultado del bloque 1404 se indica como `max_high`.

Luego, como se ilustra en el bloque 1406, la condición de amplitud del pico es verdadera, cuando la segunda amplitud espectral máxima es mayor que la primera amplitud espectral máxima ponderada por un número predeterminado c_3 que es mayor de o igual a 1. c_3 preferiblemente se ajusta a un valor de 1,5 o a un valor de 3 dependiendo de diferentes tasas donde, generalmente, se ha probado que los valores entre 1,0 y 5,0 son útiles.

Además, como se indica en la figura 14, la determinación en los bloques 1402 y 1404 tiene lugar después de la conformación con la información de conformación de banda baja, es decir, después del procesamiento ilustrado en el bloque 804a y antes del procesamiento ilustrado por el bloque 804b o, con respecto a la figura 17, después del bloque 1702 y antes del bloque 1704.

En otras realizaciones, la condición de amplitud del pico 1106 y, en particular, el procedimiento de la figura 14, bloque 1402 no se determina a partir del valor menor en la banda de frecuencia más baja, es decir, el valor de frecuencia más baja del espectro, sino que la determinación de la primera amplitud espectral máxima en la banda más baja se determina basándose en una porción de la banda más baja donde la porción se extiende desde una frecuencia de inicio predeterminada hasta una frecuencia máxima de la banda de frecuencia más baja, donde la frecuencia de inicio predeterminada es mayor de una frecuencia mínima de la banda de frecuencia más baja. En una realización, la frecuencia de inicio predeterminada es al menos 10% de la banda de frecuencia más baja por encima de la frecuencia mínima de la banda de frecuencia más baja o, en otras realizaciones, la frecuencia de inicio predeterminada está en una frecuencia que es igual a la mitad de una frecuencia máxima de la banda de frecuencia más baja dentro de un intervalo de tolerancia de más o menos 10% de la mitad de la frecuencia máxima.

Además, se prefiere que el tercer número predeterminado c_3 dependa de una tasa de bits que ha de proporcionarse por la etapa de cuantificador/codificador, de manera que el número predeterminado es mayor para una tasa de bits más alta. En otras palabras, cuando la tasa de bits que tiene que ser proporcionada por la etapa de cuantificador y codificador 806 es alta, entonces c_3 es alta, mientras que cuando la tasa de bits se debe determinar como baja, entonces el número predeterminado c_3 es bajo. Cuando se considera la ecuación preferida en el bloque 1406, queda claro que es el número predeterminado superior c_3 , la región espectral de pico se determina más raramente. Sin embargo, cuando c_3 es pequeño, entonces se determina con más frecuencia una región espectral de pico en la que hay valores espectrales para ser finalmente atenuados.

Los bloques 1202, 1204, 1402, 1404 o 1302 y 1306 siempre determinan una amplitud espectral. La determinación de la amplitud espectral se puede realizar de modo diferente. Una manera de determinación de la envolvente espectral es la determinación de un valor absoluto de un valor espectral del espectro real. De modo alternativo, la amplitud espectral puede ser una magnitud de un valor espectral complejo. En otras realizaciones, la amplitud espectral puede ser cualquier potencia del valor espectral del espectro real o cualquier potencia de una magnitud de un espectro complejo, donde la potencia es mayor de 1. Preferiblemente, la potencia es un número entero, pero se ha demostrado que las potencias de 1,5 o 2,5 adicionalmente son útiles. Preferiblemente, no obstante, se prefieren las potencias de

2 o 3.

En general, el conformador 804 se configura para atenuar al menos un valor espectral en la región espectral del pico detectada basándose en una amplitud espectral máxima en la banda de frecuencia superior y/o basándose en una amplitud espectral máxima en la banda de frecuencia más baja. En otras realizaciones, el conformador se configura para determinar la amplitud espectral máxima en una porción de la banda de frecuencia más baja, la porción que se extiende desde una frecuencia de inicio predeterminada de la banda de frecuencia más baja hasta una frecuencia máxima de la banda de frecuencia más baja. La frecuencia de inicio predeterminada es mayor de una frecuencia mínima de la banda de frecuencia más baja y preferiblemente es al menos 10% de la banda de frecuencia más baja por encima de la frecuencia mínima de la banda de frecuencia más baja o la frecuencia de inicio predeterminada preferiblemente está en la frecuencia que es igual a la mitad de una frecuencia máxima de la banda de frecuencia más baja dentro de una tolerancia de más o menos 10% de la mitad de la frecuencia máxima.

El conformador se configura además para determinar el factor de atenuación que determina la atenuación adicional, donde el factor de atenuación se deriva de la amplitud espectral máxima de la banda de frecuencia más baja multiplicada por un número predeterminado que es mayor de o igual a uno dividido por la amplitud espectral máxima de la banda de frecuencia superior. Para este fin, se hace referencia al bloque 1602 que ilustra la determinación de una amplitud espectral máxima de la banda más baja (preferiblemente después de la conformación, es decir, después del bloque 804a en la figura 10 o después del bloque 1702 de la figura 17).

Además, el conformador se configura para determinar la amplitud espectral máxima de la banda más alta, de nuevo preferiblemente después de la conformación, por ejemplo, como se realiza en el bloque 804a de la figura 10 o bloque 1702 en la figura 17. Entonces, en el bloque 1606, el factor de atenuación fac se calcula como se ilustra, donde el número predeterminado c_3 se ajusta para ser mayor de o igual a 1. En realizaciones, c_3 de la figura 16 es el mismo número predeterminado c_3 que en la figura 14. Sin embargo, en otras realizaciones, c_3 de la figura 16 se puede ajustar diferente de c_3 en la figura 14. Adicionalmente, c_3 en la figura 16 que influye directamente el factor de atenuación también es dependiente de la tasa de bits de modo que se fija un número predeterminado más alto c_3 para que una tasa de bits más alta sea realizada por la etapa de cuantificador/codificador 806 como se ilustra en la figura 8.

La figura 17 ilustra una implementación preferida similar a la que se muestra en la figura 10 en los bloques 804a y 804b, es decir, que se realiza una conformación con la información de ganancia de banda baja aplicada a los valores espectrales por encima de la frecuencia límite tal como f_{CELP} a fin de obtener valores espectrales conformados por encima de la frecuencia límite y adicionalmente en una etapa siguiente 1704, el factor de atenuación fac calculado por el bloque 1606 de la figura 16 se aplica en el bloque 1704 de la figura 17. En consecuencia, la figura 17 y la figura 10 ilustran una situación donde el conformador se configura para conformar los valores espectrales de la región espectral detectada basándose en una primera operación de ponderación usando una porción de la información de conformación para la banda de frecuencia más baja y una segunda operación de ponderación posterior usando una información de atenuación, es decir, el ejemplo del factor de atenuación fac .

En otras realizaciones, sin embargo, el orden de las etapas de la figura 17 se invierte de modo que la primera operación de ponderación tiene lugar usando la información de atenuación y la segunda operación de ponderación posterior tiene lugar usando al menos una porción de la información de conformación para la banda de frecuencia más baja. O, de modo alternativo, la conformación se realiza usando una operación de ponderación única usando una información de ponderación combinada que depende y se deriva de la información de atenuación por una parte y al menos una porción de la información de conformación para la banda de frecuencia más baja por la otra parte.

Como se ilustra en la figura 17, la información de atenuación adicional se aplica a todos los valores espectrales en la región espectral del pico detectada. De forma alternativa, el factor de atenuación solo se aplica a, por ejemplo, el valor espectral más alto o el grupo de valores espectrales más altos, donde los miembros del grupo pueden oscilar entre 2 y 10, por ejemplo. Además, las realizaciones también aplican el factor de atenuación a todos los valores espectrales de la banda de frecuencia superior para los cuales se ha detectado la región espectral del pico mediante el detector para un marco de tiempo de la señal de audio. En consecuencia, en esta realización, se aplica el mismo factor de atenuación a la banda de frecuencia superior entera cuando solo se ha determinado un valor espectral único como una región espectral del pico.

Cuando, durante un cierto marco, no se ha detectado la región espectral del pico, entonces la banda de frecuencia más baja y la banda de frecuencia superior se conforman mediante el conformador sin atenuación adicional. De este modo, se realiza una conmutación del marco de tiempo al marco de tiempo, donde, según la implementación, se prefiere algún tipo de suavizado de la información de atenuación.

Preferiblemente, la etapa del cuantificador y codificador comprende un procesador del bucle de velocidad como se ilustra en la figura 15a y la figura 15b. En una realización, la etapa de cuantificador y codificador 806 comprende un ponderador de ganancia global 1502, un cuantificador 1504 y un codificador de entropía tal como un codificador aritmético o de Huffman 1506. Además, el codificador de entropía 1506 proporciona, para un conjunto determinado de

valores cuantificados para un marco de tiempo, una tasa de bits estimada o medida a un controlador 1508.

El controlador 1508 está configurado para recibir un criterio de terminación de bucle por una parte y/o una información de tasa de bits predeterminada por otra parte. Tan pronto como el controlador 1508 determina que no se obtiene una tasa de bits predeterminada y/o no se cumple un criterio de terminación, entonces el controlador proporciona una ganancia global ajustada al ponderador de ganancia global 1502. Entonces, el ponderador de ganancia global aplica la ganancia global ajustada a las líneas espectrales conformadas y atenuadas de un marco de tiempo. La salida ponderada de ganancia global del bloque 1502 se proporciona al cuantificador 1504 y el resultado cuantificado se proporciona al codificador de entropía 1506 que determina una vez más una tasa de bits estimada o medida para los datos ponderados con la ganancia global ajustada. En el caso de que se cumpla el criterio de terminación y/o se cumpla la tasa de bits predeterminada, entonces la señal de audio codificada se emite en la línea de salida 814. Sin embargo, cuando no se obtiene la tasa de bits predeterminada o no se cumple un criterio de terminación, entonces comienza de nuevo el bucle. Esto se ilustra con más detalle en la figura 15b.

Cuando el controlador 1508 determina que la tasa de bits es demasiado alta como se ilustra en el bloque 1510, entonces aumenta una ganancia global como se ilustra en el bloque 1512. En consecuencia, todas las líneas espectrales conformadas y atenuadas son más pequeñas debido que se dividen por la ganancia global aumentada y el cuantificador luego cuantifica los valores espectrales menores de modo que el codificador de entropía genera un número menor de bits requeridos para este marco de tiempo. En consecuencia, los procedimientos de ponderación, cuantificación y codificación se realizan con la ganancia global ajustada como se ilustra en el bloque 1514 en la figura 15b, y entonces una vez más se determina si la tasa de bits es demasiado alta. Si la tasa de bits es todavía demasiado alta, entonces se realizan de nuevo los bloques 1512 y 1514. Cuando, sin embargo, se determina que la tasa de bits no es demasiado alta, el control procede a la etapa 1516 que describe si se cumple un criterio de terminación. Cuando se cumple el criterio de terminación, el bucle de velocidad se detiene y la ganancia global final se introduce adicionalmente en la señal codificada a través de una interfaz de salida tal como la interfaz de salida 1014 de la figura 10.

Sin embargo, cuando se determina que el criterio de terminación no se cumple, entonces se disminuye la ganancia global como se ilustra en el bloque 1518, de manera que, al final, se utiliza la tasa de bits máxima permitida. Esto asegura que los marcos de tiempo que son fáciles de codificar están codificados con una mayor precisión, es decir, con menos pérdida. Por lo tanto, para tales casos, la ganancia global disminuye tal como se ilustra en el bloque 1518 y la etapa 1514 se realiza con la ganancia global disminuida y se realiza la etapa 1510 a fin de ver si la tasa de bits resultante es demasiado alta o no.

Naturalmente, la implementación específica en relación con el aumento de la ganancia global o la disminución del incremento se puede ajustar según sea necesario. Adicionalmente, el controlador 1508 se puede implementar para tener bloques 1510, 1512 y 1514 o para tener bloques 1510, 1516, 1518 y 1514. De este modo, según la implementación, y también según el valor de partida para la ganancia global, el procedimiento puede ser tal que se inicia desde una ganancia global muy alta hasta que se encuentra la ganancia global más baja que todavía cumple los requisitos de la tasa de bits. Por otra parte, el procedimiento se puede realizar de tal manera que se inicia a partir de una ganancia global relativamente baja y la ganancia global se incrementa hasta que se obtiene una tasa de bits permisible. Adicionalmente, como se ilustra en la figura 15b, incluso se puede aplicar también una mezcla entre ambos procedimientos.

La figura 10 ilustra la incrustación del codificador de audio de la invención que consiste en los bloques 802, 804a, 804b y 806 dentro de un escenario de codificador de dominio de frecuencia/dominio de tiempo conmutado.

En particular, el codificador de audio comprende un procesador común. El procesador común consiste en un controlador ACELP/TCX 1004 y el limitador de banda, tal como un remuestreador 1006 y un analizador LPC 808. Esto se ilustra mediante las cajas sombreadas indicadas por 1002.

Además, el limitador de banda alimenta el analizador de LPC que ya se ha discutido con respecto a la figura 8. A continuación, la información de conformación LPC generada por el analizador LPC 808 se envía a un codificador CELP 1008 y la salida del codificador CELP 1008 se introduce en una interfaz de salida 1014 que genera la señal finalmente codificada 1020. Además, la rama de codificación de dominio de tiempo que consiste en el codificador 1008 comprende adicionalmente un codificador de extensión de ancho de banda de dominio del tiempo 1010 que proporciona información y, típicamente, información paramétrica tal como información de la envolvente espectral para al menos la banda alta de la entrada de señal de audio de banda completa en la entrada 1001. Preferiblemente, la banda alta procesada por el codificador de extensión del ancho de banda de dominio de tiempo 1010 es una banda que comienza en la frecuencia límite que también es usada por el limitador de banda 1006. De este modo, el limitador de banda realiza un filtrado de paso bajo para obtener la banda más baja y la banda alta filtrada por el limitador de banda de paso bajo 1006 es procesada por el codificador de extensión del ancho de banda de dominio de tiempo 1010.

Por otra parte, el dominio espectral o ramificación de codificación TCX comprende un convertidor del espectro de

tiempo 1012 y, a modo de ejemplo, un enmascaramiento tonal como se ha discutido anteriormente con el fin de obtener un procesamiento del codificador de llenado de intervalo.

A continuación, el resultado del convertidor de espectro de tiempo 1012 y el procesamiento de enmascaramiento tonal opcional adicional se introducen en un conformador espectral 804a y el resultado del conformador espectral 804a se introduce en un atenuador 804b. El atenuador 804b está controlado por el detector 802 que realiza una detección usando los datos del dominio del tiempo o usando la salida del bloque del convertidor del espectro de tiempo 1012 como se ilustra en 1022. Los bloques 804a y 804b implementan juntos el conformador 804 de la figura 8 como se ha discutido anteriormente. El resultado del bloque 804 se introduce en la etapa de cuantificador y codificador 806, es decir, en una cierta realización, controlada por una tasa de bits predeterminada. Adicionalmente, cuando los números predeterminados aplicados por el detector también dependen de la tasa de bits predeterminada, entonces la tasa de bits predeterminada también se introduce en el detector 802 (no mostrado en la figura 10).

En consecuencia, la señal codificada 1020 recibe datos de la etapa de cuantificador y codificador, información de control del controlador 1004, información del codificador CELP 1008 e información del codificador de extensión del ancho de banda de dominio de tiempo 1010.

Posteriormente, las realizaciones preferidas de la presente invención se discuten incluso en más detalle.

Una opción que ahorra interoperabilidad y compatibilidad con versiones anteriores a las implementaciones existentes es realizar un preprocesamiento del lado del codificador. El algoritmo, como se explica posteriormente, analiza el espectro MDCT. En caso de que los componentes de señal significativos inferiores a f_{CELP} estén presentes y se encuentren picos altos por encima de f_{CELP} , que potencialmente destruyen la codificación del espectro completo en el bucle de velocidad, estos picos por encima de f_{CELP} se atenúan. Aunque la atenuación no se puede revertir en el lado del decodificador, la señal decodificada resultante es más agradable de forma perceptualmente significativa que antes, en donde grandes partes del espectro fueron eliminadas por completo.

La atenuación reduce el foco del bucle de velocidad sobre los picos por encima de f_{CELP} y permite que los coeficientes de MDCT de baja frecuencia significativa sobrevivan al bucle de velocidad.

El siguiente algoritmo describe el preprocesamiento del lado del codificador:

1) Detección del contenido de la banda baja (por ejemplo, 1102):

La detección del contenido de banda baja analiza si están presentes porciones significativas de señal de banda baja. Para esto, se buscan la amplitud máxima del espectro MDCT por debajo y por encima de f_{CELP} en el espectro MDCT antes de la aplicación de ganancias de conformación LPC inversa. El procedimiento de búsqueda devuelve los siguientes valores:

a) max_low_pre : el coeficiente MDCT máximo por debajo de f_{CELP} , evaluado en el espectro de valores absolutos antes de la aplicación de ganancias de conformación LPC inversa

b) max_high_pre : el coeficiente MDCT máximo por encima de f_{CELP} , evaluado en el espectro de valores absolutos antes de la aplicación de ganancias de conformación inversa LPC. Para la decisión, se evalúa la siguiente condición:

$$\text{Condición 1: } c_1 * \text{max_low_pre} > \text{max_high_pre}$$

Si la Condición 1 es verdadera, se asume una cantidad significativa de contenido de banda baja y continúa el preprocesamiento; si la Condición 1 es falsa, el preprocesamiento se interrumpe. Esto asegura que no se aplica ningún daño a señales de banda alta solamente, por ejemplo, un barrido senoidal cuando está por encima de f_{CELP} .

Pseudo-código:

```
max_low_pre = 0;
para (i=0; i<LTCX(CELP); i++)
{
    tmp = fabs(XM(i));
    si (tmp > max_low_pre)
    {
        max_low_pre = tmp;
    }
}
```

```

max_high_pre = 0;
para (i=0; i<LTCX(BW) - LTCX(CELP); i++)
{
    tmp = fabs(XM(LTCX(CELP) + i));
    si (tmp > max_high_pre)
    {
        max_high_pre = tmp;
    }
}

si (c1 * max_low_pre > max_high_pre)
{
    /* continuar con el preprocesamiento */
    ...
}

```

donde

X_M es el espectro MDCT antes de la aplicación de la conformación de ganancia LPC inversa,

$L_{TCX}^{(CELP)}$ es el número de coeficientes de MCDT hasta f_{CELP}

$L_{TCX}^{(BW)}$ es el número de coeficientes de MCDT para el espectro MDCT completo.

En un ejemplo de implementación c_1 se ajusta a 16, y fabs retorna al valor absoluto.

2) Evaluación de métrica pico-distancia (por ejemplo, 1104):

Una métrica de pico-distancia analiza el impacto de los picos espectrales por encima de f_{CELP} en el codificador aritmético. Por lo tanto, la amplitud máxima del espectro MDCT por debajo y por encima de f_{CELP} se busca en el espectro MDCT después de la aplicación de ganancias de conformación LPC inversa, es decir, en el dominio donde también se aplica el codificador aritmético. Además de la amplitud máxima, también se evalúa la distancia desde f_{CELP} . El procedimiento de búsqueda devuelve los siguientes valores:

a) max_low: el coeficiente MCDT máximo por debajo de f_{CELP} , evaluado en el espectro de valores absolutos después de la aplicación de ganancias de conformación de LPC inversa

b) dist_low: la distancia de max_low de f_{CELP}

c) max_high: el coeficiente MCDT máximo por encima de f_{CELP} , evaluado en el espectro de valores absolutos después de la aplicación de las ganancias de conformación de LPC inversa

d) dist_high: la distancia de max_high de f_{CELP}

Para la decisión, se evalúa la siguiente condición:

Condición 2: $c_2 * dist_high * max_high > dist_low * max_low$

Si la Condición 2 es verdadera, se asume una tensión significativa para el codificador aritmético, debido a un pico espectral muy alto o una alta frecuencia de este pico. El pico alto dominará el proceso de codificación en el bucle de velocidad, la alta frecuencia penalizará al codificador aritmético, ya que el codificador aritmético siempre se ejecuta de frecuencias bajas a altas, es decir, las frecuencias más altas son ineficientes para codificar. Si la Condición 2 es verdadera, se continúa con el preprocesamiento. Si la Condición 2 es falsa, el preprocesamiento se interrumpe.

```

max_low = 0;
dist_low = 0;
para (i=0; i<LTCX(CELP); i++)
{
    tmp = fabs(XM(LTCX(CELP) - 1-i));
    si (tmp > max_low)
    {
        max_low = tmp;
    }
}

```

```

        dist_low = i;
    }
}

5   max_high = 0;
    dist_high = 0;
    para (i=0; i<LTCX(BW) - LTCX(CELP); i++)
    {
        tmp = fabs( $\tilde{X}_M$ (LTCX(CELP) + i));
10   si (tmp > max_high)
        {
            max_high = tmp;
            dist_high = i;
        }
15   }

    si (c2 * dist_high * max_high > dist_low * max_low)
    {
        /* continuar con el preprocesamiento */
20   ...
    }

```

donde

$\tilde{X}_M$ es el espectro MDCT después de la aplicación de la conformación de ganancia LPC inversa,

L_{TCX}^(CELP) es el número de coeficientes de MCDT hasta f_{CELP}

L_{TCX}^(BW) es el número de coeficientes de MCDT para el espectro MDCT completo

En un ejemplo de implementación c₂ se ajusta a 4.

3) Comparación de la amplitud del pico (por ejemplo, 1106):

Finalmente, se comparan las amplitudes de pico en regiones espectrales psicoacústicamente similares. Por lo tanto, la amplitud máxima del espectro MDCT por debajo y por encima de f_{CELP} se busca en el espectro MDCT después de la aplicación de ganancias de conformación LPC inversa. La amplitud máxima del espectro MDCT por debajo de f_{CELP} no se busca en todo el espectro, sino que solo comienza con f_{bajo} > 0 Hz. Esto es para descartar las frecuencias más bajas, que son psicoacústicamente más importantes y usualmente tienen la mayor amplitud después de la aplicación de ganancias de conformación LPC inversa, y solo para comparar componentes con una importancia psicoacústica similar. El procedimiento de búsqueda devuelve los siguientes valores:

a) max_low2: el coeficiente MCDT máximo por debajo de f_{CELP}, evaluado en el espectro de valores absolutos después de la aplicación de las ganancias de conformación de LPC inversa a partir de f_{bajo}

b) max_high: el coeficiente MCDT máximo por encima de f_{CELP}, evaluado en el espectro de valores absolutos después de la aplicación de las ganancias de conformación de LPC inversa

Para la decisión, se evalúa la siguiente condición:

Condición 3: max_high > c₃ * max_low2

Si la condición 3 es verdadera, se asumen coeficientes espectrales por encima de f_{CELP}, que tienen amplitudes significativamente mayores que exactamente por debajo de f_{CELP}, y que se asumen costosas de codificar. La constante c₃ define una ganancia máxima, que es un parámetro de ajuste. Si la Condición 2 es verdadera, se continúa con el preprocesamiento. Si la Condición 2 es falsa, el preprocesamiento se interrumpe.

Pseudo-código:

```

max_low2 = 0;
para (i=Llow; i<LTCX(CELP); i++)

```

```

{
    tmp = fabs( $\tilde{X}_M(i)$ );
    si (tmp > max_low2)
    {
5         max_low2 = tmp;
    }
}

max_high = 0;
10 para (i=0; i<LTCX(BW) - LTCX(CELP); i++)
{
    tmp = fabs( $\tilde{X}_M(L_{TCX}^{(CELP)} + i)$ );
    si (tmp > max_high)
    {
15         max_high = tmp;
    }
}

20 si (max_high > c3 * max_low2)
{
    /* continuar con el preprocesamiento */
    ...
}

```

25 donde

L_{bajo} es una compensación correspondiente para f_{bajo}

30 $\tilde{X}_M$ es el espectro MDCT después de la aplicación de la conformación de ganancia LPC inversa,

$L_{TCX}^{(CELP)}$ es el número de coeficientes de MCDT hasta f_{CELP}

35 $L_{TCX}^{(BW)}$ es el número de coeficientes de MCDT para el espectro MDCT completo

En un ejemplo de implementación f_{bajo} se ajusta a $L_{TCX}^{(CELP)}/2$. En un ejemplo de implementación c_3 se ajusta a 1,5 para las tasas de bits bajas y se ajusta a 3,0 para las tasas de bits altas.

40 4) Atenuación de picos altos por encima de f_{CELP} (por ejemplo, figuras 16 y 17):

Si se halla que la condición 1–3 es verdadera, se aplica una atenuación de los picos por encima de f_{CELP} . La atenuación permite una ganancia máxima c_3 en comparación con una región espectral psicoacústicamente similar. El factor de atenuación se calcula de la siguiente manera:

45
$$\text{attenuation_factor} = c_3 * \text{max_low2}/\text{max_high}$$

El factor de atenuación posteriormente se aplica a todos los coeficientes de MCDT por encima de f_{CELP} .

50 5) Pseudo-código:

```

si ( (c1 * max_low_pre > max_high_pre) &&
    (c2 * dist_high * max_high > dist_low * max_low) &&
    (max_high > c3 * max_low2)
55 )
{
    fac = c3 * max_low2/max_high;

    para (i = LTCX(CELP); i< LTCX(BW); i++)
    {
60          $\tilde{X}_M(i) = \tilde{X}_M(i) * \text{fac};$ 
    }
}

```

}

donde

5 X_M es el espectro MDCT después de la aplicación de la conformación de ganancia LPC inversa,

$L_{TCX}^{(CELP)}$ es el número de coeficientes de MCDT hasta f_{CELP}

10 $L_{TCX}^{(BW)}$ es el número de coeficientes de MCDT para el espectro MDCT completo

El preprocesamiento del lado del codificador reduce significativamente la tensión para el bucle de codificación mientras que aún se mantienen los coeficientes espectrales relevantes por encima de f_{CELP} .

15 La figura 7 ilustra un espectro de MDCT de una trama crítica después de la aplicación de ganancias de conformación de LPC inversa y el preprocesamiento del lado del codificador anteriormente descritos. Según los valores numéricos elegidos para c_1 , c_2 y c_3 , el espectro resultante, que posteriormente se introduce en el bucle de velocidad, podría tener el aspecto anterior. Se reducen significativamente, pero todavía es probable que sobrevivan al bucle de velocidad, sin consumir todos los bits disponibles.

20 Aunque algunos aspectos se han descrito en el contexto de un aparato, está claro que estos aspectos también representan una descripción del método correspondiente, en el que un bloque o dispositivo corresponde a una etapa del método o una característica de una etapa del método. Análogamente, los aspectos descritos en el contexto de una etapa del método también representan una descripción de un bloque o elemento o característica correspondiente de un aparato correspondiente. Algunas o todas las etapas del método pueden ser ejecutados por (o usando) un aparato

25 de hardware, como por ejemplo, un microprocesador, un ordenador programable o un circuito electrónico. En algunas realizaciones, una o más de las etapas del método más importantes se pueden ejecutar mediante tal aparato.

La señal de audio codificada de la invención se puede almacenar en un medio de almacenamiento digital o se puede transmitir en un medio de transmisión tal como un medio de transmisión inalámbrico o un medio de transmisión

30 cableado, tal como Internet.

Dependiendo de ciertos requisitos de implementación, las realizaciones de la invención se pueden implementar en hardware o en software. La implementación se puede realizar usando un medio de almacenamiento no transitorio o un medio de almacenamiento digital, por ejemplo, un disquete, un DVD, un Blu-Ray, un CD, una ROM, una PROM,

35 una EPROM, una EEPROM o una memoria FLASH, que tienen señales de control legibles electrónicamente almacenadas en ellas, que actúan conjuntamente (o son capaces de actuar conjuntamente) con un sistema informático programable de tal manera que se realiza el método respectivo. Por lo tanto, el medio de almacenamiento digital puede ser legible por ordenador.

40 Algunas realizaciones según la invención comprenden un soporte de datos que tiene señales de control legibles electrónicamente, que son capaces de actuar conjuntamente con un sistema informático programable, de manera que se lleva a cabo uno de los métodos descritos en el presente documento.

45 Generalmente, las realizaciones de la presente invención se pueden implementar como un producto de programa informático con un código de programa, estando el código de programa operativo para realizar uno de los métodos cuando el producto de programa informático se ejecuta en un ordenador. El código de programa, por ejemplo, se puede almacenar en un soporte legible por máquina.

Otras realizaciones comprenden el programa informático para realizar uno de los métodos descritos en el presente documento, almacenados en un soporte legible por máquina.

50

En otras palabras, una realización del método de la invención es, por lo tanto, un programa informático que tiene un código de programa para realizar uno de los métodos descritos en el presente documento, cuando el programa informático se ejecuta en un ordenador.

55

Una realización adicional de los métodos de la invención es, por lo tanto, un soporte de datos (o un medio de almacenamiento digital, o un medio legible por ordenador) que comprende, grabado en el mismo, el programa informático para realizar uno de los métodos descritos en el presente documento. El soporte de datos, el medio de almacenamiento digital o el medio grabado normalmente son tangibles y/o no transitorios.

60

Una realización adicional del método de la invención es, por lo tanto, un flujo de datos o una secuencia de señales que representan el programa informático para llevar a cabo uno de los métodos descritos en el presente documento. El flujo de datos o la secuencia de señales, por ejemplo, se puede configurar para transferirse a través de una conexión de comunicación de datos, por ejemplo, a través de Internet.

Una realización adicional comprende un medio de procesamiento, por ejemplo, un ordenador, o un dispositivo lógico programable, configurado o adaptado para realizar uno de los métodos descritos en el presente documento.

- 5 Una realización adicional comprende un ordenador que tiene instalado en el mismo el programa informático para realizar uno de los métodos descritos en el presente documento.

- 10 Una realización adicional según la invención comprende un aparato o un sistema configurado para transferir (por ejemplo, de forma electrónica u óptica) un programa informático para realizar uno de los métodos descritos en el presente documento a un receptor. El receptor puede ser, por ejemplo, un ordenador, un dispositivo móvil, un dispositivo de memoria o similares. El aparato o sistema puede comprender, por ejemplo, un servidor de archivos para transferir el programa informático al receptor.

- 15 En algunas realizaciones, se puede usar un dispositivo lógico programable (por ejemplo, una matriz de puertas de campo programable) para realizar algunas o todas las funcionalidades de los métodos descritos en el presente documento. En algunas realizaciones, una matriz de puertas de campo programable puede actuar conjuntamente con un microprocesador con el fin de realizar uno de los métodos descritos en el presente documento. Generalmente, los métodos se realizan preferiblemente por cualquier aparato de hardware.

- 20 El aparato descrito en el presente documento se puede implementar usando un aparato de hardware, o usando un ordenador, o usando una combinación de un aparato de hardware y un ordenador.

- 25 El aparato descrito en el presente documento, o cualquiera de los componentes del aparato descrito en el presente documento, se puede implementar al menos parcialmente en hardware y/o en software.

- Los métodos descritos en el presente documento se pueden realizar usando un aparato de hardware, o usando un ordenador, o usando una combinación de un aparato de hardware y un ordenador.

- 30 Los métodos descritos en el presente documento, o cualquiera de los componentes del aparato descrito en el presente documento, se pueden realizar al menos parcialmente por hardware y/o por software.

- 35 Las realizaciones descritas anteriormente son meramente ilustrativas para los principios de la presente invención. Se entiende que las modificaciones y variaciones de las disposiciones y los detalles descritos en el presente documento serán evidentes para los expertos en la técnica. Por lo tanto, la intención está limitada solo por el alcance de las reivindicaciones de patente inminentes y no por los detalles específicos presentados a modo de descripción y explicación de las realizaciones en el presente documento.

- 40 En la descripción anterior, se puede observar que varias características se agrupan juntas en realizaciones con el propósito de racionalizar la divulgación. Este método de divulgación no se debe interpretar como reflejo de una intención de que las realizaciones reivindicadas requieran más características que las que se mencionan expresamente en cada reivindicación. Más bien, como las siguientes reivindicaciones reflejan, el contenido de la invención se puede hallar en menos de todas las características de una única realización dada a conocer. Por lo tanto, las siguientes reivindicaciones se incorporan por el presente documento a la descripción detallada, donde cada reivindicación puede estar por sí sola como una realización separada. Si bien cada reivindicación puede estar por sí sola como una realización separada, se debe observar que – aunque una reivindicación dependiente se puede referir en las reivindicaciones a una combinación específica con una o más reivindicaciones adicionales – otras realizaciones también pueden incluir una combinación de la reivindicación dependiente con el contenido de cada reivindicación dependiente adicional o una combinación de cada característica con otras reivindicaciones dependientes o independientes. En el presente documento se proponen tales combinaciones a menos que se indique que no se pretende una combinación específica. Además, se pretende incluir también las características de una reivindicación a cualquier otra reivindicación independiente, incluso si esta reivindicación no se hace directamente dependiente de la reivindicación independiente.

- 55 Además, se debe observar que los métodos dados a conocer en la memoria descriptiva o en las reivindicaciones pueden implementarse mediante un dispositivo que tiene medios para realizar cada una de las etapas respectivas de estos métodos.

- 60 Además, en algunas realizaciones, una etapa única puede incluir o se puede descomponer en subetapas múltiples. Dichas subetapas se pueden incluir y formar parte de la divulgación de esta etapa única a menos que se excluyan explícitamente.

Referencias

[1] 3GPP TS 26.445 – Codec for Enhanced Voice Services (EVS); Detailed algorithmic description

Anexo

Posteriormente, se indican las porciones de la versión estándar anterior 13 (3GPP TS 26.445 – Codec for Enhanced Voice Services (EVS); Detailed algorithmic description). La sección 5.3.3.2.3 describe una realización preferida del conformador, la sección 5.3.3.2.7 describe una realización preferida del cuantificador de la etapa de cuantificador y codificador, y la sección 5.3.3.2.8 describe un codificador aritmético en una realización preferida del codificador en la etapa de cuantificador y codificador, en el que el bucle de velocidad preferido para la tasa de bits constante y la ganancia global se describe en la sección 5.3.2.8.1.2. Las características IGF de la realización preferida se describen en la sección 5.3.3.2.11., donde se hace referencia específica a la sección 5.3.3.2.11.5.1 Cálculo del enmascaramiento tonal IGF.

5.3.3.2.3 Conformación de LPC en el dominio MDCT

5.3.3.2.3.1 Principio general

La conformación LPC se realiza en el dominio MDCT mediante la aplicación de factores de ganancia calculados a partir de coeficientes de filtro LP cuantificados ponderados al espectro MDCT. La tasa de muestreo de entrada sr_{inp} , en la que se basa la transformada MDCT, puede ser mayor que la tasa de muestreo CELP sr_{celp} , para la que se calculan los coeficientes LP. Por lo tanto, las ganancias de conformación de LPC solo se pueden calcular para la parte del espectro MDCT correspondiente al intervalo de frecuencias CELP. Para la parte restante del espectro (si existe) se utiliza la ganancia de conformación de la banda de frecuencia más alta.

5.3.3.2.3.2 Cálculo de las ganancias de conformación de LPC

Para calcular las 64 ganancias de conformación de LPC los coeficientes de filtro LP ponderados $\tilde{a}$ primero se transforman en el dominio de frecuencia usando DFT apilado de forma extraña de longitud 128:

$$X_{LPC}(b) = \sum_{i=0}^{16} \tilde{a}(i) e^{-j \frac{\pi}{128} (2b+1)i} \quad (1)$$

Las ganancias de conformación de LPC g_{LPC} luego se calculan como los valores absolutos recíprocos de X_{LPC} :

$$g_{LPC}(b) = \frac{1}{|X_{LPC}(b)|}, \quad b = 0 \dots 63 \quad (2)$$

5.3.3.2.3.3 Aplicación de las ganancias de conformación de LPC al espectro MDCT

Los coeficientes de MDCT X_M correspondientes al intervalo de frecuencia de CELP se agrupan en 64 sub-bandas. Los coeficientes de cada sub-banda se multiplican por la recíproca de la correspondiente ganancia de conformación de LPC para obtener el espectro conformado $\tilde{X}_M$. Si el número de intervalos de MDCT correspondiente al intervalo de frecuencia de CELP $L_{TCX}^{(celp)}$ no es un múltiplo de 64, el ancho de las sub-bandas varía en un intervalo tal como se define por el siguiente pseudo-código:

$$w = \lfloor L_{TCX}^{(celp)} / 64 \rfloor, \quad r = L_{TCX}^{(celp)} - 64w$$

si $r = 0$ entonces

$$s = 1, \quad w_1 = w_2 = w$$

además si $r \leq 32$ entonces

$$s = \lfloor 64/r \rfloor, \quad w_1 = w, \quad w_2 = w+1$$

además

$$s = \lfloor 64/(64-r) \rfloor, \quad w_1 = w+1, \quad w_2 = w$$

$$i = 0$$

```

    para  $j = 0, \dots, 63$ 
    {
        si  $j \bmod s \neq 0$  entonces
             $w = w_1$ 
5      además
             $w = w_2$ 

        para  $l = 0, \dots, \min(w, L_{TCX}^{(celp)} - i) - 1$ 
        {
10       $\tilde{X}_M(i) = X_M(i) / g_{LPC}(j)$ 
             $i = i + 1$ 
        }
    }

```

- 15 Los restantes coeficientes de MCDT por encima del intervalo de frecuencia de CELP (si hubiera) se multiplican por la recíproca de la última ganancia de conformación de LPC:

$$\tilde{X}_M(i) = \frac{X_M(i)}{g_{LPC}(63)}, \quad i = L_{TCX}^{(celp)} \dots L_{TCX}^{(bw)} - 1 \quad (3)$$

20 5.3.3.2.4 Acentuación de baja frecuencia adaptativo

5.3.3.2.4.1 Principio general

- 25 El propósito de los procedimientos de desacentuación y acentuación de baja frecuencia adaptativos (ALFE) es mejorar el rendimiento subjetivo del códec TCX del dominio de frecuencia a bajas frecuencias. Con este fin, las líneas espectrales MDCT de baja frecuencia se amplifican antes de la cuantificación en el codificador, de este modo aumenta su SNR de cuantificación, y este refuerzo se deshace antes del procedimiento MDCT inverso en los decodificadores internos y externos para evitar artefactos de amplificación.

- 30 Hay dos algoritmos ALFE diferentes que se seleccionan consistentemente en codificador y decodificador basándose en la elección del algoritmo de codificación aritmética y la tasa de bits. El algoritmo ALFE 1 se utiliza a 9,6 kbps (codificador aritmético basado en envolvente) y a 48 kbps y superior (codificador aritmético basado en contexto). El algoritmo ALFE 2 se utiliza desde 13,2 hasta incl. 32 kbps. En el codificador, el ALFE funciona en las líneas espectrales en el vector $x[]$ directamente antes (algoritmo 1) o después (algoritmo 2) de cada cuantificación MDCT, que se ejecuta 35 varias veces dentro de un bucle de velocidad en el caso del codificador aritmético basado en contexto (ver la subcláusula 5.3.3.2.8.1).

5.3.3.2.4.2 Algoritmo de acentuación adaptativo 1

- 40 El algoritmo ALFE 1 funciona basándose en las ganancias de la banda de frecuencia LPC, $lpcGains[]$. Primero, el mínimo y el máximo de las nueve primeras ganancias –las ganancias de baja frecuencia (LF)– se encuentran usando operaciones de comparación ejecutadas dentro de un bucle sobre los índices de ganancia de 0 a 8.

- 45 Entonces, si la relación entre el mínimo y el máximo supera un umbral de $1/32$, se realiza un refuerzo gradual de las líneas más bajas en x , de modo que la primera línea (DC) se amplifica por $(32 \min/\max)^{0.25}$ y la 33ª línea no se amplifica:

```

    tmp = 32 * min
    si ((max < tmp) && (max > 0))
    {
50      fac = tmp = pow(tmp/max, 1/128)
      para (i = 31; i >= 0; i--)
      {
          /* refuerzo gradual de 32 líneas más bajas */
          x[i] *= fac
55      fac *= tmp
      }
    }

```

5.3.3.2.4.3 Algoritmo de acentuación adaptativo 2

El algoritmo ALFE 2, a diferencia del algoritmo 1, no funciona basándose en las ganancias LPC transmitidas, sino que se señaliza mediante modificaciones a las líneas MDCT de frecuencia baja (LF) cuantificadas. El procedimiento se divide en cinco etapas consecutivas:

- 5 • Etapa 1: primero hallar la primera magnitud máxima en index i_max en el cuarto espectral inferior ($k = 0 \dots L_{TCX}^{(bw)} / 4$) que utiliza $invGain = 2/g_{TCX}$ y modifica el máximo: $xq[i_max] += (xq[i_max] < 0) ? -2 : 2$
- 10 • Etapa 2: luego comprimir el intervalo de valores del total de $x[i]$ hasta i_max mediante la recuantificación de todas las líneas en $k = 0 \dots i_max-1$ como en la subcláusula que describe la cuantificación pero que usa $invGain$ en vez de g_{TCX} como el factor de ganancia global.
- 15 • Etapa 3: primero hallar la magnitud máxima por debajo de i_max ($k = 0 \dots L_{TCX}^{(bw)} / 4$) que es la mitad de alta si $i_max > -1$ usando $invGain = 4/g_{TCX}$ y que modifica el máximo: $xq[i_max] += (xq[i_max] < 0) ? -2 : 2$
- 20 • Etapa 4: comprimir de nuevo y cuantificar el total de $[i]$ hasta la mitad de la altura i_max hallada en la etapa anterior, como en la etapa 2
- 20 • Etapa 5: terminar y siempre comprimir dos líneas en el último i_max hallado, es decir, en $k = i_max + 1, i_max + 2$, de nuevo utilizando $invGain = 2/g_{TCX}$ si el i_max inicial hallado en la etapa 1 es mayor que -1 o utilizando $invGain = 4/g_{TCX}$ de lo contrario. Todos los i_max se inicializan en -1 . Para obtener más detalles, por favor, véase `AdaptLowFreqEmph()` en `tcx_utils_enc.c`.

5.3.3.2.5 Medición del ruido del espectro en el espectro de potencia

25 Para la orientación de la cuantificación en el proceso de codificación TXX, se determina una medición de ruido entre 0 (tonal) y 1 (tipo ruido) para cada línea espectral MDCT por encima de una frecuencia especificada basada en el espectro de potencia de la transformada actual. El espectro de potencia $X_P(k)$ se calcula a partir de los coeficientes MDCT $X_M(k)$ y los coeficientes MDST $X_S(k)$ en el mismo segmento de señal de dominio de tiempo y con la misma operación de ventana:

$$X_P(k) = X_M^2(k) + X_S^2(k) \quad \text{para } k = 0 \dots L_{TCX}^{(bw)} - 1 \quad (4)$$

35 Cada medición de ruido en $noiseFlag(k)$ se calcula entonces de la siguiente manera. En primer lugar, si la longitud de la transformada cambió (por ejemplo, después de una transformada de transición TCX después de una trama ACELP) o si la trama anterior no usó la codificación TCX20 (por ejemplo, en caso de que se usara una longitud de transformada más corta en la última trama), todos $noiseFlag(k)$ hasta $L_{TCX}^{(bw)} - 1$ se ajustan de nuevo a cero. La línea de inicio de la medición de ruido k_{start} se inicializa según la siguiente tabla 1.

40 **Tabla 1: Tabla de inicialización k_{start} en las mediciones de ruido**

Tasa de bits (kbps)	9,6	13,2	16,4	24,4	32	48	96	128
bw= NB, WB	66	128	200	320	320	320	320	320
bw=SWB,FB	44	96	160	320	320	256	640	640

45 Para las transiciones ACELP a TCX, k_{start} se escala en 1,25. Entonces, si la línea de inicio de la medición de ruido k_{start} es menor de $L_{TCX}^{(bw)} - 6$, la $noiseFlag(k)$ en y por encima de k_{start} se derivan recursivamente de las sumas de corridas de las líneas espectrales de potencia:

$$s(k) = \sum_{i=k-7}^{k+7} X_P(i), \quad c(k) = \sum_{i=k-1}^{k+1} X_P(i) \quad (5)$$

$$noiseFlags(k) = \begin{cases} 1 & \text{si } s(k) \geq (1.75 - 0.5 \cdot noiseFlags(k)) \cdot c(k) \\ 0 & \text{de otra manera} \end{cases} \quad \text{para } k_{start} \dots L_{TCX}^{(bw)} - 8 \quad (6)$$

Además, a cada tiempo $noiseFlags(k)$ se da el valor cero en el bucle anterior, $lastTone$ variable se ajusta a k . Las 7 líneas superiores se tratan por separado ya que $s(k)$ no se puede actualizar más ($c(k)$, sin embargo, se calcula como antes):

$$noiseFlags(k) = \begin{cases} 1 & \text{si } s(L_{TCX}^{(bw)} - 8) \geq (1.75 - 0.5 \cdot noiseFlags(k)) \cdot c(k) \\ 0 & \text{de otra manera} \end{cases} \quad \text{para } L_{TCX}^{(bw)} - 7 \dots L_{TCX}^{(bw)} - 2 \quad (7)$$

La línea más alta en $k = L_{TCX}^{(bw)} - 1$ se define como que es de tipo ruido, en consecuencia $noiseFlags(L_{TCX}^{(bw)} - 1) = 1$. Finalmente, si $lastTone$ variable anterior (que se inicializó a cero) es mayor de cero, entonces $noiseFlags(lastTone+1) = 0$. Cabe señalar que este procedimiento solo se lleva a cabo en TCX20, no en otros modos TCX ($noiseFlags(k) = 0$ para $k = 0 \dots L_{TCX}^{(bw)} - 1$).

5.3.3.2.6 Detector del factor de paso bajo

Un factor de paso bajo c_{lpf} se determina basándose en el espectro de potencia para todas las tasas de bits por debajo de 32,0 kbps. En consecuencia, el espectro de potencia $X_P(k)$ se compara iterativamente contra un umbral t_{lpf} para el total de $k = L_{TCX}^{(bw)} - 1 \dots L_{TCX}^{(bw)} / 2$, donde $t_{lpf} = 32.0$ para las ventanas MDCT regulares y $t_{lpf} = 64.0$ para las ventanas de transición ACELP a MDCT. La iteración se detiene tan pronto como $X_P(k) > t_{lpf}$.

El factor de paso bajo c_{lpf} determina como $c_{lpf} = 0.3 \cdot c_{lpf,prev} + 0.7 \cdot (k+1) / L_{TCX}^{(celp)}$, donde $c_{lpf,prev}$ es el último factor de paso bajo determinado. En la puesta en marcha del codificador, $c_{lpf,prev}$ se ajusta a 1,0. El factor de paso bajo c_{lpf} se usa para determinar el intervalo de detención del llenado de ruido (véase la subcláusula 5.3.3.2.10.2).

5.3.3.2.7 Cuantificador uniforme con zona muerta adaptativa

Para una cuantificación uniforme del espectro MDCT $\tilde{X}_M$ después o antes del ALFE (dependiendo del algoritmo de acentuación aplicado, véase la subcláusula 5.3.3.2.4.1), los coeficientes se dividen primero por la ganancia global g_{TCX} (véase la subcláusula 5.3.3.2.8.1.1), que controla el tamaño de la etapa de cuantificación. Los resultados luego se redondean a cero con un desplazamiento de redondeo que se adapta para cada coeficiente basado en la magnitud del coeficiente (relativa a g_{TCX}) y la tonalidad (como se define por $noiseFlags(k)$ en la subcláusula 5.3.3.2.5). Para líneas espectrales de alta frecuencia con tonalidad y magnitud bajas, se utiliza un desplazamiento de redondeo de cero, mientras que para todas las otras líneas espectrales se emplea un desplazamiento de 0,375. Más específicamente, se ejecuta el siguiente algoritmo.

A partir del coeficiente de MDCT más alto codificado en el índice $k = L_{TCX}^{(bw)} - 1$, se ajustó $\tilde{X}_M(k) = 0$ y disminución k en 1 siempre que la condición $noiseFlags(k) > 0$ y $|\tilde{X}_M(k)| / g_{TCX} < 1$ se evalúa como verdadera. Entonces hacia abajo de la primera línea en el índice $k' \geq 0$ donde esta condición no se cumple (lo cual está garantizado desde $noiseFlags(k)=0$), se realizan el redondeo hacia cero con un desplazamiento de redondeo de 0,375 y la limitación de los valores de número entero resultantes en el intervalo de - 32768 a 32767:

$$\hat{X}_M(k) = \begin{cases} \min \left(\left\lfloor \frac{\tilde{X}_M(k)}{g_{TCX}} + 0.375 \right\rfloor, 32767 \right) & , \tilde{X}_M(k) > 0 \\ \max \left(\left\lfloor \frac{\tilde{X}_M(k)}{g_{TCX}} - 0.375 \right\rfloor, -32768 \right) & , \tilde{X}_M(k) \leq 0 \end{cases} \quad (8)$$

con $k=0..k'$. Finalmente, todos los coeficientes cuantificados de $\hat{X}_M(k)$ en y por encima de $k = L_{TCX}^{(bw)}$ se ajustan a cero.

5 5.3.3.2.8 Codificador aritmético

Los coeficientes espectrales cuantificados se codifican sin ruido mediante una codificación de entropía y más particularmente mediante una codificación aritmética.

10 La codificación aritmética utiliza probabilidades de precisión de 14 bits para calcular su código. La distribución de probabilidad del alfabeto se puede derivar de diferentes maneras. A tasas bajas, se deriva de la envolvente de LPC, mientras que a altas tasas se deriva del contexto pasado. En ambos casos, se puede añadir un modelo armónico para refinar el modelo de probabilidades.

15 El siguiente pseudo-código describe la rutina de codificación aritmética, que se utiliza para codificar cualquier símbolo asociado con un modelo de probabilidad. El modelo de probabilidad está representado por una tabla de frecuencia acumulativa *cum_freq[]*. La derivación del modelo de probabilidad se describe en las siguientes subcláusulas.

```

/* variables globales */
20 bajo
    alto
    bits_to_follow

ar_encode(symbol, cum_freq[])
25 {
    si (ari_first_symbol()) {
        bajo = 0;
        alto = 65535;
        bits_to_follow = 0;
30     }
    intervalo = alto-bajo+1;

    si (símbolo > 0) {
        alto = bajo + ((range*cum_freq[symbol-1])>>14) - 1;
35     }
    bajo += ((range*cum_freq[symbol-1])>>14) - 1;

    para (;;) {
        si (alto < 32768 ) {
40         write_bit(0);
        mientras que ( bits_to_follow ) {
            write_bit(1);
            bits_to_follow--;
        }
45     }
    además si (bajo >= 32768 ) {
        write_bit(1)
        mientras que ( bits_to_follow ) {
            write_bit(0);
50         bits_to_follow--;
        }
        bajo -= 32768;
        alto -= 32768;
    }
55     además si ( (bajo >= 16384) && (alto < 49152) ) {
        bits_to_follow += 1;
        bajo -= 16384;
        alto -= 16384;
    }
60     además break;

    bajo += bajo;
    alto += alto+1;
}

```

```

    si (ari_last_symbol()) /* flush bits */
    si ( bajo < 16384 ) {
6      write_bit(0);
      mientras que ( bits_to_follow > 0) {
        write_bit(1);
        bits_to_follow--;
      }
      además {
10     write_bit(1);
        mientras que ( bits_to_follow > 0) {
          write_bit(0);
          bits_to_follow--;
        }
15     }
    }
  }

```

20 Las funciones auxiliares *ari_first_symbol()* y *ari_last_symbol()* detectan el primer símbolo y el último símbolo de la palabra de código generada respectivamente.

5.3.3.2.8.1 Códec aritmético basado en el contexto

25 5.3.3.2.8.1.1 Estimador de ganancia global

La estimación de la ganancia global g_{TCX} para la trama TCX se realiza en dos etapas iterativas. La primera estimación considera una ganancia de SNR de 6dB por muestra por bit de SQ. La segunda estimación refina la estimación teniendo en cuenta la codificación de entropía.

30 La energía de cada bloque de 4 coeficientes se calcula primero:

$$E[k] = \sum_{i=0}^4 \hat{X}^2[4.k+i]$$

(9)

35 Una búsqueda de bisección se realiza con una resolución final de 0,125dB:

Inicialización: ajustar $fac = desplazamiento = 12.8$ y $blanco = 0.15(target_bits - L/16)$

Iteración: realizar el siguiente bloque de operaciones 10 veces

40 1– $fac = fac/2$

2– $desplazamiento = desplazamiento - fac$

45 2– $ener = \sum_{i=0}^{L/4} a[i]$, donde $a[i] = \begin{cases} E[k] - desplazamiento & \text{si } E[k] - desplazamiento > 0.3 \\ 0 & \text{de otra manera} \end{cases}$

3– si ($ener > target$) entonces $desplazamiento = desplazamiento + fac$

La primera estimación de la ganancia entonces está proporcionada por:

$$g_{TCX} = 10^{0.45 + desplazamiento/2}$$

(10)

50

5.3.3.2.8.1.2 Bucle de velocidad para la tasa de bits constante y la ganancia global

A fin de fijar la mejor ganancia g_{TCX} dentro de las restricciones de $used_bits \leq target_bits$, se lleva a cabo un

procedimiento de convergencia de g_{TCX} y $used_bits$ mediante el uso de los valores y constantes siguientes:

W_{Lb} y W_{Ub} indican las ponderaciones correspondientes al límite inferior y límite superior,

5 g_{Lb} y g_{Ub} indican la ganancia correspondiente al límite inferior y el límite superior, y

Lb_found y Ub_found indican indicadores que indican que se halla g_{Lb} y g_{Ub} , respectivamente.

10 μ y η son variables con $\mu = \max(1, 2.3 - 0.0025 * target_bits)$ y $\eta = 1/\mu$.

λ y ν son constantes, ajustados como 10 y 0,96.

Después de la estimación inicial del consumo de bits por codificación aritmética, $stop$ se establece 0 cuando $target_bits$ es mayor que $used_bits$, mientras que $stop$ se establece como $used_bits$ cuando $used_bits$ es mayor que $target_bits$.

Si $stop$ es mayor que 0, esto significa que $used_bits$ es mayor que $target_bits$,

20 g_{TCX} se debe modificar para ser mayor que el anterior y Lb_found se fija como VERDADERO, g_{Lb} se establece como el anterior g_{TCX} . W_{Lb} se establece como

$$W_{Lb} = stop - target_bits + \lambda, \quad (11)$$

25 Cuando se estableció Ub_found , esto significa que $used_bits$ era menor que $target_bits$, g_{TCX} se actualiza como un valor interpolado entre límite superior y el límite inferior.

$$g_{TCX} = (g_{Lb} \cdot W_{Ub} + g_{Ub} \cdot W_{Lb}) / (W_{Ub} + W_{Lb}), \quad (12)$$

30 De otra manera, esto significa que Ub_found es FALSO, la ganancia se amplifica como

$$g_{TCX} = g_{TCX} \cdot (1 + \mu \cdot ((stop/\nu) / target_bits - 1)), \quad (13)$$

35 con mayor relación de amplificación cuando la relación de $used_bits (= stop)$ y $target_bits$ es mayor para acelerar para alcanzar g_{Ub} .

Si $stop$ equivale a 0, esto significa $used_bits$ es menor que $target_bits$,

40 g_{TCX} debe ser menor que el anterior y Ub_found se establece como 1, Ub se establece como el anterior g_{TCX} y W_{Ub} se establece como

$$W_{Ub} = target_bits - used_bits + \lambda, \quad (14)$$

Si Lb_found ya se ha establecido, la ganancia se calcula como

$$45 \quad g_{TCX} = (g_{Lb} \cdot W_{Ub} + g_{Ub} \cdot W_{Lb}) / (W_{Ub} + W_{Lb}), \quad (15)$$

de otra manera, a fin de acelerar la ganancia de banda más baja g_{Lb} , la ganancia se reduce como,

$$50 \quad g_{TCX} = g_{TCX} \cdot (1 - \eta \cdot (1 - (used_bits \cdot \nu) / target_bits)), \quad (16)$$

con mayores tasas de reducción de ganancia cuando la relación de $used_bits$ y $target_bits$ es pequeña.

Después de la corrección anterior de ganancia, se realiza la cuantificación y se obtiene la estimación de $used_bits$ por codificación aritmética. Como resultado, $stop$ se ajusta a 0 cuando $target_bits$ es mayor que $used_bits$, y se establece como $used_bits$ cuando es mayor que $target_bits$. Si el recuento del bucle es menor de 4, el procedimiento de establecimiento del límite inferior o procedimiento de establecimiento del límite superior se lleva a cabo en el próximo bucle dependiendo del valor de $stop$. Si el recuento del bucle es 4, se obtienen la ganancia final g_{TCX} y la secuencia de MDCT cuantificada $X_{QMDCT}(k)$.

5.3.3.2.8.1.3 Derivación y codificación del modelo de probabilidad

Los coeficientes espectrales cuantificados X se codifican sin ruido a partir del coeficiente de menor frecuencia y progresan al coeficiente de frecuencia más alta. Están codificados por grupos de dos coeficientes a y b que se reúnen en una llamada 2-tupla $\{a, b\}$.

Cada 2-tupla $\{a, b\}$ se divide en tres partes, a saber, MSB, LSB y el signo. El signo se codifica independientemente de la magnitud usando una distribución de probabilidad uniforme. La misma magnitud se divide además en dos partes, los dos bits más significativos (MSB) y los restantes al menos dos planos de bit significativos (LSBs, si corresponde). Las 2 tuplas para las cuales la magnitud de los dos coeficientes espectrales es menor o igual a 3 son codificadas directamente por la codificación MSB. De lo contrario, se transmite primero un símbolo de escape para señalar cualquier plano de bits adicional.

En el ejemplo de la figura 1 se ilustran la relación entre 2-tupla, los valores espectrales individuales a y b de una 2-tupla, los planos de bit más significativos m y los planos de bit menos significativos restantes r . En este ejemplo, se envían tres símbolos de escape antes del valor real m , lo que indica tres planos de bits de menor significación transmitidos.

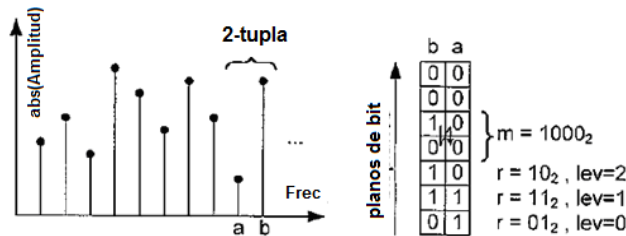


Figura 1: ejemplo de un par codificado (2-tupla) de valores espectrales a y b y su representación como m y r .

El modelo de probabilidad se deriva del contexto pasado. El contexto pasado se traduce en un índice de 12 bits y se mapea con la tabla de consulta `ari_context_lookup []` con uno de los 64 modelos de probabilidad disponibles almacenados en `ari_cf_m[]`.

El contexto pasado se deriva de dos 2-tuplas ya codificadas dentro de la misma trama. El contexto se puede derivar del vecindario directo o localizar más lejos en las frecuencias pasadas. Los contextos separados se mantienen para las regiones de pico (coeficientes que pertenecen a los picos armónicos) y otras regiones (no pico) según el modelo armónico. Si no se usa ningún modelo armónico, solo se usa el otro contexto de región (no pico).

Los valores espectrales a cero que se encuentran en la cola del espectro no se transmiten. Esto se logra mediante la transmisión del índice de la última 2-tupla no cero. Si se usa un modelo armónico, la cola del espectro se define como la cola del espectro que consiste en los coeficientes de las regiones del pico, seguidos por los otros coeficientes de región (no pico), ya que esta definición tiende a aumentar el número de ceros finales y por lo tanto mejora la eficiencia de codificación. El número de muestras para codificar se calcula de la siguiente manera:

$$lastnz = 2 \left(\max_{0 \leq k < L/2} \{ (X[ip[2k]] + X[ip[2k+1]]) > 0 \} \right) + 2 \quad (17)$$

Los siguientes datos se escriben en el flujo de bits en el siguiente orden:

- 1- $lastnz/2-1$ se codifica en $\left\lceil \log_2\left(\frac{L}{2}\right) \right\rceil$ bits.
- 2- Los MSB codificados por entropía junto con símbolos de escape.
- 3- Los signos con palabras clave de 1 bit
- 4- Los bits de cuantificación residual descritos en la sección cuando el presupuesto de bits no se utiliza completamente.
- 5- Los LSB se escriben hacia atrás desde el final del búfer de flujo de bits.

El siguiente pseudo-código describe cómo se deriva el contexto y cómo se calculan los datos de flujo de bits para los MSB, signos y LSB. Los argumentos de entrada son los coeficientes espectrales cuantificados $X[]$, el tamaño del espectro considerado L , el presupuesto de bits $target_bits$, los parámetros del modelo armónico (pi , hi) y el índice del último símbolo no cero $lastnz$.

```

ari_context_encode(X[], L, target_bits, pi[], hi[], lastnz)
{
    c[0]=c[1]=p1=p2=0;
    para (k=0; k<lastnz; k+=2) {
        ari_copy_states();
        (a1_i,p1,idx1) = get_next_coeff(pi,hi,lastnz);
        (b1_i,p2,idx2) = get_next_coeff(pi,hi,lastnz);
        t=get_context(idx1,idx2,c,p1,p2);
        esc_nb = lev1 = 0;
        a = a1 = abs(X[a1_i]);
        b = b1 = abs(X[b1_i]);
        /* codificación de signo */
        si (a1>0) save_bit(X[a1_i]>0?0:1);
        si (b1>0) save_bit(X[b1_i]>0?0:1);
        /* codificación de MSB */
        mientras que (a1 > 3 || b1 > 3) {
            pki = ari_context_lookup[t+1024*esc_nb];
            /* escribe palabra clave de escape */
            ari_encode(17, ari_cf_m[pki]);
            a1>>=1; b1 >>=1; lev1++;
            esc_nb = min(lev1,3);
        }
        pki = ari_context_lookup[t+1024*esc_nb];
        ari_encode(a1+4*b1, ari_cf_m[pki]);
        /* codificación de LSB */
        para (lev=0;lev<lev1;lev++){
            write_bit_end((a>>lev)&1);
            write_bit_end((b>>lev)&1);
        }
        /*verificar presupuesto*/
        si (nbbits>target_bits){
            ari_restore_states();
            break;
        }
        c=update_context(a,b,a1,b1,c,p1,p2);
    }
    write_sign_bits();
}

```

Las funciones auxiliares *ari_save_states()* y *ari_restore_states()* se usan para guardar y restaurar los estados codificadores aritméticos respectivamente. Permite cancelar la codificación de los últimos símbolos si viola el presupuesto de bits. Además y en caso de desbordamiento del presupuesto de bits, es capaz de llenar los bits restantes con ceros hasta llegar al final del presupuesto de bits o hasta procesar muestras *lastnz* en el espectro.

Las otras funciones auxiliares se describen en las siguientes subcláusulas.

5.3.3.2.8.1.4 Obtener próximo coeficiente

```

5      (a,p,idx) = get_next_coeff(pi, hi, lastnz)
      Si ((ii[0] ≥ lastnz - min(#pi, lastnz)) or
10     (ii[1] < min(#pi, lastnz) y pi[ii[1]] < hi[ii[0]])) entonces
      {
      p=1
      idx=ii[1]
      a=pi[ii[1]]
      }
      además
      {
15     p=0
      idx=ii[0] + #pi
      a=hi[ii[0]]
      }
      ii[p]=ii[p] + 1

```

20 Los contadores ii[0] y ii[1] se inicializan a 0 en el comienzo de *ari_context_encode()* (y *ari_context_decode()* en el decodificador).

5.3.3.2.8.1.5 Actualización del contexto

25 El contexto se actualiza como se describe en el siguiente pseudocódigo. Consiste en la concatenación de dos elementos de contexto de 4 bits.

```

      si (  $p1 \neq p2$  )
      {
30     si (  $\text{mod}(idx1,2) == 1$  )
      {
      t =  $1 + 2 \lfloor a/2 \rfloor \cdot (1 + \lfloor a/4 \rfloor)$ 
      si (  $t > 13$  )
      t =  $12 + \min(1 + \lfloor a/8 \rfloor, 3)$ 
35     c[p1] =  $2^4 \cdot (c[p1] \wedge 15) + t$ 
      }
      si (  $\text{mod}(idx2,2) == 1$  )
      {
      t =  $1 + 2 \lfloor b/2 \rfloor \cdot (1 + \lfloor b/4 \rfloor)$ 
40     si (  $t > 13$  )
      t =  $12 + \min(1 + \lfloor b/8 \rfloor, 3)$ 
      c[p2] =  $2^4 \cdot (c[p2] \wedge 15) + t$ 
      }
      }
45     además
      {
      c[p1 ∨ p2] =  $16 \cdot (c[p1 \vee p2] \wedge 15)$ 
      si (  $esc\_nb < 2$  )
      c[p1 ∨ p2] =  $c[p1 \vee p2] + 1 + (a1 + b1) \cdot (esc\_nb + 1)$ 
50     además
      c[p1 ∨ p2] =  $c[p1 \vee p2] + 12 + esc\_nb$ 
      }

```

5.3.3.2.8.1.6 Obtener contexto

55 El contexto final se enmienda de dos maneras:

```

t = c[p1 ∨ p2]
si min(idx1, idx2) > L/2 entonces
t = t + 256
5 si target_bits > 400 entonces
t = t + 512

```

El contexto t es un índice de 0 a 1023.

10 5.3.3.2.8.1.7 Estimación del consumo de bits

La estimación de consumo de bits del codificador aritmético basado en contexto es necesaria para la optimización del bucle de velocidad de la cuantificación. La estimación se realiza calculando el requisito de bits sin llamar al codificador aritmético. Los bits generados se pueden estimar con precisión mediante:

```

15 cum_freq = arith_cf_m[pki] + m
proba *= cum_freq[0] - cum_freq[1]
nlz = norm_1(proba) /*obtener el número de cero inicial */
nbits = nlz
20 proba >= 14

```

donde $proba$ es un número entero inicializado a 16384 y m es un símbolo MSB.

25 5.3.3.2.8.1.8 Modelo armónico

Para la codificación aritmética basada tanto en el contexto como en la envolvente, se utiliza un modelo armónico para una codificación más eficiente de tramas con contenido armónico. El modelo se inhabilita si se cumple cualquiera de las siguientes condiciones:

- 30 – La tasa de bits no es de 9,6, 13,2, 16,4, 24,4, 32, 48 kbps.
- La trama anterior fue codificada por ACELP.
- Se utiliza la codificación aritmética basada en envolvente y el tipo de codificador no es de voz ni genérico.
- 35 – El indicador del modelo armónico de un solo bit en el flujo de bits se establece en cero.

Cuando el modelo está habilitado, el intervalo del dominio de frecuencia de los armónicos es un parámetro clave y comúnmente se analiza y codifica para ambas variedades de codificadores aritméticos.

40 5.3.3.2.8.1.8.1 Codificación del intervalo de armónicos

45 Cuando el retardo de tono y la ganancia se utilizan para el postprocesamiento, el parámetro de retardo se utiliza para representar el intervalo de armónicos en el dominio de frecuencia. De lo contrario, se aplica la representación normal del intervalo.

5.3.3.2.8.1.8.1.1 Codificación del intervalo dependiendo del retardo del tono del dominio de tiempo

Si la parte entera del retardo de tono en el dominio de tiempo d_{int} es menor que el tamaño de la trama de MDCT L_{TCX} , la unidad del intervalo del dominio de frecuencia (entre picos armónicos correspondientes al retardo de tono) T_{UNIT} con exactitud fraccional de 7 bit está dada por

$$T_{UNIT} = \frac{(2 \cdot L_{TCX} \cdot res_max) \cdot 2^7}{(d_{\text{int}} \cdot res_max + d_{fr})} \quad (18)$$

55 donde d_{fr} indica la parte fraccional del retardo de tono en el dominio de tiempo, res_max indica el número máximo de valores fraccionales permisibles cuyos valores son 4 o 6 dependiendo de las condiciones.

Debido a que T_{UNIT} tiene intervalo limitado, el intervalo real entre los picos armónicos del dominio de frecuencia se codifica relativamente a T_{UNIT} usando los bits especificados en la tabla 2. Entre los factores de multiplicación candidatos, $Ratio()$ dados en la tabla 3 o tabla 4, el número de multiplicación se selecciona de modo que proporciona el intervalo armónico más adecuado de coeficientes de transformación del dominio MDCT.

$$Index_T = (T_{UNIT} + 2^6) / 2^7 - 2 \quad (19)$$

$$T_{MDCT} = \lfloor 4 \cdot T_{UNIT} \cdot Ratio(Index_{Bandwidth}, Index_T, Index_{MUL}) \rfloor / 4 \quad (20)$$

Tabla 2: número de bits para especificar el multiplicador dependiendo de $Index_T$

$Index_T$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
NB:	5	4	4	4	4	4	4	3	3	3	3	2	2	2	2	2
WB:	5	5	5	5	5	5	4	4	4	4	4	4	4	2	2	2

Tabla 3: candidatos del multiplicador en el orden de $Index_{MUL}$ dependiendo de $Index_T$ (NB)

$Index_T$																
0	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
	19	20	21	22	23	24	25	26	27	28	30	32	34	36	38	40
1	0,5	1	2	3	4	5	6	7	8	9	10	12	16	20	24	30
2	2	3	4	5	6	7	8	9	10	12	14	16	18	20	24	30
3	2	3	4	5	6	7	8	9	10	12	14	16	18	20	24	30
4	2	3	4	5	6	7	8	9	10	12	14	16	18	20	24	30
5	1	2	2,5	3	4	5	6	7	8	9	10	12	14	16	18	20
6	1	1,5	2	2,5	3	3,5	4	4,5	5	6	7	8	9	10	12	16
7	1	2	3	4	5	6	8	10	—	—	—	—	—	—	—	—
8	1	2	3	4	5	6	8	10	—	—	—	—	—	—	—	—
9	1	1,5	2	3	4	5	6	8	—	—	—	—	—	—	—	—
10	1	2	2,5	3	4	5	6	8	—	—	—	—	—	—	—	—
11	1	2	3	4	—	—	—	—	—	—	—	—	—	—	—	—
12	1	2	4	6	—	—	—	—	—	—	—	—	—	—	—	—
13	1	2	3	4	—	—	—	—	—	—	—	—	—	—	—	—
14	1	1,5	2	4	—	—	—	—	—	—	—	—	—	—	—	—
15	1	1,5	2	3	—	—	—	—	—	—	—	—	—	—	—	—
16	0,5	1	2	3	—	—	—	—	—	—	—	—	—	—	—	—

Tabla 4: candidatos del multiplicador en el orden dependiendo de $Index_T$ (WB)

$Index$																
0	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
	19	20	21	22	23	24	25	26	27	28	30	32	34	36	38	40
1	1	2	3	4	5	6	7	8	9	10	12	14	16	18	20	22
	24	26	28	30	32	34	36	38	40	44	48	54	60	68	78	80
2	1,5	2	2,5	3	4	5	6	7	8	9	10	12	14	16	18	20
	22	24	26	28	30	32	34	36	38	40	42	44	48	52	54	68
3	1	1,5	2	2,5	3	4	5	6	7	8	9	10	11	12	13	14

	15	16	18	20	22	24	26	28	30	32	34	36	40	44	48	54
4	1	1,5	2	2,5	3	3,5	4	4,5	5	5,5	6	6,5	7	7,5	8	9
	10	11	12	13	14	15	16	18	20	22	24	26	28	34	40	41
5	1	1,5	2	2,5	3	3,5	4	4,5	5	6	7	8	9	10	11	12
	13	14	15	16	17	18	19	20	21	22,5	24	25	27	28	30	35
6	0,5	1	1,5	2	2,5	3	3,5	4	4,5	5	5,5	6	7	8	9	10
7	1	2	2,5	3	4	5	6	7	8	9	10	12	15	16	18	27
8	1	1,5	2	2,5	3	3,5	4	5	6	8	10	15	18	22	24	26
9	1	1,5	2	2,5	3	3,5	4	5	6	8	10	12	13	14	18	21
10	0,5	1	1,5	2	2,5	3	4	5	6	8	9	11	12	13,5	16	20
11	0,5	1	1,5	2	2,5	3	4	5	6	7	8	10	11	12	14	20
12	0,5	1	1,5	2	2,5	3	4	4,5	6	7,5	9	10	12	14	15	18
13	0,5	1	1,25	1,5	1,75	2	2,5	3	3,5	4	4,5	5	6	8	9	14
14	0,5	1	2	4	–	–	–	–	–	–	–	–	–	–	–	–
15	1	1,5	2	4	–	–	–	–	–	–	–	–	–	–	–	–
16	1	2	3	4	–	–	–	–	–	–	–	–	–	–	–	–

5.3.3.2.8.1.8.1.2 Codificación del intervalo sin depender del retardo de tono del dominio de tiempo

5 Cuando el retardo de tono y la ganancia de dominio de tiempo no se usa o la ganancia de tono es menor de o igual a 0,46, se usa la codificación normal del intervalo con resolución desigual.

El intervalo unitario de los picos espectrales T_{UNIT} se codifica como

$$T_{UNIT} = index + base \cdot 2^{Res} - bias, \quad (21)$$

10

Y el intervalo real T_{MDCT} se representa con resolución fraccional de Res como

$$T_{MDCT} = T_{UNIT} / 2^{Res}. \quad (22)$$

15 Cada parámetro se muestra en la tabla 5, donde "tamaño pequeño" significa que el tamaño de la trama es menor de 256 de las tasas de bits deseadas es menor de o igual a 150.

Tabla 5: resolución desigual para la codificación de (0 ≤ índice < 256)

	<i>Res</i>	<i>base</i>	<i>sesgo</i>
<i>índice</i> < 16	3	6	0
16 ≤ <i>índice</i> < 80	4	8	16
80 ≤ <i>índice</i> < 208	3	12	80
"tamaño pequeño" o 208 ≤ <i>índice</i> < 224	1	28	208
224 ≤ <i>índice</i> < 256	0	188	224

20

5.3.3.2.8.1.8.2 Nulo

5.3.3.2.8.1.8.3 Búsqueda para intervalo de armónicos

En busca del mejor intervalo de armónicos, el codificador trata de encontrar el índice que puede maximizar la suma ponderada E_{PERIOD} de la parte del pico de los coeficientes MDCT absolutos. $E_{ABSM}(k)$ indica la suma de 3 muestras del valor absoluto de los coeficientes de transformada del dominio MDCT como

$$E_{ABSM}(k) = \sum_{j=0}^2 abs(X_M(k+j-1)) \quad (23)$$

$$E_{PERIOD}(T_{MDCT}) = \left(\frac{1}{num_peak} \right) \sum_{n=1}^{num_peak} E_{ABSM}(\lfloor n \cdot T_{MDCT} \rfloor) ((3n-2)/255)^{0.3} \quad (24)$$

donde num_peak es el número máximo que $\lfloor n \cdot T_{MDCT} \rfloor$ alcanza el límite de las muestras en el dominio de frecuencia.

En caso de que el intervalo no se base en el retardo de tono en el dominio del tiempo, se usa la búsqueda jerárquica para ahorrar costes computacionales. Si el índice del intervalo es menor de 80, la periodicidad se comprueba mediante una etapa gruesa de 4. Después de obtener el mejor intervalo, se busca una periodicidad más fina alrededor del mejor intervalo de -2 a +2. Si el índice es igual o mayor de 80, se busca la periodicidad de cada índice.

5.3.3.2.8.1.8.4 Decisión del modelo armónico

En la estimación inicial, se obtiene el número de bits usados sin modelo armónico, $used_bits$, y uno con modelo armónico, $used_bits_{hm}$ y el indicador de bits consumidos $Indicator_B$ se definen como

$$indicator_B = B_{no_hm} - B_{hm}, \quad (25)$$

$$B_{no_hm} = \max(stop, used_bits), \quad (26)$$

$$B_{hm} = \max(stop_{hm}, used_bits_{hm}) + Index_bits_{hm}, \quad (27)$$

donde $Index_bits_{hm}$ indica los bits adicionales para modelar la estructura armónica, y $stop$ y $stop_{hm}$ indican los bits consumidos cuando son más grandes que los bits objetivo. En consecuencia, cuanto mayor es el $Indicator_B$, más preferible es usar el modelo armónico. La periodicidad relativa $indicator_{hm}$ se define como la suma normalizada de valores absolutos para las regiones del pico de los coeficientes de MCDT conformados como

$$indicator_{hm} = L_M \cdot E_{PERIOD}(T_{MDCT_max}) / \sum_{n=1}^{L_M} E_{ABSM}(n), \quad (28)$$

donde T_{MDCT_max} es el intervalo armónico que alcanza el valor máximo de E_{PERIOD} . Cuando la puntuación de periodicidad de esta trama es mayor que el umbral como

$$si ((indicator_B > 2) \parallel ((abs(indicator_B) \leq 2) \& \& (indicator_{hm} > 2.6))), \quad (29)$$

esta trama se considera codificada por el modelo armónico. Los coeficientes de MCDT conformados divididos por la ganancia g_{TCX} se cuantifican para producir una secuencia de valores enteros de los coeficientes de MCDT, $\hat{X}_{TCX_hm}$, y se comprime por la codificación aritmética con el modelo armónico. Este procedimiento necesita el procedimiento de convergencia iterativa (bucle de velocidad) para obtener g_{TCX} y $\hat{X}_{TCX_hm}$ con bits consumidos B_{hm} . Al final de la convergencia, a fin de validar el modelo armónico, los bits consumidos B_{no_hm} por codificación aritmética con el modelo normal (no armónico) para $\hat{X}_{TCX_hm}$ se calcula adicionalmente y se compara con B_{hm} . Si B_{hm} es mayor que B_{no_hm} , la codificación aritmética de $\hat{X}_{TCX_hm}$ se invierte para usar el modelo normal. $B_{hm} - B_{no_hm}$ se puede usar para la cuantificación residual para las mejoras adicionales. De otra manera, se usa el modelo armónico en la codificación aritmética.

En contraste, si el indicador de periodicidad de esta trama es menor o la misma que el umbral, la cuantificación y la codificación aritmética se llevan a cabo asumiendo el modelo normal para producir una secuencia de valores enteros de los coeficientes de MCDT conformados, $\hat{X}_{TCX_no_hm}$ con los bits consumidos B_{no_hm} . Después de la convergencia del bucle de velocidad, se calculan los bits consumidos B_{hm} por codificación aritmética con el modelo armónico para $\hat{X}_{TCX_no_hm}$. Si B_{no_hm} es mayor que B_{hm} , la codificación aritmética de $\hat{X}_{TCX_no_hm}$ se cambia para usar el modelo armónico. De otra manera, el modelo normal se usa en la codificación aritmética.

5.3.3.2.8.1.9 Usos de información armónica en la codificación aritmética basada en el contexto

Para la codificación aritmética basada en el contexto, todas las regiones se clasifican en dos categorías. Una es parte de pico y consiste en 3 muestras consecutivas centradas en U^{th} pico (U es un número entero positivo hasta el límite) del pico armónico de τ_U ,

$$\tau_U = \lfloor U \cdot T_{MDCT} \rfloor. \quad (30)$$

Las otras muestras pertenecen a la parte normal o del valle. La parte del pico armónico se puede especificar mediante el intervalo de armónicos y múltiplos enteros del intervalo. La codificación aritmética utiliza diferentes contextos para las regiones de pico y valle.

Para facilitar la descripción y la implementación, el modelo armónico utiliza las siguientes secuencias de índice:

$$pi = (i \in [0..L_M - 1] : \exists U : \tau_U - 1 \leq i \leq \tau_U + 1), \quad (31)$$

$$hi = (i \in [0..L_M - 1] : i \notin pi), \quad (32)$$

$$ip = (pi, hi), \text{ la concatenación de } pi \text{ y } hi. \quad (1)$$

En caso del modelo armónico inhabilitado, estas secuencias son $pi = ()$, y $hi = ip = (0, \dots, L_M - 1)$.

5.3.3.2.8.2 Codificador aritmético basado en la envolvente

En el dominio MDCT, las líneas espectrales se ponderan con el modelo perceptual $W(z)$ de tal manera que cada línea se puede cuantificar con la misma exactitud. La varianza de las líneas espectrales individuales sigue la forma del predictor lineal $A^{-1}(z)$ ponderado por el modelo perceptual, por lo que la forma ponderada es $S(z) = W(z)A^{-1}(z)$. $W(z)$ se calcula mediante la transformación $\hat{q}'_l$ a las ganancias LPC del dominio de frecuencia como se detalla en las subcláusulas 5.3.3.2.4.1 y 5.3.3.2.4.2. $A^{-1}(z)$ se deriva de $\hat{q}'_1$ después de la conversión a los coeficientes de forma directa, y la aplicación de compensación de la inclinación $1 - \gamma z^{-1}$, y finalmente la transformación a las ganancias del dominio de frecuencia LPC. Todas las demás herramientas de conformación de frecuencia, así como la contribución del modelo armónico, se incluirán también en esta forma de envolvente $S(z)$. Se debe observar que esto solo proporciona las variaciones relativas de las líneas espectrales, mientras que la envolvente global tiene escala arbitraria, por lo que se debe comenzar por escalar la envolvente.

5.3.3.2.8.2.1 Escalado de la envolvente

Se asumirá que las líneas espectrales x_k son media de cero y se distribuyen según la distribución de Laplace, por lo que la función de distribución de probabilidad es

$$f(x_k) = \frac{1}{2b_k} \exp\left(-\frac{|x_k|}{b_k}\right) \quad (34)$$

La entropía y, por lo tanto, el consumo de bits de tal línea espectral es $bits_k = 1 + \log_2 2eb_k$. Sin embargo, esta fórmula asume que el signo está codificado también para aquellas líneas espectrales cuantificadas a cero. Para compensar esta discrepancia, en cambio se usa la aproximación

$$bits_k = \log_2 \left(2eb_k + 0.15 + \frac{0.035}{b_k} \right), \quad (35)$$

que es exacta para $b_k \geq 0.08$. Se asumirá que el consumo de bits de las líneas con $b_k \leq 0.08$ es $bits_k = \log_2(1.0224)$ que coincide con el consumo de bits a $b_k = 0.08$. Para $b_k > 255$ mayor se usa la entropía verdadera $bits_k = \log_2(2eb_k)$ por simplicidad.

La varianza de las líneas espectrales es entonces $\sigma_k^2 = 2b_k^2$. Si s_k^2 es el k th elemento de la potencia de la forma envolvente $|S(z)|^2$, entonces s_k^2 describe la energía relativa de las líneas espectrales de modo que $\gamma^2 \sigma_k^2 = b_k^2$ donde γ es el coeficiente de escalado. En otras palabras, s_k^2 describe solo la forma del espectro sin ninguna magnitud significativa y γ se utiliza para escalar esa forma para obtener la varianza real σ_k^2 .

Nuestro objetivo es que cuando se codifican todas las líneas del espectro con un codificador aritmético, entonces el

$$B = \sum_{k=0}^{N-1} bits_k$$

consumo de bits coincide con un nivel predefinido B , es decir, . Luego se puede usar un algoritmo bi-sección para determinar el factor de escalado apropiado γ de modo que se alcanza la tasa de bits B .

Una vez que la forma de la envolvente b_k se ha escalado de tal manera que el consumo de bits esperado de las señales que coinciden con esa forma produce la tasa de bits objetivo, se puede proceder a cuantificar las líneas espectrales.

5.3.3.2.8.2.2 Cuantificación del bucle de velocidad

Se asume que x_k se cuantifica a un número entero $\hat{x}_k$ de modo que el intervalo de cuantificación es $[\hat{x}_k - 0.5, \hat{x}_k + 0.5]$ entonces la probabilidad de una línea espectral que aparece en este intervalo es para $|\hat{x}_k| \geq 1$

$$p(\hat{x}_k) = \left(\exp\left(-\frac{|\hat{x}_k| - 0.5}{b_k}\right) - \exp\left(-\frac{|\hat{x}_k| + 0.5}{b_k}\right) \right) = \left(1 - \exp\left(-\frac{1}{b_k}\right) \right) \exp\left(-\frac{|\hat{x}_k| - 0.5}{b_k}\right). \quad (36)$$

y para $|\hat{x}_k| = 0$

$$p(\hat{x}_k) = \left(1 - \exp\left(-\frac{0.5}{b_k}\right) \right). \quad (37)$$

De ello se deduce que el consumo de bits para estos dos casos está en el caso ideal

$$\begin{cases} 1 - \frac{0.5}{b_k} \log_2 e - \log_2 \left(1 - \exp\left(-\frac{1}{b_k}\right) \right) + \frac{|\hat{x}_k|}{b_k} \log_2 e, & \hat{x}_k \neq 0 \\ \log_2 \left(1 - \exp\left(-\frac{0.5}{b_k}\right) \right), & \hat{x}_k = 0. \end{cases} \quad (38)$$

Mediante el pre-cálculo de los términos $\log_2 \left(1 - \exp\left(-\frac{1}{b_k}\right) \right)$ y $\log_2 \left(1 - \exp\left(-\frac{0.5}{b_k}\right) \right)$, se puede calcular de forma

eficiente el consumo de bits del espectro entero.

El bucle de velocidad se puede aplicar con una búsqueda bi-sección, donde se ajusta el escalado de las líneas espectrales por un factor ρ y se calcula el consumo de bits del espectro ρ^{x_k} , hasta estar suficientemente cerca de la tasa de bits deseada. Cabe señalar que los valores de caso ideal anteriores para el consumo de bits no coinciden necesariamente de forma perfecta con el consumo de bits final, ya que el códec aritmético funciona con una aproximación de precisión finita. Este bucle de velocidad se basa, por lo tanto, en una aproximación del consumo de bits, pero con el beneficio de una implementación computacionalmente eficiente.

Cuando se ha determinado el escalado óptimo σ , el espectro se puede codificar con un codificador aritmético estándar. Una línea espectral que se cuantifica a un valor $\hat{x}_k \neq 0$ se codifica al intervalo

$$\left[\exp\left(-\frac{|\hat{x}_k| - 0.5}{b_k}\right), \exp\left(-\frac{|\hat{x}_k| + 0.5}{b_k}\right) \right] \quad (39)$$

y $\hat{x}_k = 0$ se codifica en el intervalo

$$\left[1, \exp\left(-\frac{|\hat{x}_k| + 0.5}{b_k}\right) \right]. \quad (40)$$

El signo de $x_k \neq 0$ se codificará con un bit adicional.

Se debe observar que el codificador aritmético debe funcionar con una implementación de punto fijo de modo que los intervalos anteriores sean de bits exactos en todas las plataformas. Por lo tanto, todas las entradas al codificador aritmético, que incluyen el modelo predictivo lineal y el filtro de ponderación, se deben implementar en el punto fijo en todo el sistema.

5.3.3.2.8.2.3 Derivación y codificación del modelo de probabilidad

Cuando se ha determinado el escalado óptimo σ , el espectro se puede codificar con un codificador aritmético estándar. Una línea espectral que se cuantifica a un valor $\hat{x}_k \neq 0$ se codifica al intervalo

$$\left[\exp\left(-\frac{|\hat{x}_k| - 0.5}{b_k}\right), \exp\left(-\frac{|\hat{x}_k| + 0.5}{b_k}\right) \right] \quad (41)$$

y $\hat{x}_k = 0$ se codifica en el intervalo

$$\left[1, \exp\left(-\frac{|\hat{x}_k| + 0.5}{b_k}\right) \right]. \quad (42)$$

El signo de $x_k \neq 0$ se codificará con un bit adicional.

5.3.3.2.8.2.4 Modelo armónico en la codificación aritmética basada en la envolvente

En el caso de codificación aritmética basada en la envolvente, el modelo armónico se puede utilizar para mejorar la codificación aritmética. El procedimiento de búsqueda similar como en la codificación aritmética basada en el contexto se utiliza para estimar el intervalo entre armónicos en el dominio MDCT. Sin embargo, el modelo armónico se usa en combinación con la envolvente LPC como se muestra en la figura 2. La forma de la envolvente se produce según la información del análisis armónico.

La forma armónica en k en la muestra de datos de frecuencia se define como

$$Q(k) = h \cdot \exp\left(-\frac{(k-\tau)^2}{2\sigma^2}\right), \quad (43)$$

cuando $\tau-4 \leq k \leq \tau+4$ de otra manera $Q(k)=1.0$, donde τ indica la posición central de U^{th} armónico.

$$\tau = \lfloor U \cdot T_{MDCT} \rfloor \quad (44)$$

h y σ son altura y ancho de cada armónico que depende del intervalo unitario como se muestra,

$$h = 2.8(1.125 - \exp(-0.07 \cdot T_{MDCT} / 2^{Res})) \quad (45)$$

$$\sigma = 0.5(2.6 - \exp(-0.05 \cdot T_{MDCT} / 2^{Res})) \quad (46)$$

La altura y el ancho se hacen más grandes cuando el intervalo aumenta.

La envolvente espectral $S(k)$ se modifica mediante la forma armónica $Q(k)$ a k como

$$S(k) = S(k) \cdot (1 + g_{harm} \cdot Q(k)), \quad (47)$$

donde la ganancia para los componentes armónicos g_{harm} siempre se establece como 0,75 para el modo genérico,

y g_{harm} se selecciona de {0,6, 1,4, 4,5, 10,0} que minimiza E_{norm} para el modo de voz usando 2 bits,

$$E_{ABSres} = \sum_{k=0}^{L_M-1} (|X_M(k)| / S(k)), \quad (48)$$

$$E_{norm} = \sum_{k=0}^{L_M-1} (|X_M(k)| / S(k) / E_{ABSres})^4. \quad (49)$$

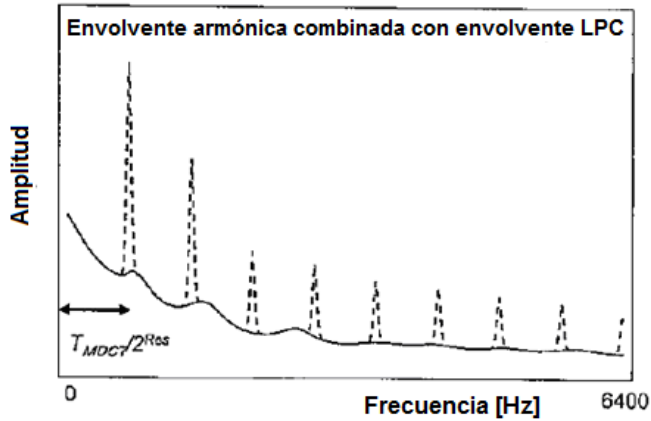


Figura 2: ejemplo de la envolvente armónica combinada con la envolvente de LPC usada en la codificación aritmética basada en la envolvente.

5.3.3.2.9 Codificación de la ganancia global

5.3.3.2.9.1 Optimización de la ganancia global

La ganancia global óptima g_{opt} se calcula a partir de los coeficientes de MCDT cuantificados y no cuantificados. Para las tasas de bit hasta 32 kbps, la desacentuación de baja frecuencia adaptativa (véase la subcláusula 6.2.2.3.2) se aplica a los coeficientes de MCDT cuantificados antes de esta etapa. En caso de que el cálculo produzca una ganancia óptima menor de o igual a cero, se usa la ganancia global g_{TCX} determinada antes (por estimación y bucle de velocidad).

$$g'_{opt} = \frac{\sum_{k=0}^{L_{TCX}^{(bw)}-1} X_M(k) \hat{X}_M(k)}{\sum_{k=0}^{L_{TCX}^{(bw)}-1} (\hat{X}_M(k))^2} \quad (50)$$

$$g_{opt} = \begin{cases} g'_{opt} & \text{if } g'_{opt} \geq 0 \\ g_{TCX} & \text{if } g'_{opt} < 0 \end{cases} \quad (51)$$

5.3.3.2.9.2 Cuantificación de ganancia global

Para la transmisión al decodificador, la ganancia global óptima g_{opt} se cuantifica a un índice de 7 bits $I_{TCX,gain}$:

$$I_{TCX,gain} = \left\lfloor 28 \log_{10} \left(\sqrt{\frac{L_{TCX}^{(bw)}}{160}} g_{opt} \right) + 0.5 \right\rfloor \quad (52)$$

La ganancia global decuantificada $\hat{g}_{TCX}$ se obtiene como se define en la subcláusula 6.2.2.3.3).

5.3.3.2.9.3 Codificación residual

La cuantificación residual es una capa de cuantificación de refinamiento que refina la primera etapa SQ. Aprovecha los bits no usados finales $target_bits - nbbits$, donde $nbbits$ es el número de bits consumidos por el codificador de entropía. La cuantificación residual adopta una estrategia ambiciosa y ninguna codificación de entropía para detener la codificación cada vez que el flujo de bits alcanza el tamaño deseado.

La cuantificación residual puede refinar la primera cuantificación por dos medios. El primer medio es el refinamiento de la cuantificación de la ganancia global. El refinamiento de la ganancia global solo se realiza para las tasas de y por encima de 13,2kbps. Se le asignan como máximo tres bits adicionales. La ganancia cuantificada $\hat{g}_{TCX}$ se refina secuencialmente a partir de $n = 0$ y aumenta n por uno después de cada iteración siguiente:

si $(g_{opt} < \hat{g}_{TCX})$ entonces
 write_bit(0)
 $\hat{g}_{TCX} = \hat{g}_{TCX} \cdot 10^{-2^{-n-2}/28}$
 además entonces
 write_bit(1)
 $\hat{g}_{TCX} = \hat{g}_{TCX} \cdot 10^{2^{-n-2}/28}$

El segundo medio de refinamiento consiste en la re-cuantificación de la línea del espectro cuantificada por línea. En primer lugar, las líneas cuantificadas no cero se procesan con un cuantificador residual de 1 bit:

si $(X[k] < \hat{X}[k])$ entonces
 write_bit(0)
 además entonces
 write_bit(1)

Finalmente, si quedan bits, las líneas de cero se consideran y cuantifican con en 3 niveles. El desplazamiento de

redondeo de SQ con zona muerta se tuvo en cuenta en el diseño del cuantificador residual:

$fac_z = (1 - 0.375) \cdot 0.33$
si $(|X[k]| < fac_z \cdot \hat{X}[k])$ *entonces*
 $write_bit(0)$
además entonces
 $write_bit(1)$
 $write_bit((1 + \text{sgn}(X[k])) / 2)$

5 5.3.3.2.10 Llenado de ruido

En el lado del decodificador se aplica el llenado de ruido para llenar intervalos en el espectro MDCT donde los coeficientes se han cuantificado a cero. El llenado de ruido inserta ruido pseudo-aleatorio en los intervalos, comenzando en el intervalo $k_{NFstart}$ hasta el intervalo k_{NFstop} . Para controlar la cantidad de ruido insertada en el decodificador, se calcula un factor de ruido en el lado del codificador y se transmite al decodificador.

5.3.3.2.10.1 Inclinación de llenado de ruido

Para compensar la inclinación LPC, se calcula un factor de compensación de inclinación. Para las tasas de bits por debajo de 13,2 kbps, la compensación de la inclinación se calcula a partir de los coeficientes LP cuantificados en forma directa $\hat{a}$, mientras que para las tasas de bits más altas, se usa un valor constante:

$$t'_{NF} = \begin{cases} 0.5625 & , \text{ si la tasa de datos es } \geq 13200 \\ \min \left(1, \frac{\sum_{i=0}^{15} \hat{a}(i+1)\hat{a}(i)}{\sum_{i=0}^{15} (\hat{a}(i))^2} + 0.09375 \right) & , \text{ si la tasa de datos es } < 13200 \end{cases} \quad (53)$$

$$t_{NF} = \max(0.375, t'_{NF})^{\frac{1}{L_{TCX}^{(celp)}}} \quad (54)$$

5.3.3.2.10.2 Intervalos de inicio y detención del llenado de ruido

Los intervalos de inicio y detención del llenado de ruido se calculan de la siguiente manera:

$$k_{NFstart} = \begin{cases} L_{TCX}^{(celp)} / 6 & , \text{ si la tasa de datos es } \geq 13200 \\ L_{TCX}^{(celp)} / 8 & , \text{ si la tasa de datos es } < 13200 \end{cases} \quad (55)$$

$$k_{NFstop} = \begin{cases} t(0) & \text{ si se usa IGF} \\ L_{TCX}^{(bw)} & \text{ además} \end{cases}$$

$$k_{NFstop,LP} = \begin{cases} \min(t(0), \text{round}(c_{lpf} \cdot L_{TCX}^{(celp)})) & , \text{ si se usa IGF} \\ \min(L_{TCX}^{(bw)}, \text{round}(c_{lpf} \cdot L_{TCX}^{(celp)})) & , \text{ además} \end{cases} \quad (56)$$

5.3.3.2.10.3 Ancho de la transición del ruido

En cada lado de un segmento de llenado de ruido se aplica un debilitamiento de transición al ruido insertado. El ancho de las transiciones (número de intervalos) se define como:

$$w_{NF} = \begin{cases} 8 & , \text{ si la tasa de datos es } < 48000 \\ 4 + \lfloor 12.8 \cdot g_{LTP} \rfloor & , \text{ si } (tasa \text{ de datos} \geq 48000) \wedge TCX20 \wedge (HM = 0 \vee previous = ACELP) \\ 4 + \lfloor 12.8 \cdot \max(g_{LTP}, 0.3125) \rfloor & , \text{ si } (tasa \text{ de datos} \geq 48000) \wedge TCX20 \wedge (HM \neq 0 \wedge previous \neq ACELP) \\ 3 & , \text{ si } (tasa \text{ de datos} \geq 48000) \wedge TCX10 \end{cases} \quad (57)$$

donde HM indica que se usa el modelo armónico para el códec aritmético y $previous$ indica el modo de códec previo.

5 5.3.3.2.10.4 Cálculo de segmentos de ruido

Se determinan los segmentos de llenado de ruido, que son los segmentos de intervalos sucesivos del espectro MDCT entre $k_{NFstart}$ y $k_{NFstop,LP}$ para los cuales todos los coeficientes están cuantificados a cero. Los segmentos se determinan tal como se define mediante el siguiente pseudo-código:

```

10      k = kNFstart
      mientras (k > kNFstart / 2) y (X̂M(k) = 0) hacen k = k - 1
      k = k + 1
      k'NFstart = k

      j = 0
      mientras (k < kNFstop,LP) {
        mientras (k < kNFstop,LP) y (X̂M(k) ≠ 0) hacen k = k + 1
        kNF0(j) = k

        mientras (k < kNFstop,LP) y (X̂M(k) = 0) hacen k = k + 1
        kNF1(j) = k

        if (kNF0(j) < kNFstop,LP) entonces j = j + 1
      }
      nNF = j

```

donde $k_{NF0}(j)$ y $k_{NF1}(j)$ son los intervalos de inicio y detención del segmento de llenado de ruido j , y n_{NF} es el número de segmentos.

15 5.3.3.2.10.5 Cálculo del factor del ruido

El factor de ruido se calcula a partir de los coeficientes MDCT no cuantificados de los intervalos para los que se aplica el llenado de ruido.

20 Si el ancho de transición de ruido w_{NF} es 3 o menos intervalos, se calcula un factor de atenuación basado en la energía de los intervalos de MDCT pares e impares:

$$E_{NF\text{even}} = \sum_{i=0}^{\left\lfloor \frac{k_{NFstop,LP}}{2} \right\rfloor - \left\lfloor \frac{k'_{NFstart}}{2} \right\rfloor - 1} \left(X_M \left(2 \left\lfloor \frac{k'_{NFstart}}{2} \right\rfloor + 2i \right) \right)^2 \quad (58)$$

$$E_{NF\text{odd}} = \sum_{i=0}^{\left\lfloor \frac{k_{NFstop,LP}}{2} \right\rfloor - \left\lfloor \frac{k'_{NFstart}}{2} \right\rfloor - 1} \left(X_M \left(2 \left\lfloor \frac{k'_{NFstart}}{2} \right\rfloor + 2i + 1 \right) \right)^2 \quad (59)$$

$$f_{NFatt} = \begin{cases} \sqrt{\frac{2 \min(E_{even}, E_{odd})}{E_{even} + E_{odd}}} & , \text{ si } w_{NF} \leq 3 \\ 1 & , \text{ si } w_{NF} > 3 \end{cases} \quad (60)$$

Para cada segmento se calcula un valor de error a partir de los coeficientes de MCDT no cuantificados, aplicando ganancia global, compensación de inclinación y transiciones:

$$E'_{NF}(j) = \frac{1}{g_{TCX}} \sum_{i=k_{NF0}}^{k_{NF1}-1} \left(|X_M(i)| \frac{\min(i - k_{NF0}(j) + 1, w_{NF})}{w_{NF}} \frac{\min(k_{NF1}(j) - i, w_{NF})}{w_{NF}} \left(\frac{1}{t_{NF}} \right)^i \right) \quad (61)$$

Una ponderación para cada segmento se calcula basándose en el ancho del segmento:

$$e_{NF}(j) = \begin{cases} k_{NF1}(j) - k_{NF0}(j) - w_{NF} + 1 & , (w_{NF} \leq 3) \wedge (k_{NF1}(j) - k_{NF0}(j) > 2w_{NF} - 4) \\ \frac{0.28125}{w_{NF}} (k_{NF1}(j) - k_{NF0}(j))^2 & , (w_{NF} \leq 3) \wedge (k_{NF1}(j) - k_{NF0}(j) \leq 2w_{NF} - 4) \\ k_{NF1}(j) - k_{NF0}(j) - 7 & , (w_{NF} > 3) \wedge (k_{NF1}(j) - k_{NF0}(j) > 12) \\ 0.03515625 (k_{NF1}(j) - k_{NF0}(j))^2 & , (w_{NF} > 3) \wedge (k_{NF1}(j) - k_{NF0}(j) \leq 12) \end{cases} \quad (62)$$

El factor de ruido luego se calcula de la siguiente manera:

$$f_{NF} = \begin{cases} \frac{\sum_{i=0}^{n_{NF}-1} E'_{NF}(i)}{\sum_{i=0}^{n_{NF}-1} e_{NF}(i)} & , \text{ if } \sum_{i=0}^{n_{NF}-1} e_{NF}(i) > 0 \\ 0 & , \text{ else} \end{cases} \quad (63)$$

5.3.3.2.10.6 Cuantificación del factor de ruido

Para la transmisión, el factor de ruido se cuantifica para obtener un índice de 3 bit:

$$I_{NF} = \min(\lfloor 10.75 f_{NF} + 0.5 \rfloor, 7) \quad (64)$$

5.3.3.2.11 Llenado de intervalos inteligente

La herramienta de *Llenado de intervalos inteligente* (IGF) es una técnica de llenado de ruido mejorada para llenar intervalos (regiones de los valores cero) en los espectros. Estos intervalos se pueden producir debido a la cuantificación gruesa en el procedimiento de codificación en el que grandes porciones de un espectro dado se pueden ajustar a cero para cumplir con las restricciones de bits. Sin embargo, con la herramienta IGF estas porciones de señal que faltan se reconstruyen en el lado receptor (RX) con información paramétrica calculada en el lado de transmisión (TX). IGF se utiliza solo si el modo TCX está activo.

A continuación, véase la tabla 6 de todos los puntos de operación de IGF:

Tabla 6: modos de aplicación IGF

Tasa de bits	Modo
9,6 kbps	WB
9,6 kbps	SWB
13,2 kbps	SWB
16,4 kbps	SWB
24,4 kbps	SWB
32,2 kbps	SWB
48,0 kbps	SWB
16,4 kbps	FB
24,4 kbps	FB

32,0 kbps	FB
48,0 kbps	FB
96,0 kbps	FB
128,0 kbps	FB

En el lado de transmisión, el IGF calcula los niveles en las bandas de factor de escala, usando un espectro de TCX de valor real o complejo. Adicionalmente, los índices de blanqueamiento espectral se calculan usando una medición de planitud espectral y un factor de cresta. Un codificador aritmético se utiliza para la codificación sin ruido y la transmisión eficiente al lado del receptor (RX).

5.3.3.2.11.1 Funciones auxiliares de IGF

5.3.3.2.11.1.1 Valores de mapeo con el factor de transición

Si existe una transición de la codificación de CELP a TCX ($isCelpToTCX = true$) o se señala una trama TCX 10 ($isTCX10 = true$), la longitud de la trama TCX puede cambiar. En caso de cambio de longitud de trama, todos los valores que están relacionados con la longitud de trama se mapean con la tF :

$$tF: N \times P \rightarrow N,$$

$$tF(n, f) := \begin{cases} \left\lfloor nf + \frac{1}{2} \right\rfloor, & \text{si } \left\lfloor nf + \frac{1}{2} \right\rfloor \text{ es par} \\ \left\lfloor nf + \frac{1}{2} \right\rfloor + 1, & \text{si } \left\lfloor nf + \frac{1}{2} \right\rfloor \text{ es impar} \end{cases} \quad (65)$$

donde n es un número natural, por ejemplo, un desplazamiento de banda del factor de escala, y f es un factor de transición, véase la tabla 11.

5.3.3.2.11.1.2 Espectro de potencia TCX

El espectro de potencia $P \in P^n$ de la trama TCX actual se calcula con:

$$P(sb) := R(sb)^2 + I(sb)^2, \quad sb = 0, 1, 2, \dots, n-1 \quad (66)$$

donde n es la longitud de ventana de TCX real, $R \in P^n$ es el vector que contiene la parte de valor real (transformada por cos) del espectro TCX actual, e $I \in P^n$ es el vector que contiene la parte imaginaria (transformada por sin) del espectro TCX actual.

5.3.3.2.11.1.3 La función de medición de la planitud espectral SFM

Se permite que $P \in P^n$ sea el espectro de potencia TCX calculado según la subcláusula 5.3.3.2.11.1.2 y b la línea de inicio y e la línea de detención del intervalo de medición SFM.

La función SFM , aplicada con IGF, se define con:

$$SFM: P^n \times N \times N \rightarrow P,$$

$$SFM(P, b, e) := \left\lfloor \frac{1}{2^{1+P}} \right\rfloor \left(\frac{1}{e-b} \left(1 + \sum_{sb=b}^{e-1} P(sb) \right) \right)^{-1}, \quad (67)$$

donde n es la longitud de ventana TCX real y P se define con:

$$p := \frac{1}{e-b} \sum_{sb=b}^{e-1} \lfloor \max(0, \log_2(P(sb))) \rfloor \quad (68)$$

5 5.3.3.2.11.1.4 La función del factor de cresta $CREST$

Se permite que P e P^n sea el espectro de potencia TCX calculado según la subcláusula 5.3.3.2.11.1.2 y b la línea de inicio y e la línea de detención del intervalo de medición del factor de cresta.

10 La función $CREST$, aplicada con IGF, se define con:

$$CREST: P^n \times N \times N \rightarrow P, \\ CREST(P, b, e) = \max \left(1, E_{max} \left(\frac{1}{e-b} \sum_{sb=b}^{e-1} \lfloor \max(0, \log_2(P(sb))) \rfloor^2 \right)^{\frac{1}{2}} \right), \quad (69)$$

donde n es la longitud de ventana TCX real y E_{max} se define con:

$$E_{max} := \max_{sb \in [b, e] \subset N} (0, \log_2(P(sb))). \quad (70)$$

5.3.3.2.11.1.5 La función de mapeo hT

20 La función de mapeo hT se define con:

$$hT: P \times N \rightarrow (0, 1, 2), \\ hT(s, k) = \begin{cases} 0 & \text{para } s \leq ThM_k \\ 1 & \text{para } ThM_k < s \leq ThS_k \\ 2 & \text{para } s > ThS_k \end{cases}, \quad (71)$$

donde S es un valor de planitud espectral calculado y k es la banda de ruido en alcance. Para los valores umbral ThM_k , ThS_k refiérase a la siguiente tabla 7.

Tabla 7: umbrales para el blanqueamiento para nT , ThM y ThS

Tasa de bits	Modo	nT	ThM	ThS
9,6 kbps	WB	2	0,36, 0,36	1,41, 1,41
9,6 kbps	SWB	3	0,84, 0,89, 0,89	1,30, 1,25, 1,25
13,2 kbps	SWB	2	0,84, 0,89	1,30, 1,25
16,4 kbps	SWB	3	0,83, 0,89, 0,89	1,31, 1,19, 1,19
24,4 kbps	SWB	3	0,81, 0,85, 0,85	1,35, 1,23, 1,23
32,2 kbps	SWB	3	0,91, 0,85, 0,85	1,34, 1,35, 1,35
48,0 kbps	SWB	1	1,15	1,19
16,4 kbps	FB	3	0,63, 0,27, 0,36	1,53, 1,32, 0,67
24,4 kbps	FB	4	0,78, 0,31, 0,34, 0,34	1,49, 1,38, 0,65, 0,65
32,0 kbps	FB	4	0,78, 0,31, 0,34, 0,34	1,49, 1,38, 0,65, 0,65

48,0 kbps	FB	1	0,80	1,0
96,0 kbps	FB	1	0	2,82
128,0 kbps	FB	1	0	2,82

5.3.3.2.11.1.6 Nulo

5.3.3.2.11.1.7 Tablas del factor de escala IGF

5

Las tablas del factor de escala IGF están disponibles para todos los modos donde se aplica IGF.

Tabla 8: tabla de desplazamiento de la banda del factor de escala

Tasa de bits	Modo	Número de bandas (nB)	Desplazamientos de banda del factor de escala ($t[0], t[1], \dots, t[nB]$)
9,6 kbps	WB	3	164, 186, 242, 320
9,6 kbps	SWB	3	200, 322, 444, 566
13,2 kbps	SWB	6	256, 288, 328, 376, 432, 496, 566
16,4 kbps	SWB	7	256, 288, 328, 376, 432, 496, 576, 640
24,4 kbps	SWB	8	256, 284, 318, 358, 402, 450, 508, 576, 640
32,2 kbps	SWB	8	256, 284, 318, 358, 402, 450, 508, 576, 640
48,0 kbps	SWB	3	512, 534, 576, 640
16,4 kbps	FB	9	256, 288, 328, 376, 432, 496, 576, 640, 720, 800
24,4 kbps	FB	10	256, 284, 318, 358, 402, 450, 508, 576, 640, 720, 800
32,0 kbps	FB	10	256, 284, 318, 358, 402, 450, 508, 576, 640, 720, 800
48,0 kbps	FB	4	512, 584, 656, 728, 800
96,0 kbps	FB	2	640, 720, 800
128,0 kbps	FB	2	640, 720, 800

10

La tabla 8 anterior se refiere a la ventana de longitud TCX 20 y un factor de transición 1,00.

Para todos las longitudes de ventana se aplican el siguiente remapeo

15

$$t(k) := t^F(t(k), f), \quad k = 0, 1, 2, \dots, nB \quad (72)$$

donde t^F es la función del mapeo del factor de transición descrito en la subcláusula 5.3.3.2.11.1.1.

5.3.3.2.11.1.8 La función de mapeo m

20

Tabla 9: sub-banda de fuente mínima IGF, $minSb$

Tasa de bits	modo	$minSb$
9,6 kbps	WB	30
9,6 kbps	SWB	32
13,2 kbps	SWB	32
16,4 kbps	SWB	32
24,4 kbps	SWB	32
32,2 kbps	SWB	32
48,0 kbps	SWB	64
16,4 kbps	FB	32
24,4 kbps	FB	32
32,0 kbps	FB	32
48,0 kbps	FB	64
96,0 kbps	FB	64
128,0 kbps	FB	64

25

Para cada modo se define una función de mapeo para acceder a las líneas fuente desde una línea de destino dada en el intervalo IGF.

Tabla 10: funciones de mapeo para cada modo

Tasa de bits	Modo	nT	Función de mapeo
9,6 kbps	WB	2	$m2a$
9,6 kbps	SWB	3	$m3a$
13,2 1 kbps	SWB	2	$m2b$
16,4 kbps	SWB	3	$m3b$
24,4 kbps	SWB	3	$m3c$
32,2 kbps	SWB	3	$m3c$
48,0 kbps	SWB	1	$m1$
16,4 kbps	FB	3	$m3d$
24,4 kbps	FB	4	$m4$
32,0 kbps	FB	4	$m4$
48,0 kbps	FB	1	$m1$
96,0 kbps	FB	1	$m1$
128,0 kbps	FB	1	$m1$

5 La función de mapeo $m1$ se define con:

$$m1(x) := \min Sb + 2t(0) - t(nB) + (x - t(0)), \text{ para } t(0) \leq x < t(nB) \quad (73)$$

10 La función de mapeo $m2a$ se define con:

$$m2a(x) := \begin{cases} \min Sb + (x - t(0)) & \text{para } t(0) \leq x < t(2) \\ \min Sb + (x - t(2)) & \text{para } t(2) \leq x < t(nB) \end{cases} \quad (74)$$

La función de mapeo $m2b$ se define con:

$$m2b(x) := \begin{cases} \min Sb + (x - t(0)) & \text{para } t(0) \leq x < t(4) \\ \min Sb + tF(32, f) + (x - t(4)) & \text{para } t(2) \leq x < t(nB) \end{cases} \quad (75)$$

La función de mapeo $m3a$ se define con:

$$m3a(x) := \begin{cases} \min Sb + (x - t(0)) & \text{para } t(0) \leq x < t(1) \\ \min Sb + tF(32, f) + (x - t(1)) & \text{para } t(1) \leq x < t(2) \\ \min Sb + tF(46, f) + (x - t(2)) & \text{para } t(2) \leq x < t(nB) \end{cases} \quad (76)$$

La función de mapeo $m3b$ se define con:

$$m3b(x) := \begin{cases} \min Sb + (x - t(0)) & \text{para } t(0) \leq x < t(4) \\ \min Sb + tF(48, f) + (x - t(4)) & \text{para } t(4) \leq x < t(6) \\ \min Sb + tF(64, f) + (x - t(6)) & \text{para } t(6) \leq x < t(nB) \end{cases} \quad (77)$$

La función de mapeo $m3c$ se define con:

$$m3c(x) := \begin{cases} \min Sb + (x - t(0)) & \text{para } t(0) \leq x < t(4) \\ \min Sb + tF(32, f) + (x - t(4)) & \text{para } t(4) \leq x < t(7) \\ \min Sb + tF(64, f) + (x - t(7)) & \text{para } t(7) \leq x < t(nB) \end{cases} \quad (78)$$

La función de mapeo $m3d$ se define con:

$$m3d(x) := \begin{cases} \min Sb + (x - t(0)) & \text{para } t(0) \leq x < t(4) \\ \min Sb + t(x - t(4)) & \text{para } t(4) \leq x < t(7) \\ \min Sb + (x - t(7)) & \text{para } t(7) \leq x < t(nB) \end{cases} \quad (79)$$

La función de mapeo m^4 se define con:

$$m4(x) := \begin{cases} \min Sb + (x - t(0)) & \text{para } t(0) \leq x < t(4) \\ \min Sb + tF(32, f) + (x - t(4)) & \text{para } t(4) \leq x < t(6) \\ \min Sb + (x - t(6)) & \text{para } t(6) \leq x < t(9) \\ \min Sb + (t(9) - t(8)) + (x - t(9)) & \text{para } t(9) \leq x < t(nB) \end{cases} \quad (80)$$

El valor f es el factor de transición apropiado, véase la tabla 11 y t^F se describe en la subcláusula 5.3.3.2.11.1.1.

Por favor, obsérvese que todos los valores $t(0), t(1), \dots, t(nB)$ se mapearán ya con la función t^F , como se describe en la subcláusula 5.3.3.2.11.1.1. Los valores para nB se definen en la tabla 8.

Las funciones de mapeo descritas en el presente documento se referirán en el texto como "función de mapeo m" asumiendo que se selecciona la función apropiada para el modo actual.

5.3.3.2.11.2 Elementos de entrada de IGF (TX)

El módulo codificador IGF espera los siguientes vectores e indicadores como entrada:

R : vector con parte real del espectro TCX actual X_M

I : vector con parte imaginario del espectro TCX actual X_S

P : vector con valores del espectro de potencia TCX X_P

$isTransiert$: indicador, que señala si la trama actual contiene un transitorio, véase la subcláusula 5.3.2.4.1.1

$isTCX10$: indicador, que señala una trama TCX 10

$isTCX20$: indicador, que señala una trama TCX 20

$isCelpToTCX$: indicador, que señala la transición de CELP a TCX; generar el indicador por prueba si la última trama fue CELP

$isIndepFlag$: indicador, que señala que la trama actual es independiente de la trama previa

Listado en la tabla 11, las siguientes combinaciones señalizadas a través de los indicadores $isTCX10$, $isTCX20$ y $isCelpToTCX$ se permiten con IGF:

Tabla 11: transiciones TCX, factor de transición f , longitud de ventana n

Tasa de bits/Modo	$isTCX10$	$isTCX20$	$isCelpToTCX$	Factor de transición f	Longitud de ventana n
9,6 kbps / WB	falso	verdadero	falso	1,00	320
	falso	verdadero	verdadero	1,25	400
9,6 kbps / SWB	falso	verdadero	falso	1,00	640
	falso	verdadero	verdadero	1,25	800

13,2 kbps / SWB	falso	verdadero	falso	1,00	640
	falso	verdadero	verdadero	1,25	800
16,4 kbps / SWB	falso	verdadero	falso	1,00	640
	falso	verdadero	verdadero	1,25	800
24,4 kbps / SWB	falso	verdadero	falso	1,00	640
	falso	verdadero	verdadero	1,25	800
32,0 kbps / SWB	falso	verdadero	falso	1,00	640
	falso	verdadero	verdadero	1,25	800
48,0 kbps / SWB	falso	verdadero	falso	1,00	640
	falso	verdadero	verdadero	1,00	640
	verdadero	falso	falso	0,50	320
16,4 kbps / FB	falso	verdadero	falso	1,00	960
	falso	verdadero	verdadero	1,25	1200
24,4 kbps / FB	falso	verdadero	falso	1,00	960
	falso	verdadero	verdadero	1,25	1200
32,0 kbps / FB	falso	verdadero	falso	1,00	960
	falso	verdadero	verdadero	1,25	1200
48,0 kbps / FB	falso	verdadero	falso	1,00	960
	falso	verdadero	verdadero	1,00	960
	verdadero	falso	falso	0,50	480
96,0 kbps / FB	falso	verdadero	falso	1,00	960
	falso	verdadero	verdadero	1,00	960
	verdadero	falso	falso	0,50	480
128,0 kbps / FB	falso	verdadero	falso	1,00	960
	falso	verdadero	verdadero	1,00	960
	verdadero	falso	falso	0,50	480

5.3.3.2.11.3 Funciones IGF en el lado de transmisión (TX)

Toda la declaración de la función asume que los elementos de entrada se proporcionan en forma de trama a trama.
5 Las únicas excepciones son dos tramas consecutivas TCX 10s, donde la segunda trama está codificada dependiendo de la primera trama.

5.3.3.2.11.4 Cálculo del factor de escala IGF

- 10 Esta subcláusula describe cómo el vector del factor de escala IGF $g(k)$, $k = 0, 1, \dots, nB-1$ se calcula en el lado de transmisión (TX).

5.3.3.2.11.4.1 Cálculo de valores complejos

- 15 En caso de que el espectro de potencia TCX P está disponible, se calculan los valores g del factor de escala IGF usando P :

$$E(k)_{\text{cplx, target}} := \sqrt{\frac{1}{t(k+1)-t(k)} \sum_{tb=t_k}^{t(k+1)-1} P(tb)}, \quad k = 0, 1, \dots, nB-1, \quad (81)$$

- 20 y se permite que $m: N \rightarrow N$ sea la función de mapeo que mapea el intervalo objetivo de IGF en el intervalo fuente de IGF descrito en la subcláusula 5.3.3.2.11.1.8. calcular:

$$E(k)_{\text{cplx, source}} := \sqrt{\frac{1}{t(k+1)-t(k)} \sum_{tb=t_k}^{t(k+1)-1} P(m(tb))}, \quad k = 0, 1, \dots, nB-1, \quad (82)$$

$$E(k)_{\text{real, source}} := \sqrt{\frac{1}{t(k+1)-t(k)} \sum_{tb=t_k}^{t(k+1)-1} R(m(tb))^2}, \quad k = 0, 1, \dots, nB-1, \quad (83)$$

- 25 donde $t(0), t(1), \dots, t(nB)$ ya se mapeará con la función tF , véase la subcláusula 5.3.3.2.11.1.1. y nB son el número

de bandas del factor de escala IGF, véase la tabla 8.

Calcular $g(k)$ con:

$$g(k) := \left\lfloor \frac{1}{2} + 4 \log_2 \left(\max \left(\frac{9}{10}, 16 \left(\frac{E(k)}{E(k)_{cplx,source}} \right) \right) \right) \right\rfloor, k = 0, 1, \dots, nB-1 \quad (84)$$

y limitar $g(k)$ en el intervalo $[0,91] \subset \mathbb{Z}$ con

$$g(k) \boxminus \max(0, g(k)), \quad (85)$$

Los valores $g(k)$, $k = 0, 1, \dots, nB-1$, se transmitirán al lado del receptor (RX) después de la compresión sin pérdidas adicionales con un codificador aritmético descrito en la subcláusula 5.3.3.2.11.8.

5.3.3.2.11.4.2 Cálculo de valores reales

Si el espectro de potencia TCX no está disponible calcular:

$$E(k)_{real} := \sqrt{\frac{1}{t(k+1)-t(k)} \sum_{tb=t(k)}^{t(k+1)-1} R(tb)^2}, k = 0, 1, \dots, nB-1 \quad (86)$$

donde $t(0), t(1), \dots, t(nB)$ ya se mapeará con la función t^F , véase la subcláusula 5.3.3.2.11.1.1. y nB son el número de bandas, véase la tabla 8.

Calcular $g(k)$ con:

$$g(k) := \left\lfloor \frac{1}{2} + 4 \log_2 \left(\max \left(\frac{9}{10}, 16 E(k)_{real} \right) \right) \right\rfloor, k = 0, 1, \dots, nB-1 \quad (87)$$

y limitar $g(k)$ en el intervalo $[0,91] \subset \mathbb{Z}$ con

$$\begin{aligned} g(k) &= \max(0, g(k)), \\ g(k) &\boxminus \min(91, g(k)). \end{aligned} \quad (88)$$

Los valores $g(k)$, $k = 0, 1, \dots, nB-1$, se transmitirán al lado del receptor (RX) después de la compresión sin pérdidas adicionales con un codificador aritmético descrito en la subcláusula 5.3.3.2.11.8.

5.3.3.2.11.5 Enmascaramiento tonal IGF

Con el fin de determinar qué componentes espectrales se deben transmitir con el codificador de núcleo, se calcula un enmascaramiento tonal. Por lo tanto, todo el contenido espectral significativo se identifica mientras que el contenido que es adecuado para la codificación paramétrica a través de IGF se cuantifica a cero.

5.3.3.2.11.5.1 Cálculo del enmascaramiento tonal IGF

En el caso de que el espectro de potencia TCX P no está disponible, se suprime el contenido espectral total anterior $t(0)$:

$$R(tb) := 0, t(0) \leq tb < t(nB) \quad (89)$$

donde R es el espectro TCX de valor real después de aplicar TNS y n es la longitud de ventana TCX actual.

En caso de que el espectro de potencia TCX P está disponible, calcular:

5

$$E_{HP} = \frac{1}{2t(0)} \sum_{i=0}^{t(0)-1} i P(i) \quad (90)$$

donde $t(0)$ es la primera línea espectral en el intervalo IGF.

10 Dado E_{HP} , aplicar el siguiente algoritmo:

Inicializar $last$ y $next$:

$last := R(t(0)-1)$

$next := \begin{cases} 0 & \text{if } P(t(0)-1) < E_{HP} \\ R(t(0)) & \text{else} \end{cases}$

15

para ($i = t(0)$; $i < t(nB)-1$; $i++$) {

si ($P(i) < E_{HP}$) {

$last := R(i)$

$R(i) := next$

$next := 0$

} además si ($P(i) \geq E_{HP}$) {

$R(i-1) := last$

$last := R(i)$

$next := R(i+1)$

25

}

}

si $P(t(nB)-1) < E_{HP}$, ajustar $R(t(nB)-1) := 0$

30 5.3.3.2.11.6 Cálculo de planitud espectral IGF

Tabla 12: número de mosaicos nT y ancho del mosaico wT

Tasa de bits	Modo	nT	wT
9,6 kbps	WB	2	$t(2)-t(0), t(nB)-t(2)$
9,6 kbps	SWB	3	$t(1)-t(0), t(2)-t(1), t(nB)-t(2)$
13,2 kbps	SWB	2	$t(4)-t(0), t(nB)-t(4)$
16,4 kbps	SWB	3	$t(4)-t(0), t(6)-t(4), t(nB)-t(6)$
24,4 kbps	SWB	3	$t(4)-t(0), t(7)-t(4), t(nB)-t(7)$
32,2 kbps	SWB	3	$t(4)-t(0), t(7)-t(4), t(nB)-t(7)$
48,0 kbps	SWB	1	$t(nB)-t(0)$
16,4 kbps	FB	3	$t(4)-t(0), t(7)-t(4), t(nB)-t(7)$
24,4 kbps	FB	4	$t(4)-t(0), t(6)-t(4), t(9)-t(6), t(nB)-t(9)$
32,0 kbps	FB	4	$t(4)-t(0), t(6)-t(4), t(9)-t(6), t(nB)-t(9)$
48,0 kbps	FB	1	$t(nB)-t(0)$

96,0 kbps	FB	1	$t(nB)-t(0)$
128,0 kbps	FB	1	$t(nB)-t(0)$

Para el cálculo de la planitud espectral de IGF se necesitan dos matrices estáticas $prevFIR$ y $prevIIR$, y ambos de tamaño nT son necesarios para mantener los estados de filtro sobre tramas. Adicionalmente, se necesita un indicador estático $wasTransient$ para guardar la información del indicador de entrada $isTransient$ de la trama anterior.

5.3.3.2.11.6.1 Restablecimiento de los estados de filtro

Los vectores $prevFIR$ y $prevIIR$ son ambas matrices estáticas de tamaño en el módulo IGF y ambas matrices se inicializan con ceros:

$$\left. \begin{array}{l} prevFIR(k) := 0 \\ prevIIR(k) := 0 \end{array} \right\} \text{for } k = 0, 1, \dots, nT - 1 \quad (91)$$

Esta inicialización se realizará

- con inicio de códec
- con cualquier conmutador de tasa de bits
- con cualquier conmutador de tipo códec
- con una transición de CELP a TCX, por ejemplo, $isCelpToTCX = true$
- si la trama actual tiene propiedades transitorias, por ejemplo, $isTransient = true$

5.3.3.2.11.6.2 Restablecimiento de los niveles de blanqueamiento actuales

El vector $currWLevel$ se inicializará con cero para todos los mosaicos,

$$currWLevel(k) = 0, \quad k = 0, 1, \dots, nT - 1 \quad (92)$$

- con inicio de códec
- con cualquier conmutador de tasa de bits
- con cualquier conmutador de tipo códec
- con una transición de CELP a TCX, por ejemplo, $isCelpToTCX = true$

5.3.3.2.11.6.3 Cálculo de los índices de plenitud espectral

Las siguientes etapas 1) a 4) se ejecutarán de modo consecutivo:

- 1) Actualizar búfer de nivel previo e inicializar niveles actuales:

$$\left. \begin{array}{l} prevWLevel(k) := currWLevel(k), \quad k = 0, 1, \dots, nT - 1 \\ currWLevel(k) := 0, \quad k = 0, 1, \dots, nT - 1 \end{array} \right\} \quad (93)$$

En caso de que $prevIsTransient$ o $isTransient$ es verdadera, aplicar

$$currWLevel(k) = 1, \quad k = 0, 1, \dots, nT - 1 \quad (94)$$

además, si la espectro de potencia P está disponible, calcular

$$tmp(k) := \frac{SFM(P, e(k), e(k+1))}{CREST(P, e(k), e(k+1))}, k = 0, 1, \dots, nT - 1 \quad (95)$$

con

$$e(k) := \begin{cases} t(0) & k = 0 \\ e(k-1) + wT(k) & k = 1, \dots, nT - 1 \end{cases} \quad (96)$$

donde SFM es una función de medición de la planitud espectral, descrita en la subcláusula 5.3.3.2.11.1.3 y $CREST$ es un factor de cresta descrito en la subcláusula 5.3.3.2.11.1.4.

Calcular:

$$s(k) := \min \left(2.7, tmp(k) + prevFIR(k) + \frac{1}{2} prevIIR(k) \right) \quad (97)$$

Después del cálculo del vector $s(k)$, los estados de filtro se actualizan con:

$$\begin{aligned} prevFIR(k) &= tmp(k), k = 0, 1, \dots, nT - 1 \\ prevIIR(k) &= s(k), k = 0, 1, \dots, nT - 1 \\ prevIsTransient &= isTransient \end{aligned} \quad (98)$$

- 2) Una función de mapeo $hT: N \times P \rightarrow N$ se aplica a los valores calculados para obtener un vector del índice de nivel de blanqueamiento $currWLevel$. La función de mapeo $hT: N \times P \rightarrow N$ se describe en la subcláusula 5.3.3.2.11.1.5.

$$currWLevel(k) = hT(s(k), k), k = 0, 1, \dots, nT - 1 \quad (99)$$

- 3) Con los modos seleccionados, véase la tabla 13, aplicar el siguiente mapeo final:

$$currWLevel(nT - 1) := currWLevel(nT - 2) \quad (100)$$

Tabla 13: modos para la etapa 4) mapeo

Tasa de bits	modo	mapeo
9,6 kbps	WB	aplicar
9,6 kbps	SWB	aplicar
13,2 kbps	SWB	NOP
16,4 kbps	SWB	aplicar
24,4 kbps	SWB	aplicar
32,2 kbps	SWB	aplicar
48,0 kbps	SWB	NOP
16,4 kbps	FB	aplicar
24,4 kbps	FB	aplicar
32,0 kbps	FB	aplicar
48,0 kbps	FB	NOP
96,0 kbps	FB	NOP
128,0 kbps	FB	NOP

Después de ejecutar la etapa 4) el vector del índice de nivel de blanqueamiento $currWLevel$ está listo para la transmisión.

5.3.3.2.11.6.4 Codificación de los niveles de blanqueamiento de IGF

Los niveles de blanqueamiento de IGF, definidos en el vector *currWLevel*, se transmiten usando 1 o 2 bits por mosaico. El número exacto de bits totales requeridos depende de los valores reales contenidos en *currWLevel* y el valor del indicador *isIndep*. El proceso detallado se describe en el pseudo-código a continuación:

```

5      isSame = 1;
      nTiles = nT ;
      k = 0;
      si (isIndep) {
10     isSame = 0;
      } además {
      para (k = 0; k < nTiles ; k++) {
          si (currWLevel(k) != prevWLevel(k)) {
15     isSame = 0;
          romper;
          }
      }

      si (isSame) {
20     write_bit(1);
      } además {
          si (!isIndep) {
              write_bit(0);
          }

25     encode_whitening_level(currWLevel(0)) ;
      para (k = 1; k < nTiles ; k++) {
          isSame = 1;
          si (currWLevel(k) != currWLevel(k-1)) {
30     isSame = 0;
          romper;
          }

          si (!isSame) {
              write_bit(1);
35     para (k = 1; k < nTiles ; k++) {
              encode_whitening_level(currWLevel(k)) ;
              }
          } además {
              write_bit(0);
40     }
      }

```

donde el vector *prevWLevel* contiene los niveles de blanqueamiento de la trama previa y la función *encode_whitening_level* se ocupa del mapeo real del nivel de blanqueamiento *currWLevel*(*k*) a un código binario. La

función se implementa según el pseudo-código siguiente:

```

      si (currWLevel(k) == 1) {
          write_bit(0);
      } además {
50     write_bit(1);
      si (currWLevel(k) == 0) {
          write_bit(0);
      } además {

```

```

write_bit(1);
}
}

```

5 5.3.3.2.11.7 Indicador de la planitud temporal IGF

La envolvente temporal de la señal reconstruida por el IGF se aplana en el lado del receptor (RX) según la información transmitida sobre la planitud de la envolvente temporal, que es un indicador de planitud IGF.

- 10 La planitud temporal se mide como la ganancia de predicción lineal en el dominio de frecuencia. En primer lugar, se realiza la predicción lineal de la parte real del espectro TCX actual y luego se calcula la ganancia de predicción η_{igf} :

$$\eta_{igf} = \frac{1}{\prod_{i=1}^8 (1 - k_i^2)} \quad (101)$$

- 15 donde k_i = coeficiente i -th PARCOR obtenido por la predicción lineal.

A partir de la ganancia de predicción η_{igf} y la ganancia de predicción η_{tns} descrito en la subcláusula 5.3.3.2.2.3, el indicador de la planitud temporal IGF $isIgfTemFlat$ se define como

$$isIgfTemFlat = \begin{cases} 1 & \eta_{igf} < 1.15 \text{ y } \eta_{tns} < 1.15 \\ 0 & \text{de otra manera} \end{cases} \quad (102)$$

20

5.3.3.2.11.8 Codificación sin ruido IGF

- 25 El vector de factor de escala de IGF g se codifica sin ruido con un codificador aritmético con el fin de escribir una representación eficiente del vector al flujo de bits.

El módulo utiliza las funciones comunes de codificador aritmético bruto de la infraestructura, que son proporcionados por el codificador de núcleo. Las funciones usadas son, $ari_encode_14bits_sign(bit)$ que codifica el valor bit , $ari_encode_14bits_ext(value, cumulativeFrequencyTable)$ que codifica $value$ desde un alfabeto de 27 símbolos ($SYMBOLS_IN_TABLE$) utilizando la tabla de frecuencia acumulativa, $cumulativeFrequencyTable$ que inicializa el
 30 codificador aritmético y $ari_finish_encoding_14bits()$ que finaliza el codificador aritmético.

5.3.3.2.11.8.1 Indicador de independencia IGF

- 35 El estado interno del codificador aritmético se restablece en caso de que el indicador $isIndepFlag$ tiene el valor $true$. Este indicador se puede ajustar a $false$ solo en los modos donde las ventanas TCX10 (véase la tabla 11) se usan para la segunda trama de dos tramas TCX 10 consecutivas.

5.3.3.2.11.8.2 Indicadores todo cero IGF

40

El indicador IGF todo cero señala que todos los factores de escala IGF son cero

$$allZero = \begin{cases} 1 & \text{si } g(k) = 0, \text{ para } 0 \leq k < nB \\ 0 & \text{además} \end{cases} \quad (103)$$

- 45 El indicador $allZero$ se escribe en el primer flujo de bits. En el caso de que el indicador es $true$, el estado del codificador se restablece y no se escriben más datos en el flujo de bits, de lo contrario el vector del factor de escala

codificado aritmético g sigue en el flujo de bits.

5.3.3.2.11.8.3 Funciones auxiliares que codifican IGF aritmético

5 5.3.3.2.11.8.3.1 La función de restablecimiento

Los estados codificadores aritméticos consisten en $t \in \{0,1\}$, y el vector $prev$, que representa el valor del vector g conservado de la trama anterior. Al codificar el vector g , el valor 0 para medios t significa que no hay trama previa disponible, por lo tanto $prev$ no está definido y no se usa. Por lo tanto, el valor 1 para medios t que existe una trama
 10 previa disponible $prev$ tiene datos válidos y se utiliza, este el caso solo en los modos en los que se utilizan ventanas TCX10 (véase la tabla 11) para la segunda trama de dos tramas TCX 10 consecutivas. Para restablecer el estado del codificador aritmético, es suficiente para establecer $t = 0$.

Si se ha establecido una trama $isIndepFlag$, el estado del codificador se restablece antes de codificar el vector del
 15 factor de escala g . Cabe señalar que la combinación $t = 0$ y $isIndepFlag = false$ es válida, y puede suceder para la segunda trama de dos tramas TCX 10 consecutivas, cuando la primera trama tenía $allZero=1$. En este caso particular, la trama no usa información de contexto de la trama previa (el vector $prev$), porque $t = 0$, y está codificado actualmente como una trama independiente.

20 5.3.3.2.11.8.3.2 La función arith_encode_bits

La función *arith_encode_bits* codifica un número entero sin signo X , de longitud $nBits$ bits, mediante la escritura de un bit a la vez.

```

25 arith_encode_bits(x, nBits)
{
  para (i = nBits - 1; i >= 0; --i) {
    bit = (x >> i) & 1;
    ari_encode_14bits_sign(bit);
30  }
}
```

5.3.3.2.11.8.3.2 Las funciones guardar y restaurar del estado del codificador

35 El guardado del estado del codificador se obtiene usando la función *iisIGFSCFEncoderSaveContextState*, que copia t y vector $prev$ en $tSave$ y vector $prevSave$, respectivamente. La restauración del estado del codificador se realiza usando la función complementaria *iisIGFSCFEncoderRestoreContextState*, que copia de nuevo $tSave$ y el vector $prevSave$ en t y el vector $prev$, respectivamente.

40 5.3.3.2.11.8.4 Codificación aritmética IGF

Se debe observar que el codificador aritmético debe ser capaz de contar bits solamente, por ejemplo, realizar la codificación aritmética sin escribir bits en el flujo de bits. Si se pide el codificador aritmético con una solicitud de recuento, mediante el uso del parámetro ajustado *doRealEncoding* ajustado a *false*, el estado interno del codificador
 45 aritmético se deberá guardar antes de la llamada a la función de nivel superior *iisIGFSCFEncoderEncode* y se restaurar y después de la llamada por la persona que llama. En este caso particular, los bits generados internamente por el codificador aritmético no se escriben en el flujo de bits.

La función *arith_encode_residual* codifica el residuo de predicción de valor entero X , usando la tabla de frecuencia
 50 acumulativa *cumulativeFrequencyTable*, y el desplazamiento de la tabla *tableOffset*. El desplazamiento de la tabla *tableOffset* se usa para ajustar el valor X antes de codificar, a fin de minimizar la probabilidad total de que un valor muy pequeño o muy grande sea codificado usando la codificación de escape, que es ligeramente menos eficiente. Los valores que están entre $MIN_ENC_SEPARATE=-12$ y $MAX_ENC_SEPARATE=12$, inclusive, están

codificados directamente usando la tabla de frecuencia acumulativa *cumulativeFrequencyTable*, y un tamaño de alfabeto de *SYMBOLS_IN_TABLE=27*.

Para el alfabeto anterior de símbolos *SYMBOLS_IN_TABLE*, los valores 0 y *SYMBOLS_IN_TABLE-1* están reservados como códigos de escape para indicar que un valor es demasiado pequeño o demasiado grande para caber en el intervalo predeterminado. En estos casos, el valor *extra* indica la posición del valor en una de las colas de la distribución. El valor *extra* se codifica usando 4 bits si está en el intervalo $\{0, \dots, 14\}$, o usando 4 bits con valor 15 seguido de 6 bits extra si está en el intervalo $\{15, \dots, 15+62\}$, o usando 4 bits con valor 15 seguido de 6 bits extra con valor 63 seguido por 7 bits extra si es mayor o igual que $15+63$. El último de los tres casos es principalmente útil para evitar la rara situación en la que una señal artificial construida a propósito puede producir una condición de valor residual inesperadamente grande en el codificador.

```

arith_encode_residual(x, cumulativeFrequencyTable, tableOffset)
{
15   x += tableOffset;
    si ((x >= MIN_ENC_SEPARATE) && (x <= MAX_ENC_SEPARATE)) {
        ari_encode_14bits_ext((x - MIN_ENC_SEPARATE) + 1, cumulativeFrequencyTable);

    volver;
20   } además si (x < MIN_ENC_SEPARATE) {
        extra = (MIN_ENC_SEPARATE - 1) - x;
        ari_encode_14bits_ext(0, cumulativeFrequencyTable);
    } además { /* x > MAX_ENC_SEPARATE */
        extra = x - (MAX_ENC_SEPARATE + 1);
25   ari_encode_14bits_ext(SYMBOLS_IN_TABLE - 1, cumulativeFrequencyTable);
    }

    si (extra < 15) {
        arith_encode_bits(extra, 4);
30   } además { /* extra >= 15 */
        arith_encode_bits(15, 4);
        extra -= 15;

        si (extra < 63) {
35   arith_encode_bits(extra, 6);
        } además { /* extra >= 63 */
            arith_encode_bits(63, 6);
            extra -= 63;
            arith_encode_bits(extra, 7);
40   }
        }
    }
}

```

La función *encode_sfe_vector* codifica el vector del factor de escala g , que consiste en valores enteros nB . El valor t y el vector *prev*, que constituye el estado del codificador, se usan como parámetros adicionales para la función. Cabe señalar que la función del nivel superior *iisIGFSCFEncoderEncode* debe llamar la función de inicialización del codificador aritmético común *ari_start_encoding_14bits* antes de llamar la función *encode_sfe_vector*, y también llamar la función de finalización del codificador aritmético *ari_done_encoding_14bits* después.

La función *quant_ctx* se usa para cuantificar un valor de contexto *ctx*, mediante su limitación a $\{-3, \dots, 3\}$, y se define como:

```

quant_ctx(ctx)
{
55   si (abs(ctx) <= 3) {
        volver ctx;
    }
}

```

```

    } además si (ctx > 3) {
        volver 3;
    } además { /* ctx < -3 */
        volver -3;
5    }
    }

```

Las definiciones de los nombres simbólicos indicados en los comentarios del pseudo-código, utilizados para calcular los valores de contexto, se enumeran en la siguiente tabla 14:

Tabla 14: definición de nombres simbólicos

La trama previa (cuando está disponible)	la trama actual
$a = \text{prev}[f]$	$x = g[f]$ (el valor para codificar)
$c = \text{prev}[f-1]$	$b = g[f-1]$ (cuando está disponible)
	$e = g[f-2]$ (cuando está disponible)

```

15 encode_sfe_vector(t, prev, g, nB)
    para (f = 0; f < nB; f++) {
        si (t == 0) {
            si (f == 0) {
                ari_encode_14bits_ext(g[f] >> 2, cf_se00);
20         arith_encode_bits(g[f] & 3, 2); /* LSBs como 2 bit sin procesar*/
            }
            además si (f == 1) {
                pred = g[f-1]; /* pred = b */
                arith_encode_residual(g[f] - pred, cf_se01, cf_off_se01);
            } además { /* f >= 2 */
25         pred = g[f-1]; /* pred = b */
                ctx = quant_ctx(g[f-1] - g[f-2]); /* Q(b - e) */
                arith_encode_residual(g[f] - pred, cf_se02[CTX_DESPLAZAMIENTO + ctx],
                    cf_off_se02[IGF_CTX_DESPLAZAMIENTO + ctx]);
            }
30     }
    además { /* t == 1 */
        si (f == 0) {
            pred = prev[f]; /* pred = a */
            arith_encode_residual(x[f] - pred, cf_se10, cf_off_se10);
35     } además { /* (t == 1) && (f >= 1) */
            pred = prev[f] + g[f-1] - prev[f-1]; /* pred = a + b - c */
            ctx_f = quant_ctx(prev[f] - prev[f-1]); /* Q(a - c) */
            ctx_t = quant_ctx(g[f-1] - prev[f-1]); /* Q(b - c) */
            arith_encode_residual(g[f] - pred,
40         cf_sel1[CTX_OFFSET + ctx_t][CTX_OFFSET + ctx_f],
            cf_off_sel1[CTX_OFFSET + ctx_t][CTX_OFFSET + ctx_f]);
        }
    }
45 }

```

Existen cinco casos en la función anterior, que dependen del valor de t y también de la posición f de un valor en el vector:

- 50 - cuando $t = 0$ y $f = 0$, el primer factor de escala de una trama independiente se codifica, mediante su división en los bits más significativos que están codificados usando la tabla de frecuencia acumulativa cf_se00 , y por lo menos dos bits significativos codificados directamente.
- cuando $t = 0$ y $f = 1$, el segundo factor de escala de una trama independiente se codifica (como un residuo

de predicción) usando la tabla de frecuencia acumulativa cf_se01 .

- cuando $t = 0$ y $f \geq 2$, se codifican el tercer y los siguientes factores de escala de una trama independiente (como residuos de predicción) usando la tabla de frecuencia acumulativa $cf_se02[CTX_OFFSET + ctx]$, determinado por el valor de contexto cuantificado ctx .
 - cuando $t = 1$ y $f = 0$, el primer factor de escala de una trama dependiente se codifica (como un residuo de predicción) usando la tabla de frecuencia acumulativa cf_se10 .
 - cuando $t = 1$ y $f \geq 1$, se codifican el segundo y los siguientes factores de escala de una trama dependiente (como residuos de predicción) usando la tabla de frecuencia acumulativa $cf_se11[CTX_OFFSET + ctx_t][CTX_OFFSET + ctx_f]$, determinado por los valores de contexto cuantificados ctx_t y ctx_f .
- 15 Cabe señalar que las tablas de frecuencias acumulativas predefinidas cf_se01 , cf_se02 , y los desplazamientos de la tabla cf_off_se01 , cf_off_se02 dependen del punto operativo actual e implícitamente de la tasa de bits, y se seleccionan del conjunto de opciones disponibles durante la inicialización del codificador para cada punto operativo determinado. La tabla de frecuencia acumulativa cf_se00 es común para todos los puntos operativos, y tablas de frecuencia acumulativa cf_se10 y cf_se11 , y los correspondientes desplazamientos de la tabla cf_off_se10 y cf_off_se11 también son comunes pero se usan solo para los puntos operativos correspondiente a las tasas de bits más grandes o iguales que 48 kbps, en el caso de las tramas TCX 10 dependientes (cuando $t = 1$).

5.3.3.2.11.9 Escritor de flujo de bits IGF

- 25 Los factores de escala IGF codificados aritméticamente, los niveles de blanqueamiento IGF y el indicador de planitud temporal IGF se transmiten consecutivamente al lado del decodificador a través del flujo de bits. La codificación de los factores de escala IGF se describe en la subcláusula 5.3.3.2.11.8.4. Los niveles de blanqueamiento de IGF se codifican como se presenta en la subcláusula 5.3.3.2.11.6.4. Finalmente, el indicador de planitud temporal IGF, representado como un bit, se escribe en el flujo de bits.
- 30 En el caso de una trama TCX20, es decir, $isTCX20 = true$, y ninguna solicitud de recuento se señala al escritor del flujo de bits, la salida del escritor del flujo de bits alimenta directamente al flujo de bits. En el caso de una trama TCX10 ($isTCX10 = true$), en la que dos subtramas están codificadas dependientemente dentro de una trama de 20 ms, la salida del escritor de flujo de bits para cada subtrama se escribe en un búfer temporal, que produce un flujo de bits que contiene la salida del escritor del flujo de bits para las subtramas individuales. El contenido de este búfer temporal se escribe finalmente en el flujo de bits.
- 35

REIVINDICACIONES

1. Codificador de audio para codificar una señal de audio que tiene una banda de frecuencia más baja y una banda de frecuencia superior, que comprende:

un detector (802) para detectar un componente de señal significativo en la banda de frecuencia más baja y una región espectral del pico en la banda de frecuencia superior de la señal de audio;

un conformador (804) para conformar la banda de frecuencia más baja usando la información de conformación para la banda de frecuencia más baja y para conformar la banda de frecuencia superior usando al menos una porción de la información de conformación para la banda de frecuencia más baja, en el que el conformador (804) se configura para atenuar adicionalmente los valores espectrales en la región espectral del pico detectada en la banda de frecuencia superior, cuando se ha detectado el componente de señal significativo en la banda de frecuencia más baja; y

una etapa de cuantificador y codificador (806) para cuantificar una banda de frecuencia más baja conformada y una banda de frecuencia superior conformada y para codificar por entropía los valores espectrales cuantificados de la banda de frecuencia más baja conformada y la banda de frecuencia superior conformada.

2. Codificador de audio según la reivindicación 1, que comprende además:

un analizador de predicción lineal (808) para derivar los coeficientes de predicción lineal para un marco de tiempo de la señal de audio mediante el análisis de un bloque de muestras de audio en el marco de tiempo, siendo las muestras de audio de banda limitada para la banda de frecuencia más baja,

en el que el conformador (804) está configurado para conformar la banda de frecuencia más baja usando los coeficientes de predicción lineal como la información de conformación, y

en el que el conformador (804) está configurado para usar al menos la porción de los coeficientes de predicción lineal derivados del bloque de muestras de audio de banda limitada a la banda de frecuencia más baja para conformar la banda de frecuencia superior en el marco de tiempo de la señal de audio.

3. Codificador de audio según la reivindicación 1 o 2, en el que el conformador (804) está configurado para calcular una pluralidad de factores de conformación para una pluralidad de sub-bandas de la banda de frecuencia más baja usando coeficientes de predicción lineal derivados de la banda de frecuencia más baja de la señal de audio,

en el que el conformador (804) está configurado para ponderar, en la banda de frecuencia más baja, los coeficientes espectrales en una sub-banda de la banda de frecuencia más baja usando un factor de conformación calculado para la correspondiente sub-banda, y

para ponderar los coeficientes espectrales en la banda de frecuencia superior usando un factor de conformación calculado para una de las sub-bandas de la banda de frecuencia más baja.

4. Codificador de audio según una de las reivindicaciones anteriores,

en el que el detector (802) está configurado para determinar una región espectral del pico en la banda de frecuencia superior, cuando al menos uno de un grupo de condiciones es válido, comprendiendo el grupo de condiciones al menos las siguientes:

una condición de distancia del pico (1104), y una condición de amplitud del pico (1106).

5. Codificador de audio según una de las reivindicaciones anteriores,

en el que el conformador (804) está configurado para atenuar al menos un valor espectral de la región espectral del pico detectada basándose en una amplitud espectral máxima en la banda de frecuencia superior o basándose en una amplitud espectral máxima en la banda de frecuencia más baja.

6. Codificador de audio según una de las reivindicaciones anteriores,

en el que el conformador (804) está configurado para conformar los valores espectrales de la región espectral del pico detectada basándose en:

una primera operación de ponderación (1702, 804a) usando al menos la porción de la información de

conformación para la banda de frecuencia más baja y una segunda operación de ponderación posterior (1704, 804b) usando una información de atenuación; o

5 una primera operación de ponderación usando la información de atenuación y una segunda operación de ponderación posterior usando al menos una porción de la información de conformación para la banda de frecuencia más baja, o

10 una operación de ponderación única usando una información de ponderación combinada derivada de la información de atenuación y al menos la porción de la información de conformación para la banda de frecuencia más baja.

7. Codificador de audio según la reivindicación 6,

15 en el que la información de conformación para la banda de frecuencia más baja es un conjunto de factores de conformación, asociándose cada factor de conformación con una sub-banda de la banda de frecuencia más baja,

20 en el que la al menos la porción de la información de conformación para la banda de frecuencia más baja usada en la operación de conformación para la banda de frecuencia mayor es un factor de conformación asociado con una sub-banda de la banda de frecuencia más baja que tiene una frecuencia central más alta de todas las sub-bandas en la banda de frecuencia más baja, o

25 en el que la información de atenuación es un factor de atenuación aplicada al al menos un valor espectral en la región espectral detectada o a todos los valores espectrales en la región espectral detectada o a todos los valores espectrales de la banda de frecuencia superior para los cuales la región espectral del pico se ha detectado mediante el detector (802) para un marco de tiempo de la señal de audio, o

30 en el que el conformador (804) está configurado para realizar la conformación de la banda de frecuencia más baja y superior sin ninguna atenuación adicional cuando el detector (802) no ha detectado ninguna región espectral del pico en la banda de frecuencia superior de un marco de tiempo de la señal de audio.

8. Codificador de audio según una de las reivindicaciones anteriores,

35 en el que la etapa de cuantificador y codificador (806) comprende un procesador del bucle de velocidad para estimar una característica del cuantificador de modo que se obtiene una tasa de bits predeterminada de una señal de audio codificada por entropía.

9. Codificador de audio según la reivindicación 8, en el que la característica del cuantificador es una ganancia global,

40 en el que la etapa de cuantificador y codificador (806) comprende:

45 un ponderador (1502) para ponderar valores espectrales conformados en la banda de frecuencia más baja y valores espectrales conformados de la banda de frecuencia superior por la misma ganancia global,

un cuantificador (1504) para cuantificar valores ponderados por la ganancia global; y

50 un codificador de entropía (1506) para codificar por entropía los valores cuantificados, en el que el codificador de entropía comprende un codificador aritmético o un codificador de Huffman.

10. Codificador de audio según una de las reivindicaciones anteriores, que comprende además:

55 un procesador de enmascaramiento tonal (1012) para determinar, en la banda de frecuencia superior, un primer grupo de valores espectrales para cuantificar y codificar por entropía y un segundo grupo de valores espectrales para codificar paramétricamente mediante un procedimiento de llenado de intervalos, en el que el procesador de enmascaramiento tonal está configurado para fijar el segundo grupo de valores espectrales a valores cero.

60 11. Codificador de audio según una de las reivindicaciones anteriores, que comprende además:

un procesador común (1002);

un codificador del dominio de frecuencia (1012, 802, 804, 806); y

un codificador de predicción lineal (1008),

en el que el codificador del dominio de frecuencia comprende el detector (802), el conformador (804) y la etapa de cuantificador y codificador (806), y

en el que el procesador común está configurado para calcular datos para usar mediante el codificador del dominio de frecuencia y el codificador de predicción lineal.

12. Codificador de audio según la reivindicación 11,

en el que el procesador común está configurado para remuestrear (1006) la señal de audio para obtener una banda de señal de audio remuestreada limitada a la banda de frecuencia más baja para un marco de tiempo de la señal de audio, y

en el que el procesador común (1002) comprende un analizador de predicción lineal (808) para derivar coeficientes de predicción lineal para el marco de tiempo de la señal de audio mediante el análisis de un bloque de muestras de audio en el marco de tiempo, siendo las muestras de audio de banda limitada para la banda de frecuencia más baja, o

en el que el procesador común (1002) está configurado para controlar que el marco de tiempo de la señal de audio es para representar por una salida del codificador de predicción lineal o una salida del codificador del dominio de frecuencia.

13. Codificador de audio según una de las reivindicaciones 11 a 12,

en el que el codificador del dominio de frecuencia comprende un conversor tiempo–frecuencia (1012) para convertir un marco de tiempo de la señal de audio en una representación de frecuencia que comprende la banda de frecuencia más baja y la banda de frecuencia superior.

14. Método para codificar una señal de audio que tiene una banda de frecuencia más baja y una banda de frecuencia superior, que comprende:

detectar (802) una región espectral del pico en la banda de frecuencia superior de la señal de audio;

conformar (804) la banda de frecuencia más baja de la señal de audio usando la información de conformación para la banda de frecuencia más baja y conformar (1702) la banda de frecuencia superior de la señal de audio usando al menos una porción de la información de conformación para la banda de frecuencia más baja, en el que la conformación de la banda de frecuencia superior comprende una atenuación adicional (1704) de un valor espectral en la región espectral del pico detectada en la banda de frecuencia superior; y

cuantificar una banda de frecuencia más baja conformada y una banda de frecuencia más alta conformada y codificar por entropía valores espectrales cuantificados a partir de la banda de frecuencia más baja conformada y la banda de frecuencia más alta conformada.

15. Un programa informático que comprende instrucciones que, cuando un ordenador o un procesador ejecuta el programa, hacen que el ordenador o el procesador lleven a cabo el método de la reivindicación 14.

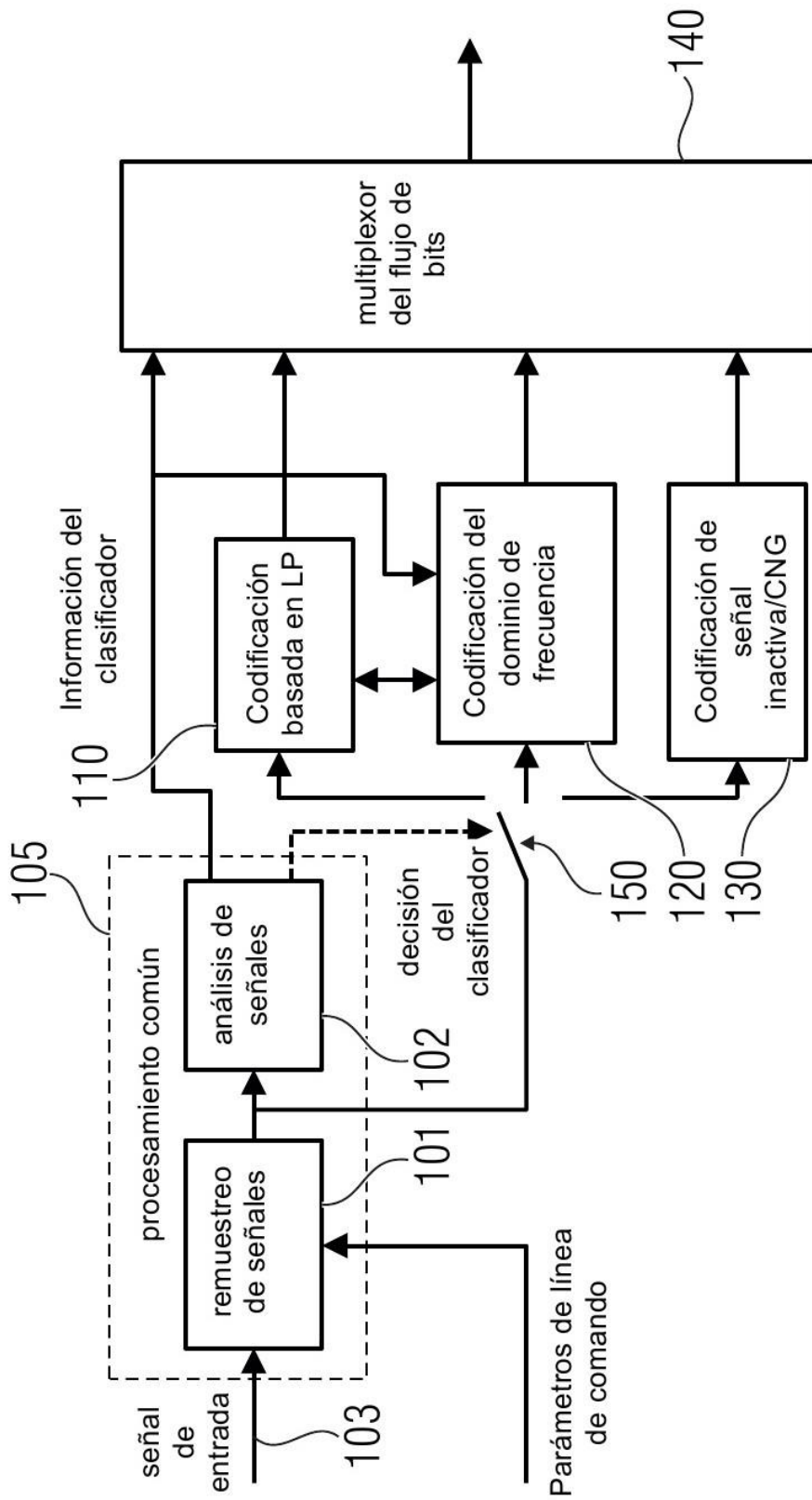


Fig. 1

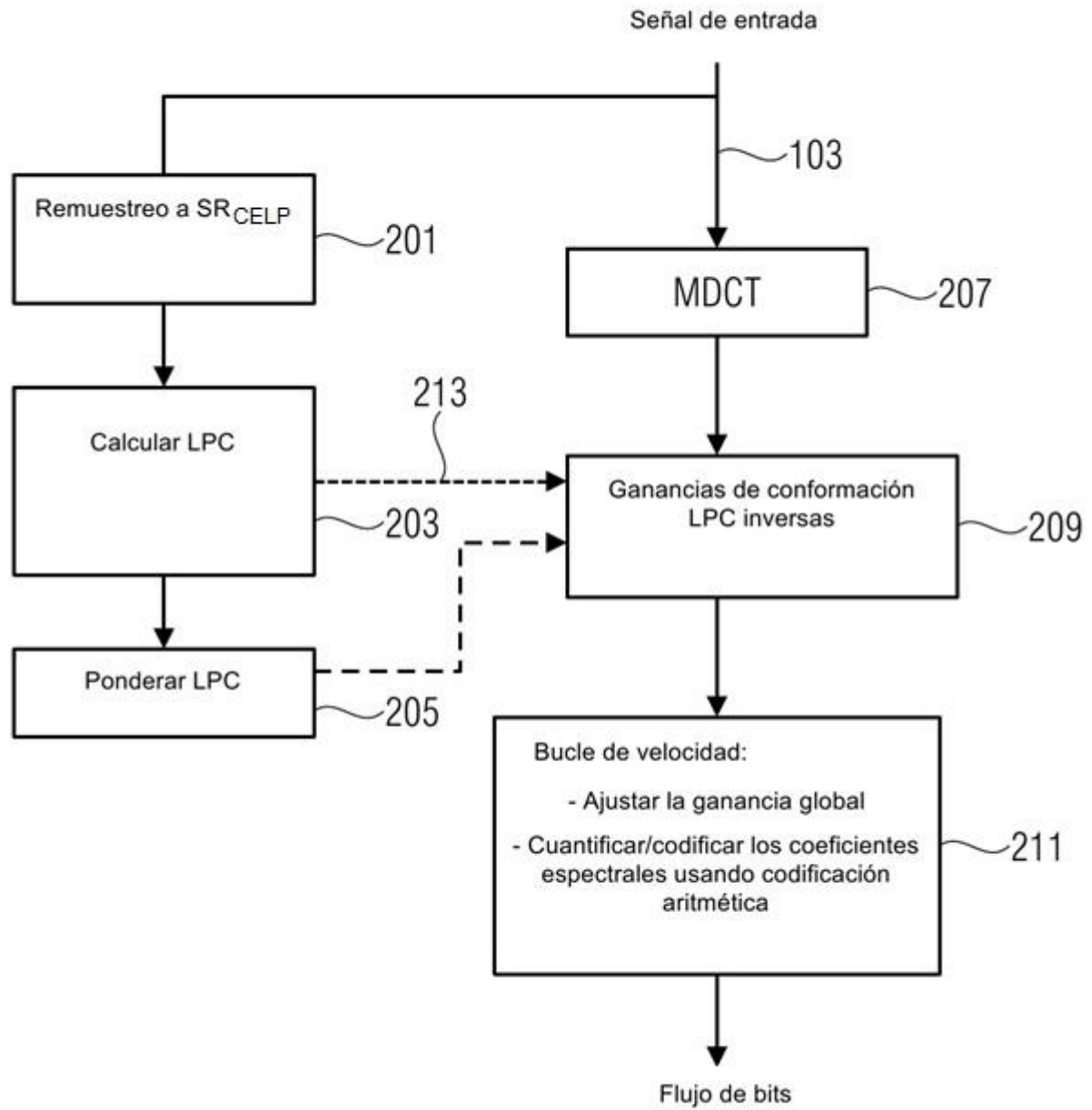


Fig. 2

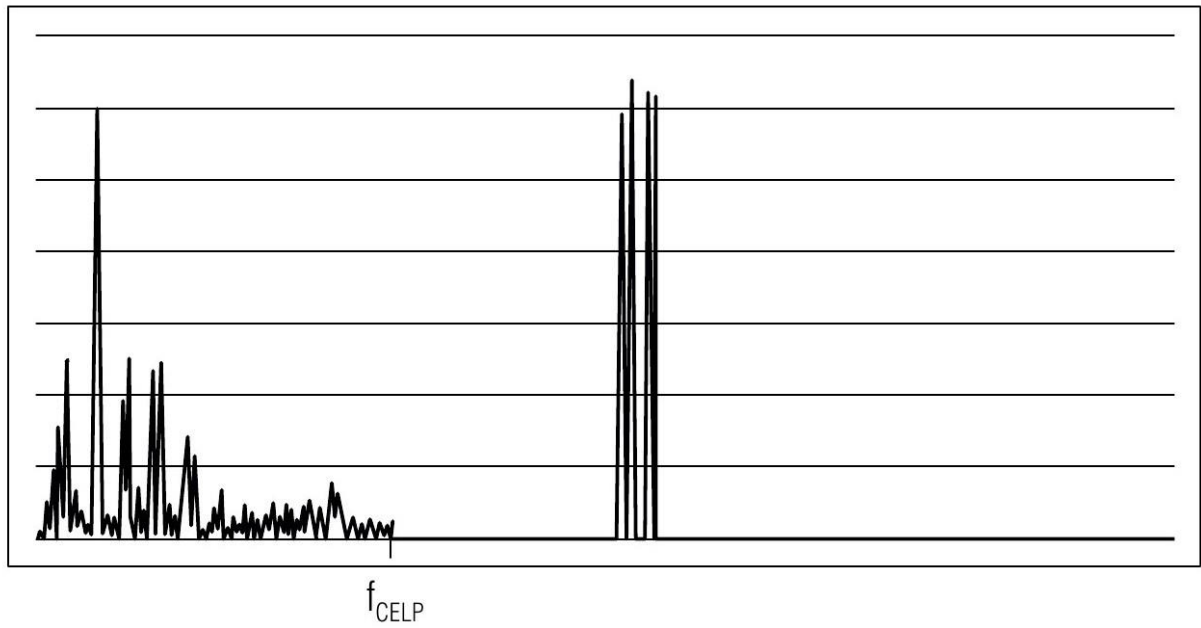


Fig. 3

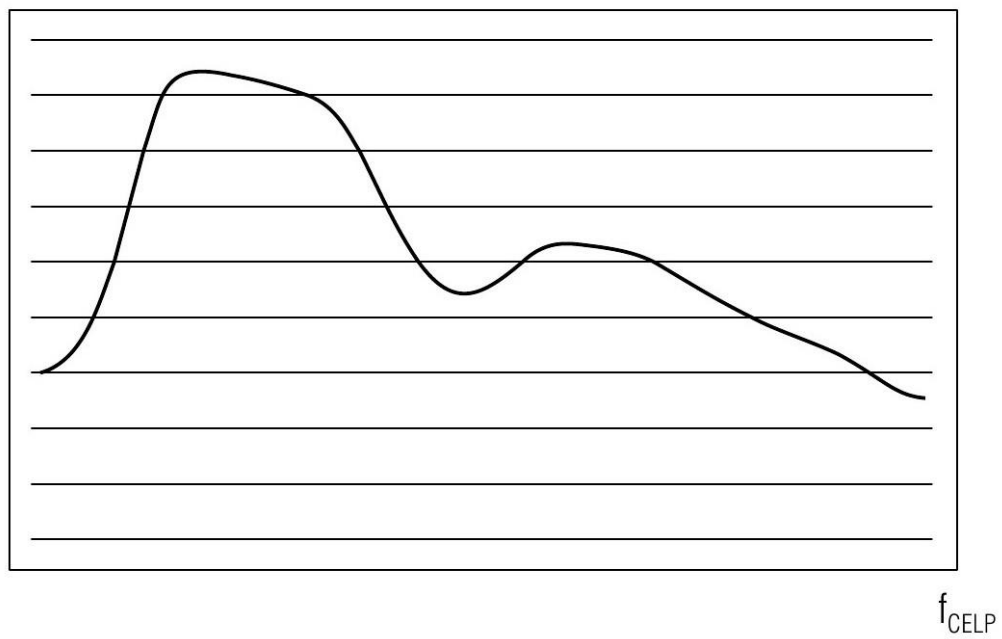


Fig. 4

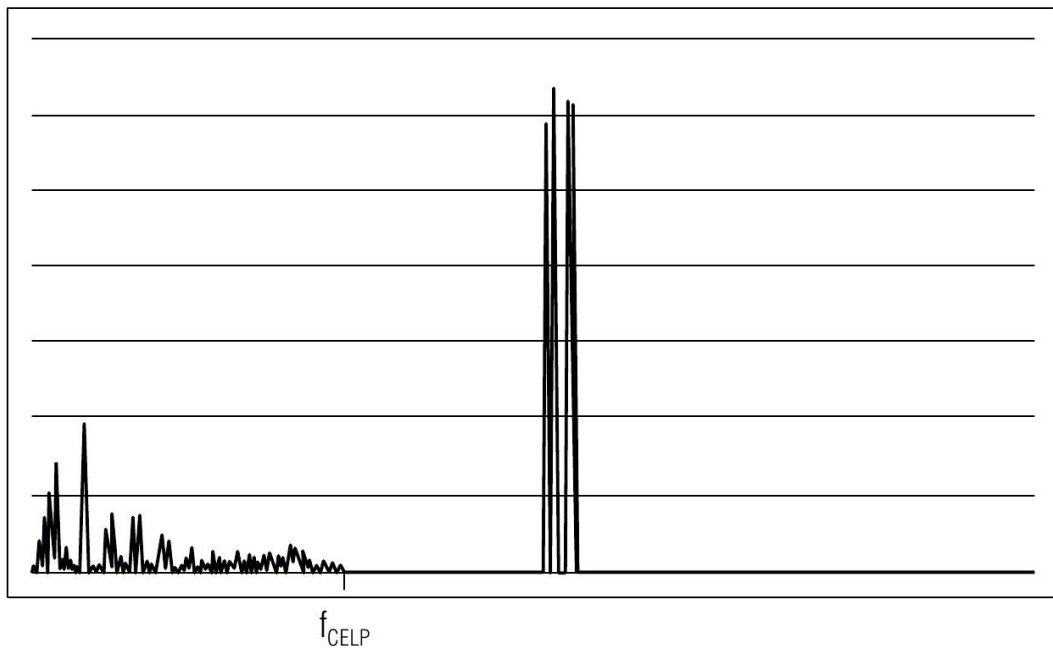


Fig. 5

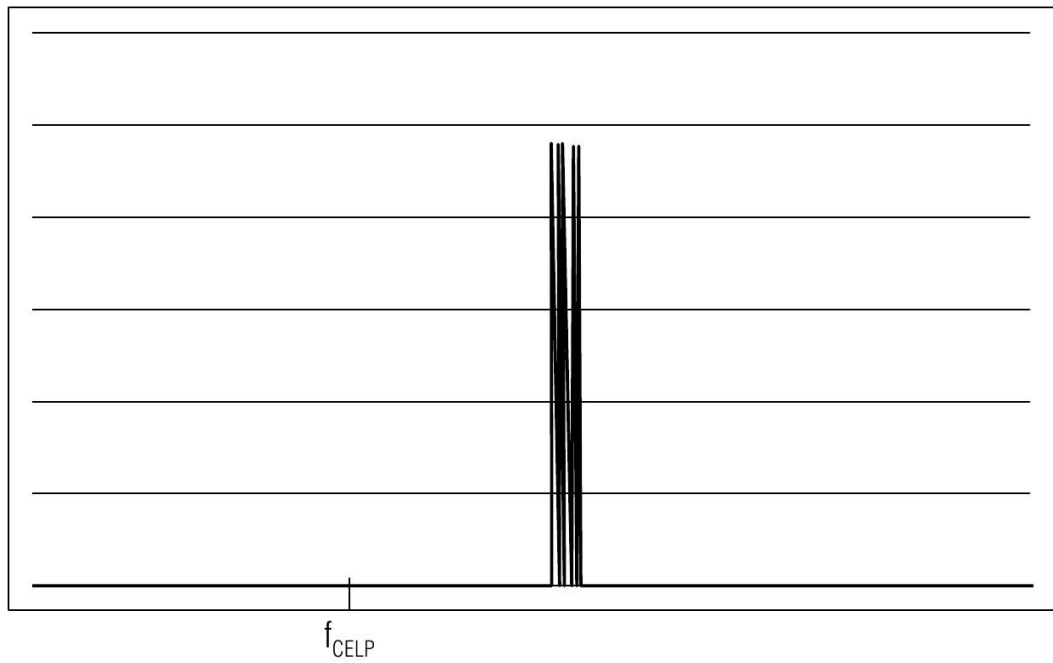


Fig. 6

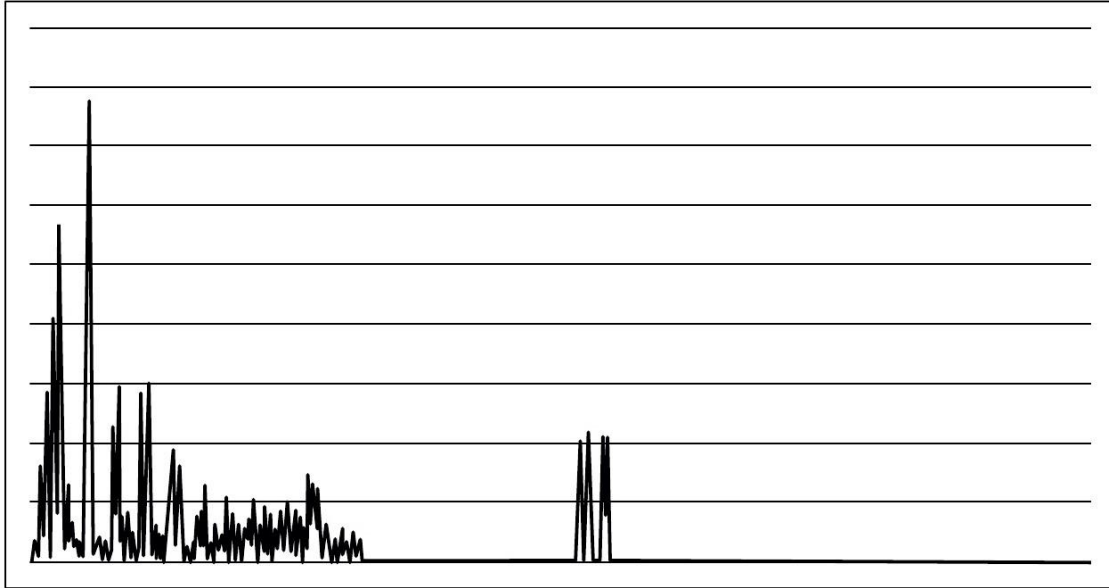


Fig. 7

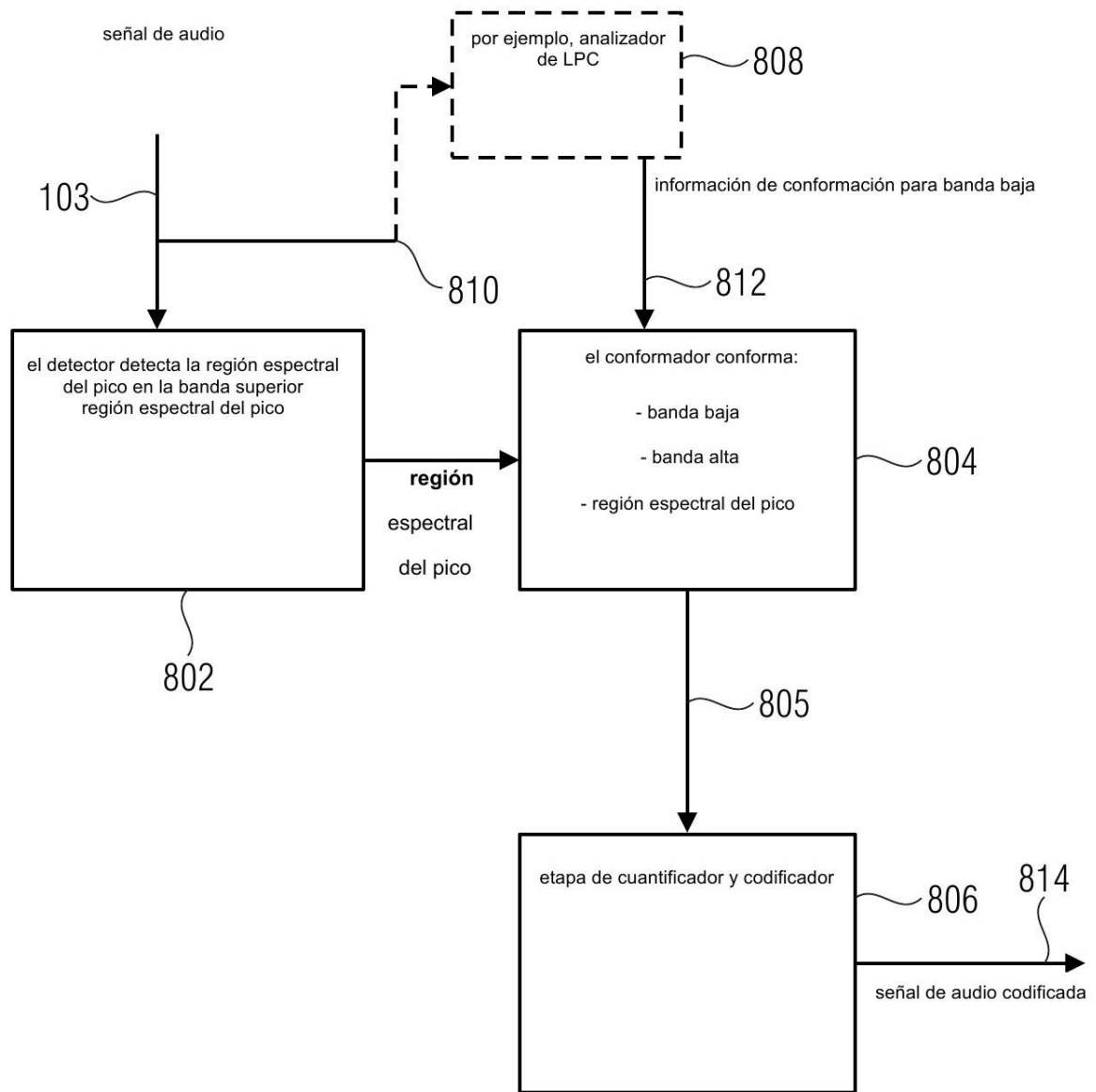
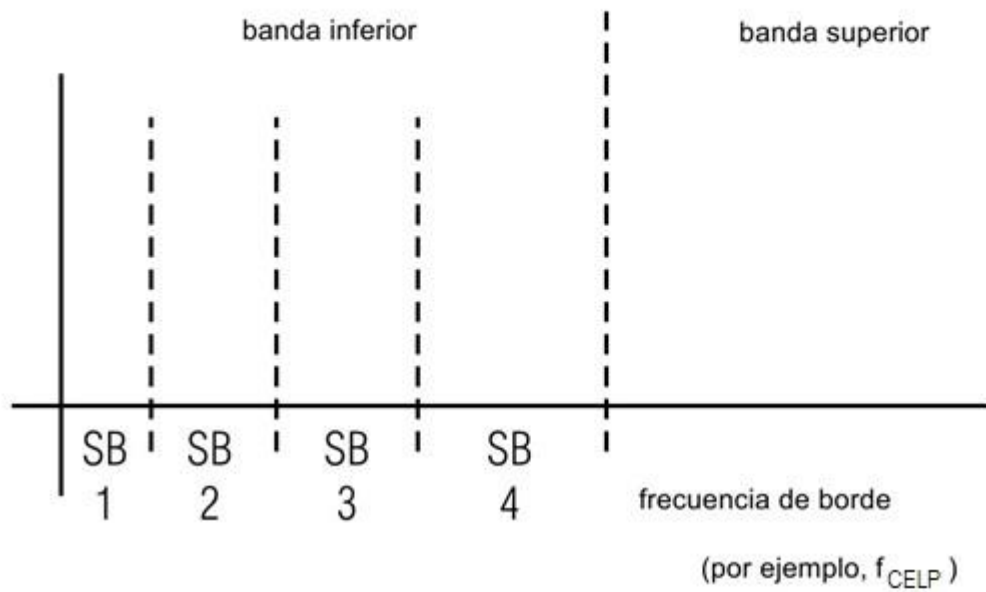


Fig. 8



- información de conformación para SB1 usada para conformación SB1
- información de conformación para SB2 usada para conformación SB2
- información de conformación para SB3 usada para conformación SB3
- información de conformación para SB4 usada para conformación SB4
- información de conformación para SB4 usada para conformación de banda superior

Fig. 9

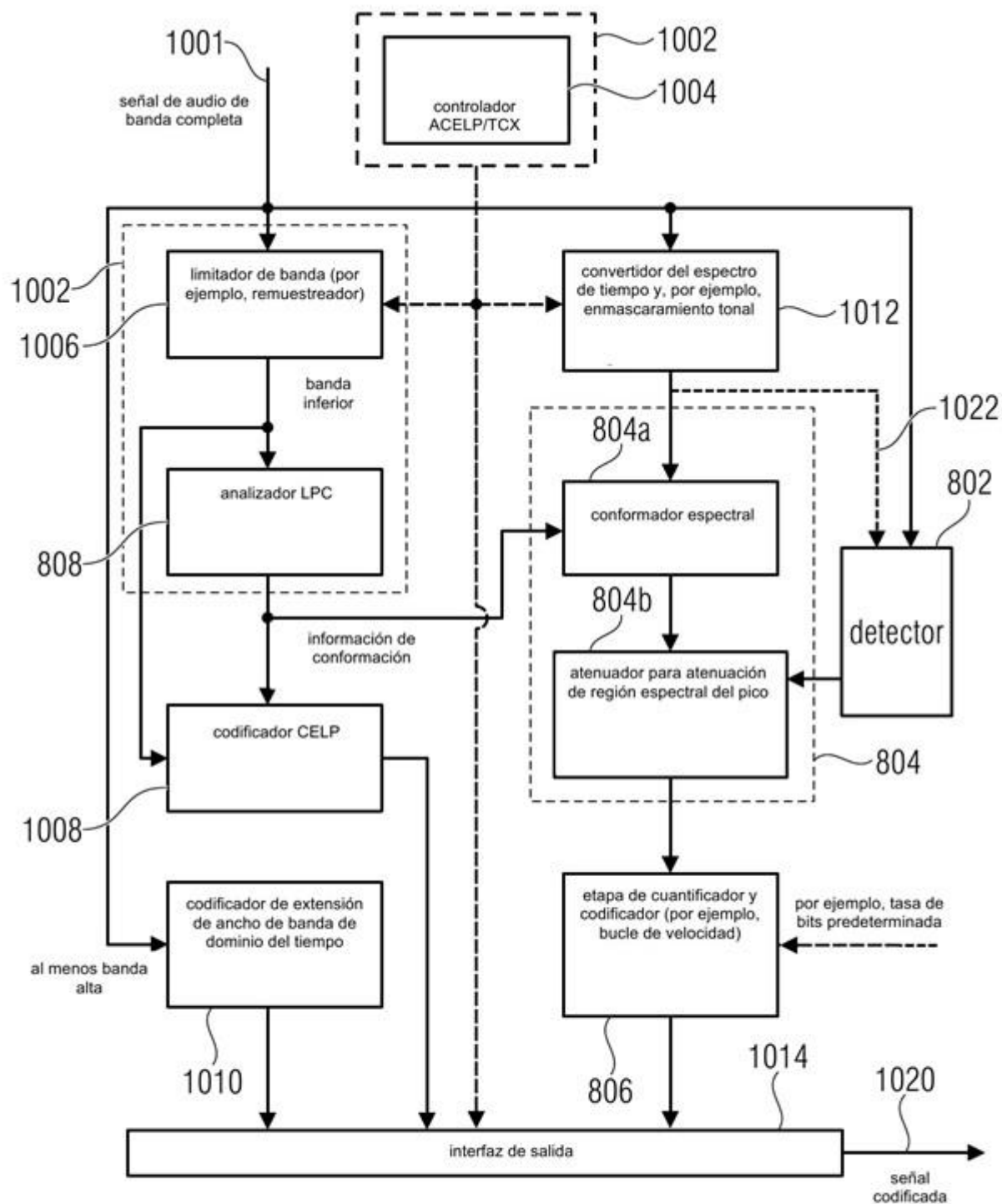


Fig. 10

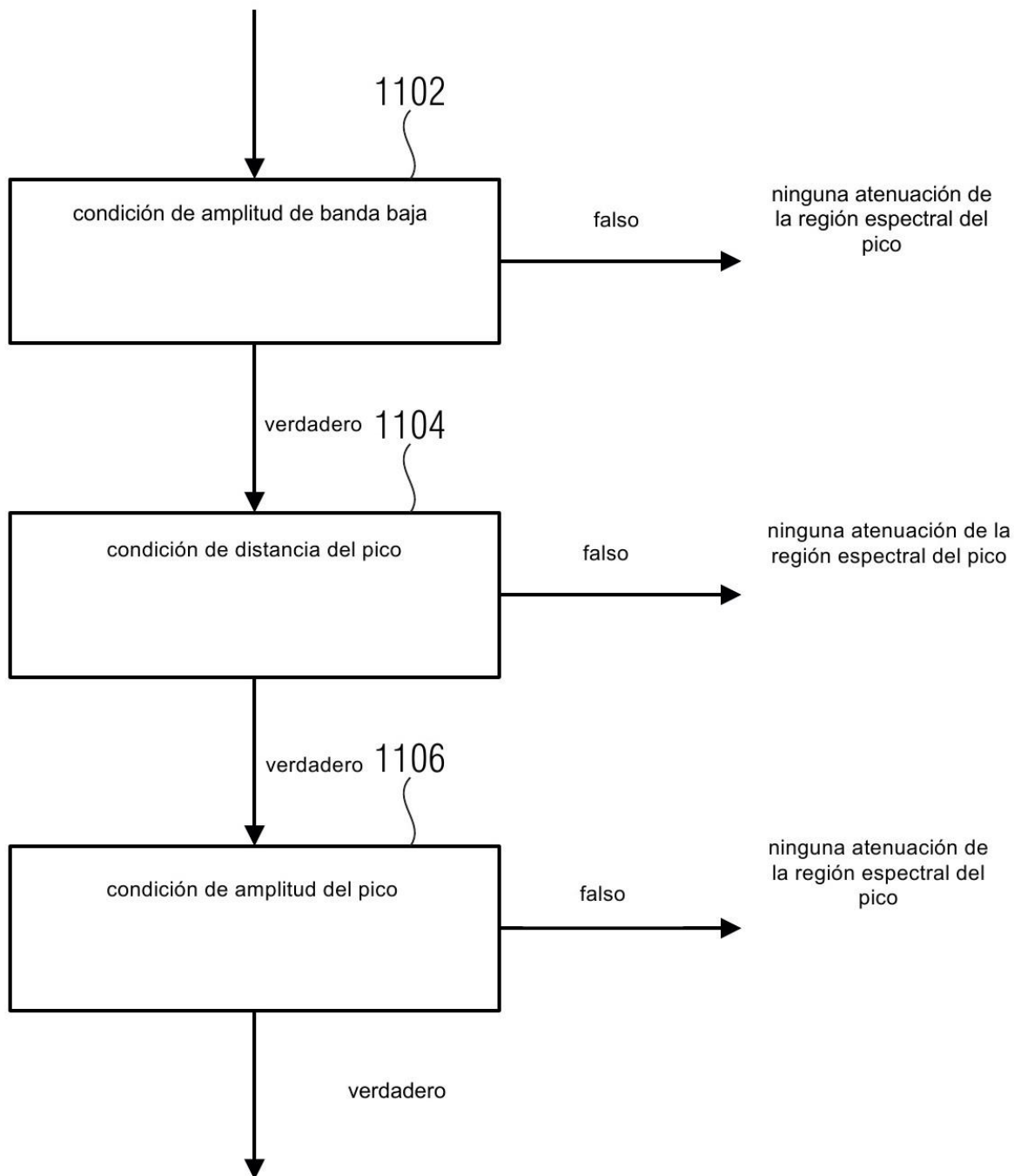


Fig. 11

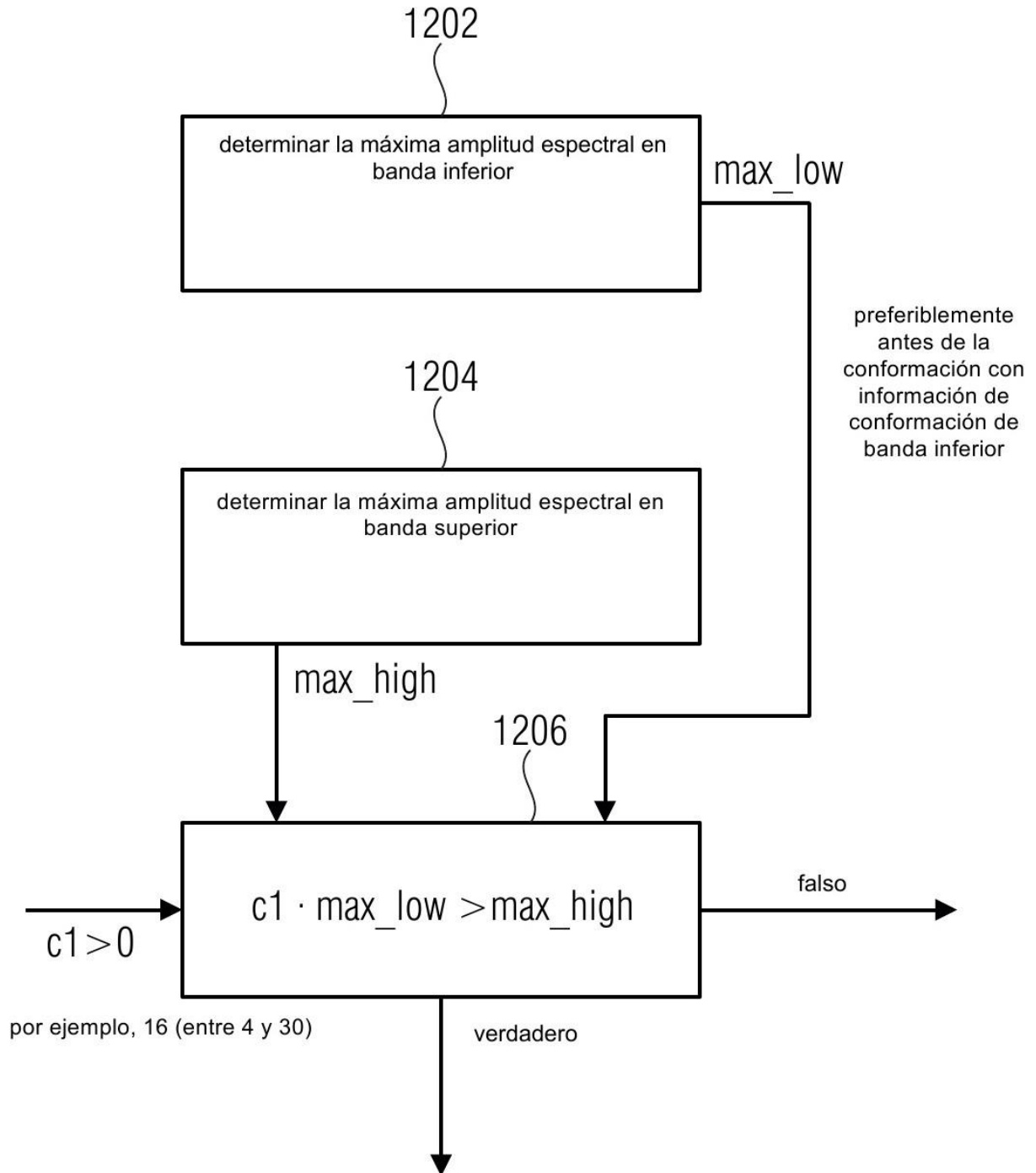


Fig. 12

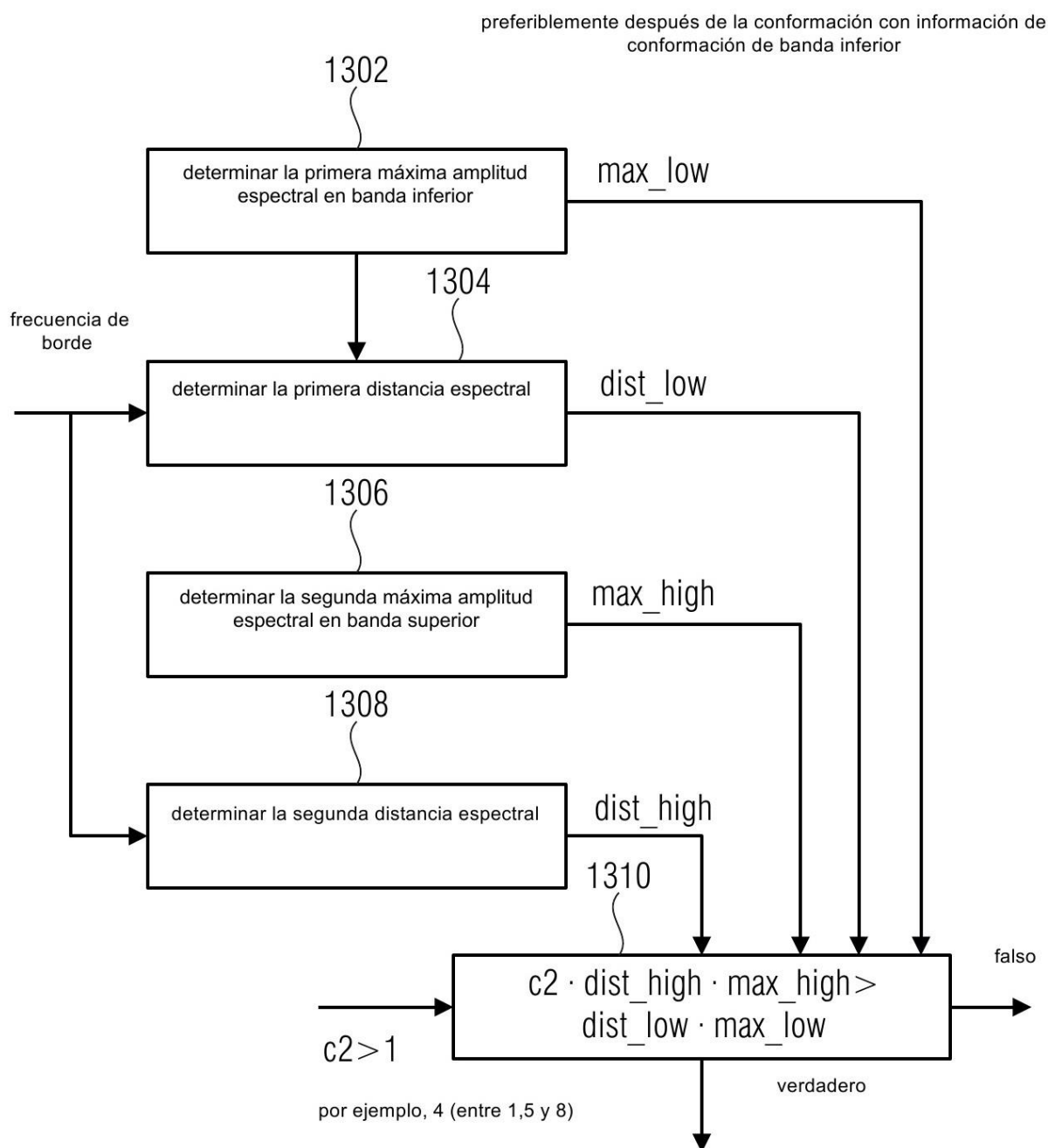


Fig. 13

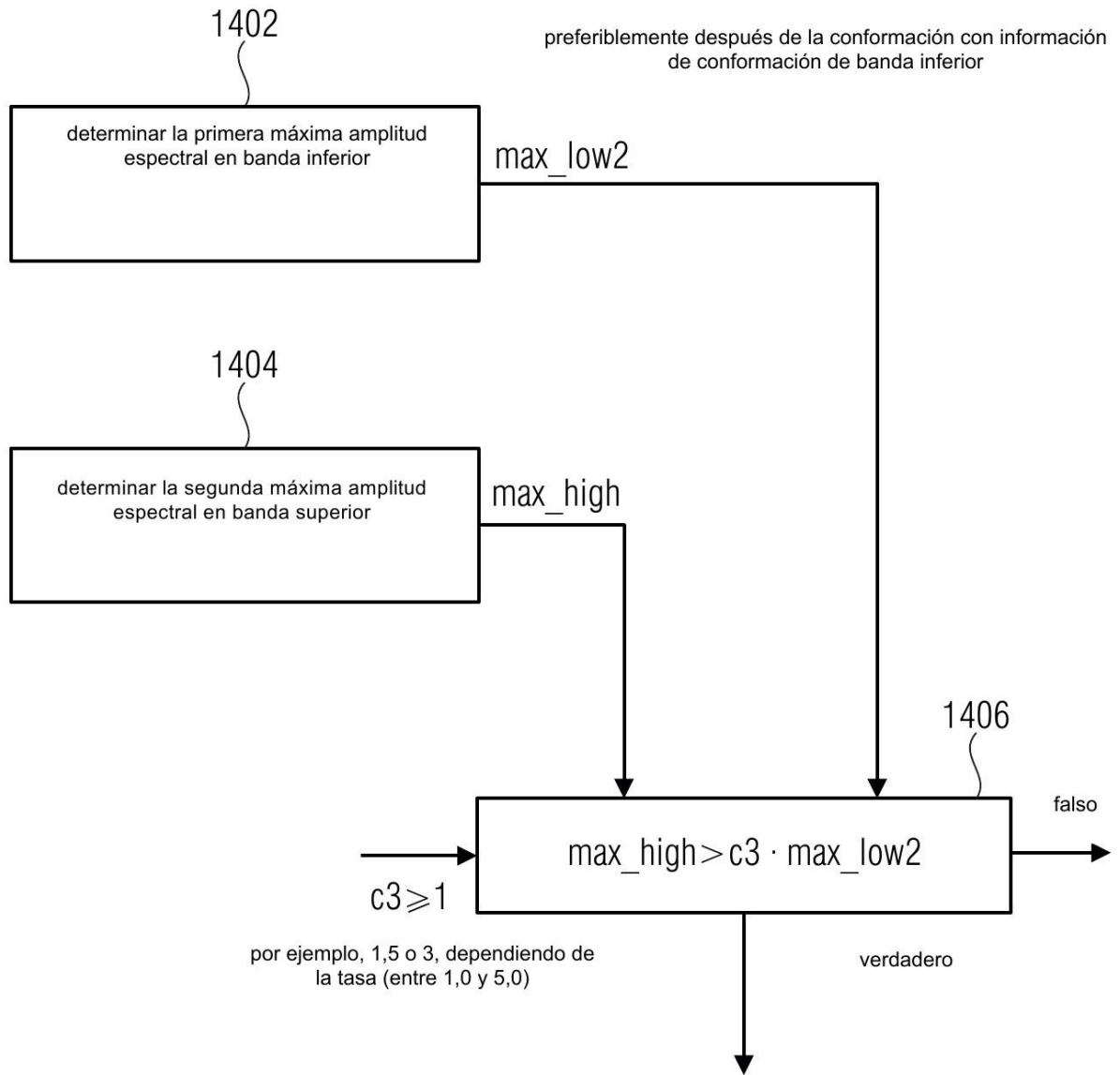


Fig. 14

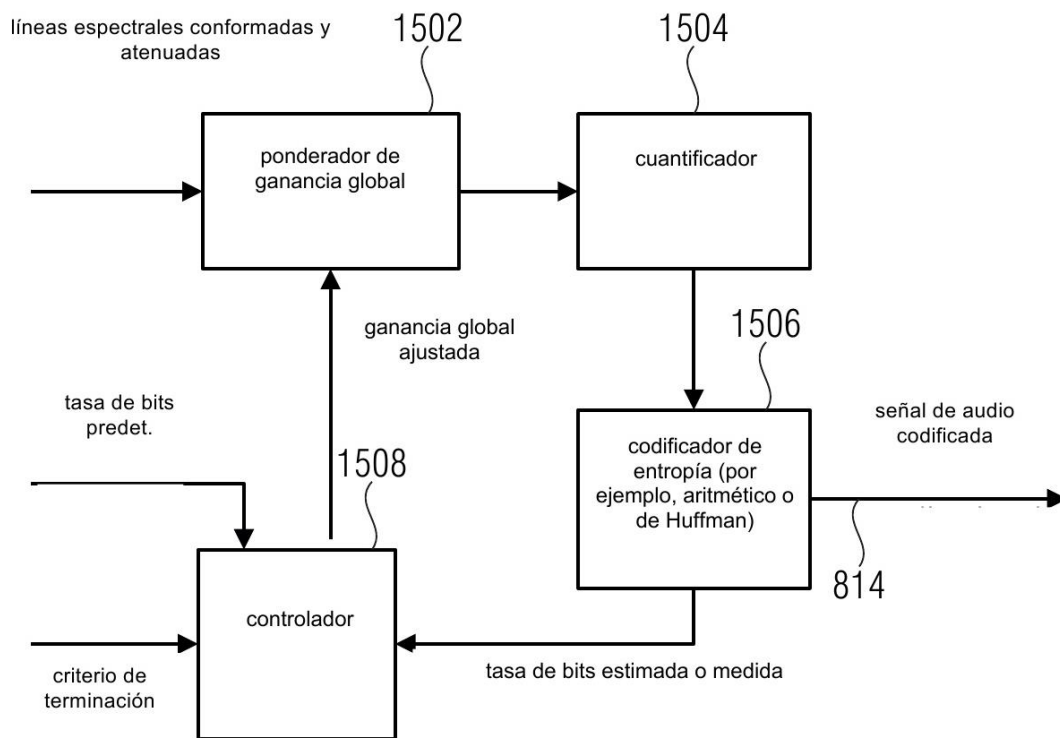


Fig. 15a

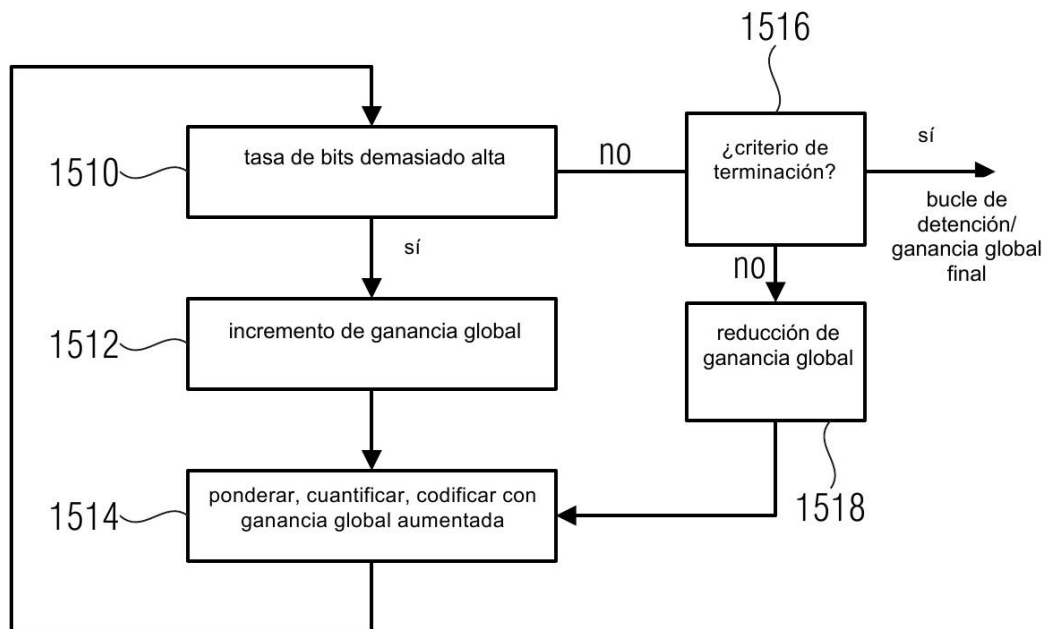


Fig. 15b

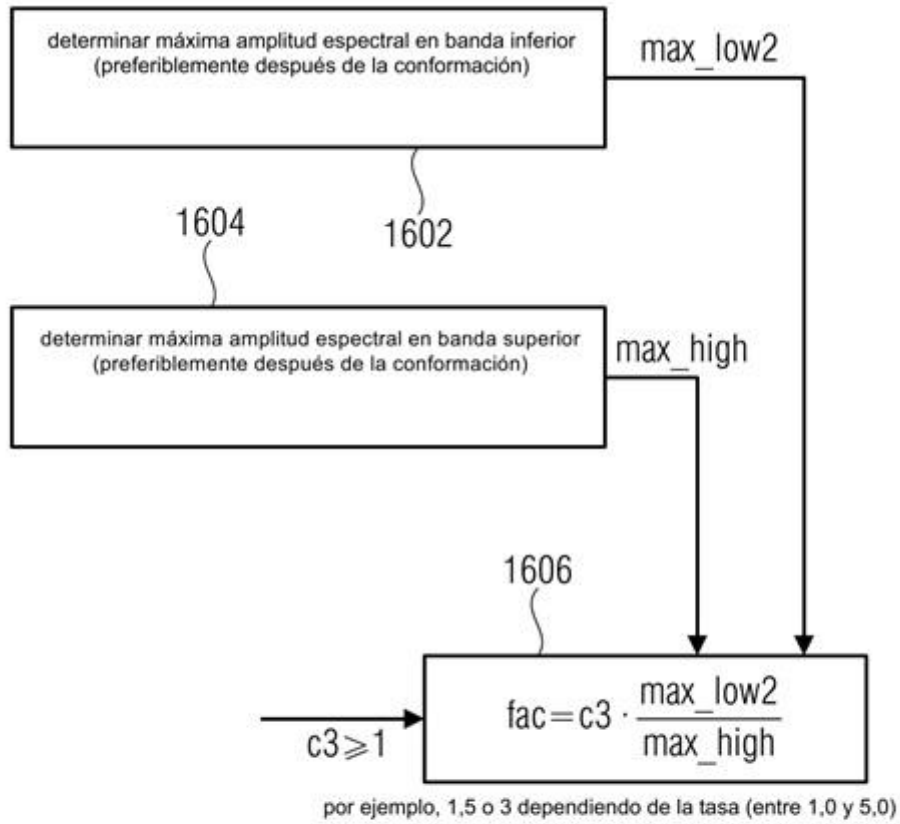


Fig. 16

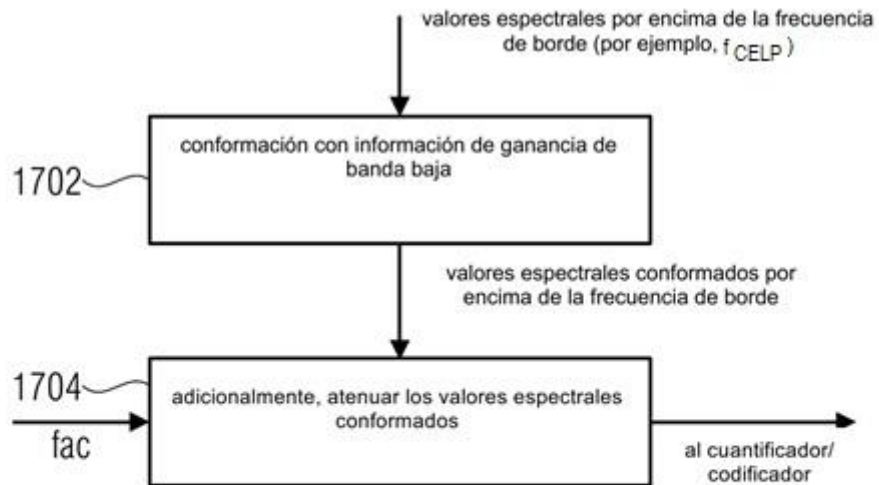


Fig. 17