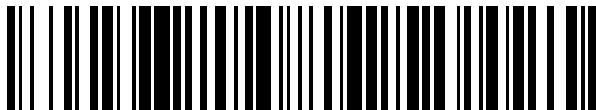


(19)



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS
ESPAÑA



(11) Número de publicación: **2 687 044**

(51) Int. Cl.:

G10L 19/16 (2013.01)
G10L 19/008 (2013.01)
H03G 7/00 (2006.01)

(12)

TRADUCCIÓN DE PATENTE EUROPEA

T3

(86) Fecha de presentación y número de la solicitud internacional: **27.03.2014 PCT/US2014/032079**

(87) Fecha y número de publicación internacional: **02.10.2014 WO14160895**

(96) Fecha de presentación y número de la solicitud europea: **27.03.2014 E 14723227 (6)**

(97) Fecha y número de publicación de la concesión europea: **13.06.2018 EP 2956937**

(54) Título: **Control de rango dinámico controlado por metadatos**

(30) Prioridad:

**29.03.2013 US 201361806628 P
24.07.2013 US 201361857966 P
16.10.2013 US 201361891687 P
26.03.2014 US 201414226596**

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
23.10.2018

(73) Titular/es:

**APPLE INC. (100.0%)
One Apple Park Way
Cupertino CA 95014, US**

(72) Inventor/es:

BAUMGARTE, FRANK M.

(74) Agente/Representante:

FORTEA LAGUNA, Juan José

ES 2 687 044 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

Control de rango dinámico controlado por metadatos

5 ASUNTOS RELACIONADOS

[0001] Esta solicitud reivindica el beneficio de las fechas de presentación anteriores de la solicitud provisional de Estados Unidos n.^o 61/806,628, presentada el 29 de marzo de 2013; la solicitud provisional de Estados Unidos n.^o 61/857,966 presentada el 24 de julio de 2013; y la solicitud provisional de Estados Unidos n.^o 61/891,687 presentada el 16 de octubre de 2013.

CAMPO

[0002] Un modo de realización de la invención en general se refiere a un procedimiento para codificar el Control/Compresión de Rango Dinámico (DRC) en una señal de audio. Además, el procedimiento descrito en el presente documento tiene en cuenta los requisitos de DRC de los nuevos códecs en desarrollo en MPEG-H (Audio 3D). También se describen otros modos de realización.

ANTECEDENTES

[0003] El control/compresión de rango dinámico (DRC) reduce el rango dinámico de una señal de audio en cierta medida al (1) hacer que las partes suaves de la señal de audio sean más fuertes; (2) hacer que las partes fuertes de la señal de audio se suavicen; o (3) tanto haciendo que las partes suaves sean más fuertes y haciendo que las partes fuertes sean más suaves. Un rango dinámico reducido puede ser deseable en varias situaciones, incluso para sistemas de reproducción de audio que solo pueden reproducir un pequeño rango dinámico manteniendo al mismo tiempo bajas distorsiones, entornos de escucha con sonidos que distraen y situaciones en las que el oyente no quiere distraer a los demás.

[0004] Aunque DRC es una característica importante para los códecs de audio actuales, varios códecs de audio recientes no soportan DRC. Por ejemplo, DRC está ausente en el estándar Codificación de audio y voz unificada (USAC) del Grupo de expertos en imágenes en movimiento (MPEG). La codificación de audio avanzada (Advanced Audio Coding, AAC) incorpora una herramienta DRC, pero esta herramienta DRC tiene inconvenientes, incluida la resolución temporal limitada y las distorsiones de aparición de alias. DRC también está presente en "Práctica recomendada de ATSC: Técnicas para establecer y mantener la sonoridad de audio para la televisión digital", documento A/85: 2011, Comité de sistemas avanzados de televisión, 25 de julio de 2011.

RESUMEN

[0005] Se propone un procedimiento de codificación de valores de ganancia de Control de Rango Dinámico, DRC, en un flujo de bits que representa un fragmento de contenido de programa de sonido de acuerdo con la reivindicación independiente 1.

[0006] El resumen anterior no incluye una lista exhaustiva de todos los aspectos de la presente invención. Se contempla que la invención incluye todos los sistemas y procedimientos que pueden practicarse a partir de todas las combinaciones adecuadas de los diversos aspectos resumidos anteriormente, así como los divulgados en la descripción detallada a continuación y particularmente señalados en las reivindicaciones presentadas con la solicitud. Tales combinaciones tienen ventajas particulares que no se enumeran específicamente en el resumen anterior.

50 BREVE DESCRIPCIÓN DE LOS DIBUJOS

[0007] Los modos de realización de la invención se ilustran a modo de ejemplo y no a modo de limitación en las figuras de los dibujos adjuntos en los que referencias similares indican elementos similares. Debe observarse que las referencias a "un" modo de realización de la invención en esta divulgación no son necesariamente al mismo modo de realización, y significan al menos uno.

La figura 1 muestra un diagrama de bloques del posprocesamiento de compresión que sigue a un descodificador de audio de acuerdo con un modo de realización.

La figura 2 muestra una representación de ganancia de Control/Compresión de Rango Dinámico (DRC) de acuerdo con un modo de realización.

La figura 3 muestra las características del codificador DRC de acuerdo con un modo de realización.

La figura 4 muestra un conjunto de ejemplos de metadatos de DRC generados en el transmisor de acuerdo con un modo de realización.

5 La figura 5 muestra ejemplos de tipos de interpolación de acuerdo con un modo de realización.

La figura 6 muestra dos modos de retardo de acuerdo con un modo de realización.

10 La figura 7 muestra la topología de filtros de cruce Linkwitz-Riley de acuerdo con un modo de realización.

15 La figura 8 muestra ejemplos de coeficientes de ponderación para un DRC de 4 bandas que utiliza un banco de filtros con 64 sub-bandas de acuerdo con un modo de realización.

15 La figura 9 muestra formas de ventana de Control/Compresión de Rango Dinámico (DRC) que se usarán para ciertas formas de ventana de descodificador correspondientes de acuerdo con un modo de realización.

20 La figura 10 muestra valores de ganancia de DRC aplicados a bloques pequeños separados de una ventana de acuerdo con un modo de realización.

20 La figura 11 muestra valores de ganancia de DRC aplicados a través de una señal de audio de acuerdo con un modo de realización.

DESCRIPCIÓN DETALLADA

25 [0008] Se explican ahora varios modos de realización de la invención con referencia a los dibujos adjuntos. Siempre que las formas, posiciones relativas y otros aspectos de las partes descritas en los modos de realización no estén claramente definidas, el alcance de la invención no está limitado solo a las partes mostradas, que están destinadas simplemente a efectos de ilustración. Además, aunque se exponen numerosos detalles, se entiende que algunos modos de realización de la invención pueden practicarse sin estos detalles. En otros casos, no se han mostrado en detalle circuitos, estructuras y técnicas bien conocidas para no entorpecer la comprensión de esta descripción.

30 [0009] Los sistemas de metadatos que incorporan metadatos de control/compresión de rango dinámico (DRC) en el flujo de bits/formato proporcionan varias ventajas sobre los sistemas que determinan los valores de ganancia de DRC en el extremo del oyente (es decir, en la reproducción). Estas ventajas incluyen (1) menor complejidad en la reproducción de la señal de audio; (2) la complejidad de DRC es un problema menor durante la reproducción, lo cual permite implementar procedimientos de DRC más complejos; y (3) un dispositivo de reproducción de audio en el extremo del oyente puede decidir si aplica el DRC. Aunque el uso de los sistemas de metadatos DRC ofrece varias ventajas, los sistemas tradicionales de metadatos de DRC, como los provistos por el Comité de sistemas de televisión avanzados (ATSC) y el Grupo de expertos en imágenes en movimiento (MPEG) también ofrecen varias desventajas.

40 [0010] Los sistemas tradicionales de metadatos de DRC (por ejemplo, aquellos definidos por los estándares ATSC y MPEG) soportan la compresión ligera y pesada como se muestra en la Tabla 1. En la mayoría de los casos, la frecuencia de actualizaciones de valores de ganancia de DRC es de una por trama. A una frecuencia de muestreo de 48 kHz, esto equivale a un intervalo de actualización entre 21 y 43 ms. AC-3 en modo de compresión ligera tiene una velocidad seis veces más rápida, a aproximadamente 5 ms a 48 kHz. Además, los valores de ganancia de DRC en estos sistemas tradicionales de metadatos de DRC se actualizan a frecuencias más bajas para frecuencias de muestreo de audio más bajas.

50 **Tabla 1: Los parámetros de DRC ganan metadatos en estándares de audio**

Estándar	ATSC: AC-3	MPEG: (HE) AAC
Compresión ligera	"Modo de línea"	MPEG "Control de rango dinámico"
Intervalo	-24... + 24 dB	-31,75... + +31,75 dB
Granularidad	0,25 dB	0,25 dB
Velocidad	1 valor por 256 muestras	1 valor por trama (1024 o 2048 muestras)
Compresión pesada	"Modo de RF"	"Valor de compresión" DVB
Intervalo	-48... + +48 dB	-48... + +48 dB
Granularidad	0,5 dB	0,5 dB
Velocidad	1 valor por 1536 muestras	1 valores por trama

[0011] La sintonización real de DRC sugiere que los cambios de ganancia deberían ser mucho más rápidos para ciertas señales de audio que lo que se puede lograr con los estándares actuales.

[0012] Otro problema con los estándares y sistemas DRC actuales, tales como MPEG-AAC y ATSC, surge del hecho de que la ganancia de DRC se aplica en el dominio de la frecuencia antes de aplicar un banco de filtros MDCT inverso. El banco de filtros MDCT es una transformación basada en la cancelación de alias de dominio del tiempo. La cancelación de alias no se puede lograr si se aplican diferentes valores de ganancia a bloques superpuestos consecutivos. Un cambio de ganancia puede producir distorsiones audibles, como ecos previos. Esto puede mostrarse fácilmente para una grabación de castañuelas.

[0013] En un modo de realización, las distorsiones de MDCT pueden evitarse si la ganancia de DRC se aplica en el dominio del tiempo después de que la señal de audio sea reconstruida por el descodificador. En el dominio de la frecuencia, la ganancia puede modificarse como máximo una vez por bloque largo o corto. Por el contrario, el enfoque de dominio del tiempo descrito en el presente documento soporta la resolución de tiempo superior deseada.

[0014] Los enfoques de dominio del tiempo actualmente soportan DRC multibanda (disponible mediante compresión ligera MPEG), pero los modos de realización analizados en el presente documento pueden mejorarse para soportar DRC multibanda. El esquema propuesto puede aumentar ligeramente la complejidad del descodificador debido a la interpolación y la aplicación de las ganancias de DRC. Sin embargo, estos inconvenientes parecen irrelevantes, ya que se pueden evitar distorsiones innecesarias, especialmente con contenidos que pueden tener una velocidad de transferencia de bits alta y que pueden reproducirse con un sistema de reproducción de alta calidad.

Herramienta DRC

Panorama general

[0015] La herramienta DRC descrita en el presente documento se basa en una codificación de ganancia de DRC unificada que puede aplicarse a una señal de audio de dominio del tiempo o sub-banda tal como las sub-bandas del banco de filtros QMF de un descodificador HE-AAC. La siguiente descripción cubre primero la aplicación de dominio del tiempo. Para la aplicación de dominio de sub-banda, solo se describen modificaciones al enfoque de dominio del tiempo.

Aplicación de dominio del tiempo

[0016] Esta sección describe cómo se aplica la herramienta de compresión dinámica a una señal de audio en el dominio del tiempo después de la descodificación, como se muestra en la Fig. 1. La Fig. 1 muestra un diagrama de bloques del posprocesamiento de compresión que sigue a un descodificador de audio. En un modo de realización, la parte descodificadora de la herramienta DRC está impulsada por metadatos que representan de manera eficiente las muestras y parámetros de ganancia de compresión para la interpolación. En algunos modos de realización, las muestras de ganancia se pueden actualizar tan rápido como sea necesario para representar cambios de ganancia con precisión hasta intervalos de actualización de al menos 1 ms. Si la ganancia es virtualmente constante, puede ser suficiente usar solo una única muestra de ganancia por trama DRC. Para minimizar la velocidad de transferencia de bits, el codificador puede elegir suficientes muestras de ganancia de DRC para garantizar una precisión suficiente de la ganancia de DRC reconstruida en una señal de audio después de la descodificación. En la práctica, esto puede significar intervalos de actualización más pequeños cuando hay cambios de ganancia más grandes.

[0017] Dado que el codificador proporciona solo valores de ganancia dispersamente muestreados, el descodificador puede aplicar interpolación para lograr una transición de ganancia suave entre las muestras. La frecuencia de muestreo de la ganancia interpolada es la frecuencia de muestreo de audio. La técnica de interpolación utilizada puede basarse en splines. Los valores interpolados de un segmento entre dos muestras de ganancia posteriores se obtiene de las dos muestras de ganancia en ambos extremos del segmento y su pendiente (derivada). Por lo tanto, al pasar de un segmento al siguiente, la primera derivada es continua, ya que ambos segmentos tienen la misma pendiente en el punto de transición.

[0018] La Fig. 2 ilustra la interpolación basándose en muestras cuantificadas de ganancia de DRC. La curva superior A muestra la ganancia de salida de DRC a una frecuencia de muestreo alta, como la frecuencia de muestreo de audio. Las muestras de la ganancia de DRC y la pendiente (es decir, círculos y flechas) se toman dispersamente basándose en una cuadrícula de tiempo uniforme. El gráfico inferior B muestra las coordenadas de muestra de ganancia cuantificada (tiempo y valor) y la pendiente cuantificada que se transmiten a la herramienta descodificadora DRC. La herramienta del descodificador interpola la curva de ganancia antes de que se aplique a la señal de audio como se ilustra mediante la línea punteada.

[0019] En un modo de realización, el intervalo de tiempo más pequeño posible para muestrear la curva de ganancia es un valor fijo entre 0,5 y 1,0 ms y el intervalo de tiempo más grande posible es una muestra de ganancia por trama DRC.

[0020] Además del modo spline descrito anteriormente, se puede usar un modo "simple" para transmitir solo un valor de ganancia de DRC por trama DRC sin parámetros de temporización y pendiente. Este modo es el más adecuado para tramas con ganancia de DRC virtualmente constante y consume la menor cantidad de bits.

[0021] Para las aplicaciones de la herramienta DRC en conjunto con un códec de audio, se proporcionan los siguientes parámetros para ajustar el tamaño de trama DRC y la resolución de tiempo para que el códec y el procesamiento de DRC se puedan realizar de la manera más eficiente en términos de complejidad y retardo. Los parámetros son:

- Tamaño de trama DRC en unidades del intervalo de muestra de audio
- *delta_t_min* en unidades del intervalo de muestra de audio
- modo de retardo

[0022] Estos parámetros tienen valores predeterminados, pero una especificación de códec puede sobrescribir los valores predeterminados.

Modificar la característica de DRC

[0023] La herramienta DRC soporta modificaciones de la ganancia de DRC descodificada de varias maneras:

- Factor de refuerzo
- Factor de compresión
- Característica personalizada de DRC

[0024] El factor de refuerzo es un valor entre 0 y 1 que se aplica a valores de ganancia positivos en dB para reducir la amplificación. El factor de compresión es un valor entre 0 y 1 aplicado a los valores de ganancia negativos para reducir la atenuación.

[0025] La configuración de DRC que incluye el codificador DRC se puede denominar en lo sucesivo "Descripción de muestra". Por ejemplo, las primeras seis características DRC estáticas se muestran en la **Fig. 3**. Conceptualmente, dicha característica de compresor estático se puede medir usando una sinusoida de 1 kHz, si la característica no está explícitamente disponible a partir del algoritmo de DRC. El nivel de sinusoida se define como -3 dBFS cuando el pico está a escala completa. Las características que se muestran en la **Fig. 3** tienen diversos grados de compresión hasta ninguna compresión en absoluto. En el caso más simple, se elige una característica de acuerdo con el efecto de compresión deseado. Cuando las sobrecargas también deben controlarse, por ejemplo para una mezcla descendente, opcionalmente solo se puede aplicar un limitador en el codificador, lo cual puede no tener un efecto de compresión estática. Por lo tanto, la característica con una ganancia constante de cero dB puede ser útil, si solo se aplica un limitador pero no DRC. En general, la ganancia de DRC transmitida en el flujo de bits puede ser el resultado de compresión o limitación dinámica o ambas.

[0026] Un ejemplo para la generación de metadatos de DRC en el transmisor se muestra en la **Fig. 4**. El DRC se configura basándose en la descripción de muestra de acuerdo con el borrador de ISO/IEC 14496-12. La señal de audio puede normalizarse en sonoridad a -31 LKFS antes de que entre en el DRC. Los metadatos de DRC pueden transmitirse junto con el flujo de bits de audio.

[0027] El receptor puede modificar la característica de DRC estática basándose en la característica de DRC del transmisor tal como se transmite en la descripción de la muestra y basándose en una característica de DRC objetivo personalizada. Comenzando con el valor de ganancia de DRC recibido (*gainQuant*), el receptor puede aplicar la característica de DRC del transmisor inverso y luego aplicar una nueva característica de DRC objetivo como se muestra en la **Tabla 2**:

Tabla 2: Mapa de ganancia de DRC de acuerdo con una característica de DRC objetivo

```
mapGain(gainQuant) {
    inLevel = inverseCompressorIO(gainQuant);
    outgain = targetCompressorIO(inLevel);
    return outgain;
}
```

[0028] La inversa de las características del transmisor 1 a 6 se puede calcular de acuerdo con la **Tabla 3** y la **Tabla 4**. Tenga en cuenta que la característica 2 no tiene una inversa útil porque la ganancia es siempre 0 dB.

Tabla 3: Cálculo de las características de DRC del codificador inverso 1 a 6

```

inverseCompressorIO(gainQuant) {
    drcInputLoudnessTarget = -31;
    gainDbMin = -32.0;
    gainDbMax = 32.0;
    if (gainQuant >= 0.0)
        tmp = gainQuant / pow(1 - pow(gainQuant / gainDbMax, expLo), 1 /
    expLo) ;
    else
        tmp = gainQuant / pow(1 - pow(gainQuant / gainDbMin, expHi), 1 /
    expHi);
    inLev = drcInputLoudnessTarget - tmp / ioRatio;
    return inLev;
}

```

5

Tabla 4: Parámetros de las características de DRC 1 a 6

Parámetro	DrcCharacteristic					
	1	2	3	4	5	6
ioRatio	0,8	0,0	0,2	0,4	0,6	1,0
expLo	6,0	9,0	9,0	9,0	9,0	5,0
ExpHi	8,0	12,0	12,0	12,0	12,0	6,0

[0029] Las características del objetivo DRC descodificador no se consideran estandarizadas. Se pueden definir opcionalmente mediante un implementador para lograr características de compresión personalizadas. Las siguientes secciones explican cómo se aplica la asignación de ganancia con más detalle.

[0030] La descripción de la muestra puede incluir un total de 11 características de DRC codificador. Para la compatibilidad con los sistemas existentes, la descripción de la muestra contiene, además de las primeras 6 características descritas anteriormente y que se muestran en la **Fig. 3**, otras cinco características que pueden estar disponibles en los sistemas ATSC, como se muestra en la **Tabla 5**.

Tabla 5: Índices de las características del codificador DRC 7 a 11

Índice de característica (DRC_characteristic)	Nombre del perfil ⁵
7	Película suave
8	Película estándar
9	Música suave
10	Música estándar
11	Voz

20

Segmentos Spline

[0031] La interpolación de la ganancia de DRC en el descodificador se basa en pares de muestras de ganancia. Cada par tiene información de coordenadas de ganancia (tiempo y valor en dB) y pendiente. El descodificador elegirá uno de los tres tipos de interpolación disponibles como se ilustra en la **Fig. 5**. En la mayoría de los casos, se elige la interpolación cúbica, que se muestra mediante el segmento spline A en la **Fig. 5**. Sin embargo, bajo ciertas condiciones, en lugar de eso se aplica una interpolación híbrida que combina la interpolación lineal y cuadrática como se muestra mediante los segmentos spline B y C en la **Fig. 5**. Para la interpolación híbrida, se inserta un nodo entre las dos coordenadas de ganancia (que se muestran como cuadrados en los segmentos spline B y C de la **Fig. 5**). En un lado de ese nodo, se aplica interpolación lineal y se aplica interpolación cuadrática en el otro. Este procedimiento está completamente especificado a continuación.

Tramas

[0032] La información de ganancia de DRC está organizada en tramas DRC. Cada trama DRC contiene datos DRC para generar la ganancia de DRC durante la duración de una trama DRC. La duración de trama DRC es constante para un elemento de audio dado y es un múltiplo del intervalo de muestra de audio. Las tramas DRC no se superponen. En la práctica, siempre que sea adecuado, se recomienda que el tamaño de trama DRC sea idéntico al tamaño de trama del códec para minimizar el retardo y la complejidad. Esta puede ser la configuración predeterminada.

Resolución de tiempo

- 10 **[0033]** La herramienta DRC utiliza una cuadrícula de tiempo uniforme para generar una representación dispersa de la ganancia de DRC. El espaciado de esta cuadrícula define la mayor resolución de tiempo disponible δ_t_{min} . La unidad de δ_t_{min} es un intervalo de muestra a la frecuencia de muestreo de audio. Por razones de complejidad, δ_t_{min} se elige como un múltiplo entero del intervalo de muestreo de audio con una duración correspondiente entre [0.5... 1.0] ms. Preferiblemente, δ_t_{min} es una potencia entera de 2, por lo que las velocidades de muestreo se pueden convertir de manera eficiente entre audio y DRC. Los valores predeterminados se calculan basándose en la siguiente ecuación:
- 15

$$\delta_t_{min} = 2^M \text{ donde } f_s 0,0005s < \delta_t_{min} \leq f_s 0,001s \quad (1)$$

- 20 **[0034]** En la ecuación anterior, la frecuencia de muestreo de audio f_s está en Hz, y el exponente M es un entero no negativo.

Anticipo en descodificador

- 25 **[0035]** El descodificador de herramientas DRC se puede hacer funcionar en uno de los dos modos de retardo. El modo de bajo retardo aplica inmediatamente la ganancia de DRC descodificada mientras que el modo predeterminado aplica la ganancia de DRC con un retardo de una trama DRC. El modo predeterminado soporta la interpolación de muestra de ganancia desde cualquier posición de la trama DRC actual a cualquier posición de la 30 siguiente trama DRC. El modo de bajo retardo requiere que una muestra de valor de ganancia esté ubicada al final de la trama DRC.

- [0036]** La Fig. 6 ilustra los dos modos de retardo: un retardo bajo A y un retardo predeterminado B. El diagrama superior A muestra que cada trama DRC tiene un nodo spline al final de la trama, de modo que toda la curva de ganancia de DRC para esa trama se puede generar inmediatamente mediante interpolación. El diagrama inferior B muestra que la curva de ganancia interpolada se aplica con un retardo de una trama DRC, ya que la interpolación para la trama n-1 (representada por un círculo) solo puede completarse después de que se reciba el primer nodo de la trama n (representado por un cuadrado).

- 40 **[0037]** Para los códecs de percepción comunes, el modo de retardo B predeterminado no requerirá un retardo de descodificador adicional. El retardo ya es necesario debido a la operación de agregación de superposición.

[0038] El modo de bajo retardo puede ser adecuado para descodificadores que no tienen retardo inherente, como un retardo debido a la agregación de superposición. Por ejemplo, este es el caso de algunos códecs sin pérdida.

Descodificación

[0039] El proceso de descodificación de las coordenadas de ganancia y pendientes consiste en la siguiente secuencia de tareas:

- 50 • Recopilar la información de configuración de DRC
• Analizar el flujo de bits DRC
• Aplicar las tablas de códigos, incluida la descodificación Huffman para descodificar los valores cuantificados
• Deshacer la codificación diferencial
- 55 **[0040]** La información de configuración de DRC puede ser parte de la descripción de la muestra. La información de configuración de DRC puede incluir los siguientes parámetros relevantes para la descodificación:
- 60 • El número de secuencias de ganancia: nDrcGainSequences
• La asignación de una secuencia de ganancia a cada canal. Los canales que usan la misma secuencia se conocen como grupos de canales. El número total de grupos es nDrcChannelGroups
• El número de bandas de DRC en un grupo: nDrcBands

[0041] Dados estos parámetros, el flujo de bits DRC se puede analizar de acuerdo con la **Tabla 20** y la **Tabla 21**. A continuación, el pseudo código se limita a una secuencia de ganancia para mayor claridad. Para el caso general, se puede agregar un bucle externo para procesar cada secuencia de ganancia en la **Tabla 6** y la **Tabla 9**.

5 [0042] Los valores codificados se descodifican aplicando la **Tabla 22** y la **Tabla 25**. Esta operación se expresa en la **Tabla 6** mediante las pseudo-funciones *decode_initial_gain()*, *decode_delta_gain()*, *decode_time_delta()* y *decode_slope()*. Los valores codificados diferencialmente se convierten en valores absolutos de acuerdo con la **Tabla 6**. El resultado descodificado se representa mediante los valores de ganancia $gDRC[g][b][k]$, los valores de tiempo $tDRC[g][b][k]$, y los valores de pendiente $sDRC[g][b][k]$ donde g es el índice del grupo de canales, b es el índice de la banda y k es el índice del nodo spline. Los valores de tiempo son números enteros relativos al comienzo de la trama DRC en unidades de *delta_t_min*. La muestra de audio que coincide con el comienzo de la trama DRC tiene un valor de tiempo de $tDRC=0$.

Tabla 6: Descodificación de pendientes y coordenadas de muestra de ganancia de DRC en el dominio de dB.

```

for(g=0; g<nDrcChannelGroups; g++) {
    for(b=0; b<nDrcBands[g]; b++) {
        gDRC[g][b][0] = decode_initial_gain(gain_initial_code[g][b]);
        if (drcGainCodingMode[g][b] == 0) {
            /* "simple" mode */
            tDRC[g][b][0] = drcFrameSize - 1;
            sDRC[g][b][0] = 0.0;
        }
        else
        {
            for (k=1; k<nNodes[g][b]; k++) {
                gDRC[g][b][k] = gDRC[g][b][k-1] + decode_delta_gain(gain_delta_code[g][b][k-1]);
            };
            tDRC[g][b][0] = delta_t_min * decode_time_delta(time_delta_code[g][b][0]) - 1;
            for (k=1; k<nNodes[g][b]; k++) {
                tDRC[g][b][k] = tDRC[g][b][k-1] +
                    delta_t_min * decode_time_delta(time_delta_code[g][b][k]);
            }
            for (k=0; k<nNodes[g][b]; k++) {
                sDRC[g][b][k] = decode_slope(slope_code[g][b][k]);
            }
        }
    }
}

```

Interpolación y modificaciones de ganancia

20 [0043] Como se mencionó anteriormente bajo la cabecera "Modificación de la característica de DRC", hay varias formas de adaptar las características de DRC en el descodificador de herramientas de DRC. Estos ajustes se aplican a las muestras de ganancia descodificadas en el dominio de dB.

25 [0044] La función *toLinear()* se presenta en la **Tabla 7** para incluir todos los pasos necesarios para generar una muestra de ganancia lineal a partir del valor logarítmico en dB (consulte la **Tabla 7**). Esta función contiene una función de asignación opcional *mapGain()* (ver **Tabla 2**) que soporta modificaciones de los valores de ganancia de DRC con el propósito de lograr una característica de compresión diferente a la utilizada en el codificador. La asignación está controlada por el índice *characteristicIndex* que seleccionará una de las características de DRC del descodificador personalizado si es mayor que 0. De lo contrario, la característica del codificador no será reemplazada. Se puede generar una característica modificada basándose en la característica de compresión del codificador que se transmite en la descripción de la muestra. Además, se soporta un factor de compresión y refuerzo para escalar las ganancias negativas y positivas, respectivamente. Estos factores tienen un valor de 1.0, a menos que el usuario proporcione valores en el rango [0,1]. Finalmente, se aplica la ganancia de normalización del sonoridad.

35 [0045] Antes de que la ganancia se pueda aplicar a la señal de audio, la señal de audio se debe convertir al dominio lineal y los valores de ganancia entre las muestras de ganancia se deben interpolar. Para lograr una complejidad menor, la conversión de dB a lineal puede realizarse antes de la interpolación. Por lo tanto, el proceso de interpolación se realiza completamente en el dominio lineal. Tanto la modificación de ganancia como la conversión al dominio lineal se realizan utilizando el pseudo código de la **Tabla 7**. Las variables de entrada son las

muestras de ganancia y las pendientes en el dominio de dB. El resultado consiste en las muestras de ganancia y las pendientes en el dominio lineal. Para la normalización de la sonoridad, se puede suministrar un valor de ganancia de normalización de la sonoridad en dB (*loudnessNormalizationGainsDb*) al descodificador mediante una herramienta de control de sonoridad u otros medios. Si no se proporciona, se usa un valor predeterminado de 0.0. En un modo de realización, la ganancia de normalización se calcula como la diferencia entre la sonoridad objetivo y la sonoridad del contenido en dB FS. La sonoridad objetivo es el nivel de sonoridad de salida deseado. La sonoridad del contenido es igual a la sonoridad del programa o la sonoridad de fijación según se define en ISO/MPEG, "14496-12 PDAM 3 - Audio mejorado (Formato de archivo)", 106.^a reunión de MPEG Ginebra, Suiza, octubre de 2013. Si no se incluyen la sonoridad del programa y la sonoridad de fijación, se puede usar un valor predeterminado para la sonoridad del contenido.

Tabla 7: Conversión de una muestra de ganancia de DRC y pendiente asociada de dB a dominio lineal

```

toLinear (gainDb, slopeDb) {
    SLOPE_FACTOR_DB_TO_LINEAR = 0.1151f /* ln(10) / 20 */
    gainRatio = 1.0;
    if (characteristicIndex > 0) {
        gainRatio = mapGain(gainDb) / gainDb;
    }
    if (gainDb < 0) {
        gainRatio *= compress;
    }
    else {
        gainRatio *= boost;
    }
    gainLin = pow(2.0, (gainRatio*gainDb+loudnessNormalizationGainDb) / 6);
    slopeLin = SLOPE_FACTOR_DB_TO_LINEAR * gainRatio * gainLin * slopeDb;
    }return (gainLin, slopeLin)
}

```

[0046] La interpolación de ganancia se implementa mediante el pseudo código en la **Tabla 8**. Las variables de entrada son:

- la diferencia de tiempo entre las dos muestras de ganancia en unidades del intervalo de frecuencia de muestreo objetivo *tGainStep*
- un par de muestras de ganancia posteriores *gain0* y *gain1* en dB
- un par de valores correspondientes de inclinación de la pendiente *slope0* y *slope1* en el dominio de dB.

[0047] Esta función usa *toLinear()* para convertir las variables al dominio lineal. El resultado es una secuencia suave de valores de ganancia a la frecuencia de muestreo objetivo localizada entre el par de muestras de ganancia. La frecuencia de muestreo objetivo es la frecuencia de muestreo de la señal de audio comprimida.

30

35

40

45

Tabla 8: Interpolación de la ganancia de DRC para un segmento spline

```

interpolateDrcGain(tGainStep, gain0, gain1, slope0, slope1)
{
    int n;
    float k1, k2, a, b, c, d;
    float slopeLeft;
    float slopeRight;
    float gainLeft;
    float gainRight;
    (gainLeft, slopeLeft) = toLinear(gain0, slope0);
    (gainRight, slopeRight) = toLinear(gain1, slope1);
    slopeLeft = slopeLeft * delta_t_min;
    slopeRight = slopeRight * delta_t_min;
    bool useCubicInterpolation = TRUE;
    int tConnect;
    float tConnectFloat;
    if (abs(slopeLeft) > abs(slopeRight)) {
        tConnectFloat = 2.0 * (gainRight - gainLeft - slopeRight *
tGainStep) /
                           (slopeLeft - slopeRight);
        tConnect = (floor)(0.5 + tConnectFloat);
        if ((tConnect >= 0) && (tConnect < tGainStep)) {
            useCubicInterpolation = FALSE;
            result[0] = gainLeft;
            result[tGainStep] = gainRight;
            a = (slopeRight - slopeLeft) / (tConnectFloat + tConnectFloat);
            b = slopeLeft;
            c = gainLeft;
            for (n=1; n<tConnect; n++) {
                float t = (float)n;
                result[n] = (a * t + b) * t + c;
            }
            a = slopeRight;
            b = gainRight;
            for (; n<tGainStep; n++) {
                float t = (float)(n - tGainStep);
                result[n] = a * t + b;
            }
        }
    }
    else if (abs(slopeLeft) < abs(slopeRight))
    {
        tConnectFloat = 2.0 * (gainLeft - gainRight + slopeLeft *
tGainStep) /
                           (slopeLeft - slopeRight);
        tConnectFloat = tGainStep - tConnectFloat;
        tConnect = (floor)(0.5 + tConnectFloat);
        if ((tConnect >= 0) && (tConnect < tGainStep)) {
            useCubicInterpolation = FALSE;
            result[0] = gainLeft;
            result[tGainStep] = gainRight;
            a = slopeLeft;
            b = gainLeft;
            for (n=1; n<tConnect; n++) {
                float t = (float)n;
                result[n] = a * t + b;
            }
            a = (slopeRight - slopeLeft) / (2.0 * (tGainStep -
tConnectFloat));
            b = -slopeRight;
        }
    }
}

```

```

        c = gainRight;
        for (; n<tGainStep; n++) {
            float t = (float) (tGainStep-n);
            result[n] = (a * t + b) * t + c;
        }
    }
}
if (useCubicInterpolation == TRUE)
{
    float tGainStepInv = 1.0 / (float)tGainStep;
    float tGainStepInv2 = tGainStepInv * tGainStepInv;
    k1 = (gainRight - gainLeft) * tGainStepInv2;
    k2 = slopeRight + slopeLeft;
    a = tGainStepInv * (tGainStepInv * k2 - 2.0 * k1);
    b = 3.0 * k1 - tGainStepInv * (k2 + slopeLeft);
    c = slopeLeft;
    d = gainLeft;
    result[0] = gainLeft;
    result[tGainStep] = gainRight;
    for (n=1; n<tGainStep; n++) {
        float t = (float) n;
        result[n] = (((a * t + b) * t + c) * t) + d;
    }
}
return result;
}

```

Aplicación de compresión

- 5 [0048] Los valores de ganancia interpolados de cada segmento de spline se concatenan para generar un vector de ganancia completa $gain[g][b][t]$ para toda la trama DRC. Finalmente, el vector de ganancia se aplica como se muestra en la **Tabla 9**. La función *channelInDrcGroup()* devuelve TRUE si el canal actual c pertenece al grupo de canales DRC actual como se especifica en la descripción de la muestra. Tenga en cuenta que la programación de los segmentos spline depende del modo de retardo (consulte la sección anterior titulada "Anticipo en descodificador") como se indica en la **Tabla 9**.
- 10

Tabla 9: Concatenación de segmentos spline a un vector de ganancia y aplicación del vector de ganancia de DRC a los canales de audio.

```

for (g=0; g<nDrcChannelGroups; g++) {
    for (b=0; b<nDrcBands[g]; b++) {
        if (delayMode == DELAY_MODE_DEFAULT) {
            for (k=0; k<nNodesPrev[g][b]-1; k++) {
                duration = tDRCprev[g][b][k+1] - tDRCprev[g][b][k];
                splineSegment = interpolateDrcGain(duration, gDRCprev[g][b][k],
                                                    gDRCprev[g][b][k+1], sDRCprev[g][b][k],
                                                    sDRCprev[g][b][k+1]);
                for (t=0; t<duration; t++)
                {

```

```

        gain[g][b][t+tDRCprev[g][b][k]] = splineSegment[t];
    }

    k = nNodesPrev[g][b]-1;
    duration = drcFrameSize + tDRC[g][b][0] - tDRCprev[g][b][k];
    splineSegment = interpolateDrcGain(duration, gDRCprev[g][b][k],
                                         gDRC[g][b][0], sDRCprev[g][b][k],
                                         sDRC[g][b][0]);
    for (t=0; t<duration; t++)
    {
        gain[g][b][t+tDRCprev[g][b][k]] = splineSegment[t];
    }
}

else
{
    for (k=0; k<nNodes[g][b]-1; k++) {
        duration = tDRC[g][b][k+1] - tDRC[g][b][k];
        splineSegment = interpolateDrcGain(duration, gDRC[g][b][k],
                                             gDRC[g][b][k+1], sDRC[g][b][k],
                                             sDRC[g][b][k+1]);
        for (t=0; t<duration; t++)
        {
            gain[g][b][t+tDRC[g][b][k]] = splineSegment[t];
        }
    }
}

/* Apply gain to DRC bands of audio in each channel */
for (c=0; c<nChannels; c++) {
    if (channelInDrcGroup(c, g)) {
        for (t=0; t<drcFrameSize; t++) {
            audioBandOut[c][b][t] = audioBandIn[c][b][t] * gain[g][b][t];
        }
    }
}

if (delayMode == DELAY_MODE_DEFAULT)
{
    for (k=0; k<nNodes; k++) {
        gDRCprev[g][b][k] = gDRC[g][b][k];
        sDRCprev[g][b][k] = sDRC[g][b][k];
        tDRCprev[g][b][k] = tDRC[g][b][k];
    }

    nNodesPrev[g][b] = nNodes[g][b];
    for (t=0; t<drcFrameSize; t++) {
        gain[g][b][t] = gain[g][b][t + drcFrameSize];
    }
}
}

for(g=0; g<nDrcChannelGroups; g++) {
    for(c=0; c<nChannels; c++) {
        if (channelInDrcGroup(c, g)) {
            for (t=0; t<drcFrameSize; t++) {
                sum = 0.0;
                for(b=0; b<nDrcBands[g]; b++) {
                    sum = sum + audioBandOut[c][b][t];
                }
                audioSampleOut[c][t] = sum;
            }
        }
    }
}
}

```

- splineSegment es un vector que contiene los valores de ganancia de un segmento spline.
- la duración es un número entero que describe la duración del segmento spline en unidades de intervalos de muestra de audio.
- nNodes es el número de valores de ganancia en la trama DRC actual.
- 5 • drcFrameSize es la cantidad de intervalos de muestra de audio en una trama DRC.
- Inicia las siguientes variables si delayMode == DELAY_MODE_DEFAULT: gDRCprev[g] [b] [0]=0.0, sDRCprev[g] [b] [0]=0.0; tDRCprev[g] [b] [0]=drcFrameSize; nNodesPrev[g] [b]=1.

Banco de filtros DRC multibanda

10 [0050] Cuando las ganancias de DRC se aplican en el dominio del tiempo y se utiliza un DRC multibanda, la señal de audio en el dominio del tiempo se debe dividir en sub-bandas antes de aplicar la ganancia de DRC a las bandas. Los parámetros de configuración del filtro pueden ser transportados por las *DRCInstructions()* definidas en el formato 15 de archivo MPEG. El formato de archivo MPEG puede proporcionar la sintaxis del flujo de bits para el número de bandas y los índices de frecuencia de cruce entre las bandas.

20 [0051] La señal de audio en el dominio del tiempo se divide en el número especificado de bandas mediante los filtros Linkwitz-Riley (LR) con una topología que se muestra en la **Fig. 7**. La topología de los filtros cruzados Linkwitz-Riley para 2, 3 y 4 bandas se muestra en la **Fig. 7**. Como se muestra en la **Fig. 7**, el índice de banda b aumenta con 25 la frecuencia de la banda. Las frecuencias de cruce $f_{c,b}$ aumentan con el índice b (es decir, $f_{c,b+1} > f_{c,b}$). Las frecuencias de cruce entre paréntesis de un filtro de paso completo especifican el correspondiente filtro de paso bajo LR con la respuesta de fase correspondiente. Si hay más de dos bandas, se agregan filtros de todo paso para compensar el retardo de las diferentes salidas, de modo que estén todas en fase. Los filtros paso bajo y paso alto se implementan como secciones de segundo orden (biquads).

25 [0052] Como se muestra en la **Fig. 7**, cada filtro de cruce Linkwitz-Riley (LR) se compone de un par de filtros de 30 paso bajo y paso alto complementarios que dan como resultado una respuesta de frecuencia plana. Cada filtro de paso bajo de LR se crea mediante una cascada de dos filtros de paso bajo Butterworth (BW) idénticos. De manera similar, cada filtro de paso alto de LR es una cascada de dos filtros de paso alto BW idénticos con el mismo orden y frecuencia de corte que los filtros de paso bajo BW.

[0053] Cada filtro BW y cada filtro de todo paso se implementa como una sección de segundo orden con la siguiente función de transferencia.

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{a_0 + a_1 z^{-1} + a_2 z^{-2}} \quad (2)$$

35 [0054] Basándose en los índices de frecuencia de cruce en la **Tabla 26**, el descodificador puede buscar las 40 frecuencias de cruce normalizadas $f_{c,Norm}$ o los parámetros del coeficiente del filtro y γ y δ . A continuación, los coeficientes del filtro se calculan utilizando la **Tabla 10** para los filtros BW y la **Tabla 11** para los filtros de todo paso. Las 45 frecuencias de cruce f_c en Hz se calculan mediante:

$$f_c = f_s \cdot f_{c,Norm} \quad (3)$$

[0055] En el caso de configuraciones de descodificador de múltiples velocidades, como HE-AAC de frecuencia dual, f_s es la frecuencia de muestreo de la señal de salida final.

Tabla 10: Fórmulas del coeficiente del filtro Butterworth

	Paso bajo BW	Paso alto BW
Frecuencia de corte normalizada	$\omega_0 = \tan(\pi f_{c,Norm})$	
Parámetros intermedios	$\delta = \frac{1}{1 + \sqrt{2}\omega_0 + \omega_0^2}$	$\gamma = \omega_0^2 \delta$
Coeficientes de filtro BW final	$a_{LP,0} = 1$ $a_{LP,1} = 2(\gamma - \delta)$ $a_{LP,2} = 2(\gamma + \delta) - 1$ $b_{LP,0} = \gamma$	$a_{HP,0} = 1$ $a_{HP,1} = 2(\gamma - \delta)$ $a_{HP,2} = 2(\gamma + \delta) - 1$ $b_{HP,0} = \delta$

	Paso bajo BW	Paso alto BW
	$b_{LP,1} = 2\gamma$	$b_{HP,1} = -2\delta$
	$b_{LP,2} = \gamma$	$b_{HP,2} = \delta$

- 5 [0056] Los filtros de todo paso en la **Fig. 7** se utilizan para generar la misma respuesta de fase que uno de los filtros de paso bajo LR (con nivel de gris correspondiente y f_c correspondiente en la **Fig. 7**) de modo que las señales de todas las bandas están en fase a la salida del banco de filtros. Los coeficientes de filtro de todo paso se obtienen de los coeficientes del filtro de paso bajo BW correspondiente, como se muestra en la **Tabla 11**.

Tabla 11: Fórmulas de coeficientes de filtro de todo paso

$a_{AP,0} = a_{LP,0}$
$a_{AP,1} = a_{LP,1}$
$a_{AP,2} = a_{LP,2}$
$k_{AP,0} = a_{LP,2}$
$b_{AP,1} = a_{LP,1}$
$b_{AP,2} = a_{LP,0}$

- 10 [0057] Después de que las ganancias de DRC se apliquen a las bandas individuales, la señal de audio final se calcula sumando todas las bandas.

DRC aplicado al dominio de sub-banda del descodificador

- 15 [0058] Si bien la aplicación de las ganancias de DRC en el dominio del tiempo es obligatoria para AAC, otros códigos MPEG utilizan el dominio de sub-banda DRC. El concepto de dominio de sub-banda DRC significa que las señales de sub-banda existentes del descodificador están sujetas a la aplicación de ganancia de DRC. Por lo tanto, no es necesario agregar una división de bandas en el dominio del tiempo para un DRC multibanda y es posible aplicar ganancias de DRC antes de renderizar y/o mezclar en el dominio de la frecuencia. La **Tabla 12** contiene una lista no exhaustiva de códigos y el dominio donde se aplica la ganancia de DRC. El dominio puede depender de la configuración del descodificador y no del flujo de bits. Por ejemplo, si MPEG-Surround se descodifica con un descodificador AAC simple, las ganancias de DRC se aplican en el dominio del tiempo. Además, el dominio de sub-banda puede no ser el dominio MDCT de un código central. En cambio, el dominio de sub-banda suele ser el dominio QMF.
- 20

- 25 [0059] Para lograr la compresión multibanda, las bandas del compresor se asignan a grupos de sub-bandas del descodificador. No es necesario hacer un filtrado adicional. Las frecuencias de cruce de DRC están asignadas a la frecuencia de cruce de sub-banda del descodificador más cercana disponible. Dadas las frecuencias de cruce de sub-banda normalizadas $f_{c,Norm,SB}(s)$ para sub-banda s , la frecuencia de cruce asignada de $f_{c,Norm}(b)$ es:

Descodificador	Dominio del tiempo DRC	Sub-banda DRC
AAC	✓	
HE-AAC		✓
MPEG-Surround		✓
SAOC		✓
USAC		✓
Audio 3D		✓

- 30 [0059] Para lograr la compresión multibanda, las bandas del compresor se asignan a grupos de sub-bandas del descodificador. No es necesario hacer un filtrado adicional. Las frecuencias de cruce de DRC están asignadas a la frecuencia de cruce de sub-banda del descodificador más cercana disponible. Dadas las frecuencias de cruce de sub-banda normalizadas $f_{c,Norm,SB}(s)$ para sub-banda s , la frecuencia de cruce asignada de $f_{c,Norm}(b)$ es:

$$\text{if } f_{c,Norm,SB}(s) \leq f_{c,Norm}(b) \leq f_{c,Norm,SB}(s+1): \\ f_{c,Norm,Mapped}(b) = \begin{cases} f_{c,Norm,SB}(s); \text{ if } f_{c,Norm}(b) < 0,5(f_{c,Norm,SB}(s) + f_{c,Norm,SB}(s+1)) \\ f_{c,Norm,SB}(s+1); \quad \quad \quad \text{en caso contrario} \end{cases} \quad (4)$$

[0060] Las ganancias de DRC pueden descodificarse como se describe anteriormente. Las ganancias DRC se pueden interpolar utilizando la misma técnica que se describe en la **Tabla 8** y **Tabla 9**, sin embargo, la frecuencia de muestreo del resultado de interpolación se reduce para que coincida con la frecuencia de muestreo de las señales de sub-banda. Esto se puede lograr submuestreando las ganancias de DRC en el dominio del tiempo interpoladas por un factor de L o interpolando directamente utilizando la frecuencia de muestreo de sub-banda como objetivo.

[0061] Para evitar transiciones espectrales agudas entre las bandas de DRC, puede haber una "atenuación cruzada" entre las ganancias de las bandas de DRC contiguas. Esta operación se llama superposición. La superposición se controla mediante coeficientes de ponderación w , uno para cada sub-banda. Los coeficientes de ponderación w pueden calcularse de acuerdo con la **Tabla 14** y determinar la contribución de la ganancia de DRC de la banda actual y la de la banda siguiente. La **Fig. 8** muestra un ejemplo de coeficientes de ponderación para un DRC de cuatro bandas que utiliza un banco de filtros con 64 sub-bandas. Los bordes de las bandas en la **Fig. 8** se indican mediante líneas discontinuas, los centros de las bandas se denotan mediante líneas continuas, y las ponderaciones se denotan mediante líneas de puntos.

[0062] Después de la superposición, las ganancias de DRC de cada banda de compresor se aplican a cada grupo de sub-bandas correspondiente a la banda del compresor. Un pequeño retardo de tiempo D para tener en cuenta el retardo del banco de filtros se agrega a las ganancias de DRC para lograr una alineación de tiempo adecuada con la señal de audio. Las operaciones de muestreo descendente y retardo se pueden expresar mediante la primera parte del pseudo código en la **Tabla 13**. Los valores de los dos parámetros se analizan a continuación como valores específicos del códec. El significado de las variables y funciones del pseudo código en la **Tabla 13** se explica en la **Tabla 15**. La descripción supone que las frecuencias de muestreo en todas las sub-bandas son iguales. Si este no es el caso, el factor L de muestreo descendente puede ajustarse para las diferentes frecuencias de muestreo de sub-banda.

Tabla 13: Muestreo descendente de ganancia de DRC, y aplicación en las sub-bandas del decodificador

```

/* resample DRC gain */
for (g=0; g<nDrcChannelGroups; g++) {
    for (b=0; b<nDrcBands[g]; b++) {
        for (m=0; m<drcFrameSizeSb; m++) {
            gainLr[g][b][m]=gain[g][b][m*L-D];
        }
    }
}
/* Overlap DRC gains in crossover regions
for (g=0; g<nDrcChannelGroups; g++) {
    s=0;
    for (b=0; b<nDrcBands-2; b++) {
        while (fCenterSubband[s] < 0.5*(fCross[g][b] + fCross[g][b+1])) {
            for (m=0; m<drcFrameSizeSb; m++) {
                gainSb[g][s][m] = w[g][s]*gainLr[g][b][m]+(1-
w[g][s])*gainLr[g][b+1][m];
            }
            s++;
        }
    }
    for (; s<nDecoderSubbands; s++) {
        for (m=0; m<drcFrameSizeSb; m++) {
            gainSb[g][s][m] = w[g][s]*gainLr[g][b][m]+(1-
w[g][s])*gainLr[g][b+1][m];
        }
    }
}
/* apply DRC gain in sub-bands */
for (g=0; g<nDrcChannelGroups; g++) {
    for (c=0; c<nChannels; c++) {
        if (channelInDrcGroup(c, g)) {
            for (s=0; s<nDecoderSubbands; s++) {
                for (m=0; m<drcFrameSizeSb; m++) {
                    audioSampleSbOut[c][s][m]=gainSb[g][s][m]*audioSampleSbIn[c][s][m];
                }
            }
        }
    }
}

```

Tabla 14: Cálculo de ponderaciones de superposición.

```

/* Overlap DRC gains in crossover regions
olapSize = 0.15;
for (g=0; g<nDrcChannelGroups; g++) {
    bwLeft = 2*tCross[g] [0];
    s=0;
    for (b=0; b<nDrcBands-2; b++) {
        olap = olapSize * fCross[g] [b];
        fStart = fCross[g][b] - min(0.5*bwLeft, olap);
        bwRight = fCross[g][b+1] - fCross[g][b];
        fStop = fCross[g][b] + min(0.5*bwRight, olap);
        while (fCenterSubband[s] <= fStart) {
            w[g][s] = 1.0;
            s++;
        }
        while (fCenterSubband[s] <= fStop) {
            w[g][s] = (fCross[g][b] + olap - fCenterSubband[s]) / (2 * olap);
            s++;
        }
        while (fCenterSubband[s] <= 0.5 * (fCross[g][b] + fCross[g][b+1])) {
            w[g][s] = 0.0;
            s++;
        }
        bwLeft = bwRight;
    }
    olap = olapSize * fCross[g][b];
    fStart = fCross[g][b] - min(0.5*bwLeft, olap);
    bwRight = 0.5 - fCross[g][b];
    fStop = fCross[g][b] + min(bwRight, olap);
    while (fCenterSubband[s] <= fStart) {
        w[g][s] = 1.0;
        s++;
    }
    while (fCenterSubband[s] <= fStop) {
        w[g][s] = (fCross[g][b] + olap - fCenterSubband[s]) / (2 * olap);
        s++;
    }
    while (s < nDecoderSubbands) {
        w[g][s] = 0.0;
        s++;
    }
}

```

Tabla 15: Explicación de los elementos del código

Elemento de código	Significado
gainSb	Ganancia de DRC para ser aplicada a las sub-bandas del descodificador
gainLr	Ganancia de DRC de baja velocidad (remuestreo)
fCross	Frecuencia de cruce normalizada
drcFrameSizeSb	Número de muestras de sub-banda por sub-banda en una trama de audio
ndecoderSubbands	Número de sub-bandas del descodificador
fCenterSubband	Frecuencia central de la sub-banda del descodificador
w0, w1	Ponderaciones para superposición de ganancia
olapSize	Tamaño de la región de superposición relativa a la frecuencia de cruce

Elemento de código	Significado
olap	Tamaño de la región de superposición en unidades de frecuencia normalizadas
audioSampleSbln	Muestra de audio de sub-banda descodificada antes de la compresión dinámica
audioSampleSbOut	Muestra de audio de sub-banda descodificada después de la compresión dinámica

Configuración de DRC para escenarios heredados de transmisión

- 5 [0063] La información de configuración de DRC puede transmitirse mediante la sintaxis del formato de archivo MPEG. Sin embargo, si se usa un formato de transmisión heredado como ADTS para transportar una transmisión de audio MPEG que no soporta el formato de archivo MPEG, la información de configuración puede integrarse en la transmisión de audio. Esto se puede lograr agregando la sintaxis AudioSampleEntry() (o una versión comprimida de la misma) del Formato de archivo a la sintaxis uni_drc_info(). Dado que la información de entrada de muestra solo se requiere a una frecuencia menor que la frecuencia de tramas, se puede usar un indicador de presencia que indique cuándo está disponible esta información. La sintaxis extendida se da en la **Tabla 16**.
- 10

Tabla 16: Sintaxis de la carga útil extendida de uni_drc_info() con el campo de entrada de muestra.

```
uni_drc_info()
sampleEntryPresent; 1 uimsbf
if (sampleEntryPresent == 1) {
    AudioSampleEntry()
}
for (s=0; s<nDrcGainSequences; s++) {
    drc_gain_sequence()
}
```

- 15 [0064] Para este caso, la información de DRC solo puede descodificarse después de que el descodificador haya recibido la entrada de muestra. La velocidad de repetición de la información de entrada de muestra determina el retardo de descodificación.

Precedencia

- 20 [0065] Si un flujo de bits contiene los metadatos de DRC propuestos y otros metadatos de DRC como compresión Ligera o Pesada de MPEG, los metadatos propuestos tendrán prioridad a menos que se instruya al descodificador para que aplique los otros metadatos de DRC.

Información específica del descodificador

Codificación avanzada de audio (AAC)

30 Extensión de metadatos DRC para AAC

- 35 [0066] Para AAC, se puede utilizar una nueva carga útil de extensión con una nueva ID para transportar los metadatos de DRC propuestos en un elemento de relleno. Los ID están codificados con un código de 4 bits y actualmente solo se definen 7. Integrar esta información de DRC en una nueva carga útil de extensión garantiza la compatibilidad con los descodificadores existentes que omitirán la nueva carga útil. El nuevo extension_type propuesto se da en la **Tabla 17**. Contiene uni_drc_info() como se muestra en la **Tabla 20**.

Tabla 17: Definición de nuevo extension_type para AAC

Símbolo	Valor del tipo de extensión	Finalidad
UNI_DRC	1111	DRC unificado

Modo de retardo para AAC

- 40 [0067] AAC usa el modo de retardo predeterminado.

Tamaño de trama DRC y resolución de tiempo para AAC

[0068] El tamaño de trama DRC tiene el tamaño predeterminado (es decir, tiene la misma duración de tiempo que el tamaño de trama AAC).

5 [0069] El valor de delta_t_min en el número de muestras a la frecuencia de audio se calcula como se especifica en la sección anterior titulada "Resolución de tiempo". Los valores específicos se proporcionan aquí por conveniencia basándose en la siguiente fórmula y la **Tabla 18**:

$$\Delta t_{\min} = 2^M \quad (5)$$

10 [0070] El exponente M aplicable se puede encontrar buscando en el rango de frecuencia de muestreo de audio que cumple:

$$f_{s,\min} \leq f_s < f_{s,\max} \quad (6)$$

15 **Tabla 18: Tabla para determinar la resolución de tiempo DRC para AAC**

$f_{s,\min}[\text{Hz}]$	$f_{s,\max}[\text{Hz}]$	M
8000	16000	3
16000	32000	4
32000	64000	5
64000	128000	6

20 [0071] Dado el tamaño de trama de códec $N_{\text{Códec}}$, el tamaño de trama DRC en unidades de muestras de DRC a una frecuencia de delta_t_min es:

$$N_{DRC} = N_{\text{Códec}} 2^M \quad (7)$$

MPEG-D USAC

25 **Extensión de metadatos DRC**

30 [0072] En USAC, la nueva carga útil de extensión puede transportarse en el elemento de carga útil de extensión UsacExtElement. Para tal fin, se define un nuevo tipo de elemento de extensión de acuerdo con la **Tabla 19**. Se aplica la configuración predeterminada de DRC dependiente del códec.

Tabla 19: Definición de nuevo usacExtElementType para USAC

Símbolo	Valor del tipo de extensión	Finalidad
ID_EXT_ELE_DRC	3	DRC unificado

35 **MPEG-4 HE-AAC, HE-AACv2, MPEG-D Surround, MPEG-D SAOC**

Extensión de metadatos DRC

40 [0073] Los metadatos DRC pueden transportarse con la secuencia central AAC como se ha descrito anteriormente.

45 [0074] MPEG-4 HE-AAC, HE-AACv2, MPEG-D Surround y MPEG-D SAOC están compuestos por un descodificador central como AAC-LC y una o más capas adicionales en la parte superior de este descodificador central. Las capas adicionales mejoran el ancho de banda de audio o la cantidad de canales de audio en comparación con el núcleo. Para estos descodificadores, las ganancias de DRC deben aplicarse a las sub-bandas de la capa más alta inmediatamente antes del banco de filtros de síntesis, pero no después de la etapa de renderizado/mezcla.

DRC gana aplicaciones en sub-bandas

50 [0075] Para la aplicación de ganancia de DRC en el dominio de QMF, las ganancias de DRC en el dominio del tiempo pueden retardarse por intervalos de muestra en el dominio del tiempo D y muestrearse descendentemente un factor de L. Los valores de D y L dependen de la configuración, como HE-AAC. de frecuencia única con respecto a

frecuencia doble. Se debe lograr una alineación de tiempo adecuada entre las ganancias de DRC y la señal de audio para todas las configuraciones.

Sintaxis de flujo de bits

[0076] El flujo de bits de DRC se define en la **Tabla 20** y la **Tabla 21**. Normalmente, el flujo de bits de DRC time_domain_drc_info () se transporta en un campo de carga útil de extensión del códec de host.

Tabla 20: Sintaxis de la carga útil de uni_drc_info()

```
uni_drc_info()  
  
for (s=0; s<nDrcGainSequences; s++) {  
    drc_gain_sequence()  
}  
}
```

```

drc_gain_sequence()
for(b=0; b<nDrcBands[g]; b++) {
    drcGainCodingMode[g][b];                                1 uimsbf
    if (drcGainCodingMode[g][b] == 0)                         9 uimsbf
        gain_initial_code[g][b];
    }
    else {
        k=0;
        do {
            slope_code[g][b][k++];                           1..10 vlclbf
        }
        while (slope_code[g][b][k-1] != slope_code_end_marker);
        nNodes[g][b] = k;
        for (k=0; k<nNodes[g][b]; k++) {
            time_delta_code[g][b][k];                        1..12 vlclbf
        }
        gain_initial_code[g][b];
        for (k=1; k<nNodes[g][b]; k++) {
            gain_delta_code[g][b][k-1];                      1..11 vlclbf
        }
    }
}

```

Tabla 21: Sintaxis de drc_gain_sequence()

15

Tabla 22: Codificación del valor de ganancia de DRC inicial

Codificación	Tamaño	gainInitial en [dB]	Intervalo
$\{s, m_1\}$	{1 bit, 8 bits}	$g_{DRC}(0) = (-1)^s m_1 2^{-3}$	-31,875... 31,875 dB, tamaño de paso de 0,125 dB

20

Tabla 23: Codificación de las diferencias de ganancia de DRC

Tamaño de la palabra del código [bits]	Codificación binaria gainValueDelta	gainDelta [dB]
4	0x000	-2,0
9	0x039	-1,875
11	0x0E2	-1,750
11	0x0E3	-1,625
10	0x070	-1,500
10	0x1AC	-1,375
10	0x1AD	-1,250
9	0x0D5	-1,125

Tamaño de la palabra del código [bits]	Codificación binaria gainValueDelta	gainDelta [dB]
7	0x00F	-1,000
7	0x034	-0,875
6	0x036	-0,750
5	0x019	-0,625
5	0x002	-0,500
5	0x00F	-0,375
3	0x001	-0,250
2	0x003	-0,125
3	0x002	0,000
2	0x002	0,125
6	0x018	0,250
6	0x006	0,375
7	0x037	0,500
8	0x01D	0,625
9	0x0D7	0,750
9	0x0D4	0,875
5	0x00E	1,000

Tabla 24: Codificación de la inclinación de la pendiente

Tamaño de la palabra del código [bits]	Codificación binaria de la inclinación de la pendiente	Inclinación de la pendiente
7	0x058	-3,0518
9	0x142	-1,2207
8	0x0B2	-0,4883
6	0x02A	-0,1953
6	0x029	-0,0781
6	0x02D	-0,0312
3	0x004	-0,0050
2	0x003	0,000
5	0x017	0,0050
6	0x02B	0,0312
7	0x051	0,0781
10	0x287	0,1953
10	0x286	0,4883
8	0x0A0	1,2207
8	0x0B3	3,0518
1	0x000	n/a (marcador final)

Tabla 25: Codificación de diferencias de tiempo con nNodesMax = N_{DRC}

Tamaño de la palabra del código [bits]	Codificación binaria de diferencia de tiempo	Diferencia tDRC_delta horaria
1	0x000	nNodesMax
3	0x004	1
5	0x014+(a-2)	a=[2..5]
6	0x030+(a-6)	a=[6..13]
12	0xE00+(a-14)	a=[14..nNodesMax-1]

Tabla 26: Codificación de frecuencias de cruce normalizadas y parámetros de coeficientes de filtro asociados

crossover_freq_index	$f_{c,Norm}$	γ	δ
0	2/1024	0,0000373252	0,9913600345
1	3/1024	0,0000836207	0,9870680830
2	4/1024	0,0001480220	0,9827947083
3	5/1024	0,0002302960	0,9785398263
4	6/1024	0,0003302134	0,9743033527
5	2/256	0,0005820761	0,9658852897
6	3/256	0,0012877837	0,9492662926
7	2/128	0,0022515827	0,9329321561
8	3/128	0,0049030350	0,9010958535
9	2/64	0,0084426929	0,8703307793
10	3/64	0,0178631928	0,8118317459
11	2/32	0,0299545822	0,7570763753
12	3/32	0,0604985076	0,6574551915
13	2/16	0,0976310729	0,5690355937
14	3/16	0,1866943331	0,4181633458
15	2/8	0,2928932188	0,2928932188

Codificación de DRC de interpolación lineal

10 [0077] Aunque se describió anteriormente en relación con la codificación y descodificación de valores de ganancia basándose en la interpolación spline, en algunos modos de realización, los valores de ganancia pueden codificarse y aplicarse con el uso de interpolación lineal. Por ejemplo, en un modo de realización, los valores DRC pueden codificarse para una parte del contenido del programa de sonido que utiliza nodos spline como se describió anteriormente. En este modo de realización, los valores de pendiente entre cada uno de los nodos spline pueden omitirse del flujo de bits. En cambio, la interpolación lineal se puede realizar entre nodos spline en lugar de interpolación spline. De esta manera, la codificación de los valores de ganancia de DRC se puede simplificar evitando la generación de valores de pendiente.

Interpolación de ganancia de DRC de superposición-agregación basada en ventanas

20 [0078] En un modo de realización, se puede usar un procedimiento de interpolación de ganancia de superposición basada en ventanas para descodificar valores de ganancia de DRC. En este procedimiento, los valores de ganancia se codifican y reciben de una manera similar a la descrita anteriormente. Sin embargo, cada valor de ganancia se usa como un multiplicador para una ventana (por ejemplo, un vector de coeficientes de ventana) durante la descodificación. La curva de ganancia interpolada puede obtenerse posteriormente utilizando un procedimiento de superposición-agregación. Por ejemplo, la curva de ganancia de DRC interpolada para una ventana puede ser el producto de un valor de ganancia multiplicado por una ventana. Una razón para usar ventanas es que se puede

generar una curva de ganancia idéntica en comparación con una curva de ganancia que resulta de descodificadores estandarizados, que aplican valores de ganancia en sub-bandas. Además, el procedimiento de interpolación de ganancia de superposición-agregación basada en ventanas no genera distorsiones de aparición de alias. Una descripción más detallada de este procedimiento de interpolación de ganancia de superposición basada en ventanas se describe a continuación a modo de ejemplo.

[0079] La Fig. 9 muestra formas de ventana de DRC (líneas continuas) que se utilizarán para ciertas formas de ventana de descodificador correspondientes (líneas discontinuas). De arriba a abajo, la Fig. 9 muestra un ejemplo de ventana larga, una ventana de transición de larga a corta, una ventana corta y una ventana de transición de corta a larga. Las ventanas de DRC se pueden calcular mediante el cuadrado de la ventana de síntesis del descodificador. La ventana de DRC se puede aplicar con la misma temporización que la ventana de síntesis del descodificador correspondiente.

[0080] La siguiente ecuación ilustra cómo se calcula una ventana DRC a partir de una ventana de síntesis AAC larga con el tamaño de trama AAC N :

$$w_{DRC,largo}(n) = w_{AAC,largo}^2(n) \text{ para } n = [0,2N-1] \quad (8)$$

[0081] Las ventanas cortas y transitorias se pueden calcular de manera similar. La ecuación (9) siguiente muestra el proceso de superposición-agregación de ventanas de DRC subsiguientes ponderadas por los valores de ganancia de DRC obtenidos del flujo de bits. Los índices de tiempo y la trama se denotan n y k , respectivamente. El índice de tiempo de 0 se encuentra al comienzo de la ventana de síntesis actual (la primera muestra de salida de la trama actual).

$$g(n) = g_{DRC}(k-1)w_{DRC}(k-1,N+n) + g_{DRC}(k)w_{DRC}(k,n) \text{ para } n = [0,N-1] \quad (9)$$

[0082] La ganancia de DRC puede aplicarse a continuación a la señal x_{AAC} de salida del descodificador de acuerdo con la ecuación (10) siguiente para generar la salida de audio comprimida final de x_{DRC} . La ganancia de DRC no se aplica en el dominio MDCT.

$$x_{DRC}(n) = g(n)x_{AAC}(n) \text{ para } n = [0,N-1] \quad (10)$$

[0083] Cuando se utiliza compresión ligera en MPEG, los metadatos DRC multibanda se pueden usar para aplicar valores de ganancia de DRC independientes a bloques cortos individuales o agrupados. A diferencia de la etiqueta "Multibanda", cada ganancia de DRC se puede codificar de forma que se aplique a todo el espectro de MDCT de un bloque corto. Por lo tanto, cada ganancia de DRC funciona como un DRC de banda única. Si este es el caso, la operación de DRC puede realizarse en el dominio del tiempo de manera similar a la descrita anteriormente.

[0084] Por ejemplo, si se dan 5 valores de ganancia de DRC para los 8 bloques cortos como se muestra en el gráfico superior de la **Fig. 10**, las ventanas de DRC correspondientes se muestran como líneas continuas. El gráfico inferior muestra el uso de 8 ventanas DRC cortas con forma idéntica y los valores de ganancia de DRC correspondientes g_0 a g_4 . Las ventanas de DRC pueden obtenerse a partir de una forma de ventana corta usando la ecuación (8) con un parámetro de $N' = N/8$. Las ecuaciones (9) y (10) se aplican de forma correspondiente con el parámetro N' en lugar de N .

[0085] En general, la sintaxis del flujo de bits permite la elección independiente de DRC de una o varias bandas para tramas individuales. Con la implementación de DRC en el dominio del tiempo descrita anteriormente, el descodificador cambiará al procesamiento DRC de dominio MDCT siempre que haya verdadera información de ganancia de DRC multibanda ("multibanda verdadera"), lo cual significa que hay valores desiguales de ganancia de DRC para diferentes sub-bandas).

[0086] Esta propuesta incluye una implementación modificada de MPEG-AAC DRC que evita las distorsiones de aparición de alias de una manera compatible hacia atrás para los DRC de una sola banda. Aunque se describe en relación con MPEG-AAC DRC, en otros modos de realización se puede usar cualquier tipo de audio de flujo de bits que incluya valores de ganancia de DRC en el dominio de la frecuencia.

[0087] En el modo de realización descrito anteriormente, el descodificador se modifica para aplicar DRC en el dominio del tiempo. En otro modo de realización, se puede añadir un campo adicional al flujo de bits para aumentar la variabilidad de la aplicación de valor de ganancia de DRC a la señal de audio en el dominio del tiempo. El nuevo campo para los valores de ganancia de DRC se puede definir en varias ubicaciones en la sintaxis del flujo de bits. Para los estándares MPEG, una opción es la definición de una carga útil adicional de extensión transportada en un Elemento de relleno como se muestra en la **Tabla 27**. En este modo de realización, los canales de audio del contenido del programa se pueden agrupar en grupos DRC, donde cada grupo tiene un conjunto independiente de información DRC, es decir, se aplica un DRC independiente separado a cada grupo de canales. Un canal de audio

solo puede pertenecer a un grupo de DRC o ninguno. La información de agrupamiento se puede agregar a la descripción de muestra, que se produce una vez al comienzo de una pista. En este modo de realización, el número de grupos de DRC se llama nDrcChannelGroups.

Secuencia de metadatos	
For (g=0; g<nDrcChannelGroups-1; g++) {	
k=0;	
nDrcGainInfoBlocks[g];	2 uimsbf
for (b=0; b<nDrcGainInfoBlocks[g]; b++) {	
drcGainCodingMode[g][b];	2 uimsbf
if (b==0) {	
gainInitial[g];	9 uimsbf
for (n=1; n<nDrcGainValues[g][b]; n++) {	
gainDelta[g][k++];	1..11 vlclbf
}	
}	
else {	
for (n=0; n<nDrcGainValues[g][b]; n++) {	
gainDelta[g][k++];	1..11 vlcbf
}	
}	
}	
}	

Tabla 27: Carga útil de extensión DRC en el dominio del tiempo

[0088] Al observar la ganancia variable en el tiempo de implementaciones prácticas, se puede observar que la ganancia puede cambiar muy lentamente a veces, mientras que puede exhibir cambios dramáticos cuando la señal de audio exhibe ataques. La velocidad de transferencia de bits necesaria para codificar los valores de ganancia de DRC se reduce soportando una resolución de tiempo seleccionable individualmente para cada uno de los denominados drcGainInfoBlock. Una trama de audio se divide uniformemente en hasta 8 de estos bloques de información como se muestra en la **Tabla 28** y cada uno de los bloques puede contener hasta 16 valores de ganancia.

[0089] El aumento de la velocidad de transferencia de bits asociado con una mayor resolución de tiempo de los valores de ganancia se reduce aún más mediante el uso de un esquema adaptativo con codificación por entropía de los cambios de ganancia. Los valores de ganancia de DRC se pueden transmitir en cada trama de audio usando la sintaxis definida en la **Tabla 27**.

Tabla 28: Tabla de búsqueda para nDrcGainInfoBlocks

Palabra de código	nDrcGainInfoBlocks	Comentario
0x0	1	1 bloque por trama
0x1	2	2 bloques por trama
0x2	4	4 bloques por trama
0x3	8	8 bloques por trama

[0090] La entrada drcGainCodingMode determina la cantidad de valores de ganancia para un bloque de información como se muestra en la **Tabla 29**. Puede haber al menos un valor de ganancia por trama para soportar el

allanamiento aleatorio. El primer valor de ganancia se codifica de acuerdo con la **Tabla 30**. Los valores de ganancia restantes se codifican de forma diferencial usando la **Tabla 31** o la **Tabla 32** (dependiendo del drcGainCodingMode seleccionado).

5

Tabla 29: Tabla de búsqueda de drcGainCodingMode

drcGainCodingMode (Palabra de código)	nDrcGainValues	drcDiffGainFactor	Comentario
0x0	1	no disponible	1 valores por trama
0x1	4	1	4 valores por trama
0x2	16	1	16 valores por trama
0x3	16	4	16 valores por trama

Tabla 30: Representación de gainInitial

Codificación	Tamaño	gainInitial en [dB]	Intervalo
$\{s, m_1\}$	{1 bit, 8 bits}	$g_{DRC}(0) = (-1)^s m_1 2^{-3}$	-31,875... 31,875 dB, tamaño de paso de 0,125 dB

10

Tabla 31: Codificación de diferencias de ganancia de DRC para drcGainCodingMode=1

gainValueDelta (código binario)	gainDelta [dB] (Resolución)
00011000010	-1,5
000110001	-1,0
0001101	-0,75
000111	-0,5
0000	-0,375
001	-0,25
10	-0,125
11	0,0
01	0,125
00010	0,25
00011001	0,375
000110000	0,5
00011000011	1,0

15

Tabla 32: Codificación de diferencias de ganancia de DRC para drcGainCodingModeE[2,3]

gainValueDelta (código binario)	gainDelta [dB] (Resolución)
1100001110	-4,0
110000110	-3,0
1100000	-2,0
110001	-1,5
11010	-1,0

gainValueDelta (código binario)	gainDelta [dB] (Resolución)
11011	-0,75
1111	-0,5
100	-0,25
101	-0,125
0	0,0
1110	0,125
11001	0,25
110000100	0,5
110000101	0,75
1100001111	1,25

Tabla 33: Descodificación de valores de ganancia de DRC logarítmicos [dB]

```

float gDRC[][];
for(g=0; g<nDrcChannelGroups-1; g++) {
    int k=0;
    for (b=0; b<nDrcGainInfoBlocks[g]; b++) {
        if (b==0) {
            gDRC[g][k++] = gainInitial[g];
            for (n=1; n<nDrcGainValues[g][b]; n++) {
                gDRC[g][k] = gDRC[g][k-1] + drcDiffGainFactor[g][b] *
                    gainDelta[g][k-1];
                k++;
            }
        }
        else {
            for (n=0; n<nDrcGainValues[g][b]; n++) {
                gDRC[g][k] = gDRC[g][k-1] + drcDiffGainFactor[g][b] *
                    gainDelta[g][k-1];
                k++;
            }
        }
    }
    float gDRCPrev[g] = gDRC[g][k-1];
}

```

5

[0091] La resolución no uniforme de los valores de diferencia está motivada por psicoacústica, como la observación de que las desviaciones en el cambio de ganancia son menos audibles cuanto mayor es el cambio de ganancia. Viceversa, si la ganancia es casi constante (y la envolvente de audio también), las desviaciones en los cambios de ganancia son más audibles. El rango asimétrico se adapta a la reducción de ganancia de DRC de acción rápida para ataques repentinos en la señal de audio. Los aumentos de ganancia suelen ser más lentos.

10

[0092] Un descodificador de audio típico reconstruye la señal de audio usando un procedimiento de superposición y agregación con una superposición del 50 % de los bloques subsiguientes. Cada uno de los bloques está ponderado por una ventana que se estrecha en cada extremo. Por ejemplo, un tamaño de trama típico de MPEG-AAC es 1024 muestras. Para cada nueva trama, el descodificador reconstruye 2048 muestras, las primeras 1024 de las cuales se agregan a las últimas 1024 muestras del bloque anterior y el resultado es la salida del descodificador. Los bloques de información que vienen con la trama k se programan uniformemente durante la segunda mitad del bloque reconstruido. Los valores de ganancia dentro de cada bloque de información se distribuyen uniformemente a lo largo de la duración del bloque de información. Este esquema asegura que todos los valores de ganancia de DRC necesarios estén disponibles cuando comienza y finaliza la descodificación, así como para la interpolación.

20

[0093] Se muestra un ejemplo en la figura 11, donde la trama n del flujo de bits contiene ganancias de DRC para la segunda mitad de la ventana de síntesis. La trama n tiene 4 bloques de información con 1, 8, 2 y 4 valores de ganancia de DRC, respectivamente. El tiempo de los valores de ganancia de DRC se calcula basándose en una

distribución uniforme de valores de ganancia dentro de cada bloque de información. Posteriormente, la interpolación lineal se usa para generar valores de ganancia para cada muestra de audio en el dominio del tiempo.

5

Tabla 34: Calcular las ubicaciones de tiempo de los valores de ganancia de DRC

```

for (g=0; g<nDrcChannelGroups-1; g++) {
    int k=0;
    float tGainPrev = 0.0;
    float deltaPrev = 0.0;
    float samplesPerIblock = nFrame/nDrcGainInfoBlocks[g];
    for (b=0; b<nDrcGainInfoBlocks[g]; b++) {
        float delta = samplesPerIblock / nDrcGainValues[g][b];
        for (n=0; n<nDrcGainValues[g][b]; n++) {
            float tGain[g][k] = tGainPrev + 0.5 * (delta + deltaPrev);
            tGainPrev = tGain[g][k];
            deltaPrev = delta;
            k++;
        }
    }
}

```

[0094] El cálculo del tiempo del valor de ganancia se da en la **Tabla 34**. El resultado $tGain[g][k]$ indica la ubicación de la muestra en unidades de intervalos de muestra comenzando con 0,0 en la primera muestra de la trama de salida actual. El tamaño de trama se denota como N_{trama} en muestras.

[0095] Dados los valores de ganancia y su sincronización, se puede construir una curva de ganancia suave para todas las muestras de la trama de salida actual mediante interpolación lineal de los valores de ganancia lineal como se muestra en la **Tabla 35**. $gDRCPrev$ es el último valor de ganancia de DRC de la trama anterior. En este modo de realización, el primer valor de ganancia de la siguiente trama es necesario para interpolar los valores de ganancia de la trama para la salida. Debido al proceso de superposición-agregación, ese valor de ganancia está disponible sin lectura adicional en el flujo de bits. La función `toLinear()` se introduce para incluir todos los pasos necesarios para generar un valor de ganancia lineal a partir del valor logarítmico en dB.

20

Tabla 35: Interpolación de los valores de ganancia de DRC

```

for(g=0; g<nDrcChannelGroups-1; g++) {
    int k=0;
    float tLeft = tGanPrev[g] - nFrame;
    float tRight = tGain[g][0];
    float gainLeft = toLinear(gDRCPrev[g]);
    float gainRight = toLinear(gDRC[g][0]);
    float factor = (gainRight-gainLeft)/(tRight - tLeft);
    for (t=0; t<nFrame; t++) {
        while (tTarget > tRight) {
            k++;
            tLeft = tRight;
            tRight = tGain[g][k];
            gainLeft = gainRight;
            gainRight = toLinear(gDRC[g][k]);
            factor = (gainRight-gainLeft)/(tRight - tLeft);
        }
        gain[g][t] = gainLeft + factor * (t - tLeft);
    }
}

```

25

[0096] Finalmente, las ganancias de DRC interpoladas se aplican como se muestra en la Tabla 36.

30

Tabla 26: Aplicación de los valores de ganancia de DRC

```

for(g=0; g<nDrcChannelGroups-1; g++)
    for (ch=0; ch<nChannels; ch++) {
        if (channelInDrcGroup(ch)) {
            for (t=0; t<nFrame; t++) {
                audioSampleOut[ch][t] = audioSampleIn[ch][t] *gain[g][t];
            }
        }
    }
}

```

- 5 **[0097]** Este modo de realización descrito anteriormente incluye codificación y procesamiento de metadatos de DRC mejorados para estándares de audio tales como MPEG-Audio. Se han abordado las deficiencias de los estándares actuales, como la generación de distorsiones de aparición de alias y la insuficiente resolución temporal de los metadatos de DRC.
- 10 **[0098]** Como se analizó anteriormente, se pueden usar múltiples técnicas para codificar y aplicar valores de ganancia de DRC para una parte del contenido del programa de sonido. En algunos modos de realización, un procedimiento para aplicar valores de ganancia de control de rango dinámico (DRC) en el dominio de la frecuencia a una señal de audio en el dominio del tiempo comprende: recibir un flujo de bits, en el que el flujo de bits incluye una señal de audio codificada y valores de ganancia de DRC en el dominio de la frecuencia; decodificar, mediante un descodificador en un dispositivo de reproducción, la señal de audio codificada para producir una señal de audio descodificada en el dominio del tiempo; determinar, mediante el descodificador, ponderaciones de ventana de DRC para aplicar los valores de ganancia de DRC de frecuencia a la señal de audio descodificada en el dominio del tiempo; determinar los valores de ganancia de DRC en el dominio del tiempo basándose en los valores de ganancia de DRC en el dominio de la frecuencia y las ponderaciones de ventana de DRC; y aplicar los valores de ganancia de DRC en el dominio del tiempo para las tramas correspondientes de la señal de audio descodificada en el dominio del tiempo.
- 15 **[0099]** En un modo de realización, las ponderaciones de ventana de DRC se determinan basándose en una ventana de síntesis del descodificador. En un modo de realización, las ponderaciones de ventana de DRC se calculan como el cuadrado de la ventana de síntesis del descodificador con la misma temporización que la ventana de síntesis del descodificador. En un modo de realización, las ponderaciones de ventana DRC se determinan basándose en el producto de la ventana de síntesis del descodificador y una ventana del codificador. En un modo de realización, los valores de ganancia de DRC en el dominio del tiempo para una trama actual de la señal de audio descodificada se determinan basándose en los valores de ganancia de DRC en el dominio de la frecuencia para la trama actual con una ponderación de ventana DRC correspondiente aplicada y los valores de ganancia de DRC en el dominio de la frecuencia para la trama anterior con una ponderación de ventana DRC correspondiente aplicada. En un modo de realización, aplicar los valores de ganancia de DRC en el dominio del tiempo para producir la señal de audio DRC en el dominio del tiempo se basa en el producto de los valores de ganancia de DRC en el dominio del tiempo y las divisiones de tiempo correspondientes de la señal de audio descodificada. En un modo de realización, uno o más de los valores de ganancia de DRC de dominio del tiempo se aplican a una ventana de DRC completa para la señal de audio descodificada. En un modo de realización, la señal de audio codificada es una señal de audio DRC de Grupo de expertos en imágenes en movimiento-codificación de audio avanzada (MPEG-AAC). En un modo de realización, la señal de audio codificada es una señal de audio DRC del Comité de sistemas de televisión avanzados (ATSC).
- 20 **[0100]** En otro modo de realización, un procedimiento de codificación de valores de ganancia de Control de Rango Dinámico (DRC) en un flujo de bits que representa una parte de contenido de programa de sonido comprende: agrupar cada canal de audio del contenido de programa de sonido en un único grupo de DRC de un conjunto de grupos de DRC; e insertar metadatos de ganancia de DRC en el flujo de bits para cada grupo de DRC, en el que los metadatos de ganancia de DRC para cada grupo de DRC se usan para aplicar de forma variable valores de ganancia de DRC correspondientes a cada trama en el grupo de DRC. En un modo de realización, los metadatos de ganancia de DRC para cada grupo de DRC incluyen: un primer valor de datos que representa un modo de codificación seleccionado para un valor de ganancia de DRC inicial; un segundo valor de datos que representa el valor de ganancia de DRC inicial; y un tercer valor de datos que representa las diferencias aplicadas al valor de ganancia de DRC inicial para generar los valores de ganancia de DRC para cada trama del grupo de DRC. En un modo de realización, el primer valor de datos representa el número de valores de ganancia para aplicar a cada trama del grupo de DRC basándose en el valor de ganancia de DRC inicial. En un modo de realización, el modo de codificación seleccionado representado por el primer valor de datos se elige de un conjunto predefinido de modos de codificación. En un modo de realización, los valores de ganancia de DRC se aplican usando la interpolación. En un modo de realización, la interpolación es una interpolación lineal en el dominio lineal. En un modo de realización, se asignan múltiples canales a un único grupo de DRC. En un modo de realización, se usa una resolución de tiempo no uniforme para frecuencias de actualización de valores de ganancia de DRC basándose en la varianza de ganancia generada por el codificador DRC para minimizar la velocidad de transferencia de bits del flujo de bits. En un modo de

realización, el primer valor de datos que representa el valor de ganancia inicial se codifica usando una escala de cuantificación no uniforme basada en psicoacústica para minimizar la velocidad de transferencia de bits del flujo de bits. En un modo de realización, el primer valor de datos que representa el valor de ganancia inicial se codifica utilizando un código de longitud variable para minimizar la velocidad de transferencia de bits del flujo de bits. En un modo de realización, el tercer valor de datos que representa las diferencias aplicadas al valor de ganancia de DRC inicial para cada trama del grupo de DRC se codifica para minimizar la velocidad de transferencia de bits del flujo de bits. En un modo de realización, el tercer valor de datos que representa las diferencias aplicadas al valor de ganancia de DRC inicial son códigos de longitud variable codificados para minimizar la velocidad de transferencia de bits del flujo de bits.

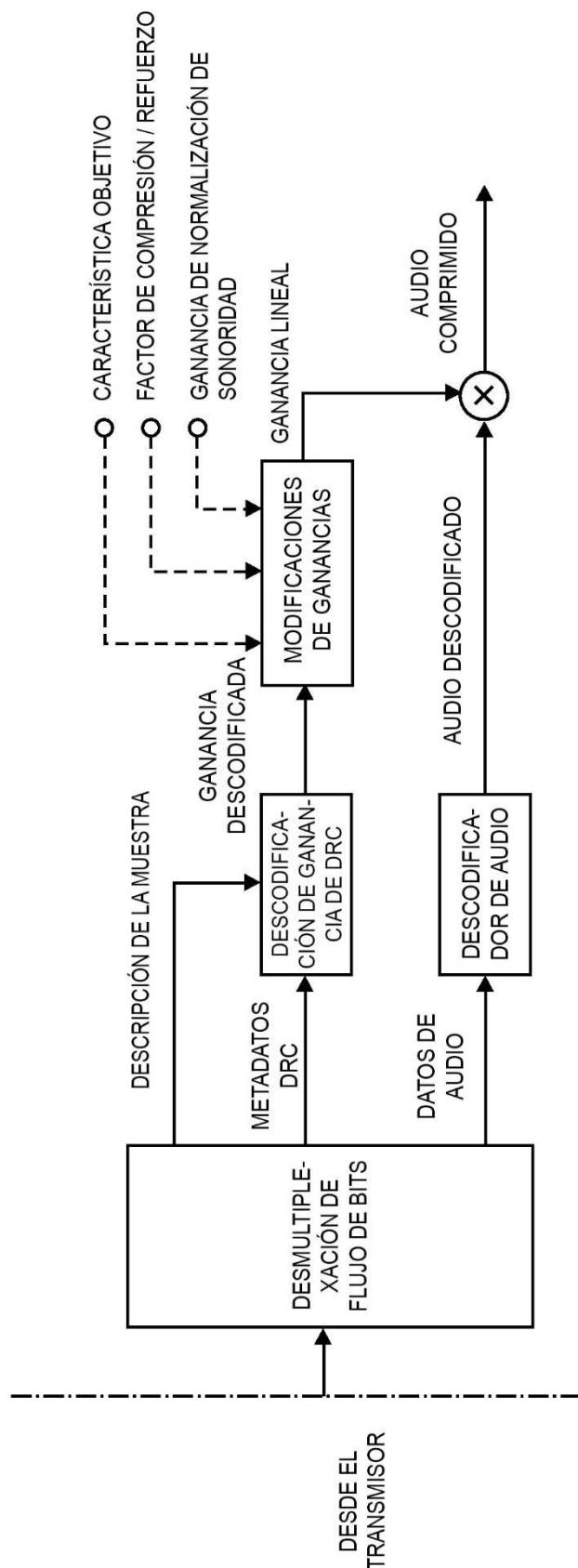
[0101] Como se explicó anteriormente, un modo de realización de la invención puede ser un medio legible por máquina tal como uno o más dispositivos de memoria de estado sólido que tienen almacenadas instrucciones en el mismo que programan uno o más componentes de procesamiento de datos (denominados genéricamente "procesador" o "sistema informático") para realizar algunas de las operaciones descritas anteriormente. En otros modos de realización, algunas de estas operaciones pueden ser realizadas por componentes de hardware específicos que contienen lógica cableada. Esas operaciones pueden realizarse de forma alternativa mediante cualquier combinación de componentes de procesamiento de datos programados y componentes de circuitos fijos cableados.

[0102] Aunque se han descrito y mostrado ciertos modos de realización en los dibujos adjuntos, se debe entender que tales modos de realización son meramente ilustrativos de la amplia invención y no restrictivos, y que la invención no se limita a las construcciones y disposiciones específicas mostradas y descritas, ya que a los expertos en la técnica se les pueden presentar otras modificaciones. Por lo tanto, la descripción se debe considerar como ilustrativa en lugar de limitativa.

25

REIVINDICACIONES

1. Un procedimiento de codificación de valores de ganancia de control de rango dinámico, DRC, en un flujo de bits que representa una parte del contenido del programa de sonido, que comprende:
 - 5 agrupar cada canal de audio del contenido del programa de sonido en un solo grupo de DRC de un conjunto de grupos de DRC; e
 - 10 insertar metadatos de ganancia de DRC en el flujo de bits para cada grupo de DRC, en el que los metadatos de ganancia de DRC para cada grupo de DRC se usan para aplicar los valores de ganancia de DRC correspondientes a cada trama en el grupo de DRC.
2. El procedimiento según la reivindicación 1, en el que los metadatos de ganancia de DRC para cada grupo de DRC incluyen:
 - 15 un primer valor de datos que representa un modo de codificación seleccionado para un valor de ganancia de DRC inicial;
 - 20 un segundo valor de datos que representa el valor de ganancia de DRC inicial; y
 - 25 un tercer valor de datos que representa las diferencias aplicadas al valor de ganancia de DRC inicial para generar los valores de ganancia de DRC para cada trama del grupo de DRC.
3. El procedimiento según la reivindicación 2, en el que el primer valor de datos representa el número de valores de ganancia para aplicar a cada trama del grupo de DRC basándose en el valor de ganancia de DRC inicial.
4. El procedimiento según la reivindicación 1, en el que se asignan múltiples canales a un único grupo de DRC.
5. El procedimiento según la reivindicación 2, en el que se usa una resolución de tiempo no uniforme para velocidades de actualización de valores de ganancia de DRC basándose en la varianza de ganancia generada por el codificador DRC, para minimizar la velocidad de transferencia de bits del flujo de bits.
- 30 35 40 6. El procedimiento según la reivindicación 2, en el que el primer valor de datos que representa el valor de ganancia inicial se codifica usando una escala de cuantificación no uniforme basada en psicoacústica para minimizar la velocidad de transferencia de bits del flujo de bits.
7. El procedimiento según la reivindicación 2, en el que el primer valor de datos que representa el valor de ganancia inicial y el tercer valor de datos que representa las diferencias a aplicar al valor de ganancia de DRC inicial se codifican usando códigos de longitud variable para minimizar la velocidad de transferencia de bits del flujo de bits.
8. Un medio legible por máquina que tiene almacenados en el mismo instrucciones que programan un sistema informático para realizar un procedimiento de acuerdo con cualquiera de las reivindicaciones 1 a 7.

FIG. 1

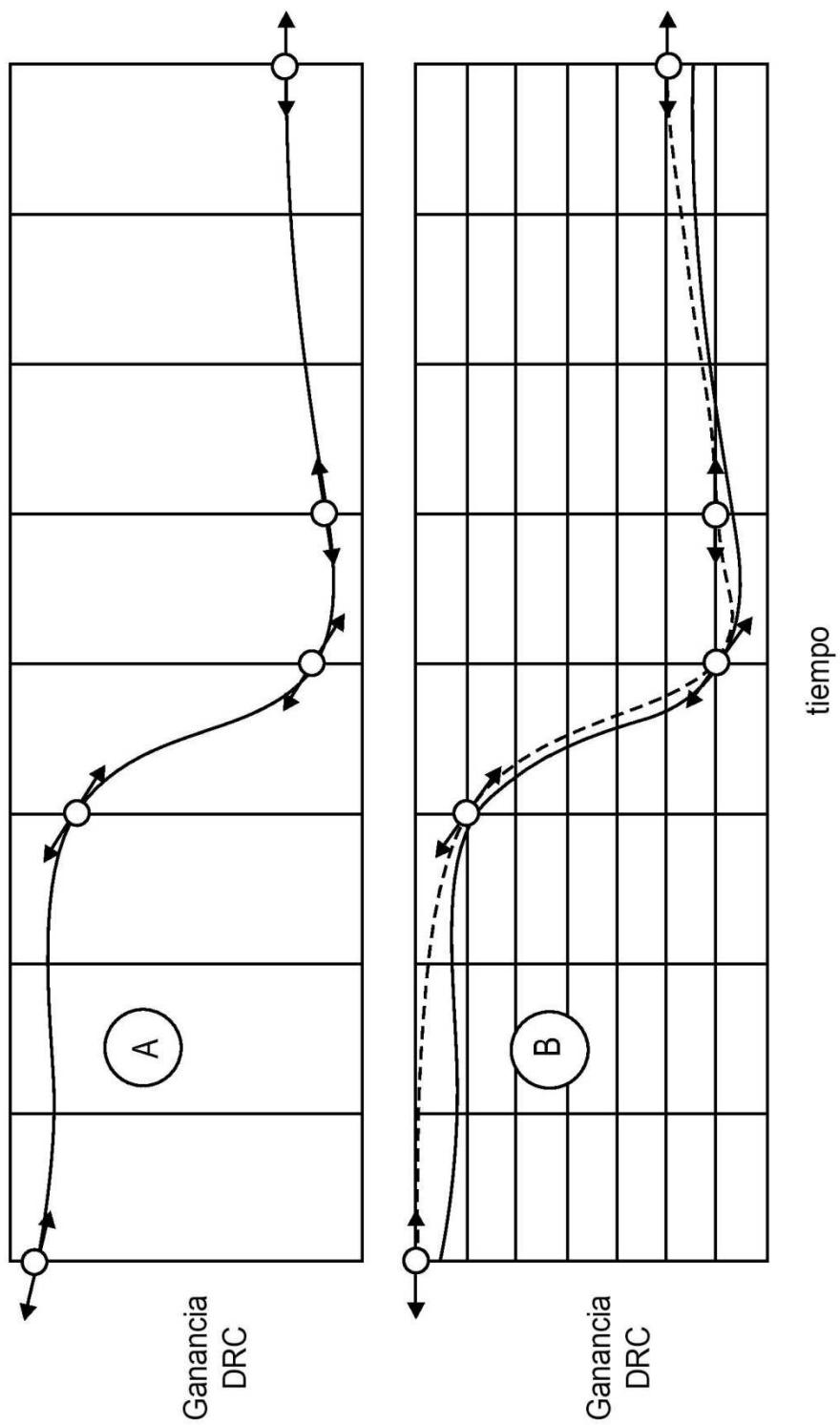


FIG. 2

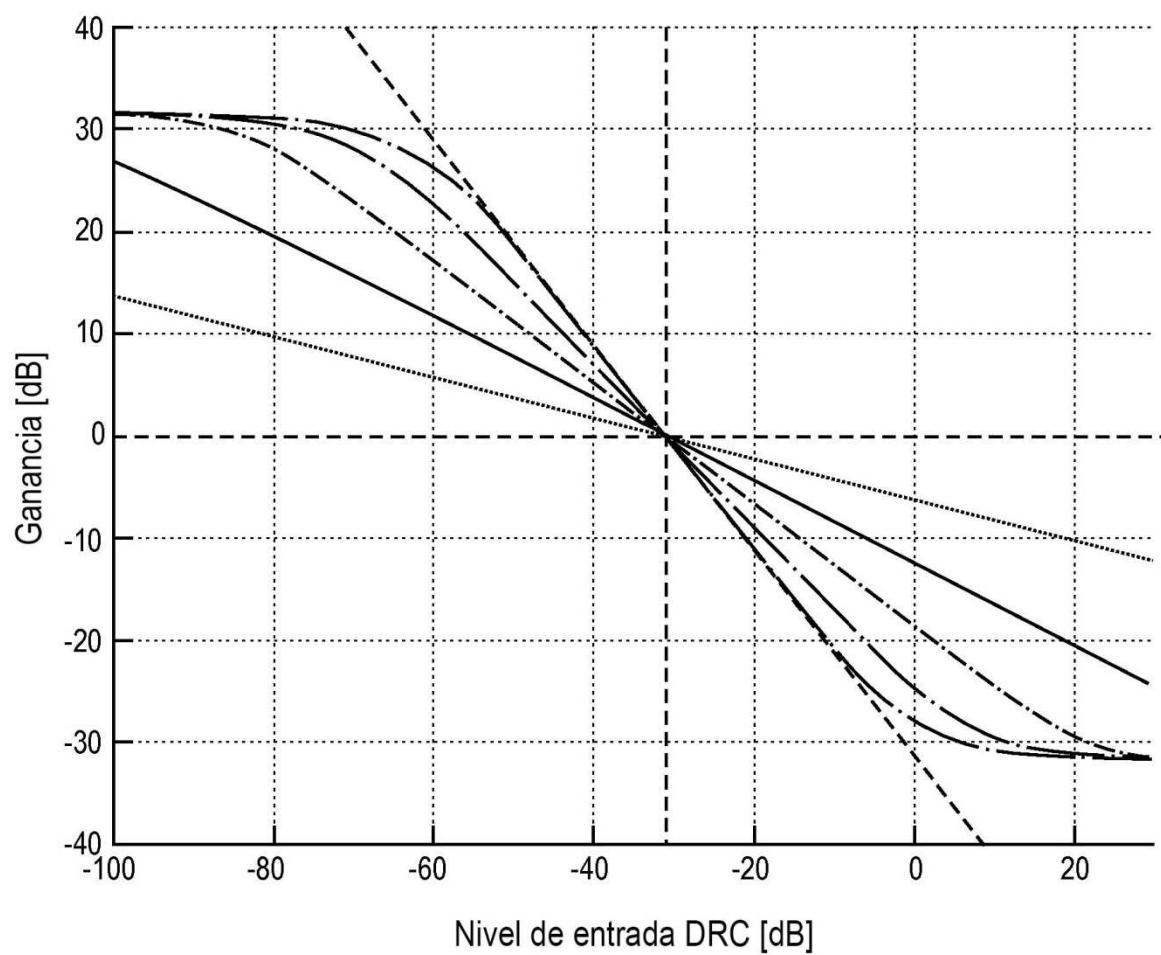
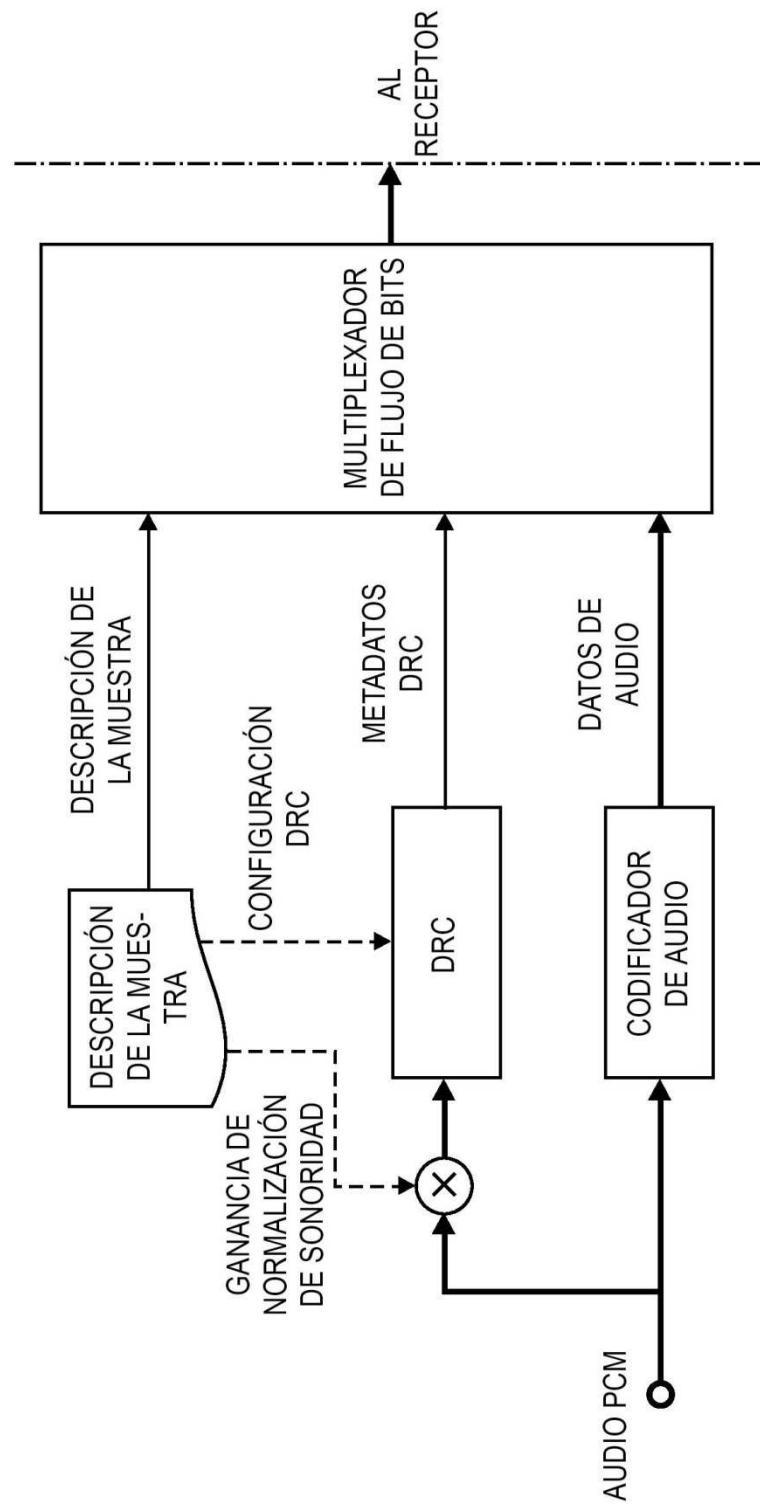


FIG. 3

**FIG. 4**

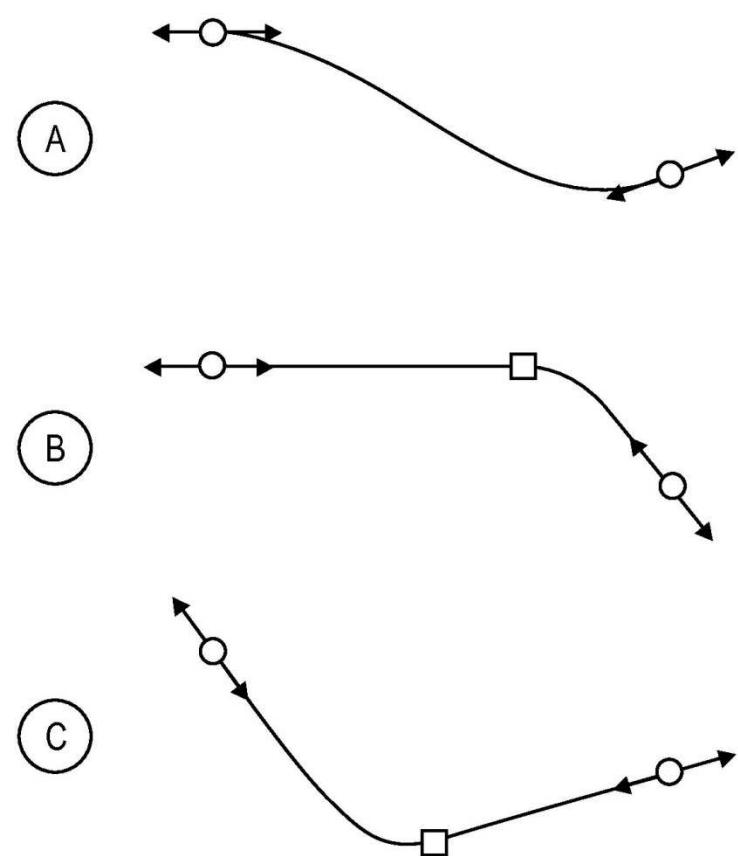


FIG. 5

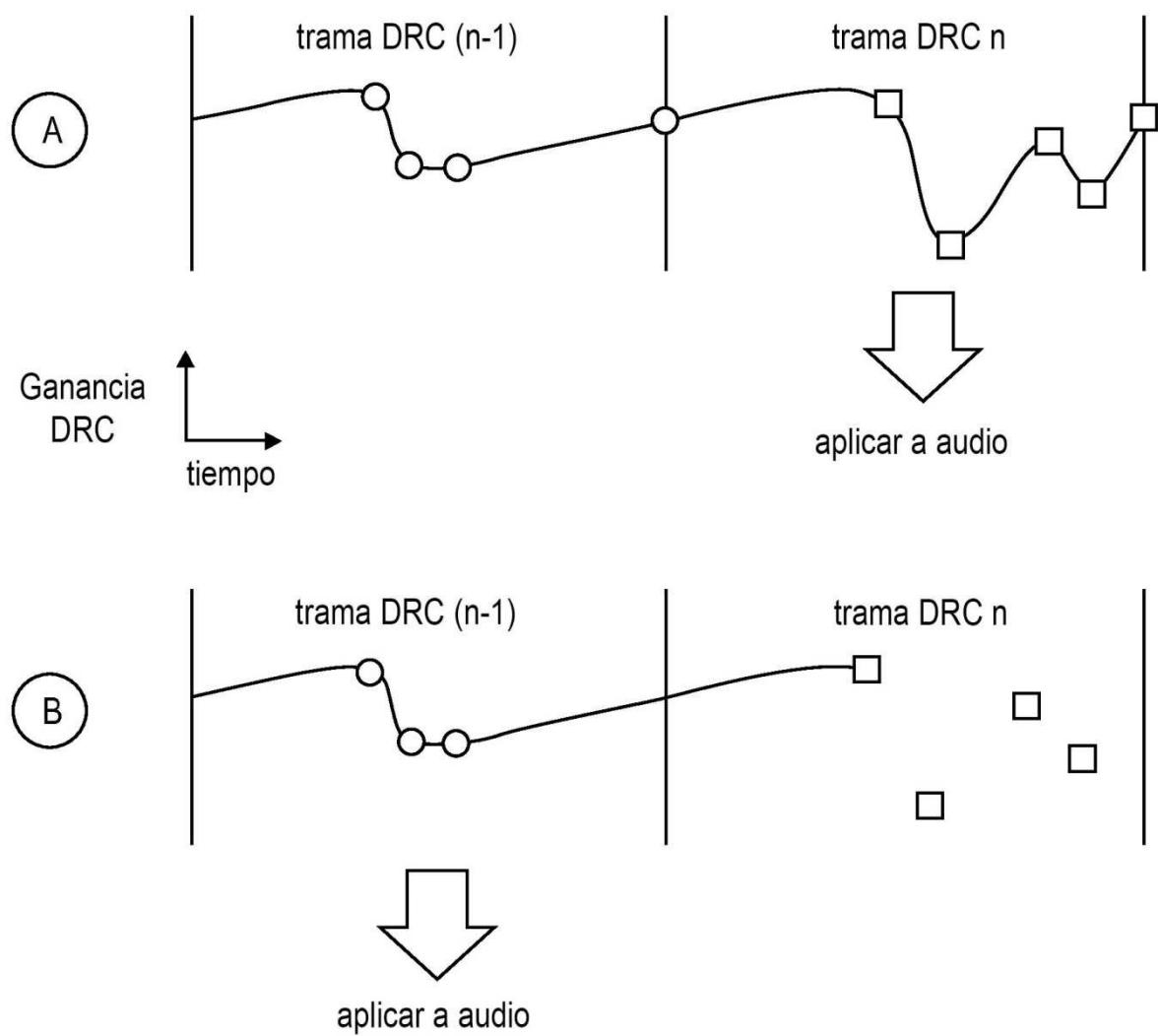
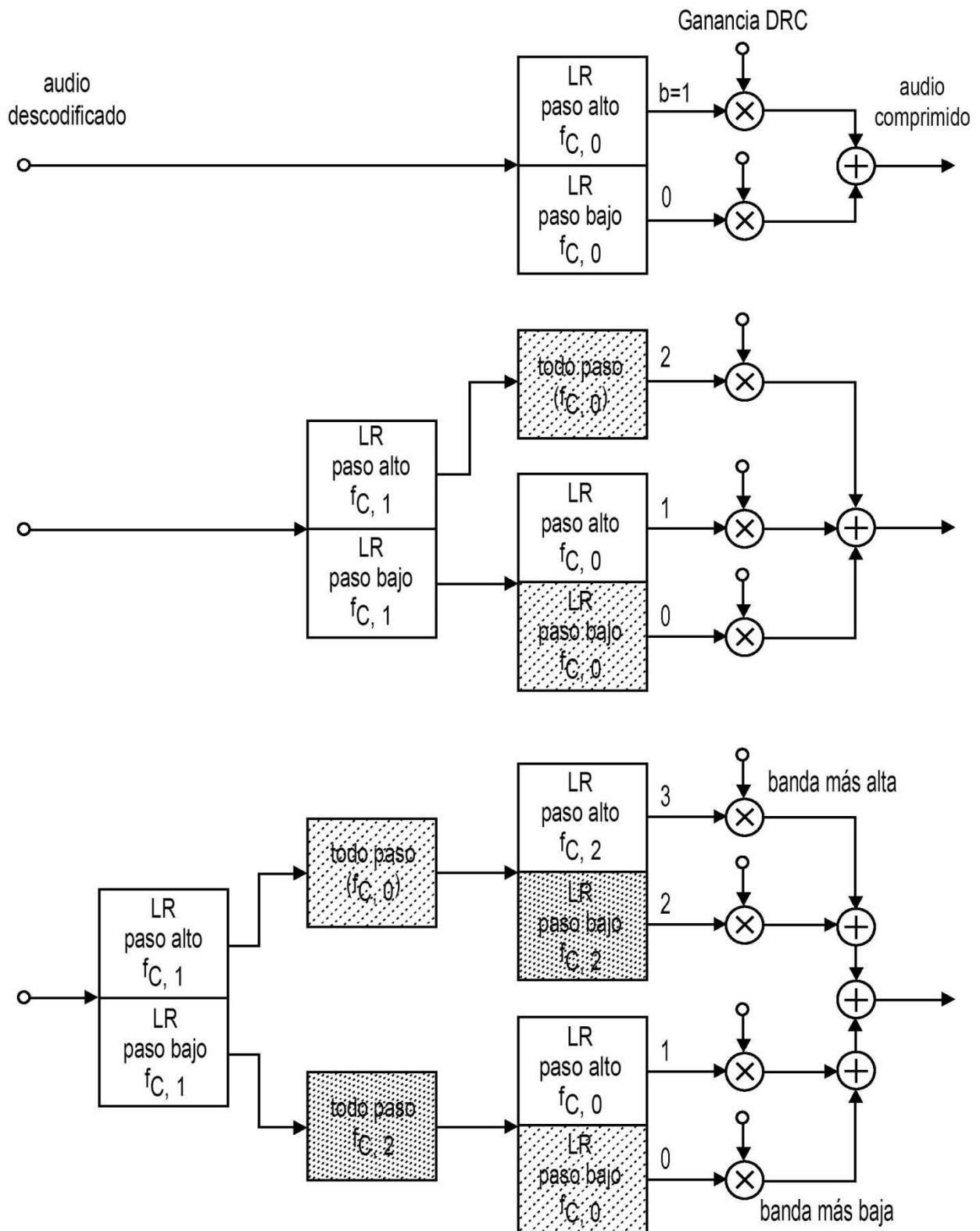
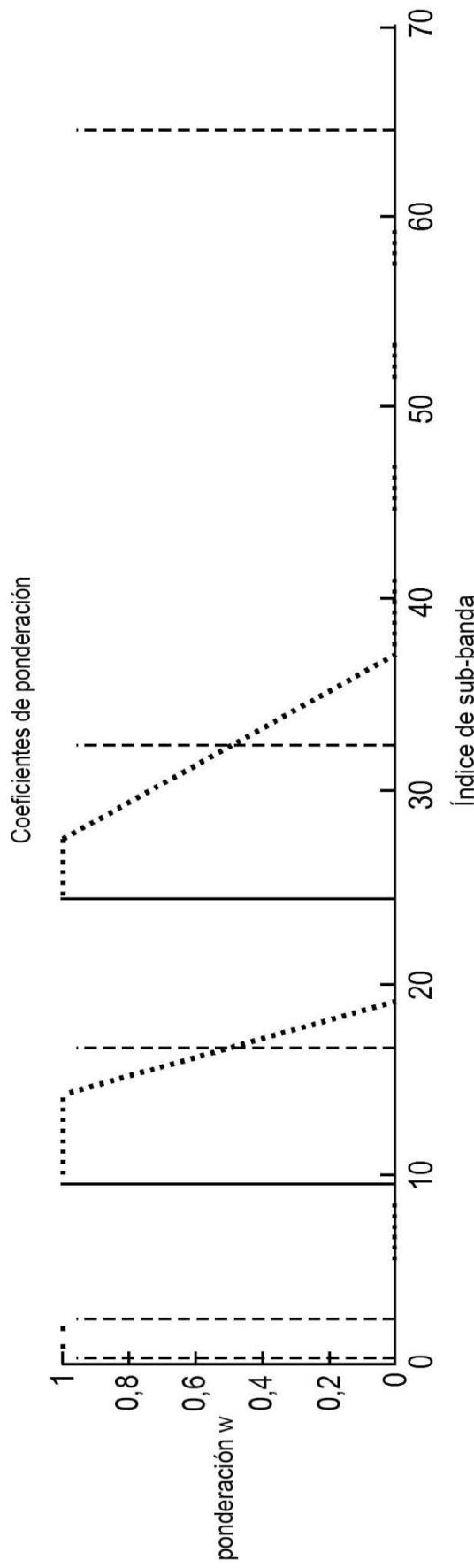


FIG. 6

**FIG. 7**

**FIG. 8**

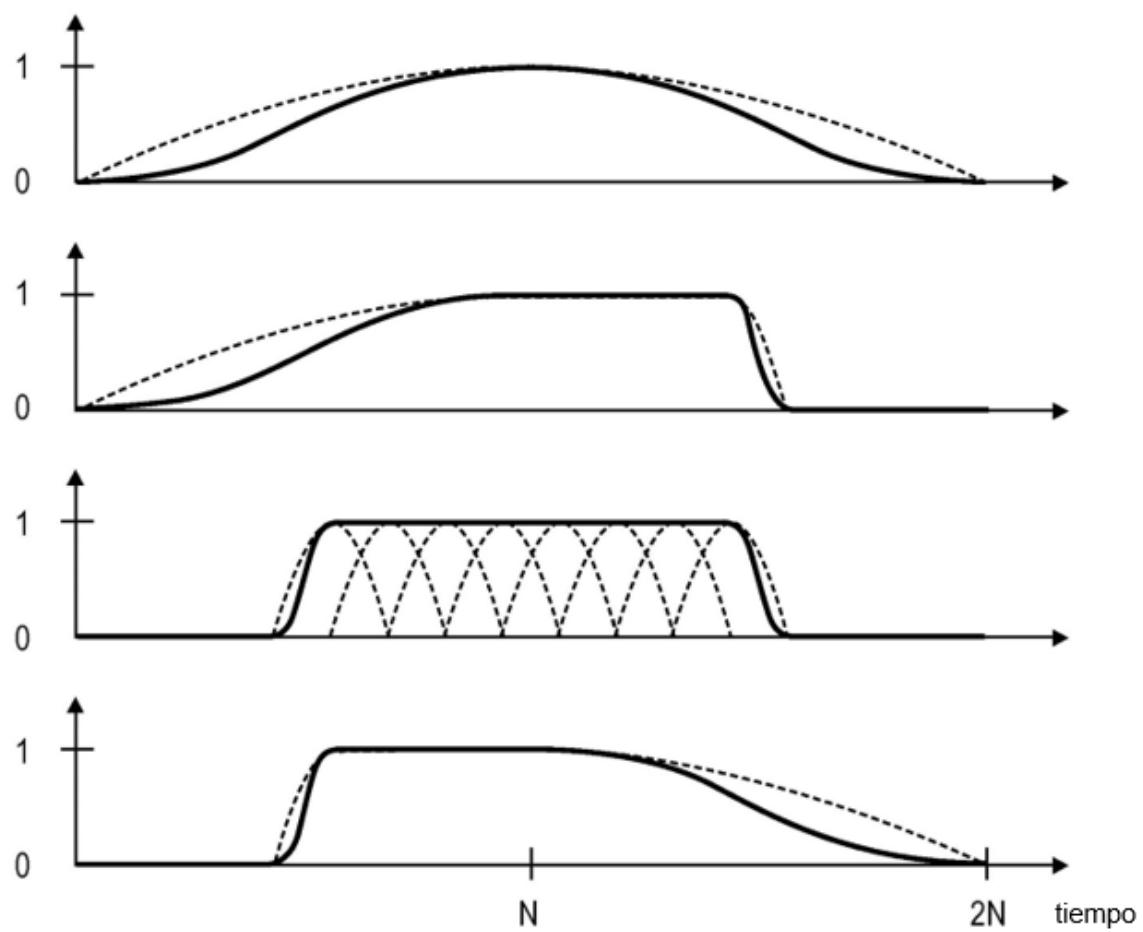


FIG. 9

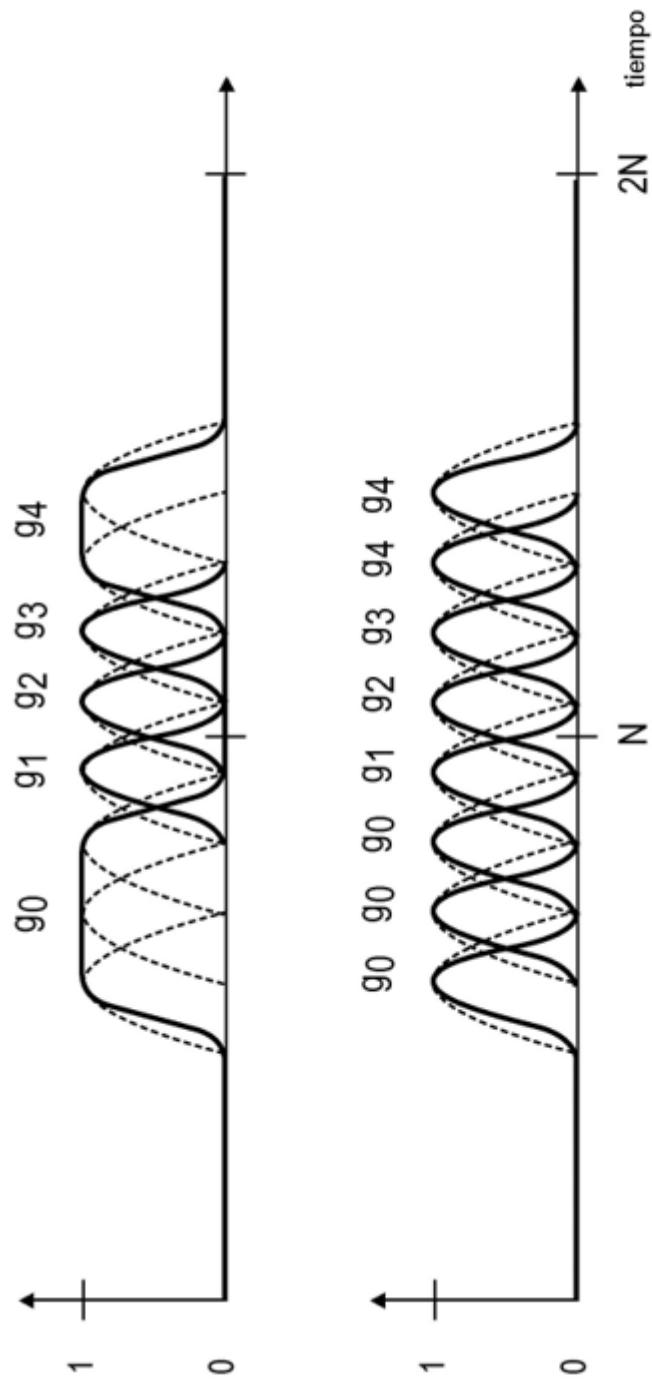
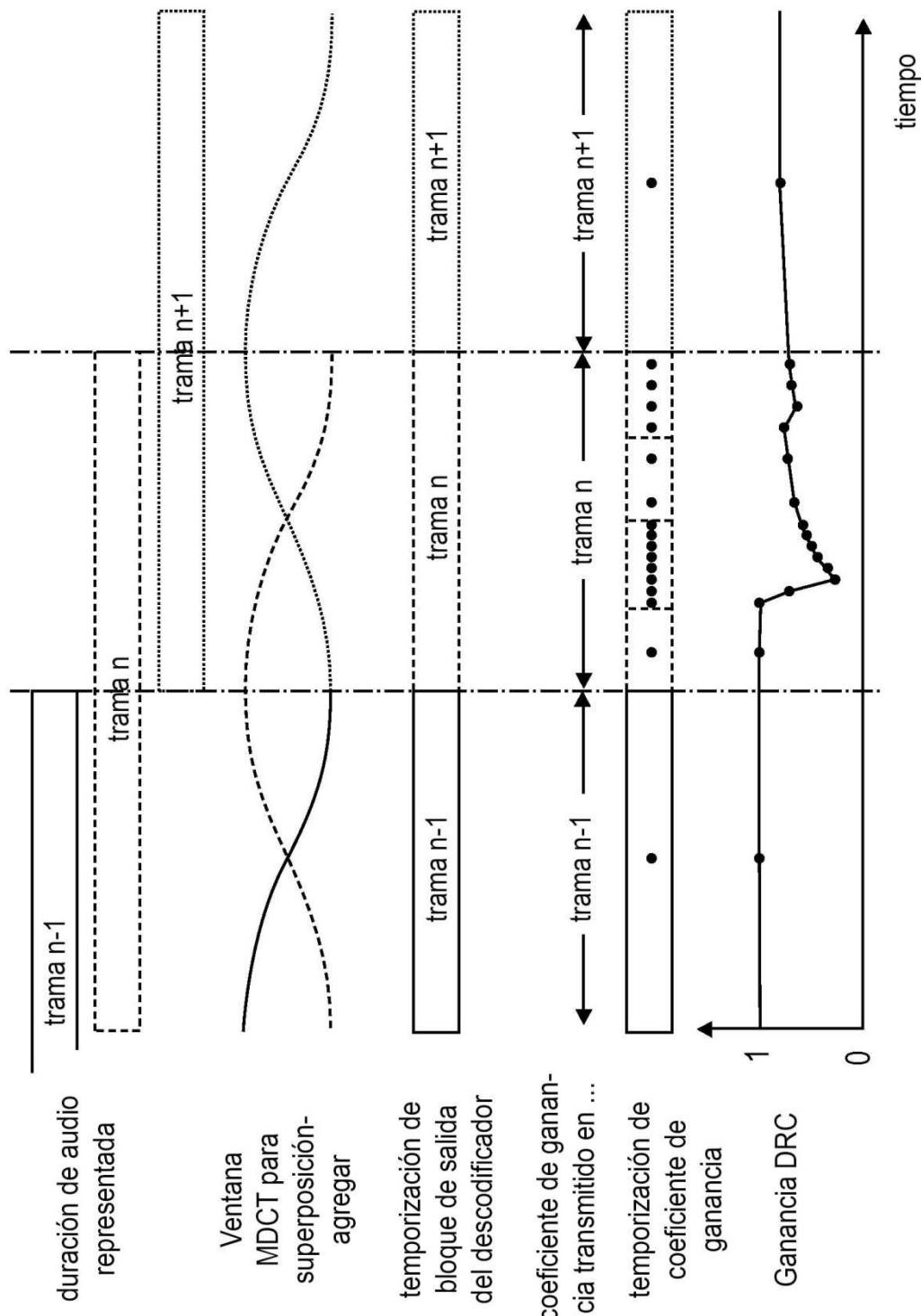


FIG. 10

**FIG. 11**