INTEGRACIÓN DE TRANSPOSITOR ARMÓNICO COMPATIBLE CON VERSIONES ANTERIORES PARA LA RECONSTRUCCIÓN DE ALTA FRECUENCIA DE SEÑALES DE AUDIO

CAMPO TÉCNICO

Las formas de realización se refieren al procesamiento de señales de audio y, más específicamente, a la codificación, decodificación o transcodificación de flujos de bits de audio con datos de control que especifican se ha de ejecutar una forma básica de reconstrucción de alta frecuencia ("HFR", por sus siglas en inglés, High-Frequency Reconstruction) o una forma mejorada de HFR de los datos de audio.

ANTECEDENTES DE LA INVENCIÓN

Un flujo de bits de audio típico incluye tanto datos de audio (por ej., datos de audio codificados) indicativos de uno o más canales de contenido de audio, como metadatos indicativos de por lo menos una característica de los datos de audio o el contenido de audio. Uno de los formatos muy conocidos para generar un flujo de bits de audio codificado es el formato de Codificación Avanzada de Audio (AAC, por sus siglas en inglés Advanced Audio Coding) de MPEG–4 (del inglés Moving Picture Expert Group 4, Grupo de Expertos en Imágenes en Movimiento 4), descripto en la norma MPEG ISO/IEC 14496-3:2009. En la norma de MPEG-4, AAC indica "codificación avanzada de audio" y HE-AAC significa "codificación avanzada de audio de alta eficiencia"

La norma de AAC MPEG-4 define varios perfiles de audio, que determinan qué objetos y herramientas de codificación están presentes en un codificador o decodificador de conformidad. Tres de estos perfiles de audio son (1) el perfil de AAC, (2) el perfil de HE-AAC y (3) el perfil de HE-AAC v2. El perfil de AAC incluye el tipo de objeto de AAC de baja complejidad (o “AAC-LC” del inglés Advanced Audio Coding–Low Complexity). El objeto de AAC-LC es la contraparte del perfil de baja complejidad AAC MPEG-2, con algunos ajustes, y no incluye el tipo de objeto de la replicación de la banda espectral (por sus siglas en inglés, SBR, Spectral Band Replication) ni el tipo de objeto estéreo paramétrico (por sus siglas en inglés, “PS”, Parametric Stereo). El perfil HE-AAC es una superserie del perfil de AAC e incluye además el tipo de objeto de SBR. El perfil de HE-AAC v2 es una superserie del perfil de HE-AAC e incluye además el tipo de objeto PS.

El tipo de objetos SBR contiene la herramienta de replicación de la banda espectral, que es una importante herramienta de codificación para la reconstrucción de alta frecuencia (“HFR”) que mejora significativamente la eficiencia de compresión de los códecs de audio perceptual. La SBR reconstruye los componentes de alta frecuencia de una señal de audio del lado del receptor (por ej., en el decodificador). De esa manera, el codificador sólo necesita codificar y transmitir los componentes de baja frecuencia, dando lugar a una calidad de audio muy superior a bajas tasas de datos. La SBR se basa en la replicación de secuencias de armónicos, previamente truncas a fin de reducir la tasa de datos, a partir de la señal limitada en ancho de banda existente y datos de control obtenidos del codificador. La relación entre los componentes tonales y de ruido se mantiene mediante el filtrado inverso adaptativo, como así también la adición opcional de ruido y sinusoides. En la norma MPEG-4 AAC, la herramienta de SBR ejecuta el emparchado (“patching”) espectral (también

llamado traslación lineal o traslación espectral), en el cual se copia (o “emparcha”) un número de subbandas consecutivas de Filtros Espejo en Cuadratura (por sus siglas en inglés, QMF, Quadrature Mirror Filter de una porción de baja banda transmitida de una señal de audio a una porción de banda alta de la señal de audio, que se genera en el decodificador.

El emparchado espectral o traslación lineal puede no ser ideal para ciertos tipos de audio, tales como el contenido musical con frecuencias de cruce relativamente bajas. Por lo tanto, se necesitan técnicas para mejorar la replicación de la banda espectral.

Breve Descripción de las Formas de Realización de la invención

Una primera clase de formas de realización se relaciona con un método para decodificar un flujo de bits de audio que se da a conocer. El método incluye recibir el flujo de bits de audio codificado y decodificar los datos de audio para generar una señal de audio de banda baja decodificada. El método incluye además extraer metadatos de reconstrucción de alta frecuencia y filtrar la señal de audio de banda baja decodificada con un banco de filtros de análisis para generar una señal de audio de baja banda filtrada. El método incluye asimismo extraer una bandera que indica si se ha de ejecutar la traslación espectral o transposición armónica en los datos de audio y regenerar una porción de banda alta de la señal de audio usando la señal de audio de baja banda filtrada y los metadatos de reconstrucción de alta frecuencia de acuerdo con la bandera. Por último, el método incluye combinar la señal de audio de baja banda filtrada y la porción de banda alta regenerada para formar una señal de audio de banda ancha.

Una segunda clase de formas de realización se relaciona con un decodificador de audio para decodificar un flujo de bits de audio codificado. El decodificador incluye una interfaz de entrada para recibir el flujo de bits de audio codificado, donde el flujo de bits de audio codificado incluye datos de audio que representan una porción de baja banda de una señal de audio y un decodificador del núcleo para decodificar los datos de audio y generar una señal de audio de banda baja decodificada. El decodificador incluye además un demultiplexor para extraer del flujo de bits de audio codificando metadatos de reconstrucción de alta frecuencia, donde los metadatos de reconstrucción de alta frecuencia incluyen parámetros operativos para un proceso de reconstrucción de alta frecuencia que traslada linealmente un número de subbandas consecutivas de una porción de baja banda de la señal de audio a una porción de banda alta de la señal de audio y un banco de filtros de análisis para filtrar la señal de audio de banda baja decodificada a fin de generar una señal de audio de baja banda filtrada. El decodificador incluye asimismo un demultiplexor para extraer del flujo de bits de audio codificado una bandera que indica si se ha de ejecutar una traslación lineal o una transposición armónica en los datos de audio y un regenerador de alta frecuencia para regenerar una porción de banda alta de la señal de audio usando la señal de audio de baja banda filtrada y los metadatos de reconstrucción de alta frecuencia de acuerdo con la bandera. Por último, el decodificador incluye un banco de filtros de síntesis para combinar la señal de audio de baja banda filtrada y la porción de banda alta regenerada a fin de formar una señal de audio de banda ancha.

Otras clases de formas de realización se relacionan con la codificación y transcodificación de flujos de bits de audio que contienen metadatos que identifican

si se debe ejecutar o no el procesamiento de replicación de la banda espectral intensificado (por sus siglas en inglés, eSBR, enhanced spectral band replication).

Breve Descripción de los Dibujos

La FIG. 1 es un diagrama de bloques de una forma de realización de un sistema que puede ser configurado para ejecutar una forma de realización del método de la invención.

La FIG. 2 es un diagrama de bloques de un codificador que constituye una forma de realización de la unidad de procesamiento de audio de la invención.

La FIG. 3 es un diagrama de bloques de un sistema que incluye un decodificador que constituye una forma de realización de la unidad de procesamiento de audio de la invención, y opcionalmente también un post-procesador acoplado al mismo.

La FIG. 4 es un diagrama de bloques de un decodificador que constituye una forma de realización de la unidad de procesamiento de audio de la invención.

La FIG. 5 es un diagrama de bloques de un decodificador que es otra forma de realización de la unidad de procesamiento de audio de la invención.

La FIG. 6 es un diagrama de bloques de otra forma de realización de la unidad de procesamiento de audio de la invención.

La FIG. 7 es un diagrama de un bloque de un flujo de bits de AAC de MPEG-4, que incluye los segmentos en los cuales éste se divide.

Notación y Nomenclatura

En toda esta revelación, inclusive en las reivindicaciones, la expresión realizar una operación “en” una señal o datos (por ej., filtrado, escalado, transformación o aplicación de ganancia a la señal o los datos) se utiliza en un sentido amplio para indicar la ejecución de la operación directamente en la señal o los datos, o en una versión procesada de la señal o los datos (por ej., en una versión de la señal que ha sido objeto de filtrado preliminar o procesamiento previo anterior a la ejecución de la operación en la misma).

En toda esta revelación, inclusive en las reivindicaciones, la expresión “unidad de procesamiento de audio” o “procesador de audio” se utiliza en un sentido amplio para indicar un sistema, dispositivo o aparato configurado para procesar datos de audio. Los ejemplos de unidades de procesamiento de audio incluyen, aunque no a modo de limitación, codificadores, transcodificadores, decodificadores y códecs, sistemas de pre–procesamiento, sistemas de post–procesamiento y sistemas de procesamiento de flujos de bits (a los que en ocasiones se hace referencia como herramientas de procesamiento de flujos de bits). Prácticamente todos los dispositivos electrónicos del consumidor, tales como teléfonos móviles, televisores, laptops y computadoras tablet contienen una unidad de procesamiento de audio o procesador de audio.

En toda esta revelación, inclusive en las reivindicaciones, el término “se acopla” o “acoplado” se utiliza en un sentido amplio para referirse a una conexión directa o indirecta. Por consiguiente, si un primer dispositivo se acopla a un Segundo dispositivo, esa conexión puede ser una conexión directa o a través de una conexión indirecta a través de otros dispositivos y conexiones. Más aun, los componentes que están integrados a o con otros componentes también se acoplan entre sí.

Descripción Detallada de las Formas de Realización de la invención

La norma de MPEG–4 AAC contempla que un flujo de bits de MPEG-4 AAC codificado incluye metadatos que indican cada tipo de procesamiento de reconstrucción de alta frecuencia (“HFR”) que un decodificador ha de aplicar (si es que se debe aplicar alguna) para decodificar contenido de audio del flujo de bits, y/o que controla dicho procesamiento de HFR y/o es indicativo de por lo menos una característica o parámetro de por lo menos una herramienta de HFR que se ha de emplear para decodificar contenido de audio del flujo de bits. En la presente utilizamos la expresión “metadatos de SBR” para indicar metadatos de este tipo que se describen o mencionan en la norma de MPEG–4 AAC para usar con la replicación de la banda espectral (“SBR”). Como sabrá apreciar una persona con capacitación en la técnica, la SBR es parte de la HFR.

La SBR se utiliza preferentemente como Sistema de dos velocidades, con el códec subyacente operando a la mitad de la tasa de muestreo original, en tanto que la SBR opera a la tasa de muestreo original. El codificador de SBR funciona en paralelo con el códec del núcleo subyacente, aunque a una tasa de muestreo más elevada. Si bien la SB Res principalmente un post proceso que se lleva a cabo en el decodificador, se extraen parámetros importantes en el codificador a fin de garantizar la reconstrucción de alta frecuencia más precisa en el decodificador. El codificador estima la envolvente espectral del rango de SBR para un rango/resolución en tiempo y frecuencia adecuado para las características de los segmentos de la señal de entrada actual. La envolvente espectral es estimada por un complejo análisis de QMF y el subsiguiente cálculo de energía. Se pueden elegir las resoluciones en tiempo y frecuencia de las envolventes espectrales con un alto nivel de libertad, para

garantizar la resolución en tiempo frecuencia más adecuada para el segmento de entrada dato. La estimación de la envolvente debe tomar en cuenta que un transitorio en la original, situado principalmente en la región de alta frecuencia (por ejemplo, platillos de batería), está presente en menor medida en la banda alta generada por SBR antes del ajuste de la envolvente, puesto que la banda alta en el decodificador se basa en la banda baja, donde el transitorio es mucho menos pronunciado en comparación con la banda alta. Este aspecto impone diferentes requisitos para la resolución en tiempo frecuencia de los datos de la envolvente espectral, en comparación con el cálculo de la envolvente espectral corriente utilizado en otros algoritmos de codificación de audio.

Aparte de la envolvente espectral, se extraen varios parámetros adicionales que representan las características espectrales de la señal de entrada para diferentes regiones de tiempo y frecuencia. Dado que el codificador naturalmente tiene acceso a la señal original, como así también información sobre la manera en que la unidad de SBR del decodificador genera la banda alta, dados los parámetros de control específico, es posible que el Sistema maneje situaciones en que la banda baja constituye una fuerte serie de armónicos y la banda alta, que se ha de recrear, constituye principalmente componentes aleatorios de la señal, como así también situaciones en que hay fuertes componentes tonales presentes en la banda alta original sin contrapartes en la banda baja, sobre las cuales se basa la región de banda alta. Más aun, el codificador de SBR funciona en estrecha relación con el códec del núcleo subyacente para evaluar qué rango de frecuencias debe cubrir la SBR en un momento dado. Los datos de SBR son codificados con eficiencia antes de la transmisión aprovechando la codificación entrópica, como así también las

dependencias de los canales de los datos de control, en el caso de las señales estéreo.

Los algoritmos de extracción de parámetros de control por lo general deben adaptarse meticulosamente al códec subyacente a una tasa de bits dada y una tasa de muestreo dada. Esto se debe a que una tasa de bits más baja por lo general un rango de SBR más amplio en comparación con una alta tasa de bits, y que las diferentes velocidades de muestreo corresponden a diferentes resoluciones temporales de las tramas de SBR.

Un decodificador de SBR incluye típicamente varias partes diferentes. Comprende un módulo de decodificación del flujo de bits, un módulo de reconstrucción de alta frecuencia (HFR), un módulo adicional de componentes de alta frecuencia y un módulo ajustador de la envolvente. El Sistema se basa en un banco de filtros QMF de valor complejo. En el modulo de extracción de flujos de bits se leen los datos de control del flujo de bits y se los decodifica. Se obtiene la rejilla de alta frecuencia correspondiente a la trama en curso, antes de leer los datos de la envolvente del flujo de bits. El decodificador del núcleo subyacente decodifica la señal de audio de la trama actual (aunque a una tasa de muestreo más baja) para producir muestras de audio en el dominio del tiempo. La trama de audio así obtenido se utiliza para la reconstrucción de alta frecuencia realizada por el módulo de HFR. Luego se analiza la señal de banda baja decodificada usando un banco de filtros QMF. Seguidamente se ejecuta la reconstrucción de alta frecuencia y el ajuste de la envolvente en las muestras de subbanda del banco de filtros QMF. Las altas frecuencias se reconstruyen a partir de la banda baja de manera flexible, basándose en los parámetros de control dados. Más aun, la banda alta reconstruida es filtrada adaptativamente por canales de subbandas de acuerdo con los datos de control para

garantizar las características espectrales apropiadas de la región dada de tiempo/frecuencia.

El nivel superior de un flujo de bits de MPEG-4 AAC es una secuencia de bloques de datos (elementos “raw_data_block”), cada uno de los cuales es un segmento de datos (al que en la presente se hace referencia como “bloque”) que contiene datos de aduio (típicamente para un período de tiempo de 1024 o 960 muestras) e información relacionada y/o otros datos. En la presente empleamos el término “bloque” para indicar un segmento de un flujo de bits de MPEG-4 AAC que comprende datos de audio (y los correspondientes metadatos y, opcionalmente, también otros datos relacionados) que determinan o son indicativos de un (pero no más de uno) elemento “raw_data_block”.

Cada bloque de un flujo de bits MPEG–4 AAC puede incluir un número de elementos de sintaxis (cada uno de los cuales también se materializa en el flujo de bits en forma de segmento de datos). En la norma de MPEG–4 AAC se definen siete tipos de esos elementos de sintaxis. Cada elemento de sintaxis se identifica mediante un valor diferente del elemento de datos “id_syn_ele.” Los ejemplos de elementos de sintaxis incluyen un “single_channel_element()”, un “channel_pair_element(),” y un “fill_element().” Un elemento de canal único es un contenedor que incluye datos de audio de un canal de audio único (una señal de audio monofónica). Un elemento de par de canales incluye datos de audio de dos canales de audio (es decir, una señal de audio estéreo).

Un elemento de relleno es un contenedor de información que incluye un identificador (por ej., el valor del elemento antes citado “id_syn_ele”) seguido por datos, a los que se hace referencia como “datos de relleno”. Los elementos de

relleno se han utilizado históricamente para ajustar la tasa de bits instantánea de los flujos de bits que se deben transmitir a través de un canal de tasa constante. Mediante la adición de la cantidad apropiada de datos de relleno a cada bloque, se puede obtener una tasa de datos constante.

De conformidad con las formas de realización de la invención, los datos de relleno pueden incluir una o más cargas útiles de extensión que extienden el tipo de datos (por ej., metadatos) aptos para ser transmitidos en un flujo de bits. Un dispositivo (por ej., un decodificador) que recibe el flujo de bits puede utilizar, opcionalmente, un decodificador que recibe flujos de bits con datos de relleno que contienen un nuevo tipo de datos para extender la funcionalidad del dispositivo. Por consiguiente, como ha de apreciar una persona con capacitación en la técnica, los elementos de relleno son un tipo especial de estructura de datos y son diferentes de las estructuras de datos empleadas típicamente para transmitir datos de audio (por ej., cargas útiles de audio que contienen datos de canales).

En algunas formas de realización de la invención, el identificador empleado para identificar un elemento de relleno puede consistir en un entero de tres bits sin signo transmitido con el bit más significativo primero (“uimsbf”) con un valor de 0x6. En un bloque se pueden presentar varios casos del mismo tipo de elemento de sintaxis (por ej., varios elementos de relleno).

Otra norma para codificar flujos de bits de audio es la norma de Codificación Unificada de Voz y Audio (USAC, por sus siglas en inglés Unified Speech and Audio Coding) de MPEG (ISO/IEC 23003-3:2012). La norma MPEG USAC describe la codificación y decodificación de contenido de audio utilizando el procesamiento de replicación de la banda espectral (incluyendo el procesamiento SBR descripto en la

norma de MPEG–4 AAC, e incluyendo asimismo otras formas mejoradas de procesamiento de replicación de la banda espectral). Este procesamiento aplica herramientas de replicación de la banda espectral (a las que en ocasiones se hace referencia en la presente como “herramientas de SBR mejorada” o “herramientas de eSBR”) de una versión expandida y mejorada de la serie de herramientas de SBR descriptas en la norma de MPEG–4 AAC. Por consiguiente, la eSBR (según la definición de la norma de USAC) constituye una mejora de la SBR (según la definición de la norma MPEG-4 AAC).

En la presente, utilizamos la expresión “procesamiento de SBR mejorada” (o “procesamiento de eSBR”) para indicar procesamiento de replicación de la banda espectral usando por lo menos una herramienta de eSBR (por ej., por lo menos una herramienta de eSBR que se describe o menciona en la norma MPEG USAC) que no se describe ni menciona en la norma de MPEG–4 AAC. Los ejemplos de esas herramientas de eSBR son la transposición armónica y el pre–procesamiento adicional de emparchado por QMF o “pre-ecualización de nivel o aplanamiento previo (“flattening”)”.

Un transpositor armónico del orden entero T mapea una sinusoide con frecuencia ω con una sinusoide con frecuencia Tω, preservando a la vez la duración de la señal. Por lo general se utilizan tres órdenes, T = 2, 3, 4, en secuencia para producir cada parte del rango de frecuencias de salida deseado utilizando el orden de transposición más bajo posible. En caso de requerirse una salida superior al cuarto orden del rango de transposición, se la puede generar mediante desplazamientos en frecuencia. Siempre que resulta posible, se generan dominios temporales de banda base con muestreo crítico para el procesamiento a fin de minimizar la complejidad informática.

Un flujo de bits generado de conformidad con la norma de MPEG USAC (al que en ocasiones se hace referencia en la presente como "flujo de bits USAC") incluye contenido de audio codificado y por lo general incluye metadatos indicativos de cada tipo de procesamiento de replicación de la banda espectral que ha de aplicar un decodificador para decodificar el contenido de audio del flujo de bits USAC, y/o metadatos que controlan dicho procesamiento de replicación de la banda espectral y/o que son indicativos de por lo menos una característica o parámetro de por lo menos una herramienta de SBR y/o herramienta de eSBR que se ha de emplear para decodificar contenido de audio del flujo de bits USAC.

En la presente, utilizamos la expresión "metadatos de SBR mejorada" (o "metadatos de eSBR") para indicar metadatos indicativos de cada tipo de procesamiento de replicación de la banda espectral que ha de aplicar un decodificador para decodificar contenido de audio de un flujo de bits de audio codificado (por ej., un flujo de bits USAC) y/o que controla dicho procesamiento de replicación de la banda espectral, y/o que es indicativo de por lo menos una característica o parámetro de por lo menos una herramienta de SBR y/o herramienta de eSBR que se ha de emplear para decodificar dicho contenido de audio, pero que no ha sido descripto ni mencionado en la norma de MPEG–4 AAC. Un ejemplo de metadatos de eSBR es el de los metadatos (indicativos del procesamiento de replicación de la banda espectral o que lo controlan) que se describe o menciona en la norma de MPEG USAC, pero no en la norma de MPEG–4 AAC. Por consiguiente, los metadatos de eSBR indican, en la presente, metadatos que no son metadatos de SBR, y los metadatos de SBR indican, en la presente, metadatos que no son metadatos de eSBR.

Un flujo de bits USAC puede incluir tanto metadatos de SBR como metadatos de eSBR. Más específicamente, un flujo de bits USAC puede incluir metadatos de SBR que controlan la eficiencia del procesamiento de eSBR ejecutada por un decodificador, y metadatos de SBR que controlan la eficiencia del procesamiento de SBR ejecutado por el decodificador. De conformidad con las formas de realización típicas de la presente invención, los metadatos de eSBR (por ej., datos de configuración específicos de la eSBR) están incluidos (de conformidad con la presente invención) en un flujo de bits MPEG–4 AAC (por ej., en el contenedor sbr_extension() al final de una carga útil de SBR).

La eficiencia del procesamiento de eSBR, durante la decodificación de un flujo de bits codificado usando una serie de herramientas de eSBR (que comprenden por lo menos una herramienta de eSBR), ejecutado por un decodificador regenera la banda de alta frecuencia de la señal de audio, sobre la base de la replicación de las secuencias de armónicos que se truncaran durante la codificación. Dicho procesamiento de eSBR típicamente ajusta la envolvente espectral de la banda de alta frecuencia generada y aplica filtrado inverso, y agrega ruido y componentes sinusoides a fin de recrear las características espectrales de la señal de audio original.

De conformidad con las formas de realización típicas de la invención, se incluyen metadatos de eSBR (por ej., se incluye un pequeño número de bits de control que son metadatos de eSBR) en uno más segmentos de metadaatos de un flujo de bits de audio codificado (por ej., un flujo de bits MPEG–4 AAC) que también incluye datos de audio codificados en otros segmentos (segmentos de datos de audio). Por lo general, por lo menos uno de esos segmentos de metadatos de cada bloque del flujo de bits es (o incluye) un elemento de relleno (que incluye un

identificador que indica el comienzo del elemento de relleno), y los metadatos de eSBR están incluidos en el elemento de relleno después del identificador.

La FIG. 1 es un diagrama de bloques de una cadena de procesamiento de audio ilustrativo (un sistema de procesamiento de datos de audio), en el cual uno o más de los elementos del sistema puede estar configurado de conformidad con una forma de realización de la presente invención. El sistema incluye los siguientes elementos, acoplados entre sí de acuerdo con lo expuesto: el codificador 1, el subsistema de transporte 2, el decodificador 3 y la unidad de post-procesamiento 4. En variaciones del sistema expuesto, se omite uno o más de los elementos, o se incluyen unidades adicionales de procesamiento de datos de audio.

En algunas implementaciones, el codificador 1 (que opcionalmente incluye una unidad de procesamiento previo) está configurado para aceptar muestras PCM (por sus siglas en inglés Pulse Coded Modulation, Modulación por Impulsos Codificados) (dominio temporal) que comprenden contenido de audio como entrada, y para emitir un flujo de bits de audio codificado (con un formato que cumpla con la norma de MPEG–4 AAC) que es indicativo del contenido de audio. En ocasiones, en la presente se hace referencia a los datos del flujo de bits que son indicativos del contenido de audio como “datos de audio” o “datos de audio codificados.” Si el codificador está configurado de conformidad con una forma de realización típica de la presente invención, el flujo de bits de audio emitido por el codificador incluye metadatos de eSBR (y típicamente también otros metadatos), como así también datos de audio.

Uno o más flujos de bits de audio codificados emitidos por el codificador 1 pueden ser enviados al subsistema de transporte de audio codificado 2. El subsistema 2 está configurado para almacenar y/o transmitir cada flujo de bits

codificado emitido por el codificador 1. Un flujo de bits de audio codificado emitido por el codificador 1 puede ser almacenado por el subsistema 2 (por ej., en forma de un disco DVD o Blu ray), o transmitido por el subsistema 2 (que puede implementar un vínculo o red de transmisión), o puede ser tanto almacenado como transmitido por el subsistema 2.

El decodificador 3 está configurado para decodificar un flujo de bits de audio MPEG-4 AAC codificado (generado por el codificador 1) que recibe a través del subsistema 2. En algunas formas de realización, el decodificador 3 está configurado para extraer metadatos de eSBR de cada bloque del flujo de bits, y para decodificar el flujo de bits (incluso mediante la ejecución del procesamiento de eSBR utilizando los metadatos de eSBR extraídos) para generar datos de audio decodificados (por ej., flujos de muestras de audio PCM decodificados). En algunas formas de realización, el decodificador 3 está configurado para extraer metadatos de SBR del flujo de bits (aunque para ignorar los metadatos de eSBR incluidos en el flujo de bits), y para decodificar el flujo de bits (incluso mediante la ejecución del procesamiento de SBR utilizando los metadatos de SBR extraídos) para generar datos de audio decodificados (por ej., flujos de muestras de audio PCM decodificados). Por lo general, el decodificador 3 incluye una memoria intermedia “buffer” que almacena (por ej., de manera no transitoria) segmentos del flujo de bits de audio codificado recibidos del subsistema 2.

La unidad de post-procesamiento 4 de la Fig. 1 está configurada para aceptar un flujo de datos de audio decodificados procedentes del decodificador 3 (por ej., muestras de audio PCM decodificadas), y para ejecutar el post procesamiento del mismo. La unidad de post-procesamiento también puede estar configurada para

renderizar el contenido de audio post procesado (o el audio decodificado recibido del decodificador 3) para la reproducción por uno o más parlantes.

La FIG. 2 es un diagrama de bloques de un codificador (100) que constituye una forma de realización de la unidad de procesamiento de audio de la invención. Se puede implementar cualquiera de los componentes o elementos del codificador 100 en forma de uno o más procesos y/o uno o más circuitos (por ej., ASICs [por sus siglas en inglés Application–Specific Integrated Circuit, Circuito Integrado para Aplicaciones Específicas], FPGAs [Field Programmable Gate Array, Matriz de Puertas Programables en el Campo], u otros circuitos integrados), en hardware, software, o una combinación de hardware y software. El codificador 100 incluye el codificador 105, la etapa de relleno/formateo 107, la etapa de generación de metadatos 106 y la memoria intermedia 109, conectadas según lo expuesto. Por lo general, también, el codificador 100 incluye otros elementos de procesamiento (no se ilustran). El codificador 100 está configurado para convertir un flujo de bits de audio de entrada en un flujo de bits de salida MPEG-4 AAC codificado.

El generador de metadatos 106 está acoplado y configurado para generar (y/o hacer pasar a la etapa 107) metadatos (que incluyen metadatos de SBR y metadatos de eSBR) que se han de incluir en la etapa 107 en el flujo de bits codificado que ha de ser emitido del codificador 100.

El codificador 105 está acoplado y configurado para codificar (por ej., ejecutando la compresión en el mismo) los datos de audio de entrada, y para enviar los datos codificados así obtenidos a la etapa 107 para su inclusión en el flujo de bits codificado que ha de constituir la salida de la etapa 107.

La etapa 107 está configurada para multiplexar el audio codificado recibido del codificador 105 y los metadatos (incluyendo metadatos de SBR y metadatos de

eSBR) procedentes del generador 106 para generar el flujo de bits codificado a ser emitido por la etapa 107, preferentemente de tal manera que el flujo de bits codificado tenga el formato especificado por una de las formas de realización de la presente invención.

La memoria intermedia 109 está configurada para almacenar (por ej., de manera no transitoria) por lo menos un bloque del flujo de bits de audio codificado emitido como salida de la etapa 107, y una secuencia de los bloques del flujo de bits de audio codificado es emitida luego de la memoria intermedia 109 como salida del codificador 100 a un sistema de transporte.

La FIG. 3 es un diagrama de bloques de un sistema que incluye un decodificador (200) que constituye una forma de realización de la unidad de procesamiento de audio de la invención, y opcionalmente también un post-procesador (300) acoplado al mismo. Cualquiera de los componentes o elementos del decodificador 200 y el post-procesador 300 puede ser implementado en forma de uno o más procesos y/o uno o más circuitos (por ej., ASICs, FPGAs, u otros circuitos integrados) en hardware, software, o una combinación de hardware y software. El decodificador 200 comprende la memoria intermedia 201, el desformateador (analizador) de carga útil del flujo de bits 205, el subsistema decodificador de audio 202 (al que en ocasiones se hace referencia como etapa de decodificación del “núcleo” o subsistema de decodificación del “núcleo”), la etapa de procesamiento de eSBR 203, y la etapa de generación de bits de control 204, conectadas de la manera expuesta. También en general, el decodificador 200 incluye otros elementos de procesamiento (no se ilustra).

La memoria intermedia (“buffer”) 201 almacena (por ej., de manera no transitoria) por lo menos un bloque de un flujo de bits de audio MPEG-4 AAC

codificado recibido por el decodificador 200. En funcionamiento del decodificador 200, una secuencia de los bloques del flujo de bits es enviado de la memoria intermedia 201 al desformateador 205.

En variaciones de la forma de realización de la Fig. 3 (o de la forma de realización de la Fig. 4 que se describe a continuacióna continuación), una APU que no es un decodificador (por ej., APU 500 de la Fig. 6) incluye una memoria intermedia (por ej., una memoria intermedia idéntica a la memoria intermedia 201) que almacena (por ej., de manera no transitoria) por lo menos un bloque de un flujo de bits de audio codificado (por ej., un flujo de bits de audio MPEG-4 AAC) del mismo tipo recibido por la memoria intermedia 201 de la Fig. 3 o de la Fig. 4 (es decir, un flujo de bits de audio codificado que incluye metadatos de eSBR).

Con referencia, una vez más, a la Fig. 3, el desformateador 205 está acoplado y configurado para demultiplexar cada bloque del flujo de bits para extraer metadatos de eSBR (que incluye datos de envolvente cuantificados) y metadatos de SBR (y por lo general también otros metadatos) therefrom, to assert por lo menos los metadatos de SBR y los metadatos de eSBR a la etapa de procesamiento de eSBR 203, y también típicamente para enviar otros metadatos extraídos al subsistema de decodificación 202 (y opcionalmente también al generador de bits de control 204). El desformateador 205 también está acoplado y configurado para extraer datos de audio de cada bloque del flujo de bits, y para enviar los datos de audio extraídos al subsistema de decodificación (etapa de decodificación) 202.

El sistema de la Fig. 3 incluye además, opcionalmente, el post-procesador 300. El post-procesador 300 incluye la memoria intermedia (buffer) 301 y otros elementos de procesamiento (no se ilustran) que incluyen por lo menos un elemento de procesamiento acoplado a la memoria intermedia 301. La memoria intermedia 301

almacena (por ej., de manera no transitoria) por lo menos un bloque (o trama) de los datos de audio decodificados recibidos por el post-procesador 300 del decodificador 200. Los elementos de procesamiento del post-procesador 300 están acoplados y configurados para recibir y procesar de manera adaptativa una secuencia de los bloques (o tramas) del audio decodificado emitido por el buffer 301, utilizando metadatos emitidos como salida del subsistema de decodificación 202 (y/o el desformateador 205) y/o bits de control emitidos como salida de la etapa 204 del decodificador 200.

El subsistema decodificador de audio 202 del decodificador 200 está configurado para decodificar los datos de audio extraídos por el analizador 205 (se puede hacer referencia a esa decodificación como operación de decodificación del “núcleo”) para generar datos de audio decodificados, y para enviar los datos de audio decodificados a la etapa de procesamiento de eSBR 203. La decodificación se ejecuta en el dominio de la frecuencia y por lo general incluye la cuantificación inversa seguida por el procesamiento espectral. Por lo general, una etapa final de procesamiento en el subsistema 202 aplica una transformada del dominio de la frecuencia al dominio del tiempo a los datos de audio decodificados en el dominio de la frecuencia, por lo que la salida del subsistema son datos de audio decodificados en el dominio del tiempo. La etapa 203 está configurada para aplicar herramientas de SBR y herramientas de eSBR indicadas por los metadatos de SBR y de eSBR (extraídos por el analizador 205) a los datos de audio decodificados (es decir, para ejecutar el procesamiento de SBR y eSBR de la salida del subsistema de decodificación 202 usando los metadatos de SBR y eSBR) para generar los datos de audio completamente decodificados que constituyen la salida (por ej., al post-procesador 300) del decodificador 200. Por lo general, el decodificador 200 incluye

una memoria (a la que puede acceder el subsistema 202 y la etapa 203) que almacena los datos de audio desformateados y los metadatos emitidos como salida del desformateador 205, y la etapa 203 está configurada para acceder a los datos y metadatos de audio (incluyendo los metadatos de SBR y metadatos de eSBR) que se necesiten durante el procesamiento de SBR y eSBR. El procesamiento de SBR y el procesamiento de eSBR realizados en la etapa 203 se pueden considerar como post-procesamiento de la salida del subsistema de decodificación del núcleo 202. Opcionalmente, el decodificador 200 incluye asimismo un subsistema de mezcla ascendente final (que puede aplicar herramientas de estéreo paramétrico (“PS”, por sus siglas en inglés Parametric Stereo) definidas en la norma de MPEG–4 AAC, usando los metadatos de PS extraídos por el desformateador 205 y/o bits de control generados en el subsistema 204) que está acoplado y configurado to ejecutar la mezcla ascendente de la salida de la etapa 203 para generar el audio con mezcla ascendente completamente decodificado que es la salida del decodificador 200. Por otro lado, el post-procesador 300 está configurado para ejecutar la mezcla ascendente en la salida del decodificador 200 (por ej., empleando metadatos de PS extraídos por el desformateador 205 y/o bits de control generados in el subsistema 204).

En respuesta a los metadatos extraídos por el desformateador 205, el generador de bits de control 204 puede generar datos de control, y se pueden usar los datos de control dentro del decodificador 200 (por ej., en un subsistema de mezcla ascendente final) y/o enviados como salida del decodificador 200 (por ej., al post-procesador 300 para usar en el post-procesamiento). En respuesta a metadatos extraídos de el flujo de bits de entrada (y opcionalmente también en respuesta a datos de control), la etapa 204 puede generar (y enviar al post-procesador 300) bits

de control que indican que los datos de audio decodificados emitidos como salida de la etapa de procesamiento de eSBR 203 debe ser objeto de un tipo específico de post-procesamiento. En algunas implementaciones, el decodificador 200 está configurado para enviar los metadatos extraídos por el desformateador 205 del flujo de bits de entrada al post-procesador 300, y el post-procesador 300 está configurado para ejecutar el post procesamiento de los datos de audio decodificados emitidos como salida del decodificador 200 usando los metadatos.

La FIG. 4 es un diagrama de bloques de una unidad de procesamiento de audio (“APU”) (210) que constituye otra forma de realización de la unidad de procesamiento de audio de la invención. APU 210 es un decodificador de la técnica anterior que no está configurado para ejecutar el procesamiento de eSBR. Cualquiera de los componentes o elementos de APU 210 puede ser implementado en forma de uno o más procesos y/o uno o más circuitos (por ej., ASICs, FPGAs, u otros circuitos integrados), en hardware, software, o una combinación de hardware y software. APU 210 comprende la memoria intermedia 201, el desformateador (analizador) de la carga útil del flujo de bits 215, el subsistema decodificador de audio 202 (al que en ocasiones se hace referencia como etapa de decodificación del “núcleo” o subsistema de decodificación del “núcleo”), y la etapa de procesamiento de SBR 213, conectadas según lo expuesto. Típicamente además, la APU 210 incluye otros elementos de procesamiento (no se ilustran). La APU 210 puede representar, por ejemplo, un codificador, decodificador o transcodificador de audio.

Los elementos 201 y 202 de la APU 210 son idénticos a los elementos identificados con números idénticos del decodificador 200 (de la Fig. 3) y no se repite la descripción de los mismos. En operación de la APU 210, una secuencia de bloques de un flujo de bits de audio codificado (un flujo de bits MPEG–4 AAC)

recibido por la APU 210 es enviado de la memoria intermedia o buffer 201 al desformateador 215.

El desformateador 215 está acoplado y configurado para demultiplexar cada bloque del flujo de bits a fin de extraer metadatos de SBR (incluyendo datos de envolvente cuantificados) y típicamente también otros metadatos del mismo, aunque para ignorar los metadatos de eSBR que puedan estar incluidos en el flujo de bits de conformidad con cualquier forma de realización de la presente invención. El desformateador 215 está configurado para enviar por lo menos los metadatos de SBR a la etapa de procesamiento de SBR 213. El desformateador 215 también está acoplado y configurado para extraer datos de audio de cada bloque del flujo de bits, y para enviar los datos de audio extraídos al subsistema de decodificación (etapa de decodificación) 202.

El subsistema decodificador de audio 202 del decodificador 200 está configurado para decodificar los datos de audio extraídos por el desformateador 215 (decodificación a la que se puede hacer referencia como operación de decodificación del “núcleo”) para generar datos de audio decodificados, y para enviar los datos de audio decodificados a la etapa de procesamiento de SBR 213. La decodificación se ejecuta en el dominio frecuencial. Por lo general, una etapa final de procesamiento en el subsistema 202 aplica una transformación del dominio frecuencial al dominio temporal a los datos de audio decodificados en el dominio frecuencial, de manera que la salida del subsistema sean datos de audio decodificados en el dominio temporal. La etapa 213 está configurada para aplicar herramientas de SBR (no así herramientas de eSBR) indicadas por los metadatos de SBR (extraídos por el desformateador 215) a los datos de audio decodificados (es decir, para llevar a cabo el procesamiento de SBR en la salida del subsistema de decodificación 202 usando

los metadatos de SBR) para generar los datos de audio completamente decodificados que constituyen la salida (por ej., al post-procesador 300) de la APU 210. Por lo general, la APU 210 incluye una memoria (a la que pueden acceder el subsistema 202 y la etapa 213) que almacena los datos de audio y metadatos desformateados emitidos como salida del desformateador 215, y la etapa 213 está configurada para acceder a los datos de audio y metadatos (incluyendo metadatos de SBR) según lo necesario durante el procesamiento de SBR. Se puede considerar que el procesamiento de SBR de la etapa 213 es el post procesamiento de la salida del subsistema de decodificación del núcleo 202. Opcionalmente, la APU 210 incluye asimismo un subsistema de mezcla ascendente final (que puede aplicar herramientas de estéreo paramétrico (“PS”) definidas en la norma de MPEG–4 AAC, empleando metadatos de PS extraídos por el desformateador 215) que está acoplado y configurado para ejecutar la mezcla ascendente de la salida de la etapa 213 para generar audio con mezcla ascendente totalmente decodificado que es emitido como salida de la APU 210. Por otro lado, un post-procesador está configurado para ejecutar la mezcla ascendente en la salida de la APU 210 (por ej., empleando los metadatos de PS extraídos por el desformateador 215 y/o bits de control generados en la APU 210).

Diversas implementaciones del codificador 100, el decodificador 200 y la APU 210 están configuradas para ejecutar diferentes formas de realización del método de la invención.

De conformidad con algunas formas de realización, se incluyen metadatos de eSBR (por ej., se incluye un pequeño número de bits de control que son metadatos de eSBR) en un flujo de bits de audio codificado (por ej., un flujo de bits MPEG–4 AAC), de tal manera que los decodificadores anteriores (que no están configurados

para analizar los metadatos de eSBR, ni para usar ninguna herramienta de eSBR a la cual pertenecen los metadatos de eSBR) pueda ignorar los metadatos de eSBR pero decodificar, de todas maneras, el flujo de bits en la medida de lo posible sin usar los metadatos de eSBR ni herramienta alguna de eSBR con la cual se relacionan los metadatos de eSBR, típicamente sin penealidad de importancia en la calidad del audio. Sin embargo, los decodificadores de eSBR configurados para analizar el flujo de bits a fin de identificar los metadatos de eSBR y para usar por lo menos una herramienta de eSBR en respuesta a los metadatos de eSBR, se benefician con el uso de por lo menos una de dichas herramientas de eSBR. Por lo tanto, las formas de realización de la invención ofrecen un medio para transmitir con eficiencia los datos o metadatos de control de replicación de la banda espectral mejorada (eSBR) en forma compatible con las versiones anteriores.

Por lo general, los metadatos de eSBR en el flujo de bits es indicativo (por ej., es indicativo de por lo menos una característica o parámetro) de una o más de las siguientes herramientas de eSBR (que han sido descriptas en la norma de MPEG USAC, y que pueden o no haber sido aplicadas por un codificador durante la generación del flujo de bits):

- Transposición armónica y
- Procesamiento previo adicional de emparchado por QMF (pre-aplanamiento).

Por ejemplo, los metadatos de eSBR incluidos en el flujo de bits pueden ser indicativos de valores de los parámetros (descriptos en la norma de MPEG USAC y en la presente revelación): sbrPatchingMode[ch], sbrOversamplingFlag[ch], sbrPitchInBins[ch], sbrPitchInBins[ch], y bs_sbr_preprocessing.

En la presente, la notación X[ch], donde X es algún parámetro, denota que el parámetro se relaciona con el canal (“ch”) de contenido de audio de un flujo de bits codificado que se ha de decodificar. Para simplificar, en ocasiones omitimos la expresión [ch], y presumimos que el parámetro relevante se relaciona con un canal de contenido de audio.

En la presente, la notación X[ch][env], donde X es algún parámetro, denota que el parámetro se relaciona con la envolvente de SBR (“env”) del canal (“ch”) de contenido de audio de un flujo de bits codificado que se ha de decodificar. Para simplificar, en ocasiones omitimos las expresiones [env] y [ch], y presumimos que el parámetro relevante se relaciona con una envolvente de SBR de un canal de contenido de audio.

Durante la decodificación de un flujo de bits codificado, se controla la eficiencia de la transposición armónica durante una etapa de procesamiento de eSBR de la decodificación (por cada canal, “ch”, de contenido de audio indicado por el flujo de bits) mediante los siguientes parámetros de metadatos de eSBR: sbrPatchingMode[ch]: sbrOversamplingFlag[ch]; sbrPitchInBinsFlag[ch]; y sbrPitchInBins[ch].

El valor “sbrPatchingMode[ch]” indica el tipo de transpositor empleado en eSBR: sbrPatchingMode[ch] = 1 indica emparchado no armónico de acuerdo con lo descripto en la Sección 4.6.18.6.3 de la norma MPEG–4 AAC; sbrPatchingMode[ch] = 0 indica emparchado armónico de SBR de acuerdo con lo descripto en la Sección 7.5.3 o 7.5.4 de la norma MPEG USAC.

El valor “sbrOversamplingFlag[ch]” indica el uso de un sobremuestreo adaptativo de señales en el dominio de la frecuencia en eSBR en combinación con el

emparchado armónico de SBR basado en DFT (del inglés, Discrete Fourier Transform, Transformada de Fourier Discreta) descripto en la Sección 7.5.3 de la norma MPEG USAC. Esta bandera controla el tamaño de las DFT que se utilizan en el transpositor: 1 indican que el sobremuestreo adaptativo de señales en el dominio de la frecuencia está habilitado según lo descripto en la Sección 7.5.3.1 de la norma de MPEG USAC; 0 indica que el sobremuestreo adaptativo de señales en el dominio de la frecuencia está deshabilitado según lo descripto en la Sección 7.5.3.1 de la norma de MPEG USAC.

El valor “sbrPitchInBinsFlag[ch]” controla la interpretación del parámetro sbrPitchInBins[ch]: 1 indica que el valor de sbrPitchInBins[ch] es válido y mayor que cero; 0 indica que el valor de sbrPitchInBins[ch] está establecido en cero.

El valor “sbrPitchInBins[ch]” controla la suma de términos de productos cruzados en el transpositor armónico de SBR. El valor sbrPitchinBins[ch] es un valor entero en el rango [0,127] y representa la distancia medida en bins de frecuencia correspondientes a una DFT de 1536 líneas que actúa sobre la frecuencia de muestreo del codificador del núcleo.

En caso de que un flujo de bits MPEG–4 AAC sea indicativo de un par de canales de SBR cuyos canales no están acoplados (en lugar de un solo canal de SBR), el flujo de bits es indicativo de dos casos de la sintaxis antes expuesta (correspondientes a la transposición armónica o no armónica), uno por cada canal del sbr_channel_pair_element().

La transposición armónica de la herramienta de eSBR típicamente mejora la calidad de las señales musicales decodificadas a frecuencias de cruce relativamente bajas. La transposición no armónica (es decir, el antiguo emparchado espectral)

típicamente mejora las señales de voz. Por ende, un punto de partida en la decisión de qué tipo de transposición es preferible para codificar un contenido de audio específico es la selección del método de transposición en función de la detección de voz/música con el empleo de la transposición armónica en el contenido musical y el emparchado espectral en el contenido de voz.

La eficiencia del aplanamiento previo durante el procesamiento de eSBR se controla por el valor de un parámetro de metadatos de eSBR de un bit conocido como “bs_sbr_preprocessing”, en el sentido de que aplanamiento previo es ejecutado o no dependiendo del valor de este único bit. Cuando se utiliza el algoritmo de emparchado de SBR QMF, descripto en la Sección 4.6.18.6.3 de la norma de MPEG–4 AAC, se puede realizar el paso de aplanamiento previo (cuando así lo indica el parámetro “bs_sbr_preprocessing”) en un esfuerzo por evitar las discontinuidades en la forma de la envolvente espectral de una señal de alta frecuencia que se provee como entrada al ajustador de envolvente subsiguiente (el ajustador de la envolvente ejecuta otra etapa del procesamiento de eSBR). El aplanamiento previo mejora por lo general el funcionamiento de la siguiente etapa de ajuste de la envolvente, dando lugar a una señal de banda alta que se percibe como más estable.

Se estima que el requerimiento total de tasa de bits para incluir en un flujo de bits MPEG–4 AAC metadatos de eSBR indicativos de las herramientas de eSBR antes citada (transposición armónica y aplanamiento previo) está en el orden de unos cuantos cientos de bits por segundo, puesto que sólo se transmiten los datos de control diferenciales necesarios para ejecutar el procesamiento de eSBR de conformidad con algunas formas de realización de la invención. Los decodificadores de la técnica anterior pueden ignorar esta información porque está incluida en la

manera compatible con versiones anteriores (como se explica más adelante). Por lo tanto, el efecto deletéreo sobre la tasa de bits asociada a la inclusión de metadatos de eSBR es ínfimo, por un número de razones, entre las que se cuentan las siguientes:

- La penalidad en la tasa de bits (debida a la inclusión de los metadatos de eSBR) es una fracción muy pequeña de la tasa de bits total porque sólo se transmiten los datos de control diferenciales necesarios para ejecutar el procesamiento de eSBR (y no una transmisión simultánea de los datos de control de SBR); y
- La refinación de la información de control relacionada con la SBR por lo general no depende de los detalles de la transposición.

Por consiguiente, las formas de realización de la invención presentan un medio para transmitir eficientemente los datos de control o metadatos de replicación de la banda espectral mejorada (eSBR) de manera compatible con versiones anteriores. Esta transmisión eficiente de los datos de control de eSBR reduce los requerimientos de memoria de los decodificadores, codificadores, y transcodificadores que emplean aspectos de la invención, sin tener, en absoluto, efectos adversos tangibles sobre la tasa de bits. Más aun, también se reduce la complejidad y los requerimientos de procesamiento asociados a la realización de la eSBR de conformidad con las formas de realización de la invención, puesto que sólo se deben procesar los datos de SBR una sola vez y no en transmisión simultánea, que sería el caso si se tratara la eSBR como un tipo de objeto completamente

separado en MPEG-4 AAC en lugar de estar integrado al códec de MPEG-4 AAC de manera compatible con versiones anteriores.

Seguidamente, con referencia a la FIG. 7, describimos los elementos de un bloque (“raw_data_block”) de un flujo de bits MPEG–4 AAC en el cual se incluyen metadatos de eSBR de conformidad con algunas formas de realización de la presente invención. La FIG. 7 es un diagrama de un bloque (un “raw_data_block”) del flujo de bits MPEG–4 AAC, que expone algunos de los segmentos del mismo.

Un bloque de un flujo de bits MPEG–4 AAC puede incluir por lo menos un “single_channel_element()” (por ej., el elemento de canal único expuesto en la Fig. 7), y/o por lo menos un “channel_pair_element()” (que no se ilustra específicamente en la Fig. 7 aunque puede estar presente), que incluye datos de audio correspondientes a un programa de audio. El bloque puede incluir además un número de “elementos de relleno” (por ej., el elemento de relleno y/o el elemento de relleno 2 de la Fig. 7) que incluyen datos (por ej., metadatos) relacionados con el programa. Cada “single_channel_element()” incluye un identificador (por ej., “ID1” de la Fig. 7) que indica el comienzo de un elemento de un canal único, y puede incluir datos de audio indicativos de un canal diferente de un programa de audio multicanal. Cada “channel_pair_element incluye un identificador (no expuesto en la Fig. 7) que indica el comienzo de un elemento de par de canales, y puede incluir datos de audio indicativos de dos canales del programa.

Un fill_element (al que en la presente se hace referencia como elemento de relleno) de un flujo de bits MPEG–4 AAC incluye un identificador (“ID2” de la Fig. 7) que indica el comienzo de un elemento de relleno, y datos de relleno después del identificador. El identificador ID2 puede consistir en un entero sin signo de tres bits

que se transmite con el bit más significativo primero (“uimsbf”) que tiene un valor de 0x6. Los datos de relleno pueden incluir un elemento extension_payload()(al que en ocasiones se hace referencia en la presente como carga útil de extensión) cuya sintaxis está consignada en la Tabla 4.57 de la norma de MPEG–4 AAC. Existen varios tipos de cargas útiles de extensión y se los identifica por medio del parámetro “extension_type”, que es un entero sin signo de cuatro bits que se transmite con el bit más significativo primero (“uimsbf”).

Los datos de relleno (por ej., una carga útil de extensión de los mismos) pueden incluir una cabecera o identificador (por ej., “header1” de la Fig. 7) que indica un segmento de datos de relleno que es indicativo de un objeto de SBR (es decir, la cabecera inicializa un tipo de “objeto de SBR”, al que se hace referencia como sbr_extension_data() en la norma de MPEG–4 AAC). Por ejemplo, se identifica una carga útil de extensión de replicación de la banda espectral (SBR) con el valor de ‘1101’ o ‘1110’ correspondiente al campo extension_type de la cabecera, donde el identificador ‘1101’ identifica una carga útil de extensión con datos de SBR y ‘1110’ identifica una carga útil de extensión con datos de SBR con una Verificación de Redundancia Cíclica (CRC, por sus siglas en inglés, Cyclic Redundancy Check) para verificar la exactitud de los datos de SBR.

Cuando la cabecera (por ej., el campo extension_type) inicializa un tipo de objeto de SBR, los metadatos de SBR (a los que en ocasiones se hace referencia en la presente como “datos de replicación de la banda espectral,” y los que se mencionan como sbr_data() en la norma de MPEG–4 AAC) siguen tras la cabecera, y por lo menos un elemento de extensión de la replicación de la banda espectral (por ej., el “Elemento de extensión de SBR” del elemento de relleno 1 de la Fig. 7) puede seguir a los metadatos de SBR. Se hace referencia a ese elemento de extensión de

la replicación de la banda espectral (un segmento del flujo de bits) como contenedor “sbr_extension()” en la norma de MPEG–4 AAC. Un elemento de extensión de la replicación de la banda espectral incluye opcionalmente una cabecera (por ej., “Cabecera de extensión de SBR” del elemento de relleno 1 de la Fig. 7).

La norma de MPEG–4 AAC contempla que un elemento de extensión de la replicación de la banda espectral pueda incluir datos de PS (estéreo paramétrico) correspondientes a los datos de audio de un programa. La norma de MPEG–4 AAC contempla que cuando la cabecera de un elemento de relleno (por ej., de una carga útil de extensión del mismo) inicializa un tipo de objeto de SBR (como en el caso de “header1” de la Fig. 7) y un elemento de extensión de la replicación de la banda espectral del elemento de relleno incluye datos de PS, el elemento de relleno (por ej., la carga útil de extensión del mismo) incluye datos de replicación de la banda espectral, y un parámetro “bs_extension_id” cuyo valor (es decir, bs_extension_id = 2) indica que hay datos de PS incluidos en un elemento de extensión de la replicación de la banda espectral del elemento de relleno.

De conformidad con algunas formas de realización de la presente invención, se incluyen metadatos de eSBR (por ej., una bandera que indica si se ha de ejecutar un procesamiento de replicación de la banda espectral mejorada (eSBR) en el contenido de audio del bloque) en un elemento de extensión de la replicación de la banda espectral de un elemento de relleno. Por ejemplo, ese tipo de bandera está indicada en el elemento de relleno 1 de la Fig. 7, donde la bandera aparece después de la cabecera (la “cabecera de extensión de SBR” del elemento de relleno 1) del “elemento de extensión de SBR” del elemento de relleno 1. Opcionalmente, esa bandera y los metadatos de eSBR adicionales están incluidos en un elemento de extensión de la replicación de la banda espectral después de la cabecera del

elemento de extensión de replicación de la banda espectral (por ej., en el elemento de extensión de SBR del elemento de relleno 1 de la Fig. 7, después de la cabecera de extensión de SBR). De conformidad con algunas formas de realización de la presente invención, un elemento de relleno que incluye metadatos de eSBR también incluye un parámetro “bs_extension_id” cuyo valor (por ej., bs_extension_id = 3) indica que se incluyen metadatos de eSBR en el elemento de relleno y que se ha de ejecutar el procesamiento de eSBR en el contenido de audio del bloque pertinente.

De conformidad con algunas formas de realización de la invención, se incluyen metadatos de eSBR en un elemento de relleno (por ej., el elemento de relleno 2 de la Fig. 7) de un flujo de bits MPEG–4 AAC que no es un elemento de extensión de la replicación de la banda espectral (elemento de extensión de SBR) del elemento de relleno. Esto es así porque los elementos de relleno que contienen una extension_payload() con datos de SBR o datos de SBR con una CRC no contienen ninguna otra carga útil de extensión de ningún otro tipo de extensión. Por lo tanto, en las formas de realización en que se almacenan metadatos de eSBR en su propia carga útil de extensión, se utiliza un elemento de relleno separado para almacenar los metadatos de eSBR. Ese tipo de elemento de relleno incluye un identificador (por ej., “ID2” de la Fig. 7) que indica el comienzo de un elemento de relleno, y datos de relleno después del identificador. Los datos de relleno pueden incluir un elemento extension_payload()(al que en ocasiones se hace referencia en la presente como carga útil de extensión) cuya sintaxis está expuesta en la Tabla 4.57 de la norma de MPEG–4 AAC. Los datos de relleno (por ej., una carga útil de extensión de los mismos) incluye una cabecera (por ej., “header2” del elemento de relleno 2 de la Fig. 7) que es indicativa de un objeto de eSBR (es decir, la cabecera inicializa un tipo de objeto de replicación de la banda espectral mejorada (eSBR)), y los datos de relleno

(por ej., una carga útil de extensión de los mismos) incluyen metadatos de eSBR después de la cabecera. Por ejemplo, el elemento de relleno 2 de la Fig. 7 incluye ese tipo de cabecera (“header2”) y también incluye, después de la cabecera, metadatos de eSBR (es decir, la “bandera” del elemento de relleno 2, que es indicativa de si se ha de ejecutar procesamiento de replicación de la banda espectral mejorada (eSBR) en el contenido de audio del bloque). Opcionalmente, también se incluyen metadatos de eSBR adicionales en los datos de relleno del elemento de relleno 2 de la Fig. 7, después de header2. En las formas de realización que se describen en el presente párrafo, la cabecera (por ej., header2 de la Fig. 7) tiene un valor de identificación que no es el de los valores convencionales estipulados en la Tabla 4.57 de la norma de MPEG–4 AAC y, por el contrario, es indicativa de una carga útil de extensión de eSBR (por lo que el campo extension_type de la cabecera indica que los datos de relleno incluyen metadatos de eSBR).

En una primera clase de formas de realización, la invención es una unidad de procesamiento de audio (por ej., un decodificador), que comprende:

una memoria (por ej., la memoria intermedia o “buffer” 201 de la Fig. 3 o 4) configurada para almacenar store por lo menos un bloque de un flujo de bits de audio codificado (por ej., por lo menos un bloque de un flujo de bits MPEG–4 AAC);

un desformateador de carga útil del flujo de bits (por ej., el elemento 205 de la Fig. 3 o el elemento 215 de la Fig. 4) acoplado a la memoria y configurado para demultiplexar por lo menos una porción de dicho bloque del flujo de bits y

un subsistema de decodificación (por ej., los elementos 202 y 203 de la Fig. 3, o los elementos 202 y 213 de la Fig. 4), acoplado y configurado para decodificar por lo

menos una porción de contenido de audio de dicho bloque del flujo de bits, donde el bloque incluye:

un elemento de relleno, que incluye un identificador que indica el comienzo del elemento de relleno (por ej., el identificador “id_syn_ele” que tiene un valor 0x6, de la Tabla 4.85 de la norma de MPEG–4 AAC), y datos de relleno después del identificador, donde los datos de relleno incluyen:

por lo menos una bandera que identifica si se ha de ejecutar el procesamiento de replicación de la banda espectral mejorada (eSBR) en el contenido de audio del bloque (por ej., usando datos de replicación de la banda espectral y metadatos de eSBR incluidos en el bloque).

La bandera es metadatos de eSBR, y un ejemplo de la bandera es la bandera sbrPatchingMode. Otro ejemplo de bandera es la bandera harmonicSBR. Estas dos banderas indican si se ha de ejecutar la forma básica de replicación de la banda espectral o una forma mejorada de replicación de la banda espectral en los datos de audio del bloque. La forma básica de replicación de la banda espectrales el emparchado espectral, y la forma mejorada de replicación de la banda espectral es la transposición armónica.

En algunas formas de realización, los datos de relleno incluyen asimismo metadatos de eSBR adicionales (es decir, metadatos de eSBR aparte de la bandera).

La memoria puede ser una memoria intermedia (por ej., una implementación de la memoria intermedia 201 de la Fig. 4) que almacena (por ej., de manera no transitoria) dicho por lo menos un bloque del flujo de bits de audio codificado.

Se estima que la complejidad de la ejecución del procesamiento de eSBR (utilizando la transposición armónica de eSBR y aplanamiento previo) mediante un decodificador de eSBR durante la decodificación de un flujo de bits MPEG–4 AAC que incluye metadatos de eSBR (indicativos de estas herramientas de eSBR) sería la siguiente (en el caso de la decodificación típica con los parámetros indicados):

- Transposición armónica (16 kbps, 14400/28800 Hz)
    - basada en DFT: 3.68 WMOPS (del inglés, weighted million operations per second: millones de operaciones por segundo ponderadas);
    - basada en QMF: 0.98 WMOPS;
- Pre–procesamiento de emparchado por QMF (aplanamiento previo): 0.1WMOPS.

Se sabe que la transposición basada en DFT por lo general da mejor resultado que la transposición basada en QMF en el caso de los transitorios.

De conformidad con algunas formas de realización de la presente invención, un elemento de relleno (de un flujo de bits de audio codificado) que incluye metadatos de eSBR incluye asimismo un parámetro (por ej., un parámetro “bs_extension_id”) cuyo valor (por ej., bs_extension_id = 3) señala que hay metadatos de eSBR incluidos en el elemento de relleno y que se debe ejecutar el procesamiento de eSBR en el contenido de audio del bloque pertinente, y/o un parámetro (por ej., el mismo parámetro “bs_extension_id”) cuyo valor (por ej., bs_extension_id = 2) señala que el contenedor sbr_extension() del elemento de relleno incluye datos de PS. Por ejemplo, como se indica en la siguiente Tabla 1, ese parámetro con el valor bs_extension_id = 2 puede indicar que un contenedor

sbr_extension() del elemento de relleno incluye datos de PS, y ese parámetro con el valor bs_extension_id = 3 puede indicar que un contenedor sbr_extension() del elemento de relleno incluye metadatos de eSBR:

Tabla 1

bs_extension_id	**Significado**
0	Reservado
1	Reservado
2	EXTENSION_ID_PS
3	EXTENSION_ID_ESBR

De conformidad con algunas formas de realización de la invención, la sintaxis de cada elemento de extensión de la replicación de la banda espectral que incluye metadatos de SBR y/o datos de PS está indicada en la siguiente Tabla 2 (en la cual “sbr_extension()” denota un contenedor que es el elemento de extensión de la replicación de la banda espectral, “bs_extension_id” es como se describe en la Tabla 1 anterior, “ps_data” denota datos de PS, y “esbr_data” denote metadatos de eSBR):

Tabla 2

```
sbr_extension(bs_extension_id, num_bits_left)
{
```

```
        switch (bs_extension_id) {
        case EXTENSION_ID_PS:
                num_bits_left -= ps_data();                    Nota 1
                break;
        case EXTENSION_ID_ESBR:
                num_bits_left -= esbr_data();                  Nota 2
                break;
        default:
                bs_fill_bits;
                num_bits_left = 0;
        break;
}
}
```

Nota 1: ps_data() retorna al número de bits leído.

Note 2: esbr_data() retorna al número de bits leído.

En una forma de realización ilustrativa, los esbr_data() a los que se hace referencia en la anterior Tabla 2 son indicativos de valores de los siguientes parámetros de metadatos:

1. un parámetro de metadatos de un bit, “bs_sbr_preprocessing” y

2. por cada canal (“ch”) de contenido de audio del flujo de bits codificado que se ha de decodificar, cada uno de los parámetros antes descriptos:

“sbrPatchingMode[ch]”; “sbrOversamplingFlag[ch]”; “sbrPitchInBinsFlag[ch]” y “sbrPitchInBins[ch]”.

Por ejemplo, en algunas formas de realización, el esbr_data() puede tener la sintaxis indicada en la Tabla 3, que indica estos parámetros de metadatos:

Tabla 3

Sintaxis	No. de bits
esbr_data(id_aac, bs_coupling)	
{	
bs_sbr_preprocessing;	**1**
if (id_aac == ID_SCE) {	
if (**sbrPatchingMode[0]** == 0) {	**1**
sbrOversamplingFlag[0];	**1**
if (**sbrPitchInBinsFlag[0]**)	**1**
sbrPitchInBins[0];	**7**
else	
sbrPitchInBins[0] = 0;	
} else {	
sbrOversamplingFlag[0] = 0;	
sbrPitchInBins[0] = 0;	
}	

} else if (id_aac == ID_CPE) {	
If (bs_coupling) {	
if (**sbrPatchingMode[0,1]** == 0) {	**1**
sbrOversamplingFlag[0,1];	**1**
if (**sbrPitchInBinsFlag[0,1]**)	**1**
sbrPitchInBins[0,1];	**7**
else	
sbrPitchInBins[0,1] = 0;	
} else {	
sbrOversamplingFlag[0,1] = 0;	
sbrPitchInBins[0,1] = 0;	
}	
} else { /* bs_coupling == 0 */	
if (**sbrPatchingMode[0]** == 0) {	**1**
sbrOversamplingFlag[0];	**1**
if (**sbrPitchInBinsFlag[0]**)	**1**
sbrPitchInBins[0];	**7**
else	
sbrPitchInBins[0] = 0;	
} else {	
sbrOversamplingFlag[0] = 0;	

sbrPitchInBins[0] = 0;	
}	
if (**sbrPatchingMode[1]** == 0) {	**1**
sbrOversamplingFlag[1];	**1**
if (**sbrPitchInBinsFlag[1]**)	**1**
sbrPitchInBins[1];	**7**
else	
sbrPitchInBins[1] = 0;	
} else {	
sbrOversamplingFlag[1] = 0;	
sbrPitchInBins[1] = 0;	
}	
}	
}	
}	
Nota: *bs_sbr_preprocessing* se define como se describe en la Sección 6.2.12 de ISO/IEC 23003-3:2012. *sbrPatchingMode[ch], sbrOversamplingFlag[ch], sbrPitchInBinsFlag[ch]* y *sbrPitchInBins[ch]* se define de acuerdo con lo descripto en la Sección 7.5 de ISO/IEC 23003-3:2012.	

La sintaxis expuesta permite una eficiente implementación de una forma mejorada de replicación de la banda espectral, tal como la transposición armónica, como extensión de un decodificador de la técnica anterior. Específicamente, los

datos de eSBR de la Tabla 3 incluyen sólo los parámetros necesarios para ejecutar la forma mejorada de replicación de la banda espectral que no estén ya admitidos en el flujo de bits o directamente derivables de los parámetros ya admitidos en el flujo de bits. Todos los demás parámetros y datos de procesamiento necesarios para ejecutar la forma mejorada de replicación de la banda espectral son extraídos de parámetros preexistentes en ubicaciones ya definidas en el flujo de bits.

Por ejemplo, se puede extender un decodificador que cumple con MPEG-4 HE-AAC o HE-AAC v2 para incluir una forma mejorada de replicación de la banda espectral, tal como la transposición armónica. Esta forma mejorada de replicación de la banda espectral se suma a la forma básica de replicación de la banda espectral ya admitida por el decodificador. En el contexto de un decodificador de conformidad con MPEG-4 HE-AAC o HE-AAC v2, esta forma básica de replicación de la banda espectral es la herramienta de SBR de emparchado espectral por QMF definida en la Sección 4.6.18 de la norma de MPEG–4 AAC.

Al ejecutar la forma mejorada de replicación de la banda espectral, un decodificador HE-AAC extendido puede reutilizar muchos parámetros del flujo de bits ya incluidos en la carga útil de extensión de SBR del flujo de bits. Los parámetros específicos que se pueden reutilizar incluyen, por ejemplo, los diversos parámetros que determinan la tabla maestra de bandas de frecuencia. Estos parámetros incluyen bs_start_freq (parámetro que determina el comienzo del parámetro de tabla maestra de frecuencias), bs_stop_freq (parámetro que determina la interrupción de la tabla maestra de frecuencias), bs_freq_scale (parámetro que determina el número de bandas de frecuencia por octava) y bs_alter_scale (parámetro que modifica la escala de las bandas de frecuencia). Los parámetros que se pueden reutilizar incluyen asimismo parámetros que determinan la tabla de bandas de ruido (bs_noise_bands)

y los parámetros limitantes de la tabla de bandas (bs_limiter_bands). En consecuencia, en diversas formas de realización, se omiten por lo menos algunos de los parámetros equivalentes especificados en la norma de USAC en el flujo de bits, reduciendo de esa manera la sobrecarga de control en el flujo de bits. Por lo general, cuando en la norma de AAC hay un parámetro especificado equivalente a un parámetro especificado en la norma de USAC, el parámetro equivalente especificado en la norma de USAC tiene el mismo nombre que el parámetro especificado en la norma de AAC, por ej. el factor de escala de envolvente $E_{OrigMapped}$. Sin embargo, el parámetro equivalente especificado en la norma de USAC tiene generalmente un valor diferente que se “afina” para el procesamiento de SBR mejorado definido en la norma de USAC en lugar del procesamiento de SBR definido en la norma de AAC.

Además de los numerosos parámetros, un decodificador HE-AAC extendido puede reutilizar otros elementos al ejecutar una forma mejorada de replicación de la banda espectral de conformidad con formas de realización de la invención. Por ejemplo, los datos de envolvente y los datos de piso de ruido también se pueden extraer de los datos bs_data_env (factores de escala de envolvente) y bs_noise_env (factores de escala de piso de ruido) y reutilizar durante la forma mejorada de replicación de la banda espectral.

En esencia, estas formas de realización aprovechan los parámetros de configuración y los datos de envolvente ya admitidos por un decodificador HE-AAC o HE-AAC v2 de la técnica anterior en la carga útil de extensión de SBR para habilitar una forma mejorada de replicación de la banda espectral que requiere la menor cantidad posible de datos transmitidos adicionales. Los metadatos fueron refinados originalmente para una forma básica de HFR (por ej., el emparchado espectral de SBR), aunque de conformidad con las formas de realización, se utilizan para una

forma mejorada de HFR (por ej., la transposición armónica de eSBR). Como se mencionó anteriormente, los metadatos representan por lo general parámetros operativos (por ej., factores de escala de envolvente, factores de escala de piso de ruido, parámetros de rejilla de tiempo/frecuencia, información de adición de sinusoides, frecuencia de cruce /banda variable, modo de filtrado inverso, resolución de envolvente, modo de suavizado, modo de interpolación de frecuencias) afinados y destinados a ser utilizados con la forma básica de HFR (por ej., traslación lineal). Sin embargo, estos metadatos, combinados con otros parámetros de metadatos adicionales específicos para la forma mejorada de HFR (por ej., transposición armónica), se pueden utilizar para procesar los datos de audio, de manera eficiente y eficaz, usando la forma mejorada de HFR.

En consecuencia, se pueden generar decodificadores extendidos que admiten una forma mejorada de replicación de la banda espectral de manera muy eficiente basándose en elementos del flujo de bits ya definidos (por ejemplo, los de la carga útil de extensión de SBR) y sumando sólo los parámetros necesarios para admitir la forma mejorada de replicación de la banda espectral (en una carga útil de extensión de elemento de relleno). Esta característica de reducción de datos, combinada con la ubicación de parámetros recientemente agregados en un campo reservado de datos, tal como un contenedor de extensión, reduce sustancialmente las barreras para la creación de un decodificador que admite una forma mejorada de replicación de la banda espectral asegurando que el flujo de bits sea compatible con versiones anteriores de decodificadores antiguos que no admiten la forma mejorada de replicación de la banda espectral.

En la Tabla 3, el número de la columna derecha indica el número de bits del parámetro correspondiente de la columna izquierda.

En algunas formas de realización, se actualizan los tipos de objetos de SBR definidos en MPEG-4 AAC para que contengan la Herramienta de SBR o aspectos de la Herramienta de SBR mejorada (eSBR) señalada en el elemento de extensión de SBR (bs_extension_id== EXTENSION_ID_ESBR).

En algunas formas de realización, la invención es un método que incluye un paso de codificación de datos de audio para generar un flujo de bits codificado (por ej., un flujo de bits MPEG–4 AAC), mediante la inclusión de metadatos de eSBR en por lo menos un segmento de por lo menos un bloque del flujo de bits codificado y datos de audio en por lo menos un segmento adicional del bloque. En las formas de realización típicas, el método incluye un paso de multiplexación de los datos de audio con los metadatos de eSBR en cada bloque del flujo de bits codificado. En la decodificación típica del flujo de bits codificado en un decodificador eSBR, el decodificador extrae los metadatos de eSBR del flujo de bits (lo que incluye el análisis y demultiplexación de los metadatos de eSBR y los datos de audio) y usa los metadatos de eSBR para procesar los datos de audio a fin de generar un flujo de datos de audio decodificados.

Otro aspecto de la invención es un decodificador eSBR configurado para ejecutar el procesamiento de eSBR (por ej., usando por lo menos una de las herramientas de eSBR conocidas como transposición armónica o aplanamiento previo) durante la decodificación de un flujo de bits de audio codificado (por ej., un flujo de bits MPEG–4 AAC) que no incluye metadatos de eSBR. Se describe un ejemplo de decodificador de ese tipo con referencia a la Fig. 5.

El decodificador eSBR (400) de la Fig. 5 incluye la memoria intermedia 201 (que es idéntica a la memoria 201 de las Figs. 3 y 4), un desformateador de carga útil

de flujo de bits 215 (que es idéntico al desformateador 215 de la Fig. 4), el subsistema decodificador de audio 202 (al que en ocasiones se hace referencia como etapa de decodificación del “núcleo” o subsistema de decodificación del “núcleo”, y que es idéntico al subsistema de decodificación del núcleo 202 de la Fig. 3), el subsistema de generación de datos de control de eSBR 401, y la etapa de procesamiento de eSBR 203 (que es idéntica a la etapa 203 de la Fig. 3), conectadas de la manera ilustrada. También, por lo general, el decodificador 400 incluye otros elementos de procesamiento (no se ilustran).

En funcionamiento del decodificador 400, se envía una secuencia de bloques de un flujo de bits de audio codificado (un flujo de bits MPEG–4 AAC) recibido por el decodificador 400 de la memoria intermedia o buffer 201 al desformateador 215.

El desformateador 215 está acoplado y configurado para demultiplexar cada bloque del flujo de bits para extraer del mismo metadatos de SBR (incluyendo datos cuantificados de envolvente) y típicamente también otros metadatos. El desformateador 215 está configurado para enviar por lo menos los metadatos de SBR a la etapa de procesamiento de eSBR 203. El desformateador 215 también está acoplado y configurado para extraer datos de audio de cada bloque del flujo de bits, y para enviar los datos de audio extraídos al subsistema de decodificación (etapa de decodificación) 202.

El subsistema decodificador de audio 202 del decodificador 400 está configurado para decodificar los datos de audio extraídos por el desformateador 215 (decodificación a la que se puede hacer referencia como operación de decodificación del “núcleo”) para generar datos de audio decodificados, y para enviar los datos de audio decodificados a la etapa de procesamiento de eSBR 203. La decodificación se

ejecuta en el dominio frecuencial. Por lo general, una etapa final de procesamiento en el subsistema 202 aplica una transformación del dominio frecuencial al dominio temporal a los datos de audio decodificados en el dominio frecuencial, para que la salida del subsistema sean datos de audio decodificados en el dominio del tiempo. La etapa 203 está configurada para aplicar herramientas de SBR (y herramientas de eSBR) indicadas por los metadatos de SBR (extraídos por el desformateador 215) y por los metadatos de eSBR generados en el subsistema 401,ao los datos de audio decodificados (es decir, para ejecutar el procesamiento de SBR y de eSBR en la salida del subsistema de decodificación 202 usando los metadatos de SBR y eSBR) para generar los datos de audio completamente decodificados que constituyen la salida del decodificador 400. Por lo general, el decodificador 400 incluye una memoria (a la que puede acceder el subsistema 202 y la etapa 203) que almacena los datos de audio y metadatos de desformateados de salida del desformateador 215 (y opcionalmente también el subsistema 401), y la etapa 203 está configurada para acceder a los datos y metadatos de audio según lo necesario durante el procesamiento de SBR y eSBR. Se puede considerar el procesamiento de SBR de la etapa 203 como post procesamiento de la salida de del subsistema de decodificación del núcleo 202. Opcionalmente, el decodificador 400 incluye asimismo un subsistema de mezcla ascendente final (que puede aplicar herramientas de estéreo paramétrico (“PS”) definidos en la norma de MPEG–4 AAC, empleando los metadatos de PS extraídos por el desformateador 215), que está acoplado y configurado para ejecutar la mezcla ascendente de la salida de la etapa 203 para generar audio con mezcla ascendente completamente decodificado, que es emitido de la APU 210.

El subsistema de generación de datos de control 401 de la Fig. 5 está acoplado y configurado par detectar por lo menos una propiedad del flujo de bits de

audio codificado que se ha de decodificar, y para generar datos de control de eSBR (que pueden consistir o incluir metadatos de eSBR de cualquiera de los tipos incluidos en los flujos de bits de audio codificados de conformidad con otras formas de realización de la invención) en respuesta a por lo menos un resultado del paso de detección. Los datos de control de eSBR son enviados a la etapa 203 para activar la aplicación de herramientas de eSBR individuales o combinaciones de herramientas de eSBR al detectar una propiedad específica (o una combinación de propiedades) del flujo de bits, y/o para controlar la aplicación de dichas herramientas de eSBR. Por ejemplo, para controlar la eficiencia del procesamiento de eSBR usando transposición armónica, algunas formas de realización del subsistema de generación de datos de control 401 incluirían: un detector de música (por ej., una versión simplificada de un detector de música convencional) para establecer el parámetro sbrPatchingMode[ch] (y enviar el parámetro establecido a la etapa 203) en respuesta a la detección de que el flujo de bits es o no indicativo de música; un detector de transitorios para establecer el parámetro sbrOversamplingFlag[ch] (y enviar el parámetro establecido a la etapa 203) en respuesta a la detección de la presencia o ausencia de transitorios en el contenido de audio indicado por el flujo de bits y/o un detector de tono para establecer los parámetros sbrPitchInBinsFlag[ch] y sbrPitchInBins[ch] (y enviar los parámetros establecidos a la etapa 203) en respuesta a la detección del tono del contenido de audio indicado por el flujo de bits. Otros aspectos de la invención son métodos de decodificación de flujos de bits de audio ejecutada por cualquier forma de realización del decodificador de la invención descripto en este párrafo y el párrafo precedente.

Los aspectos de la invención incluyen un método de codificación o decodificación del tipo que cualquier forma de realización de la APU, el sistema o

dispositivo de la invención está configurado (por ej., programado) para ejecutar. Otros aspectos de la invención incluyen un sistema o dispositivo configurado (por ej., programado) para ejecutar cualquier forma de realización del método de la invención, y un medio legible por computación (por ej., un disco) que almacena un código (por ej., de manera no transitoria) para implementar cualquier forma de realización del método de la invención o los pasos del mismo. Por ejemplo, el sistema de la invención puede ser o incluir un procesador programable para uso general, un procesador o microprocesador de señales digitales, programado con software o firmware y/o de otro modo configurado para ejecutar cualquiera de una variedad de operaciones con datos, incluyendo una forma de realización de el método de la invención o los pasos del mismo. Dicho procesador para usos generales puede ser o incluir un sistema de computación que incluye un dispositivo de entrada, una memoria, y circuitos de procesamiento programados (y/o de otro modo configurado) para ejecutar una forma de realización del método de la invención (o los pasos del mismo) en respuesta a los datos enviados al mismo.

Las formas de realización de la presente invención pueden ser implementadas en hardware, firmware o software, o una combinación de estos (por ej., en forma de matriz lógica programable). A menos que se indique expresamente lo contrario, los algoritmos o procesos incluidos como parte de la invención no se relacionan de por sí con ninguna computadora específica ni otro aparato. En particular, se pueden utilizar diversas máquinas para usos generales con programas escritos de conformidad los conceptos presentados en la presente, o puede ser más conveniente construir un aparato más especializado (*por ej.*, circuitos integrados) para ejecutar los pasos necesarios del método. Por consiguiente, la invención puede ser implementada en uno o más programas de computación que ejecutan uno o más sistemas de

computación programables (por ej., una implementación de cualquiera de los elementos de la Fig. 1, o el codificador 100 de la Fig. 2 (o un elemento del mismo), o el decodificador 200 de la Fig. 3 (o un elemento del mismo), o el decodificador 210 de la Fig. 4 (o un elemento del mismo), o el decodificador 400 de la Fig. 5 (o un elemento del mismo)) cada uno de los cuales comprende por lo menos un procesador, por lo menos un sistema de almacenamiento de datos (incluyendo una memoria y/o elementos de almacenamiento volátiles y no volátiles), por lo menos un dispositivo o puerto de entrada y por lo menos un dispositivo o puerto de salida. Se aplica un código de programa a los datos de entrada para ejecutar las funciones descriptas en la presente y generar información de salida. La información de salida se aplica a uno o más dispositivos de salida, de manera conocida.

Cada uno de dichos programas pueden ser implementados en cualquier lenguaje informático deseado (incluyendo lenguajes de programación lógicos u orientados a objetos en máquinas, conjuntos o procedimientos de alto nivel) para comunicarse con un sistema de computación. En todo caso, el lenguaje puede ser un lenguaje compilado o interpretado.

Por ejemplo, mediante la implementación en un software de computación, de secuencias de instrucciones, se pueden implementar diversas funciones y pasos de las formas de realización de la invención mediante secuencias de instrucciones de software con múltiples subprocesos que corren en un hardware adecuado de procesamiento de señales digitales, en cuyo caso los diversos dispositivos, pasos y funciones de las formas de realización pueden corresponder a porciones de las instrucciones de software.

Cada uno de dichos programas de computación está preferentemente almacenado o es descargado a un medio o dispositivo de almacenamiento (*por ej.*,

memoria o medios en estado sólido o medios magnéticos u ópticos) legibles por una computadora programable para usos generales o especiales, para configurar y operar la computadora cuando el medio o dispositivo de almacenamiento es leído por el sistema de computación para ejecutar los procedimientos descriptos en la presente. El sistema de la invención también puede ser implementado en forma de medio legible por computadora, configurado con (es decir, en la que se ha almacenado) un programa de computación, donde el medio de almacenamiento así configurado hace que un sistema de computación opere de manera específica y predefinida para ejecutar las funciones descriptas en la presente.

Se ha descripto un número de formas de realización de la invención. De todos modos, se ha de entender que se pueden efectuar diversas modificaciones sin apartarse del espíritu y alcance de la invención. Numerosas modificaciones son posibles a la luz de los conceptos precedentemente vertidos. Por ejemplo, para facilitar implementaciones eficientes, se pueden usar desplazamientos de fase en combinación con complejos bancos de filtros de análisis y síntesis QMF. El banco de filtros de análisis es responsable del filtrado de la señal de baja banda en el dominio del tiempo generada por el decodificador del núcleo en una pluralidad de subbandas (por ej., subbandas de QMF). El banco de filtros de síntesis es responsable de la combinación de la banda alta regenerada producida por la técnica de HFR seleccionada (como lo indica el parámetro recibido sbrPatchingMode) con la banda baja decodificada para producir una señal de audio de salida de banda ancha. Sin embargo, una implementación de un banco de filtros dada que opera en un determinado modo de tasa de muestreo, por ej., una operación de doble tasa o un modo de SBR con muestra descendente no debe presentar desplazamientos de fase que dependan del flujo de bits. Los bancos de QMF empleados en la SBR son una

extensión con exponencial compleja de la teoría de los bancos de filtro de coseno modulado. Se puede demostrar que las restricciones de cancelación del solapamiento (“alias”) se tornan obsoletas al extender el banco de filtro de coseno modulado con modulación con exponencial compleja. Por consiguiente, en el caso de los bancos de QMF SBR, tanto los filtros de análisis, $h_k(n)$, como los filtros de síntesis, $f_k(n)$, se pueden definir según:

$$h_k(n) = f_k(n) = p_0(n) \exp\left\{i\frac{\pi}{M}\left(k+\frac{1}{2}\right)\left(n-\frac{N}{2}\right)\right\}, \quad 0 \le n \le N;\ 0 \le k < M \qquad (1)$$

donde $p_0(n)$ es un filtro prototipo simétrico o asimétrico de valor real (por lo general, un filtro prototipo de paso bajo), M indica el número de canales y N es el orden del filtro prototipo. El número de canales usados en el banco de filtros de análisis puede ser diferente del número de canales empleados en el banco de filtros de síntesis. Por ejemplo, el banco de filtros de análisis puede tener 32 canales y el banco de filtros de síntesis puede tener 64 canales. Cuando se opera el banco de filtros de síntesis en el modo de muestra descendente, el banco de filtros de síntesis puede tener sólo 32 canales. Dado que las muestras de subbandas del banco de filtros tienen valores complejos se puede anexar un paso de desplazamiento de fase posiblemente dependiente de los canales al banco de filtros de análisis. Estos desplazamientos de fase extra deben ser compensados antes del banco de filtro de síntesis. Si bien los términos de desplazamiento de fase pueden ser en principio, valores arbitrarios sin destruir la operación de la cadena análisis / síntesis de QMF, también se pueden restringir a ciertos valores para la verificación de conformidad. La señal de SBR se ve afectada por la elección de los factores de fase, en tanto que la señal de paso bajo

procedente del decodificador del núcleo no lo es. La calidad de audio de la señal de salida no resulta afectada.

Los coeficientes del filtro prototipo, $p_0(n)$, se pueden definir con una longitud, L, de 640, como se expone en la siguiente Tabla 4.

Tabla 4

n	$p_0(n)$	n	$p_0(n)$	n	$p_0(n)$
0	0,0000000000	214	0,0019765601	428	0,0117623832
1	-0,0005525286	215	-0,0032086896	429	0,0163701258
2	-0,0005617692	216	-0,0085711749	430	0,0207997072
3	-0,0004947518	217	-0,0141288827	431	0,0250307561
4	-0,0004875227	218	-0,0198834129	432	0,0290824006
5	-0,0004893791	219	-0,0258227288	433	0,0329583930
6	-0,0005040714	220	-0,0319531274	434	0,0366418116
7	-0,0005226564	221	-0,0382776572	435	0,0401458278
8	-0,0005466565	222	-0,0447806821	436	0,0434768782
9	-0,0005677802	223	-0,0514804176	437	0,0466303305
10	-0,0005870930	224	-0,0583705326	438	0,0495978676
11	-0,0006132747	225	-0,0654409853	439	0,0524093821
12	-0,0006312493	226	-0,0726943300	440	0,0550460034
13	-0,0006540333	227	-0,0801372934	441	0,0575152691
14	-0,0006777690	228	-0,0877547536	442	0,0598166570
15	-0,0006941614	229	-0,0955533352	443	0,0619602779
16	-0,0007157736	230	-0,1035329531	444	0,0639444805
17	-0,0007255043	231	-0,1116826931	445	0,0657690668
18	-0,0007440941	232	-0,1200077984	446	0,0674525021
19	-0,0007490598	233	-0,1285002850	447	0,0689664013
20	-0,0007681371	234	-0,1371551761	448	0,0703533073
21	-0,0007724848	235	-0,1459766491	449	0,0715826364
22	-0,0007834332	236	-0,1549607071	450	0,0726774642
23	-0,0007779869	237	-0,1640958855	451	0,0736406005
24	-0,0007803664	238	-0,1733808172	452	0,0744664394
25	-0,0007801449	239	-0,1828172548	453	0,0751576255
26	-0,0007757977	240	-0,1923966745	454	0,0757305756
27	-0,0007630793	241	-0,2021250176	455	0,0761748321
28	-0,0007530001	242	-0,2119735853	456	0,0765050718
29	-0,0007319357	243	-0,2219652696	457	0,0767204924
30	-0,0007215391	244	-0,2320690870	458	0,0768230011
31	-0,0006917937	245	-0,2423016884	459	0,0768173975
32	-0,0006650415	246	-0,2526480309	460	0,0767093490
33	-0,0006341594	247	-0,2631053299	461	0,0764992170

34	-0,0005946118	248	-0,2736634040	462	0,0761992479
35	-0,0005564576	249	-0,2843214189	463	0,0758008358
36	-0,0005145572	250	-0,2950716717	464	0,0753137336
37	-0,0004606325	251	-0,3059098575	465	0,0747452558
38	-0,0004095121	252	-0,3168278913	466	0,0741003642
39	-0,0003501175	253	-0,3278113727	467	0,0733620255
40	-0,0002896981	254	-0,3388722693	468	0,0725682583
41	-0,0002098337	255	-0,3499914122	469	0,0717002673
42	-0,0001446380	256	0,3611589903	470	0,0707628710
43	-0,0000617334	257	0,3723795546	471	0,0697630244
44	0,0000134949	258	0,3836350013	472	0,0687043828
45	0,0001094383	259	0,3949211761	473	0,0676075985
46	0,0002043017	260	0,4062317676	474	0,0664367512
47	0,0002949531	261	0,4175696896	475	0,0652247106
48	0,0004026540	262	0,4289119920	476	0,0639715898
49	0,0005107388	263	0,4402553754	477	0,0626857808
50	0,0006239376	264	0,4515996535	478	0,0613455171
51	0,0007458025	265	0,4629308085	479	0,0599837480
52	0,0008608443	266	0,4742453214	480	0,0585915683
53	0,0009885988	267	0,4855253091	481	0,0571616450
54	0,0011250155	268	0,4967708254	482	0,0557173648
55	0,0012577884	269	0,5079817500	483	0,0542452768
56	0,0013902494	270	0,5191234970	484	0,0527630746
57	0,0015443219	271	0,5302240895	485	0,0512556155
58	0,0016868083	272	0,5412553448	486	0,0497385755
59	0,0018348265	273	0,5522051258	487	0,0482165720
60	0,0019841140	274	0,5630789140	488	0,0466843027
61	0,0021461583	275	0,5738524131	489	0,0451488405
62	0,0023017254	276	0,5845403235	490	0,0436097542
63	0,0024625616	277	0,5951123086	491	0,0420649094
64	0,0026201758	278	0,6055783538	492	0,0405349170
65	0,0027870464	279	0,6159109932	493	0,0390053679
66	0,0029469447	280	0,6261242695	494	0,0374812850
67	0,0031125420	281	0,6361980107	495	0,0359697560
68	0,0032739613	282	0,6461269695	496	0,0344620948
69	0,0034418874	283	0,6559016302	497	0,0329754081
70	0,0036008268	284	0,6655139880	498	0,0315017608
71	0,0037603922	285	0,6749663190	499	0,0300502657
72	0,0039207432	286	0,6842353293	500	0,0286072173
73	0,0040819753	287	0,6933282376	501	0,0271859429
74	0,0042264269	288	0,7022388719	502	0,0257875847
75	0,0043730719	289	0,7109410426	503	0,0244160992
76	0,0045209852	290	0,7194462634	504	0,0230680169
77	0,0046606460	291	0,7277448900	505	0,0217467550
78	0,0047932560	292	0,7358211758	506	0,0204531793

79	0,0049137603	293	0,7436827863	507	0,0191872431
80	0,0050393022	294	0,7513137456	508	0,0179433381
81	0,0051407353	295	0,7587080760	509	0,0167324712
82	0,0052461166	296	0,7658674865	510	0,0155405553
83	0,0053471681	297	0,7727780881	511	0,0143904666
84	0,0054196775	298	0,7794287519	512	-0,0132718220
85	0,0054876040	299	0,7858353120	513	-0,0121849995
86	0,0055475714	300	0,7919735841	514	-0,0111315548
87	0,0055938023	301	0,7978466413	515	-0,0101150215
88	0,0056220643	302	0,8034485751	516	-0,0091325329
89	0,0056455196	303	0,8087695004	517	-0,0081798233
90	0,0056389199	304	0,8138191270	518	-0,0072615816
91	0,0056266114	305	0,8185776004	519	-0,0063792293
92	0,0055917128	306	0,8230419890	520	-0,0055337211
93	0,0055404363	307	0,8272275347	521	-0,0047222596
94	0,0054753783	308	0,8311038457	522	-0,0039401124
95	0,0053838975	309	0,8346937361	523	-0,0031933778
96	0,0052715758	310	0,8379717337	524	-0,0024826723
97	0,0051382275	311	0,8409541392	525	-0,0018039472
98	0,0049839687	312	0,8436238281	526	-0,0011568135
99	0,0048109469	313	0,8459818469	527	-0,0005464280
100	0,0046039530	314	0,8480315777	528	0,0000276045
101	0,0043801861	315	0,8497805198	529	0,0005832264
102	0,0041251642	316	0,8511971524	530	0,0010902329
103	0,0038456408	317	0,8523047035	531	0,0015784682
104	0,0035401246	318	0,8531020949	532	0,0020274176
105	0,0032091885	319	0,8535720573	533	0,0024508540
106	0,0028446757	320	0,8537385600	534	0,0028446757
107	0,0024508540	321	0,8535720573	535	0,0032091885
108	0,0020274176	322	0,8531020949	536	0,0035401246
109	0,0015784682	323	0,8523047035	537	0,0038456408
110	0,0010902329	324	0,8511971524	538	0,0041251642
111	0,0005832264	325	0,8497805198	539	0,0043801861
112	0,0000276045	326	0,8480315777	540	0,0046039530
113	-0,0005464280	327	0,8459818469	541	0,0048109469
114	-0,0011568135	328	0,8436238281	542	0,0049839687
115	-0,0018039472	329	0,8409541392	543	0,0051382275
116	-0,0024826723	330	0,8379717337	544	0,0052715758
117	-0,0031933778	331	0,8346937361	545	0,0053838975
118	-0,0039401124	332	0,8311038457	546	0,0054753783
119	-0,0047222596	333	0,8272275347	547	0,0055404363
120	-0,0055337211	334	0,8230419890	548	0,0055917128
121	-0,0063792293	335	0,8185776004	549	0,0056266114
122	-0,0072615816	336	0,8138191270	550	0,0056389199
123	-0,0081798233	337	0,8087695004	551	0,0056455196

124	-0,0091325329	338	0,8034485751	552	0,0056220643
125	-0,0101150215	339	0,7978466413	553	0,0055938023
126	-0,0111315548	340	0,7919735841	554	0,0055475714
127	-0,0121849995	341	0,7858353120	555	0,0054876040
128	0,0132718220	342	0,7794287519	556	0,0054196775
129	0,0143904666	343	0,7727780881	557	0,0053471681
130	0,0155405553	344	0,7658674865	558	0,0052461166
131	0,0167324712	345	0,7587080760	559	0,0051407353
132	0,0179433381	346	0,7513137456	560	0,0050393022
133	0,0191872431	347	0,7436827863	561	0,0049137603
134	0,0204531793	348	0,7358211758	562	0,0047932560
135	0,0217467550	349	0,7277448900	563	0,0046606460
136	0,0230680169	350	0,7194462634	564	0,0045209852
137	0,0244160992	351	0,7109410426	565	0,0043730719
138	0,0257875847	352	0,7022388719	566	0,0042264269
139	0,0271859429	353	0,6933282376	567	0,0040819753
140	0,0286072173	354	0,6842353293	568	0,0039207432
141	0,0300502657	355	0,6749663190	569	0,0037603922
142	0,0315017608	356	0,6655139880	570	0,0036008268
143	0,0329754081	357	0,6559016302	571	0,0034418874
144	0,0344620948	358	0,6461269695	572	0,0032739613
145	0,0359697560	359	0,6361980107	573	0,0031125420
146	0,0374812850	360	0,6261242695	574	0,0029469447
147	0,0390053679	361	0,6159109932	575	0,0027870464
148	0,0405349170	362	0,6055783538	576	0,0026201758
149	0,0420649094	363	0,5951123086	577	0,0024625616
150	0,0436097542	364	0,5845403235	578	0,0023017254
151	0,0451488405	365	0,5738524131	579	0,0021461583
152	0,0466843027	366	0,5630789140	580	0,0019841140
153	0,0482165720	367	0,5522051258	581	0,0018348265
154	0,0497385755	368	0,5412553448	582	0,0016868083
155	0,0512556155	369	0,5302240895	583	0,0015443219
156	0,0527630746	370	0,5191234970	584	0,0013902494
157	0,0542452768	371	0,5079817500	585	0,0012577884
158	0,0557173648	372	0,4967708254	586	0,0011250155
159	0,0571616450	373	0,4855253091	587	0,0009885988
160	0,0585915683	374	0,4742453214	588	0,0008608443
161	0,0599837480	375	0,4629308085	589	0,0007458025
162	0,0613455171	376	0,4515996535	590	0,0006239376
163	0,0626857808	377	0,4402553754	591	0,0005107388
164	0,0639715898	378	0,4289119920	592	0,0004026540
165	0,0652247106	379	0,4175696896	593	0,0002949531
166	0,0664367512	380	0,4062317676	594	0,0002043017
167	0,0676075985	381	0,3949211761	595	0,0001094383
168	0,0687043828	382	0,3836350013	596	0,0000134949

169	0,0697630244	383	0,3723795546	597	-0,0000617334
170	0,0707628710	384	-0,3611589903	598	-0,0001446380
171	0,0717002673	385	-0,3499914122	599	-0,0002098337
172	0,0725682583	386	-0,3388722693	600	-0,0002896981
173	0,0733620255	387	-0,3278113727	601	-0,0003501175
174	0,0741003642	388	-0,3168278913	602	-0,0004095121
175	0,0747452558	389	-0,3059098575	603	-0,0004606325
176	0,0753137336	390	-0,2950716717	604	-0,0005145572
177	0,0758008358	391	-0,2843214189	605	-0,0005564576
178	0,0761992479	392	-0,2736634040	606	-0,0005946118
179	0,0764992170	393	-0,2631053299	607	-0,0006341594
180	0,0767093490	394	-0,2526480309	608	-0,0006650415
181	0,0768173975	395	-0,2423016884	609	-0,0006917937
182	0,0768230011	396	-0,2320690870	610	-0,0007215391
183	0,0767204924	397	-0,2219652696	611	-0,0007319357
184	0,0765050718	398	-0,2119735853	612	-0,0007530001
185	0,0761748321	399	-0,2021250176	613	-0,0007630793
186	0,0757305756	400	-0,1923966745	614	-0,0007757977
187	0,0751576255	401	-0,1828172548	615	-0,0007801449
188	0,0744664394	402	-0,1733808172	616	-0,0007803664
189	0,0736406005	403	-0,1640958855	617	-0,0007779869
190	0,0726774642	404	-0,1549607071	618	-0,0007834332
191	0,0715826364	405	-0,1459766491	619	-0,0007724848
192	0,0703533073	406	-0,1371551761	620	-0,0007681371
193	0,0689664013	407	-0,1285002850	621	-0,0007490598
194	0,0674525021	408	-0,1200077984	622	-0,0007440941
195	0,0657690668	409	-0,1116826931	623	-0,0007255043
196	0,0639444805	410	-0,1035329531	624	-0,0007157736
197	0,0619602779	411	-0,0955533352	625	-0,0006941614
198	0,0598166570	412	-0,0877547536	626	-0,0006777690
199	0,0575152691	413	-0,0801372934	627	-0,0006540333
200	0,0550460034	414	-0,0726943300	628	-0,0006312493
201	0,0524093821	415	-0,0654409853	629	-0,0006132747
202	0,0495978676	416	-0,0583705326	630	-0,0005870930
203	0,0466303305	417	-0,0514804176	631	-0,0005677802
204	0,0434768782	418	-0,0447806821	632	-0,0005466565
205	0,0401458278	419	-0,0382776572	633	-0,0005226564
206	0,0366418116	420	-0,0319531274	634	-0,0005040714
207	0,0329583930	421	-0,0258227288	635	-0,0004893791
208	0,0290824006	422	-0,0198834129	636	-0,0004875227
209	0,0250307561	423	-0,0141288827	637	-0,0004947518
210	0,0207997072	424	-0,0085711749	638	-0,0005617692
211	0,0163701258	425	-0,0032086896	639	-0,0005525280
212	0,0117623832	426	0,0019765601		
213	0,0069636862	427	0,0069636862		

El filtro prototipo, $p_0(n)$, también puede ser derivado de la Tabla 4 mediante una o más operaciones matemáticas tales como redondeo, submuestreo, interpolación y decimado.

Se ha de entender que, dentro del alcance de las reivindicaciones adjuntas, la invención puede ser puesta en práctica de manera diferente de la específicamente descripta en la presente. Todo número de referencia contenido en las siguientes reivindicaciones tiene sólo fines ilustrativos y no debe ser interpretado como interpretación ni limitar de manera alguna las reivindicaciones.