

19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 386 911**

51 Int. Cl.:  
**H03M 13/29** (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

- 96 Número de solicitud europea: **07842624 .4**  
96 Fecha de presentación: **17.09.2007**  
97 Número de publicación de la solicitud: **2080271**  
97 Fecha de publicación de la solicitud: **22.07.2009**

54 Título: **Procedimiento y aparato para codificar y decodificar datos**

30 Prioridad:  
**04.10.2006 US 828213 P**  
**06.10.2006 US 539404**

45 Fecha de publicación de la mención BOPI:  
**05.09.2012**

45 Fecha de la publicación del folleto de la patente:  
**05.09.2012**

73 Titular/es:  
**Motorola Mobility, Inc.**  
**600 North US Highway 45**  
**Libertyville, IL 60048 , US**

72 Inventor/es:  
**BLANKENSHIP, Yufei W.,;**  
**BLANKENSHIP, T. Keith,;**  
**CLASSON, Brian K., y**  
**NIMBALKER, Ajit,**

74 Agente/Representante:  
**de Elzaburu Márquez, Alberto**

ES 2 386 911 T3

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

## DESCRIPCIÓN

Procedimiento y aparato para codificar y decodificar datos

**Campo de la invención**

5 La presente invención se refiere, en general, a la codificación y decodificación de datos y, en particular, a un procedimiento y a un aparato para turbo codificar y turbo decodificar datos.

**Antecedentes de la invención**

10 A veces, las transmisiones de datos digitales a través de enlaces por cable o inalámbricos pueden resultar corrompidas, por ejemplo, por ruido en el enlace o canal, por interferencia desde otras transmisiones o por otros factores ambientales. Para combatir los errores introducidos por el canal, muchos sistemas de comunicación emplean técnicas de corrección de errores para ayudar en la comunicación.

15 Una técnica utilizada para la corrección de errores es la turbo codificación de un bloque de información a transmitir. Utilizando dicha técnica, un codificador en el interior del transmisor de un sistema de comunicación codificará un bloque  $u$  de entrada de  $K$  bits de longitud en un bloque  $x$  de palabra de código de  $N$  bits. A continuación, el bloque  $x$  de palabra de código es transmitido a través del canal, posiblemente después de un procesamiento adicional, tal como intercalamiento de canales, tal como se define en las especificaciones IEEE 802.16e. En el receptor, el turbo decodificador toma el vector  $y$  de señal recibido, de longitud  $N$ , como entrada, y genera una estimación  $\hat{u}$  del vector  $u$ .

20 Típicamente, el turbo codificador está compuesto de dos codificadores convolucionales constituyentes. El primer codificador constituyente toma el bloque  $u$  de entrada como entrada en su orden original, y el segundo codificador constituyente toma el bloque  $u$  de entrada en su orden intercalado después de pasar  $u$  a través de un turbo intercalador  $\pi$ . La salida  $x$  del turbo codificador está compuesta de los bits sistemáticos (iguales al bloque  $u$  de entrada), los bits de paridad del primer codificador constituyente y los bits de paridad del segundo codificador constituyente.

25 Correspondientemente, el turbo decodificador en el interior del receptor del sistema de comunicación está compuesto de dos decodificadores convolucionales constituyentes, uno para cada código constituyente. Los decodificadores constituyentes están separados por el intercalador  $\pi$  y el des-intercalador  $\pi^{-1}$  correspondiente. Los mensajes en el formato de relación de probabilidad logarítmica (Log-Likelihood Ratios, LLRs) son pasados, iterativamente, entre los decodificadores constituyentes. La decisión  $\hat{u}$  se toma después de varias iteraciones.

30 El turbo intercalador  $\pi$  es el componente clave en el diseño del turbo código. Es responsable de la codificación del bloque  $u$  de entrada de una manera pseudo-aleatoria, proporcionando, de esta manera, las palabras de código  $x$  con una buena distribución de pesos y, por lo tanto, una buena capacidad de corrección de errores. Además del rendimiento de decodificación, el turbo intercalador  $\pi$  tiene un impacto considerable sobre la implementación del turbo decodificador en el interior del receptor. Normalmente, el rendimiento de los turbo códigos mejora con una longitud de creciente del intercalador. Sin embargo, se obtiene un retorno decreciente al incrementar el tamaño de intercalador. En la práctica, el máximo tamaño de bloque de la corrección de errores hacia adelante (Forward, Error Correction, FEC) (es decir, el tamaño del intercalador) de un turbo código está limitado a un cierto valor debido a razones de complejidad y retraso. Por lo tanto, si el tamaño del bloque de entrada (bloque de transporte concatenado o CTB) es mayor que el máximo tamaño de bloque FEC soportado por el turbo código, el CTB es segmentado (por ejemplo, usando una regla de segmentación de bloques de código) en varios segmentos pequeños, cada uno de los cuales es procesamiento, por separado, por el turbo codificador en el transmisor y, correspondientemente, por el turbo decodificador en el receptor.

40 En algunos sistemas, el turbo código puede estar diseñado para soportar solo un pequeño número de tamaños de bloque FEC por varias razones (por ejemplo, decodificación a alta velocidad, almacenamiento reducido, etc.). Por lo tanto, existe una necesidad de un procedimiento y un aparato de turbo codificación y turbo decodificación que adapte el CTB a los tamaños de bloque FEC disponibles.

45 MOTOROLA, FRANCE TELECOM, GET AND ORANGE: "EUTRA FEC Enhancement", TDOC R1-061050 OF 3GPP TSG RAN WG 1 MEETING #44BIS, 27 de Marzo de 2006 (2006-03-27), -31 de Marzo de 2006 (2006-03-31) páginas 1-14, XP002475873 Atenas, Grecia, se refiere a la turbo codificación para sistemas de comunicación 3GPP CDMA.

**Breve descripción de los dibujos**

La Fig. 1 es un diagrama de bloques de un transmisor.

La Fig. 2 es un diagrama de bloques de un receptor.

La Fig. 3 es un diagrama de bloques del turbo codificador de la Fig. 1.

La Fig. 4 es un diagrama de bloques de un formador de bloques de transporte en el lado del transmisor.

La Fig. 5 es un diagrama de bloques de un ensamblador de bloques de transporte en el lado del receptor.

La Fig. 6 es un diagrama de flujo que muestra el funcionamiento del transmisor de la Fig. 1.

La Fig. 7 es un diagrama de flujo que muestra el funcionamiento del receptor de la Fig. 2.

## 5 Descripción detallada de los dibujos

Con el propósito de abordar la necesidad indicada anteriormente, en la presente memoria se proporciona un procedimiento y un aparato para la turbo codificación y la turbo decodificación.

En una realización, se recibe un bloque de transporte concatenado (CTB) de longitud  $X$  y se determinan dos tamaños de bloque FEC,  $K_{i-1}$  y  $K_i$  a partir de un grupo de tamaños de bloque FEC, no contiguos, en la que los tamaños de bloque FEC, no contiguos, están comprendidos entre  $K_{min}$  y  $K_{max}$ , y en la que  $K_{min} \leq K_{i-1} < K_{max}$ ,  $K_{min} \leq K_i \leq K_{max}$  y en la que  $K_{i-1}$  y  $K_i$  están basados, adicionalmente, en  $X$ . El bloque de transporte concatenado de longitud  $X$  es segmentado en  $C$  segmentos, cada uno de un tamaño sustancialmente igual a  $K_{i-1}$  o  $K_i$ . Una palabra de código FEC para cada uno de los  $C$  segmentos es determinada usando los tamaños de bloque FEC  $K_i$  o  $K_{i-1}$ , y las  $C$  palabras de código FEC son transmitidas a través del canal.

El beneficio del procedimiento anterior es que reduce la inclusión de bits de relleno requeridos para codificar el CTB, mientras se usa el menor número de segmentos permitidos por los tamaños de bloque FEC, no contiguos, disponibles. En particular, el procedimiento usa dos tamaños de bloque FEC diferentes (pero contiguos) para minimizar el número de bits de relleno mientras usa el mínimo número de segmentos permitido por los tamaños de bloque FEC, no contiguos, disponibles. Además, los tamaños de bloque FEC para los tamaños de segmento y el número de segmentos pueden ser determinados usando circuitería lógica simple.

Previamente a la descripción de la codificación y la decodificación de datos, se establecen las definiciones siguientes para establecer los antecedentes necesarios:

- En aras de la simplicidad de la notación, un bloque de transporte concatenado se refiere al resultado de la concatenación de uno o más bloques de transporte, después de añadir una cabecera, tal como bits de CRC, a cada bloque de transporte.
- $X$  denota el tamaño de un bloque de transporte concatenado (por ejemplo, la longitud, en bits, del bloque de transporte concatenado).
- $Y$  denota el número total de bits de relleno añadidos a un bloque de transporte concatenado.
- $C$  denota el número de segmentos en el que un bloque de transporte concatenado es segmentado.
- $CBSS_i$  denota el tamaño del  $i$ -ésimo segmento de un bloque de transporte concatenado ( $i = 1, \dots, C$ ), donde  $C$  es el tamaño del segmento.  $CBSS$  indica el tamaño del segmento del bloque de código.
- $K_{i-1}$  y  $K_i$  indican los tamaños de bloque FEC (por ejemplo, los tamaños para los cuales se ha definido un intercalador interno del turbo código) que pueden ser usados para una codificación FEC de los segmentos de un bloque de transporte concatenado.
- $K_{tabla}$  indica un conjunto de tamaños de bloque FEC, no contiguos, disponibles (tamaños para los cuales se define un intercalador interno del turbo código).
- $K_{relleno}$  indica el número de bits de relleno añadidos a un segmento.
- $R$  indica la tasa de código madre del turbo codificador (por ejemplo,  $R = 1/3$  para el turbo código 3GPP)
- $R^{-1}$  es el inverso de la tasa de código madre del turbo codificador (por ejemplo,  $R^{-1} = 3$  para el turbo código 3GPP).
- $N_{tb}$  es el número de bits finales de la palabra de código FEC en la salida del codificador FEC. En particular,
  - $N_{tb} = 12$  para el turbo código 3GPP con bits finales.
  - $N_{tb} = 0$  para un turbo código 3GPP con tail-biting, en el que el estado inicial y el estado final son iguales.
- $\Pi$  indica el intercalador interno del turbo código.

- La operación de redondeo hacia abajo  $\lfloor x \rfloor$  indica el mayor número entero que es menor o igual que  $x$ , y la operación de redondeo hacia arriba  $\lceil x \rceil$  indica el menor número entero que es mayor o igual que  $x$ .

Ahora, con referencia a los dibujos, en los que los números similares indican componentes similares, la Fig. 1 es un diagrama de bloques de un ejemplo de un transmisor 100 que es útil para comprender la invención. Tal como se muestra, un transmisor 100 comprende circuitería 102 de segmentación de bloques de código, circuitería 103 de relleno, un turbo codificador 104, circuitería 105 de eliminación de relleno, un transmisor 108, circuitería 106 lógica y una tabla/almacenamiento 107. El transmisor 100 comprende además circuitería de recepción (no mostrada en la Fig. 1) que recibe un bloque de transporte concatenado de longitud  $X$ . La circuitería 106 lógica determina un tamaño  $K_i$  de bloque FEC disponible a partir de un grupo de tamaños 107 de bloque FEC, no contiguos, en el que los tamaños de bloque FEC, no contiguos, disponibles se encuentran entre  $K_{min}$  y  $K_{max}$ , y en el que  $K_{min} \leq K_i < K_{max}$  y en el que  $K_i$  está basado, además, en  $X$ . La circuitería 102 de segmentación de bloques de código segmenta el bloque de transporte concatenado de longitud  $X$  en  $C$  segmentos de tamaños sustancialmente iguales a  $K_i$ ; y la circuitería 104 de codificación determina una palabra de código FEC para cada uno de los  $C$  segmentos usando el tamaño de bloque FEC  $K_i$ . Finalmente, la circuitería 108 de transmisión transmite las  $C$  palabras de código FEC sobre un canal.

En una realización, el transmisor 100 comprende circuitería de recepción (no mostrada en la Fig. 1) que recibe un bloque de transporte concatenado de longitud  $X$ , circuitería 106 lógica que determina dos tamaños de bloque FEC,  $K_{i-1}$  y  $K_i$ , disponibles, a partir de un grupo de tamaños 107 de bloque FEC, no contiguos, disponibles, en el que los tamaños de bloque FEC, no contiguos, disponibles están entre  $K_{min}$  y  $K_{max}$  y en el que  $K_{min} \leq K_{i-1} < K_{max}$ ,  $K_{min} \leq K_i \leq K_{max}$ , y en el que  $K_{i-1}$  y  $K_i$  están basados, además, en  $X$ . El transmisor 100 comprende circuitería 102 de segmentación de bloques de código que segmenta el bloque de transporte concatenado de longitud  $X$  en  $C$  segmentos de tamaños sustancialmente iguales a  $K_{i-1}$  o  $K_i$ , y circuitería 104 de codificación que determina una palabra de código FEC para cada uno de los  $C$  segmentos usando un tamaño de bloque FEC  $K_i$  o  $K_{i-1}$ . Finalmente, hay provista una circuitería 108 de transmisión que transmite las  $C$  palabras de código FEC a través de un canal.

La circuitería 104 de codificación está precedida por una circuitería 103 de relleno que inserta bits de relleno en los segmentos para formar un bloque FEC de entrada. El codificador 104 FEC codifica el bloque FEC de entrada, y la circuitería 105 de eliminación de rellenos elimina los bits relacionados con los bits de relleno.

Durante el funcionamiento del transmisor 100, los datos en forma de un bloque de transporte concatenado son recibidos por una circuitería 102. La circuitería 102 prepara el bloque de transporte concatenado antes de la codificación de corrección de errores hacia adelante (FEC).

En general, el rango de los tamaños CTB (es decir,  $X$ ) puede ser diferente del rango de tamaños de bloque FEC soportados por el esquema FEC subyacente en la capa física para un sistema de comunicaciones. Por lo tanto, es necesario definir una regla que divide un CTB en segmentos que pueden ser manipulados eficientemente por el FEC. En particular, los tamaños FEC (es decir,  $X$ ) son, frecuentemente, mucho más largos que el tamaño de bloque FEC máximo que el codificador 104 FEC puede manipular. Por lo tanto, el CTB necesita ser segmentado por la circuitería 102 en un número de segmentos de menor tamaño y cada segmento debe ser codificado por un codificador 104 FEC en una palabra de código FEC separada.

La circuitería 102 usa una regla de segmentación de bloques de código que está diseñada para conseguir un buen rendimiento (es decir, el rendimiento agregado de los segmentos para un CTB determinado) con el FEC subyacente. Implica los aspectos siguientes para cualquier tamaño de CTB determinado:

- Elegir el número de segmentos  $C$ ;
- Elegir los tamaños de cada segmento;
- Insertar los bits de relleno antes de la decodificación FEC y eliminar los bits de relleno después de la codificación FEC, si el tamaño de segmento no puede ser manipulado directamente por el FEC.

Las reglas de segmentación propuestas son particularmente útiles para un sistema de acceso radio terrestre UMTS evolucionado (EUTRA) en el que un tubo codificador puede ser definido para solo un conjunto limitado de tamaños de bloque FEC (tamaños de intercalador). A diferencia del turbo codificador 3GPP Versión 6, que define 5.075 intercaladores de tamaños contiguos, uno para cada tamaño de intercalador  $K_i$  entre 40 bits y 5.114 bits, un turbo codificador EUTRA puede definir un número limitado de tamaños  $K_{tabla}$  de bloques FEC (por ejemplo, 40~50 intercaladores con tamaños no contiguos en el rango de 128 bits a 6.144 bits) para cubrir un gran número de tamaños de segmento (por ejemplo, 6.144-128+1 = 6.017 tamaños). Cuando un tamaño de segmento es igual a un tamaño de bloque FEC disponible, entonces el segmento puede ser tomado como un bloque FEC de entrada directamente (sin necesidad, de esta manera, de inserción de bits de relleno). Sin embargo, cuando el tamaño de segmento no es igual a ninguno de los tamaños de bloque FEC disponibles, puede aplicarse la inserción de bits de relleno, y puede usarse el siguiente tamaño de bloque FEC disponible,

más largo, (es decir, tamaño de intercalador) elegido de la  $K_{\text{tabla}}$  107.

Número de segmentos:

Las reglas de segmentación tienen en cuenta las propiedades siguientes de la turbo codificación.

- (a) El rendimiento del turbo código mejora según conforme aumenta el tamaño de bloque FEC.
- 5 (b) La mejora del rendimiento del turbo código mediante un aumento de los tamaños de bloque FEC ha reducido los retornos más allá de unos pocos miles de bits.
- (c) Un CTB es recibido correctamente solo si todos los segmentos son recibidos correctamente.

10 Las propiedades (a) y (c) indican que el rendimiento global es probable que esté dominado por el segmento que tiene el peor rendimiento. De esta manera, es preferente tener segmentos que son, aproximadamente, de tamaños iguales de manera que son sometidos a codificación FEC con tamaños de bloque FEC aproximadamente iguales (y, por lo tanto, con una protección contra errores aproximadamente igual correspondiente desde una perspectiva FEC).

15 La propiedad (b) sugiere que no es necesario incluir intercaladores para tamaños muy grandes en la tabla ( $K_{\text{tabla}}$ ). Sin embargo, los tamaños de bloque FEC definidos en  $K_{\text{tabla}}$  pueden depender de otros factores. Por ejemplo, i) para almacenamiento/complejidad reducidos, puede ser deseable un pequeño número de intercaladores  $K_{\text{tabla}}$ , e ii) el tamaño de intercalador máximo definido en  $K_{\text{tabla}}$  puede ser elegido para limitar el número de segmentos por CTB, limitando, de esta manera, la penalización por segmentación de un CTB. La penalización por segmentación es la pérdida de rendimiento debido a la división de un CTB en varios segmentos en lugar de codificar todo el CTB en una palabra de código FEC.

20 La propiedad (c) sugiere que debería usarse el mínimo número de segmentos para reducir la penalización por segmentación.

Teniendo en consideración todo lo anterior, el número de segmentos es  $C = \lceil X / K_{\text{max}} \rceil$ , donde  $K_{\text{max}}$  es el tamaño máximo de bloque FEC definido en  $K_{\text{tabla}}$ . Suponiendo que  $CBSS_i$  indica el tamaño de segmento del segmento  $i$ -ésimo ( $i = 1, \dots, C$ ) del bloque de transporte concatenado, la suma de todos los segmentos es igual al tamaño  $X$  de bloque de transporte concatenado, es decir, los tamaños de segmento están restringidos por la ecuación siguiente

25 
$$\sum_{i=1}^C CBSS_i = X$$

La sección siguiente describe la determinación del tamaño de bloque FEC usado para la codificación FEC, uno por cada uno de los tamaños  $C$  de segmento.

30 Determinación del tamaño de bloque FEC

Dado que un CTB de longitud  $X$  es la entrada a la función de segmentación de bloques de código, la regla para determinar el tamaño de bloque FEC (tamaño de intercalador) para un turbo codificador según se describe en la versión 6 de la norma 3GPP es tal como se indica a continuación

35 
$$\begin{aligned} C &= \lceil X / K_{\text{max}} \rceil \\ K_j &= \max(40, \lceil X / C \rceil) \\ Y &= CK_j - X. \end{aligned} \tag{1}$$

40 donde  $K_{\text{max}} = 5.114$  es el tamaño máximo de intercalador para el Turbo Código, Versión 6,  $C$  es el número de segmentos (o bloques de código),  $K_j$  es el tamaño de intercalador, e  $Y$  es el número total de bits de relleno insertados por el CTB de tamaño  $X$  cuando se usan  $C$  bloques FEC de entrada de tamaño  $K_j$ . En esencia, un CTB de tamaño  $X$  es segmentado en  $C$  segmentos de aproximadamente igual tamaño, y cada segmento es codificado usando un turbo código con un intercalador de  $K_j$  bits. Si  $Y > 0$ ,  $Y$  bits conocidos son incluidos al principio del primer segmento antes de la codificación. Debido a que los tamaños de bloque FEC (es decir, intercaladores) están definidos para todos los tamaños entre  $K_{\text{min}} = 40$  y  $K_{\text{max}} = 5.114$  en turbo código 3GPP, Versión 6, el número de bits de relleno está limitado a  $C$ , el número de segmentos usados para la segmentación de bloques de código.

45

Sin embargo, en otros sistemas, tales como el considerado para EUTRA, los tamaños de bloque FEC (tamaños de intercalador) pueden ser definidos solo para tamaños no contiguos (un conjunto más grueso de tamaños de intercaladores)  $K_{\text{tabla}}$ . En dichos casos, los tamaños de segmento que no son iguales a ninguno de los tamaños de bloque FEC (es decir, no definidos en  $K_{\text{tabla}}$ ) deben ser manipulados usando bits de relleno antes de la codificación FEC (y eliminación selectiva después de la codificación para alcanzar una tasa de código deseada).

Suponiendo que un turbo codificador soporta solo un número limitado de tamaños de bloque FEC distribuidos entre  $K_{\text{min}}$  y  $K_{\text{max}}$ , ambos incluidos, a continuación, se describen dos simples procedimientos de segmentación de bloques de código de un bloque de transporte concatenado, de longitud  $X$ , usando  $K_{\text{tabla}}$ . Estos procedimientos usan tan pocos segmentos como sea posible mientras reducen también el número de bits de relleno que se requieren para la codificación.

Permitir solo un tamaño de bloque FEC

Un procedimiento ejemplar que es útil para la comprensión de la invención es modificar (1) y permitir que todos los segmentos sean codificados con un único tamaño  $K_i$  de intercalador, donde

$$I = \arg \min_{K_i \geq \lceil X/C \rceil} (K_i - \lceil X/C \rceil), \quad (2)$$

donde  $i, 1 \leq i \leq T$ , es un índice en el grupo de tamaños de bloque FEC, no contiguos, disponibles en  $K_{\text{tabla}}$ , suponiendo que los  $T$  tamaños en  $K_{\text{tabla}}$  están ordenados en orden ascendente. En esencia, este procedimiento elige el menor  $K_i$  de  $K_{\text{tabla}}$  que es mayor o igual que  $\lceil X/C \rceil$ , es decir,  $K_i = \lceil X/C \rceil + \delta$ , donde  $0 \leq \delta < K_i - K_{i-1}$ , y  $K_{i-1} < \lceil X/C \rceil$ . Obsérvese que se asigna que  $K_{i-1} = 0$  cuando  $i=1$ . Por lo tanto, el número de bits de relleno viene dado por

$$\begin{aligned} Y &= CK_i - X \\ &= C(\lceil X/C \rceil + \delta) - X, \end{aligned} \quad (3)$$

Por lo tanto,  $Y$  es grande cuando  $\delta$  es grande. Los ejemplos siguientes ilustran la manera en la que el número de tamaños de bloque FEC disponibles ( $K_{\text{tabla}}$ ) afecta a  $Y$ .

- Si  $K_{\text{tabla}}$  tiene todos los valores entre  $Z_{\text{min}} = 40$  y  $Z_{\text{max}} = 5.114$ , el número máximo de bits de relleno es igual a  $C-1$ .
- Si  $K_{\text{tabla}}$  tiene  $T = 100$  valores distribuidos uniformemente entre  $Z_{\text{min}} = 40$  y  $Z_{\text{max}} = 5.114$ , el número total máximo de bits de relleno incluidos para todos los sectores es aproximadamente igual a  $50xC$ .

Por lo tanto, el número de bits de relleno puede ser controlado variando la granularidad de los tamaños de bloque FEC en  $K_{\text{tabla}}$ . El número de bits de relleno puede ser reducido también usando otro enfoque, tal como se describe a continuación. Sin embargo, antes de describir el procedimiento siguiente, cabe señalar que en un caso general, puede elegirse cualquier  $K_i (\geq \lceil X/C \rceil)$  de  $K_{\text{tabla}}$  para aplicar una codificación FEC con el coste de un número potencialmente mayor de bits de relleno. En este caso, los tamaños de segmento obtenidos después de la segmentación de bloques de código satisfacen  $CBSS_i \leq K_i$ , para  $i = 1, \dots, C$ . En este caso, la circuitería 106 lógica determina el número de segmentos usando la relación siguiente

$$C = \lceil X / K_{\text{max}} \rceil,$$

Permitir solo dos tamaños de bloque FEC contiguos

En una realización, en lugar de usar un tamaño  $K_i$  de bloque FEC para la codificación de todos los segmentos de un CTB determinado, se propone que se seleccionen dos tamaños de bloque FEC contiguos  $K_{i-1}$  y  $K_i$ ,  $K_{i-1} < K_i$ ,  $1 \leq i \leq T$ , desde  $K_{\text{tabla}}$ . Obsérvese que se asigna que  $K_{i-1} = 0$  cuando  $i = 1$ . El número de segmentos  $C$  y el mayor tamaño  $K_i$  de bloque FEC todavía son elegidos para ser el mismo, como en los casos anteriores, es decir,  $C$  se calcula todavía como en (1) y  $K_i$  se calcula todavía como en (2). Sin embargo, el número de segmentos codificados con tamaño  $K_{i-1}$  y tamaño  $K_i$  se determina como se indica a continuación (para una fácil comprensión, todos los cálculos implicados se repiten a continuación). En este caso, la circuitería 106 lógica realiza las siguientes operaciones para encontrar el número de segmentos,

$$\begin{aligned}
 C &= \lceil X/K_{\max} \rceil = C_I + C_{I-1}, \\
 Y &= CK_I - X, \\
 D_I &= K_I - K_{I-1}, \\
 C_{I-1} &= \lfloor Y/D_I \rfloor \\
 C_I &= C - \lfloor Y/D_I \rfloor
 \end{aligned} \tag{4}$$

y  $C_{I-1}$  y  $C_I$  son el número de segmentos que son codificados usando tamaños de bloque FEC  $K_{I-1}$  y  $K_I$ , respectivamente, donde  $K_I$  es el menor tamaño de entre los tamaños de bloque FEC disponibles que es mayor o igual que  $\lceil X/C \rceil$ , y  $D_I$  indica la diferencia entre los tamaños de intercalador  $K_{I-1}$  y  $K_I$ , contiguos.

5 Obsérvese que en (4),  $Y$  no indica el número de bits de relleno necesarios si se permiten dos tamaños contiguos, pero indica el número de bits de relleno requeridos si solo se usa un tamaño de  $K_I$  para todos los  $C$  segmentos.

De esta manera, la segmentación de bloques de código forma  $C$  segmentos, de los cuales  $C_{I-1}$  segmentos son codificados en FEC con un tamaño de bloque FEC  $K_{I-1}$ . Obsérvese que cuando  $Y < D_I$ , (4) proporciona  $C_{I-1} = 0$ , y este procedimiento degenera al uso de un tamaño de bloque FEC de  $K_I$ . (es decir, se permite el tamaño  $K_{I-1}$ , pero no se usa en realidad). Por otro lado, cuando  $Y \geq D_I$ , este procedimiento requiere menos bits de relleno que la inclusión de todos los  $C$  segmentos hasta el mayor tamaño de bloque FEC  $K_I$ . Este procedimiento es óptimo en el sentido de que el número de bits de relleno  $Y'$  añadidos por cada CTB se garantiza que es el mínimo mientras se usan los menos segmentos posibles.  $Y'$  se determina como se indica a continuación

$$Y' = C_{I-1}K_{I-1} + C_I K_I - X, \tag{5}$$

15

Puede probarse que  $Y'$  está limitado por  $D_I$ , independientemente de  $C$ ,

$$0 \leq Y' < K_I - K_{I-1}, \tag{6}$$

20 En este caso, los tamaños de segmento obtenidos después de la segmentación de bloques de código tienen las restricciones siguientes, suponiendo (sin pérdida de generalidad que los primeros  $C_I$  segmentos son codificados con  $K_I$  y el resto con  $K_{I-1}$ ).

$$CBSS_i \leq K_I, \text{ para } i = 1, \dots, C_I$$

$$CBSS_i \leq K_{I-1}, \text{ para } i = C_I + 1, \dots, C; \text{ si } C_{I-1} \geq 1.$$

25 Con referencia a la Fig. 1, tal como se ha expuesto anteriormente, debe elegirse un tamaño de bloque FEC apropiado a partir de la tabla 107 de tamaños de bloque FEC, no contiguos. La circuitería 106 lógica realiza la tarea de elegir el tamaño/los tamaños de bloque FEC apropiados, tal como se ha expuesto anteriormente. Un ejemplo de la tabla 107 se proporciona en la Tabla 1. Por ejemplo, en el primer caso, la circuitería 106 lógica elige el tamaño de bloque FEC a partir de los tamaños de bloque FEC, no contiguos, disponibles entre  $K_{\min}$  y  $K_{\max}$ , y en el que  $K_{\min} \leq K_I \leq K_{\max}$ , y en el que  $K_I$  está basado, adicionalmente, en  $X$ . Particularmente si debe usarse un único tamaño de bloque FEC  $K_I$ , la circuitería 106 lógica elige el menor  $K_I$  (desde  $K_{\text{tabla}}$ ) que no es menor que  $\lceil X/C \rceil$ , es decir,  $K_I = \lceil X/C \rceil + \delta$ , donde  $\delta \geq 0$ , y  $K_{I-1} < \lceil X/C \rceil$ . Sin embargo, si se van a usar dos tamaños de bloque FEC,  $K_{I-1}$  y  $K_I$  se determinan con la ecuación (4) que proporciona el número de segmentos que son codificados usando los tamaños de bloque FEC  $K_{I-1}$  y  $K_I$ .

35

Tabla 1: El conjunto de tamaños de bloque FEC para los cuales hay definido un intercalador interno del turbo codificador

$K_{tabla}$					
128	256	512	1.024	2.048	4.096
144	288	576	1.152	2.304	4.608
160	320	640	1.280	2.560	5.120
176	352	704	1.408	2.816	5.632
192	384	768	1.536	3.072	6.144
208	416	832	1.664	3.328	
216	440	888	1.776	3.568	
240	480	960	1.920	3.840	

5 El codificador 104 FEC subyacente soporta solo un conjunto limitado de tamaños de bloque FEC (o tamaños de entrada). Sin pérdida de generalidad, se supone que el codificador 104 FEC es un turbo codificador, y el conjunto de tamaños de bloque FEC soportados por el turbo codificador es el conjunto de tamaños de intercalador para los cuales está definido el intercalador interno del turbo código. Sin embargo, una persona con conocimientos ordinarios en la materia, reconocerá que pueden usarse otros esquemas FEC en 104, incluyendo códigos de comprobación de paridad de baja densidad (LDPC), códigos convolucionales, turbo códigos de bloque, códigos Reed-Solomon, etc.

10 Una vez determinados el número de segmentos  $C$  y el tamaño de bloque FEC para cada segmento, esta información es pasada a la circuitería 102 de segmentación de bloques de código, donde el CTB ( $X$  bits) es segmentado en  $C$  segmentos que son codificados con un tamaño de bloque FEC  $K_i$ , si sólo se permite un tamaño de bloque FEC. Como alternativa, si se permiten dos tamaños de bloque FEC contiguos, la circuitería 102 de segmentación de bloques de código puede poner en la salida  $C_i$  segmentos que deben ser codificados con un tamaño de bloque FEC  $K_i$  y  $C_{i-1}$  segmentos que deben ser codificados con un tamaño de bloque FEC  $K_{i-1}$ .

15 Inserción de bits de relleno

El número de bits de relleno (incluidos para cada segmento) puede ser determinado en base al tamaño de segmento y al tamaño del bloque FEC que está siendo usado para la codificación FEC del segmento. Hay al menos dos maneras de distribuir los bits de relleno totales en los  $C$  segmentos.

- 20 • Relleno concentrado. Colocar los bits de relleno en tan pocos segmentos como sea posible sin hacer que los tamaños de segmento sean demasiado pequeños. En un ejemplo, todos los bits de relleno pueden aparecer al comienzo del primer segmento. La ventaja es que sólo debe manipularse, por separado, un segmento (que contiene todos los bits de relleno). Además, los bits de relleno pueden ser incluidos en el segmento que es codificado con el mayor tamaño de bloque FEC  $K_i$  en lugar de un tamaño de bloque FEC más pequeño  $K_{i-1}$  cuando se usan dos tamaños de bloque FEC para un CTB. Este procedimiento es particularmente atractivo cuando se permiten dos tamaños de bloque FEC contiguos para la codificación.
- 25 • Relleno distribuido. Distribuir uniformemente los bits de relleno (tanto como sea posible) en una pluralidad de segmentos. Los bits de relleno pueden ser distribuidos hasta en todos los  $C$  segmentos.

30 Para una implementación eficiente del transmisor y del receptor, es preferente un relleno concentrado. Una realización preferente es añadir  $Y'$  (si se permiten dos tamaños de bloque FEC contiguos,  $Y$  si se permite un solo tamaño de bloque FEC) bits de relleno consecutivos en la parte delantera de uno de los segmentos (por ejemplo, el primero o el último) usando un tamaño de bloque FEC  $K_i$  antes de enviarlo al codificador. En términos de rendimiento, es equivalente a añadir los  $Y'$  bits consecutivos al final de un segmento que tiene un tamaño de bloque FEC  $K_i$ .

35 Con referencia a la Fig. 1, para cada segmento (producido por la circuitería 102), una palabra de código FEC es determinada usando las etapas de inserción de bits de relleno en el segmento para formar un bloque de entrada FEC; realizando una codificación FEC del bloque de entrada FEC, y eliminando los bits relacionados con los bits de relleno.

Cada segmento producido por la circuitería 102 es pasado a la circuitería 103 de relleno, donde tiene lugar la inserción de bits de relleno. Si no se requieren bits de relleno, entonces, la circuitería de relleno es transparente, es decir, no se añaden

bits de relleno ( $K_{relleno} = 0$ ). Los segmentos (junto con los bits de relleno) son pasados, a continuación, a un turbo codificador 104 en el que una turbo codificación de los  $C$  segmentos conduce a  $C$  palabras de código FEC. A continuación, los bits de relleno son eliminados por la circuitería 105 y las  $C$  palabras de código resultantes son transmitidas apropiadamente por la circuitería 108 de transmisión. Si la circuitería 103 no añade bits de relleno, entonces la circuitería 105 de eliminación de relleno es transparente, es decir, no se eliminan bits de relleno ( $K_{relleno} = 0$ ). Obsérvese que es posible que la circuitería 105 no elimine ningún bit correspondiente a los bits de relleno.

La Fig. 2 es un diagrama de bloques de un receptor. Durante el funcionamiento, el vector de señal recibido pasa a través de la circuitería 202 de des-segmentación de bloques de código que organiza las partes del vector de señal recibido según el segmento con el que están asociadas. El tamaño de segmento, el número de segmentos, el tamaño de bloque FEC usado para turbo-decodificar cada segmento, el número de bits de relleno pueden ser determinados usando la circuitería 213 lógica y la tabla 215 de tamaños de bloque FEC disponibles, en una manera similar a la del codificador. La circuitería 204 de manipulación de relleno usa el conocimiento de la posición de los bits de relleno en favor del turbo decodificador 206, por ejemplo, estableciendo las LLRs correspondientes a los bits de relleno a una alta magnitud. Después de la turbo decodificación, la circuitería 208 elimina los bits de relleno para obtener una estimación de un segmento. El ensamblador 211 bloques de código ensambla el transporte estimado recogiendo y organizando, de manera adecuada, las estimaciones de los segmentos obtenidas desde la circuitería 208.

#### Eliminación de los bits de paridad del codificador constituyente

Esta sección proporciona una manera específica de determinación de la palabra de código FEC. Se describe el procedimiento, que aprovecha el conocimiento de la inserción de los bits de relleno en el transmisor. En particular, el procedimiento determina qué bits (tanto bits sistemáticos como bits de paridad) pueden ser eliminados de la salida del turbo codificador con una nula o insignificante degradación del rendimiento. En general, los bits de relleno son conocidos y, por lo tanto, los bits sistemáticos de estos bits (igual a los bits conocidos) pueden ser eliminados antes de la transmisión. Sin embargo, no está claro si puede eliminarse alguno de los bits de paridad.

La Fig. 3 es un diagrama de bloques del turbo codificador 104 de la Fig. 1. Durante el funcionamiento, el bloque de entrada de  $K$  bits de longitud es introducido tanto al intercalador 301 como al codificador 302 constituyente. El intercalador 301 intercala el bloque de entrada y pasa el bloque de entrada en un orden intercalado al codificador 303 constituyente. A continuación, el codificador 303 constituyente codifica el bloque de entrada intercalado. En una manera similar, el codificador 302 constituyente codifica el bloque de entrada original. El bloque  $x$  de la palabra de código está compuesto de un bloque sistemático (igual al bloque de entrada FEC), la salida del codificador 302 constituyente y la salida del codificador 303 constituyente. A continuación, el bloque  $x$  de la palabra de código es enviado a la circuitería 105.

En un turbo codificador convencional, tal como por ejemplo, turbo códigos de cola, los estados iniciales de los codificadores constituyentes (contenidos de los registros de desplazamiento) se supone que son todos iguales a cero. En tal caso, cuando  $K_{relleno}$  bits de relleno (generalmente 0s) son insertados al comienzo del bloque de entrada del turbo código, los bits sistemáticos y los bits de paridad del codificador 302 constituyente, correspondientes a las  $K_{relleno}$  posiciones de bit, son todos igual a cero. Por lo tanto, estos bits pueden ser eliminados en el transmisor y el receptor puede utilizar este conocimiento mientras realiza una turbo decodificación. Sin embargo, en el codificador 303 constituyente, los  $K_{relleno}$  bits son codificados por el intercalador del turbo código y, por lo tanto, los bits de paridad del codificador 303 constituyente, correspondientes a los bits de relleno, no son conocidos y, de esta manera, no pueden ser eliminados de manera simple.

Cuando el turbo codificador tiene codificadores constituyentes de tipo tail-biting, es posible que el estado inicial de los codificadores constituyentes no sea siempre cero. Para los códigos de tipo tail-biting, el estado inicial y el estado final para un codificador constituyente son iguales y dependen del bloque de entrada. Por lo tanto, cuando  $K_{relleno}$  bits de relleno consecutivos (es decir, ceros) son insertados al comienzo del bloque de entrada del turbo código, los bits de paridad del codificador 302 constituyente, correspondientes a las  $K_{relleno}$  posiciones de bit, no son siempre ceros. Sin embargo, puede probarse que la mayoría de estos  $K_{relleno}$  bits de paridad del codificador 302 constituyente no transportan ninguna información.

En general, los grupos de bits de relleno consecutivos son insertados en un segmento para formar un bloque de entrada FEC en el que la longitud de grupo es un múltiplo de  $2^m - 1$  ( $= 7$  para los códigos convolucionales constituyentes dentro del turbo codificador 3GPP). A continuación, el bloque de entrada FEC es codificado en FEC y los bits de paridad relacionados con los bits de relleno son eliminados. El codificador FEC puede ser un código convolucional de tipo tail-biting usado individualmente, o un código convolucional de tipo tail-biting usado como un código constituyente de un turbo codificador.

En particular, cuando se usan para turbo códigos con códigos constitutivos de tipo tail-biting, los grupos de bits sistemáticos correspondientes a los bits de relleno pueden ser eliminados, y los bits de paridad correspondientes a los grupos de bits de relleno en la salida de un codificador constituyente pueden ser eliminados, en el que el codificador

constituyente toma el bloque de entrada FEC sin intercalación para los turbo codificadores de tipo tail-biting. Esto puede demostrarse como se indica a continuación.

5 Sea  $S(i)$  el estado del registro de desplazamiento del codificador 302 constituyente en la etapa  $i$ , sea  $m$  el número de elementos en el registro de desplazamiento, y sea  $g$  cualquier número entero mayor que 0. Cuando  $(2^m-1) \times g$  ceros son introducidos al codificador constituyente desde la etapa  $i+1$  a la etapa  $i+(2^m-1) \times g$ , la siguiente es una propiedad del codificador convolucional recursivo (tal como el usado en el turbo código 3GPP, versión 6),

$$S(i) = S(i+(2^m-1)g) \quad (7)$$

10 Obsérvese que es posible que  $S(i)$  no sea una constante. Además, es posible que los estados  $S(j)$  en el medio no sean una constante o sean iguales al estado  $S(i)$ ,  $i < j < i+(2^m-1)g$ .

15 Por lo tanto, el estado del codificador constituyente permanece inalterado entre una etapa  $i+1$  y la etapa  $i+(2^m-1)g$ . Por lo tanto, el transmisor puede aprovechar (7) eliminando la salida del codificador constituyente durante estas etapas, ya que esos bits de relleno no cambian el estado del registro de desplazamiento y, de esta manera, no proporcionan ninguna información para el decodificador. El decodificador en el interior del receptor también puede aprovechar (7), de manera similar, en base al conocimiento de las posiciones y valores de los bits de relleno. A continuación, el procedimiento anterior se describe con un ejemplo en el que  $K_{relleno}$  bits de relleno (ceros) son insertados en posiciones consecutivas en la entrada de un turbo código de tipo tail-biting.

20 Debido a que  $K_{relleno}$  bits de relleno consecutivos (ceros) son insertados en el bloque de entrada del turbo código,  $g = \lfloor K_{relleno} / (2^m-1) \rfloor$  y, por lo tanto,  $pxg \times (2^m-1)$  bits de paridad del codificador 302 constituyente pueden ser eliminados, donde  $p$  es el número bits de paridad en la salida del codificador 302 constituyente que son generados para cada bit en el bloque de entrada FEC. Por lo tanto, sólo los bits de paridad correspondientes a los grupos de bits de relleno en la salida del codificador 302 constituyente son eliminados, donde el codificador 302 constituyente toma el bloque de entrada FEC no intercalado los turbo codificadores de tipo tail-biting.

25 Para un turbo codificador 3GPP de tipo tail-biting,  $p = 1$  en codificador constituyente,  $m = 3$ . De esta manera,  $7 \lfloor K_{relleno} / (2^m-1) \rfloor$  bits de paridad pueden ser eliminados del codificador 302 constituyente para  $K_{relleno}$  bits de relleno consecutivos. Debido a que  $m = 3$ , como máximo sólo 6 bits de paridad correspondientes a los  $K_{relleno}$  bits de relleno del codificador 302 constituyente pueden necesitar ser mantenidos en la salida del codificador 302 constituyente.

30 En el codificador 303 constituyente, los  $K_{relleno}$  bits de relleno pueden resultar dispersados debido al intercalador de turbo código. Por lo tanto, puede que no sea posible eliminar los bits de paridad del codificador 303 constituyente sin afectar al rendimiento.

La sección siguiente describe algunos escenarios ejemplares en los que puede usarse la regla de segmentación de bloques de código, por ejemplo, Solicitud de repetición automática-híbrida (hybrid-Automatic Repeat reQuest, HARQ), múltiple entrada múltiple salida (MIMO), etc.

#### Formador de bloques de transporte (TB)

35 La regla de segmentación de bloques de código, descrita anteriormente, es aplicada a un bloque de transporte concatenado (CTB) en un canal ARQ híbrido (HARQ). Antes de la segmentación de bloques de código, los bits de información a enviar a un solo usuario desde la estación base dentro de un intervalo de tiempo de transmisión (TTI) pueden necesitar ser divididos en al menos un bloque de transporte, pasando, de esta manera, a través de al menos un canal HARQ. Por ejemplo, la Fig. 4 muestra un ejemplo en el que los bits de información son transmitidos usando dos canales HARQ (correspondientes a HARQ1 y HARQ2), y dos bloques de transporte TB1 y TB2. Durante el funcionamiento, los bits de información de longitud  $A$  son recibidos por la circuitería 402 de formación de TB para ser transmitidos en una o más secuencias espaciales. La circuitería 402 designa  $X$  bits como un bloque de transporte TB1, en el que  $X \leq A$ ; el procesador 404 HARQ1 adjunta los bits de CRC a los  $X$  bits para formar el bloque de transporte concatenado, de longitud  $X$ ; el bloque de transporte concatenado de longitud  $X$ , es asignado a un primer canal HARQ. El bloque de transporte concatenado es enviado a la circuitería 102 de segmentación de bloques de código.

La circuitería 402 designa  $W = A - X$  bits a partir de los bits de información como un segundo bloque de transporte TB2; el procesador 406 HARQ2 adjunta los bits de CRC a  $Y$  bits y forma un segundo bloque de transporte concatenado; el bloque de transporte concatenado es asignado a un segundo canal HARQ. El bloque de transporte concatenado es enviado a la circuitería 102 de segmentación de bloques de código.

50 Obsérvese que los circuitos 404 y 406 pueden realizar funciones adicionales, tales como otras funcionalidades relacionadas con HARQ, adición de información de control, etc.

Aunque los conceptos en la Fig. 4 se ilustran usando dos canales HARQ, los mismos pueden extenderse fácilmente a una pluralidad de canales HARQ. Si hay más de un canal HARQ soportado por un usuario dentro de un intervalo de tiempo de transmisión (TTI), la regla de segmentación de bloques de código puede aplicarse a cada TB.

5 Pueden ocurrir múltiples canales HARQ debido a que hay demasiadas palabras de código FEC (o segmentos) por TTI por usuario, tal como por un gran ancho de banda (por ejemplo, 20 MHz), una modulación de mayor orden (por ejemplo, 64 QAM), MIMO de múltiples secuencias, etc. También pueden usarse múltiples canales HARQ para TBS que tienen diferente calidad de servicio, tales como VoIP y "best-effort data".

10 Una palabra de código MIMO comprende los bits que son enviados a un solo usuario en un TTI en una secuencia MIMO. De esta manera, una palabra de código MIMO puede comprender una o más palabras de código FEC. Algunas veces, una palabra de código MIMO es usada para hacer referencia a los bits en una secuencia MIMO.

15 Pueden definirse reglas para la creación de un TB. En una realización, un TB comprenderá no más de  $x$  (por ejemplo,  $x = 8$ ) palabras de código FEC (valor de  $x$  determinado por el planificador eNodeB en EUTRA). En otra realización, si se necesitan más de  $x$  palabras de código FEC para un TB, entonces se crean dos TBs, tal como se indica a continuación. El paquete es dividido, de manera aproximadamente homogénea, entre dos TBs, teniendo cada TB casi el mismo número de palabras de código FEC de aproximadamente el mismo tamaño. En todavía otra realización, para las palabras de código FEC que deben ser enviadas a dos secuencias MIMO, cada una pertenece a un TB separado. En todavía otra realización, para las palabras de código FEC que deben ser enviadas a tres secuencias MIMO mientras se usan 2 canales HARQ simultáneos, el primero (en promedio, la secuencia de mejor calidad) pertenece a un TB y las secuencias segunda y tercera pertenecen a un segundo TB. En todavía otra realización, cuatro palabras de código MIMO a ser enviadas usando dos canales HARQ, son posibles varias combinaciones. Por ejemplo, (a) TB1 = 1,2 TB2 = 3,4 (b) TB1 = 1,3 TB2 = 2,4 (c) TB1 = 1,2 TB2 = 2,3 (d) TB1 = 1, TB2 = 2,3,4. Aquí TB<sub>*i*</sub> se refiere al TB del *i*-ésimo canal HARQ; los números 1 al 4 indican el número de la palabra de código MIMO (o secuencia).

25 La Fig. 5 es un diagrama de bloques de procesamiento del receptor cuando los bits de información son recibidos sobre al menos un canal HARQ. Los bits recibidos desde el ensamblador 211 de bloques de código son introducidos a los procesadores 504 y 506 de canal apropiados. La salida de los procesadores de canal son los bloques de transporte TB1 y TB2 estimados que son introducidos a la circuitería 502 del ensamblador de TB que combina los TBs y pone en la salida los bit de información estimados.

30 La Fig. 6 es un diagrama de flujo que muestra el funcionamiento del transmisor de la Fig. 1. El flujo de la lógica comienza en la etapa 601, donde la circuitería de segmentación recibe un bloque de transporte concatenado de longitud  $X$ . En la etapa 603, la circuitería lógica accede a la tabla 107 y elige un tamaño de bloque FEC apropiado. Tal como se ha expuesto anteriormente, en una primera realización de la presente invención, el tamaño de bloque FEC  $K_i$  es determinado a partir de un grupo de tamaños de bloque FEC, no contiguos, situados en la tabla 107, donde los tamaños de bloque FEC, no contiguos, disponibles en la tabla 107 se encuentran entre  $K_{min}$  y  $K_{max}$ , y en el que  $K_{min} \leq K_i < K_{max}$ . Tal como se ha expuesto anteriormente,  $K_i$  se basa en  $X$ .  $X$  es determinado por la circuitería 106 lógica a partir del bloque de transporte concatenado. Una vez determinado  $X$ , se determinan  $K_i = \lceil X / C \rceil + \delta$  y  $C = \lceil X / K_{max} \rceil$ . En una segunda realización de la presente invención, se determinan los tamaños de bloque FEC  $K_i$  y  $K_{i-1}$ , donde  $K_i = \lceil X / C \rceil + \delta$ .

35 Continuando, en la etapa 605, el número de segmentos  $C$  y los tamaños de bloque FEC son pasados a la circuitería 102 de segmentación y, en la etapa 607, la circuitería de segmentación segmenta el bloque de transporte concatenado de longitud  $X$  en  $C$  segmentos de tamaño sustancialmente igual a  $K_i$  (o, como alternativa,  $K_i$  y  $K_{i-1}$ ). Los bits de relleno son añadidos (si es necesario) en la etapa 609 por la circuitería 103 y, en la etapa 611, cada uno de los  $C$  segmentos es codificado (es decir, se determina una palabra de código FEC para cada uno de los  $C$  segmentos). Finalmente, en la etapa 613, las palabras de código FEC son transmitidas a través de la circuitería 108 de transmisión.

45 Tal como se ha expuesto anteriormente, la etapa de determinación de una palabra de código FEC comprende las etapas de inserción de los bits de relleno en el segmento para formar un bloque de entrada FEC, codificación FEC del bloque de entrada FEC, y eliminación de los bits relacionados con los bits de relleno. Esta etapa puede implicar la inserción de grupos de bits de relleno consecutivos en un segmento para formar un bloque de entrada FEC donde la longitud de grupo es un múltiplo de 7, la codificación FEC del bloque de entrada FEC y la eliminación de los bits relacionados con los bits de relleno. La eliminación de los bits de relleno comprende las etapas de eliminar los grupos de bits sistemáticos correspondientes a los bits de relleno y eliminar los bits de paridad correspondientes a los grupos de bits de relleno en la salida del codificador 1 constituyente, donde el codificador constituyente toma el bloque de entrada FEC no intercalado para los turbo codificadores de tipo tail-biting.

50 La Fig. 7 es un diagrama de flujo que muestra el funcionamiento del receptor de la Fig. 2. El flujo lógico comienza en la etapa 701, donde el tamaño del segmento, el número de segmentos, el tamaño de bloque FEC usado para turbo-decodificar cada segmento, y el número de bits de relleno se determinan usando circuitería 213 lógica y la tabla 215. Tal como se ha expuesto anteriormente, en un ejemplo útil para la comprensión de la presente invención, el tamaño de bloque

5 FEC  $K_i$  se determina a partir de un grupo de tamaños de bloque FEC, no contiguos, situados en la tabla 215, donde los tamaños de bloque FEC, no contiguos, disponibles en la tabla 215 se encuentran entre  $K_{min}$  y  $K_{max}$ , y en el que  $K_{min} \leq K_i < K_{max}$ . Tal como se ha expuesto anteriormente,  $K_i$  se basa en  $X$ .  $X$  es determinado por la circuitería 213 lógica a partir del vector de señal recibido. A continuación, la circuitería 213 lógica determina  $K_i = \lceil X / C \rceil + \delta$  y  $C = \lceil X / K_{max} \rceil$ . En una realización de la presente invención, se determinan los tamaños de bloque FEC  $K_i$  y  $K_{i-1}$ , donde  $K_i = \lceil X / C \rceil + \delta$ .

10 En la etapa 703, un vector de señal recibido pasa a través de la circuitería 202 de segmentación de bloques de código, que organiza las partes del vector de señal recibido según los  $C$  segmentos con los que están asociadas. En la etapa 705, la circuitería 204 de manipulación de relleno usa el conocimiento de la posición de los bits de relleno para ayudar al turbo decodificador 206, por ejemplo, estableciendo las LLRs correspondientes a los bits de relleno a una magnitud alta. Cada uno de los  $C$  segmentos es decodificado en la etapa 707. Después de la turbo decodificación, la circuitería 208 elimina los bits de relleno para obtener una estimación de un segmento (etapa 709). El ensamblador 211 de bloques de código ensambla el transporte estimado recogiendo y organizando, de manera adecuada, las estimaciones de los segmentos obtenidos desde la circuitería 208 (etapa 711).

15 Aunque la invención ha sido mostrada y descrita, particularmente, con referencia a una realización particular, las personas con conocimientos en la materia entenderán que pueden realizarse diversos cambios a la misma, en la forma y los detalles, sin alejarse del alcance de la invención, según se define en las reivindicaciones adjuntas.

**REIVINDICACIONES**

1. Procedimiento de operación de un transmisor, en el que el procedimiento comprende:

recibir un bloque de transporte concatenado de longitud  $X$ ;

5 determinar dos tamaños de bloque FEC, contiguos, disponibles  $K_{i-1}$  y  $K_i$  a partir de un grupo de tamaños de bloque FEC, no contiguos, en el que los tamaños de bloque FEC, no contiguos, disponibles se encuentran entre  $K_{min}$  y  $K_{max}$ , y en el que  $K_{min} \leq K_{i-1} < K_{max}$ ,  $K_{min} \leq K_i \leq K_{max}$ , y en el que  $K_{i-1}$  y  $K_i$  están basados, además, en  $X$ ;

segmentar el bloque de transporte concatenado de longitud  $X$  en  $C$  segmentos de tamaños

$$CBSS_i \leq K_i, \text{ para } i = 1, \dots, C_i$$

$$CBSS_i \leq K_{i-1}, \text{ para } i = C_i + 1, \dots, C; \text{ si } C_{i-1} \geq 1;$$

10 determinar una palabra de código FEC para codificar cada uno de los  $C$  segmentos usando tamaños de bloque FEC  $K_i$  o  $K_{i-1}$ ; y

transmitir las  $C$  palabras de código FEC a través del canal.

en el que  $C = \lceil X / K_{max} \rceil = C_i + C_{i-1}$ , donde

$$Y = C K_i - X,$$

15  $C_{i-1} = \lfloor Y / D_i \rfloor,$

$$C_i = C - \lfloor Y / D_i \rfloor,$$

y  $C_{i-1}$  y  $C_i$  son el número de segmentos que son codificados usando los tamaños de bloque FEC  $K_{i-1}$  y  $K_i$ , respectivamente, donde  $K_i$  es el menor tamaño, de entre los tamaños de bloque FEC disponibles, que es mayor o igual que  $\lceil X / C \rceil$ , y  $D_i$  indica la diferencia entre los tamaños de bloque FEC  $K_{i-1}$  y  $K_i$  contiguos.

20 2. Un aparato que comprende:

circuitería de recepción que recibe un bloque de transporte concatenado de longitud  $X$ ;

25 circuitería lógica para determinar dos tamaños de bloque FEC, contiguos, disponibles  $K_{i-1}$  y  $K_i$ , a partir de un grupo de tamaños de bloque FEC, no contiguos, en el que los tamaños de bloque FEC, no contiguos, disponibles se encuentran entre  $K_{min}$  y  $K_{max}$ , y en el que  $K_{min} \leq K_{i-1} < K_{max}$ ,  $K_{min} \leq K_i \leq K_{max}$ , y en el que  $K_{i-1}$  y  $K_i$  están basados, además, en  $X$ ;

circuitería (102) de segmentación de bloques de código que segmenta el bloque de transporte concatenado de longitud  $X$  en  $C$  segmentos de tamaños

$$CBSS_i \leq K_i, \text{ para } i = 1, \dots, C_i$$

$$CBSS_i \leq K_{i-1}, \text{ para } i = C_i + 1, \dots, C; \text{ si } C_{i-1} \geq 1;$$

30 circuitería (104) de codificación para la determinación de una palabra de código FEC para cada uno de los  $C$  segmentos usando el tamaño de bloque FEC  $K_i$  o  $K_{i-1}$ ; y

circuitería (108) de transmisión que transmite las  $C$  palabras de código FEC a través de un canal;

en el que  $C = \lceil X / K_{max} \rceil = C_i + C_{i-1}$ , donde

$$Y = C K_i - X,$$

35  $C_{i-1} = \lfloor Y / D_i \rfloor,$

$$C_i = C - \lfloor Y / D_i \rfloor,$$

y  $C_{i-1}$  y  $C_i$  son el número de segmentos que son codificados usando los tamaños de bloque FEC  $K_{i-1}$  y  $K_i$ , respectivamente, donde  $K_i$  es el menor tamaño, de entre los tamaños de bloque FEC disponibles, que es mayor o igual que  $\lceil X / C \rceil$ , y  $D_i$  indica la diferencia entre los tamaños de bloque FEC  $K_{i-1}$  y  $K_i$  contiguos.

40

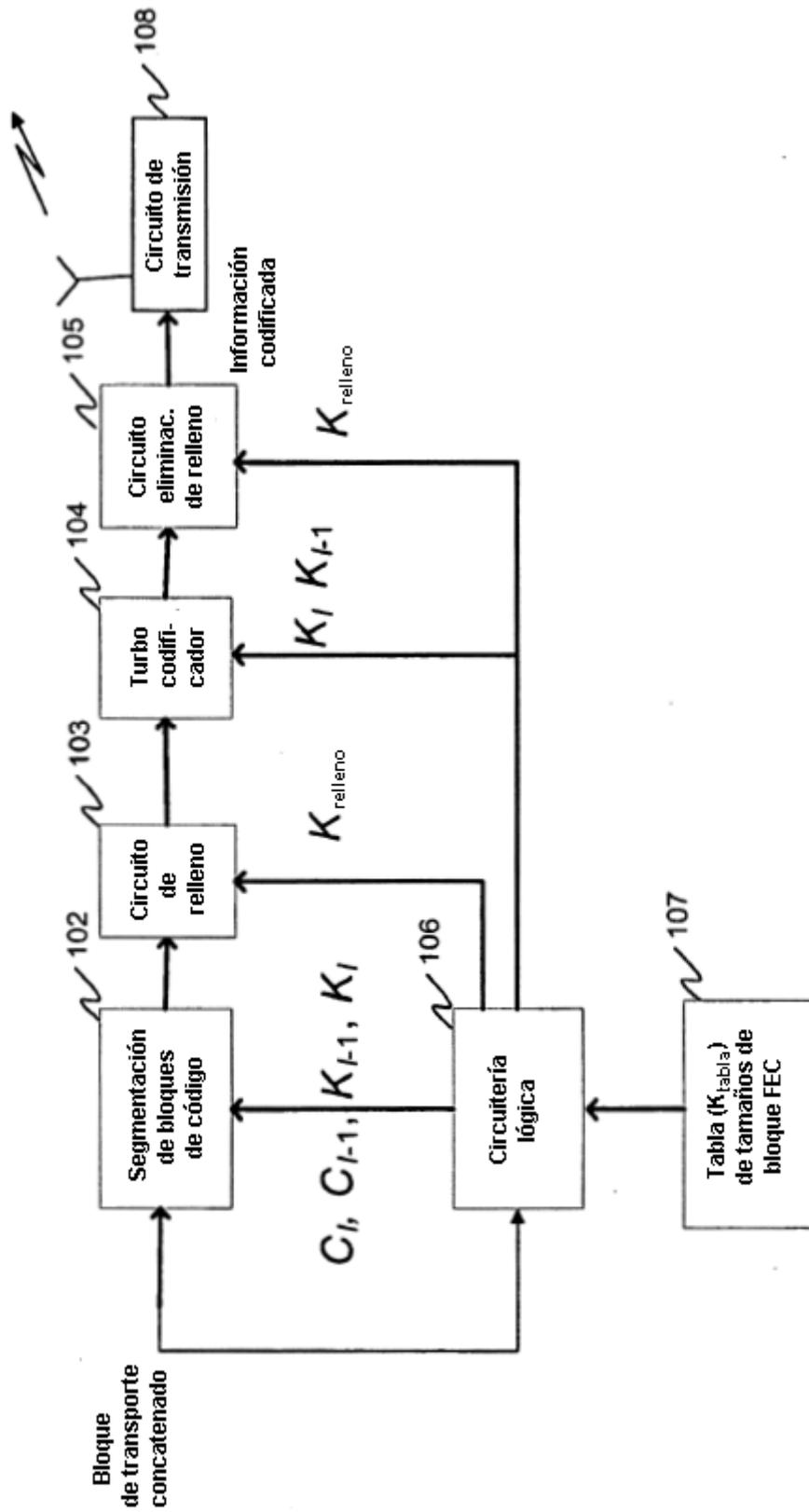


FIG. 1  
100

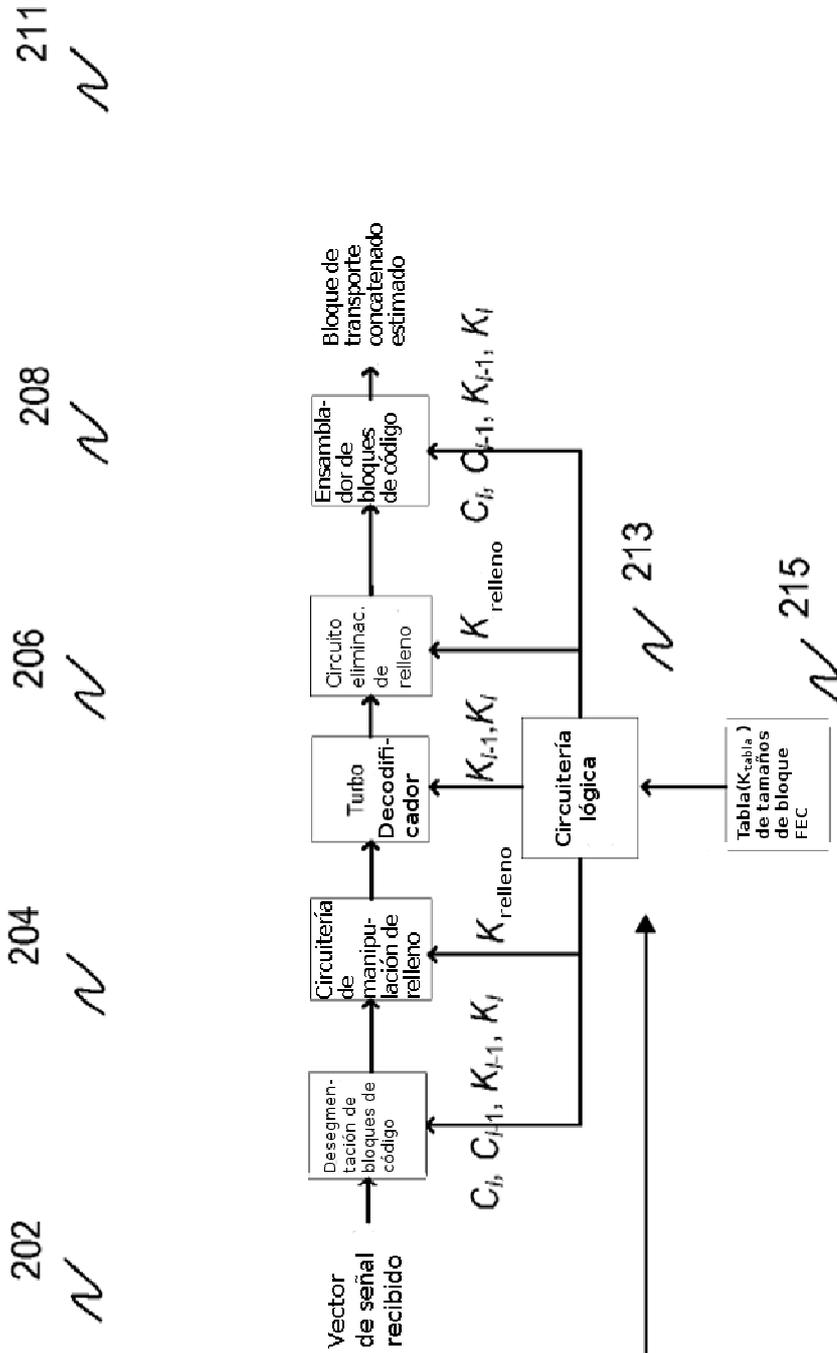


FIG. 2  
200

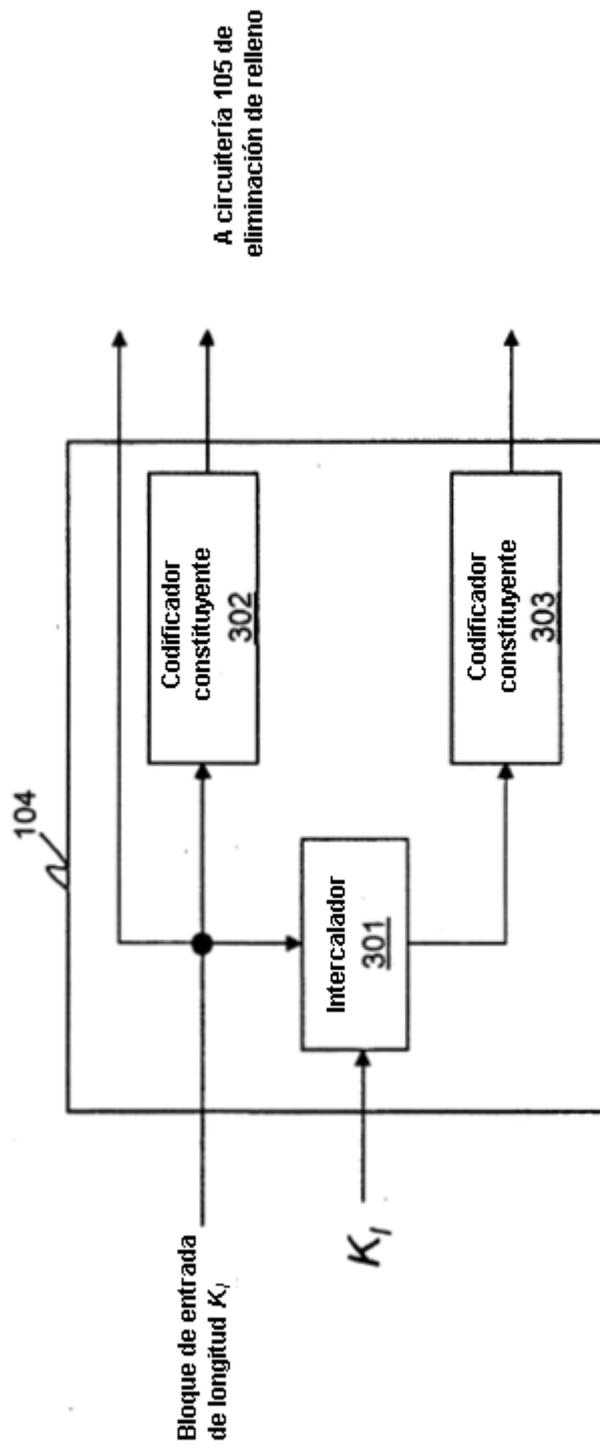


FIG. 3  
300

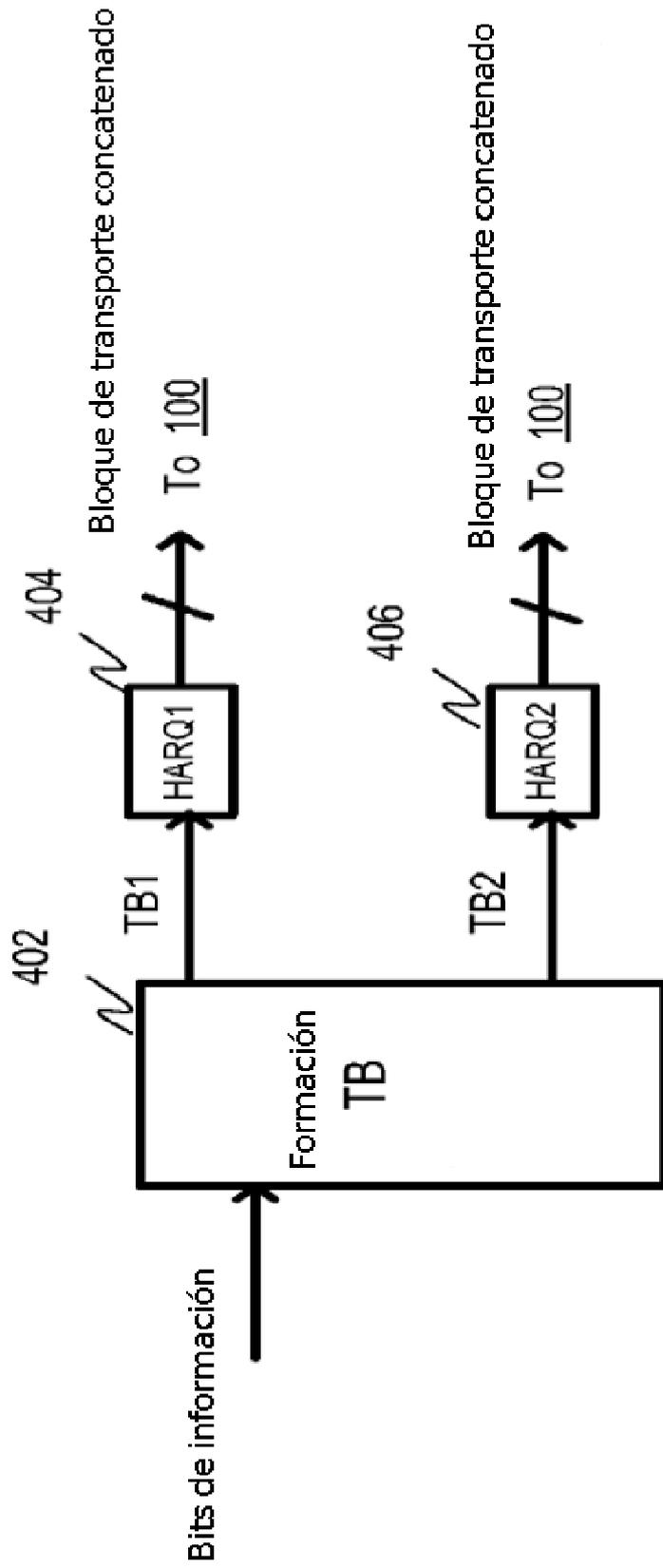


FIG. 4  
400

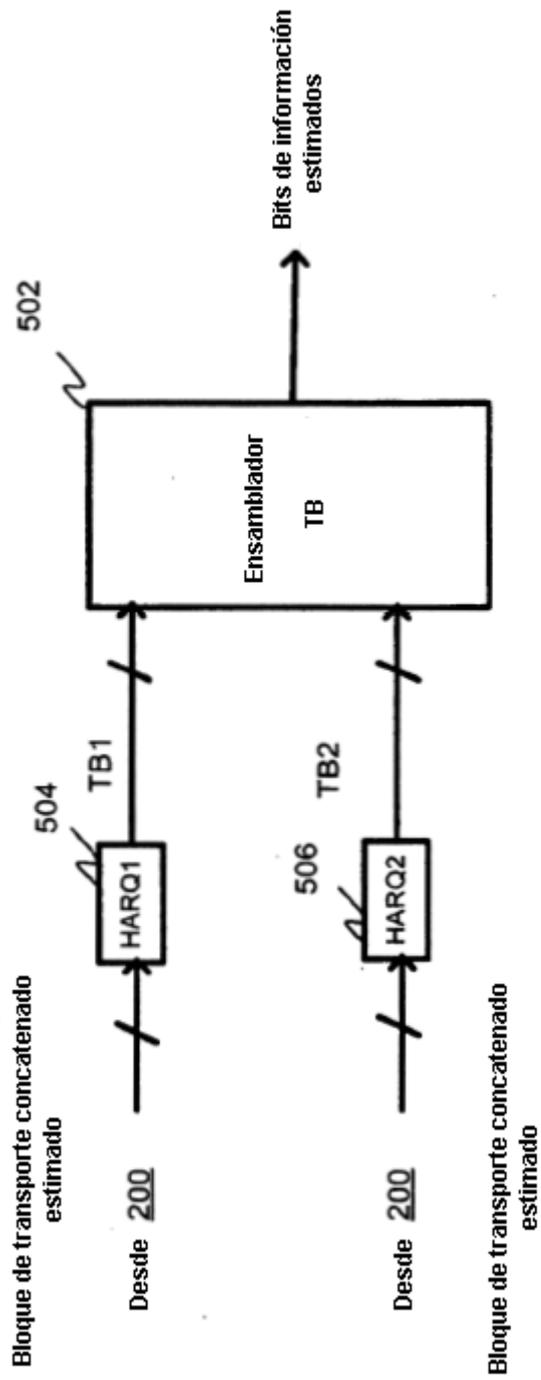


FIG. 5  
500

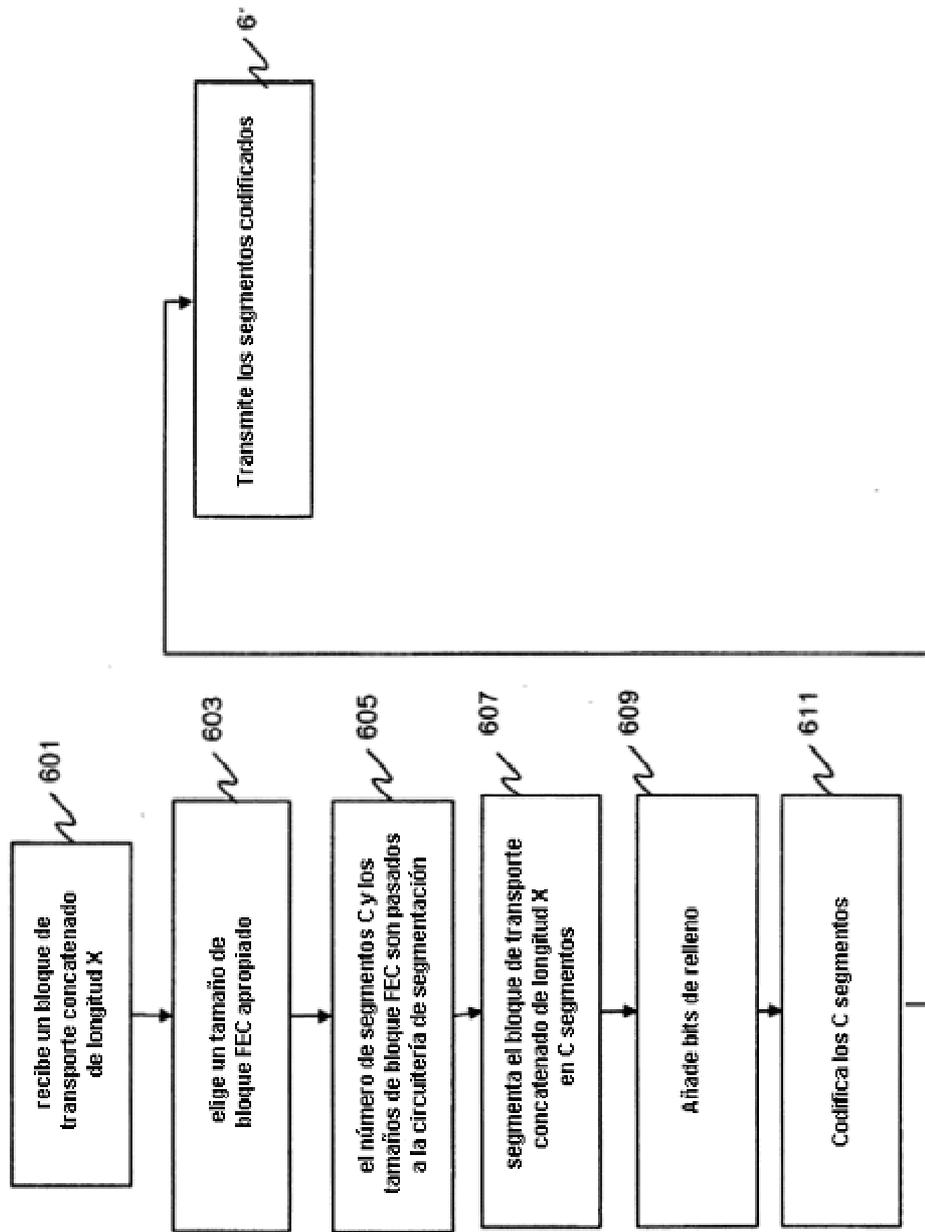


FIG. 6

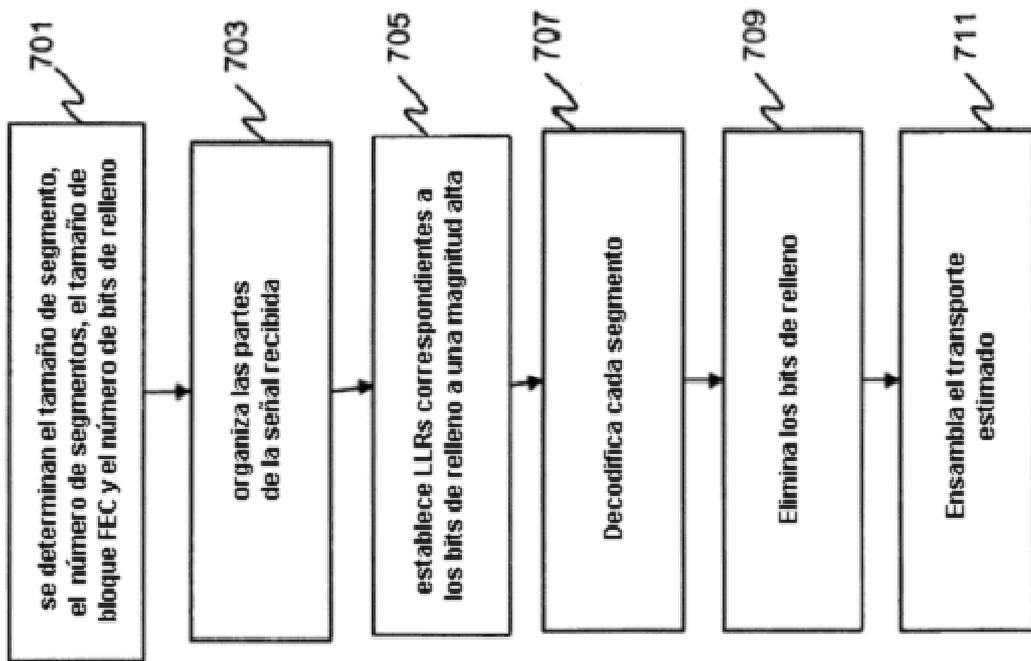


FIG. 7