(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 604 904**

(51) Int. CI.:

H04K 3/00 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(86) Fecha de presentación y número de la solicitud internacional: **13.12.2013 PCT/EP2013/076588**

(87) Fecha y número de publicación internacional: **26.06.2014 WO14095653**

(96) Fecha de presentación y número de la solicitud europea: **13.12.2013 E 13803075 (4)**

(97) Fecha y número de publicación de la concesión europea: **31.08.2016 EP 2936714**

(54) Título: **Procedimiento de perturbación de un sistema de comunicación mediante la inserción de motivos ficticios en un flujo de datos a emitir**

(30) Prioridad:

19.12.2012 FR 1203475

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
09.03.2017

(73) Titular/es:

THALES (100.0%)
Tour Carpe Diem, Place des Corolles, Esplanade Nord
92400 Courbevoie, FR

(72) Inventor/es:

DELAVEAU, FRANÇOIS;
SIRVEN, FRANÇOIS;
VIRAVAU, PHILIPPE y
MALLIER, SÉBASTIEN

(74) Agente/Representante:

CARPINTERO LÓPEZ, Mario

DESCRIPCIÓN

Procedimiento de perturbación de un sistema de comunicación mediante la inserción de motivos ficticios en un flujo de datos a emitir

La invención se refiere al campo de la perturbación en el que un objetivo consiste en neutralizar un sistema de comunicación perturbando la señal emitida por un emisor del sistema y destinado a un receptor del sistema.

Los sistemas de comunicación utilizan, durante la generación de la señal que hay que emitir, unas secuencias particulares, insertadas en la señal, que se utilizan para sincronizar los equipos emisor y receptor entre sí. Estas secuencias de sincronización se fijan de una sola vez mediante el protocolo de comunicación o el estándar de implementación del sistema y se insertan en la señal que hay que emitir con las secuencias de información moduladas.

La invención trata sobre un procedimiento de perturbación de un sistema de comunicación que busca transmitir una señal que cumple con el mismo estándar que el de dicho sistema pero que consta de varios motivos de sincronización ficticios insertados con la finalidad de hacer más difícil, para un receptor, la sincronización con un emisor.

La invención también pretende transformar un emisor estándar del sistema de comunicación que hay que aleatorizar en un aleatorizador sin modificar la cadena de emisión de este equipo. El o los motivo(s) de sincronización ficticio(s) se introducen en la señal emitida generando directamente en la secuencia de datos binarios que hay que transmitir una secuencia binaria adaptada. De este modo, la invención no necesita ninguna modificación del equipo emisor ya que interviene aguas arriba de la cadena de emisión.

El problema técnico considerado por la presente invención consiste en neutralizar un sistema de comunicaciones a partir de uno o varios terminales de dicho sistema transformado(s) en aleatorizador (aleatorizadores). Al reutilizar uno o varios terminales del sistema de comunicación que se desea neutralizar, se obtiene uno o varios aleatorizadores de bajo coste y que pueden adaptarse a una gran diversidad de redes de comunicación estandarizadas. De este modo, la invención pretende concebir una solución de perturbación que no necesita ningún equipo aleatorizador específico para este único fin.

Los documentos FR 2 858 742, US 8 055 184 y US 2010/302956 describen unos sistemas de perturbación de la sincronización.

Los sistemas de perturbación conocidos se basan, por lo general, en unos equipos aleatorizadores específicos que presentan los siguientes inconvenientes.

El consumo de energía en un aleatorizador es, por lo general, muy elevado ya que la forma de onda de perturbación utiliza no está optimizada para neutralizar de forma eficaz un sistema de comunicación.

Para limitar el consumo energético y para garantizar la eficacia de la neutralización, la forma de onda de perturbación debe, por el contrario, ser también lo más coherente posible con las señales de las que desea perturbar la recepción.

Por otra parte, las formas de onda de perturbación habituales conocidas tienen, por lo general, bien una combinatoria o una complejidad bajas y por ello son fácilmente detectables (es el caso, en particular, de las formas de onda denominadas de perturbación de bloqueo, que vuelven a estos equipos muy poco discretos), o bien muy complejas, cuando el aleatorizador pretende, por ejemplo, implementar la forma de onda exacta del sistema de telecomunicación que hay que neutralizar para realizar unos efectos de enmascaramiento o de saturación de los accesos.

Además, numerosos sistemas de perturbación conocidos padecen la mayoría de las veces una ausencia de escalabilidad, de adaptabilidad o de flexibilidad principalmente ligada a la elección de las arquitecturas de hardware y software.

Por último, los sistemas de perturbación conocidos que presentan un alcance significativo tienen también un coste, unas dimensiones y un consumo energético elevados.

La invención pretende resolver los problemas ya citados anteriormente y suprimir las limitaciones de las soluciones de la técnica anterior ofreciendo un procedimiento de perturbación de un sistema de comunicaciones a partir de una modificación de los datos útiles producidos en la entrada de un terminal de dicho sistema. La invención consiste en particular en realizar una codificación específica de los datos útiles emitidos por un terminal compatible del sistema que hay que neutralizar de modo que se genere de forma indirecta en la señal emitida finalmente uno o varios motivos de sincronización ficticios en las características estacionarias, que sean fáciles de interpretar por los receptores del sistema de comunicación considerado, y cuya recurrencia y situación en la trama emitida se seleccionan para optimizar los efectos de enmascaramiento y de saturación de la cadena de recepción considerada. De esta forma, un receptor del sistema que hay que neutralizar ya no puede sincronizarse correctamente. La

implementación del procedimiento según la invención al nivel de los datos útiles que hay que emitir y no al nivel de la cadena de emisión permite ofrecer una solución de bajo coste que no necesita modificar el equipo emisor para transformarlo en aleatorizador. Además, la utilización de varios terminales modificados permite constituir una red de aleatorizadores cooperativos cuyos rendimientos de neutralización se ven incrementados con respecto a la utilización de un único terminal modificado. En efecto, la baja potencia de emisión de un terminal se ve compensada por un enmallado del espacio que hay que cubrir y por la utilización de varios terminales que emiten unas señales de perturbación simultáneas. Cuando el procedimiento según la invención se aplica a una multitud de emisores, la invención permite reproducir, con dichos emisores, la topología del sistema de comunicación que hay que aleatorizar. La invención también permite aumentar la cobertura global en un entorno de propagación desfavorable para la red de emisores que implementan la invención. La invención también permite incrementar los efectos de enmascaramiento así como los efectos de saturación de los accesos. De manera más general, la invención permite inducir unos efectos que neutralizan el sistema de comunicación que hay que aleatorizar pero que son difíciles de diagnosticar o de interpretar, en la medida en que estos efectos reproducen casos encontrados en ingeniería de redes de radiocomunicación, difíciles o patológicos, pero no excepcionales.

La extensión de la presente invención está limitada por las reivindicaciones adjuntas.

Se mostrarán mejor otras características y ventajas de la presente invención con la lectura de la descripción que viene a continuación en relación con los dibujos adjuntos, que representan:

- La figura 1a, un esquema que ilustra la sincronización entre un receptor y un emisor compatibles de un mismo sistema de telecomunicación.
- La figura 1b, un esquema que ilustra el efecto obtenido mediante la aplicación del procedimiento de perturbación según la invención.
- La figura 2, un esquema que ilustra la transformación operada por la cadena de emisión de un terminal emisor para convertir los datos útiles que hay que emitir en símbolos modulados listos para emitirse por vía radio.
- La figura 3, un esquema que ilustra la generación, según la invención, de unos bits ficticios en el interior de los datos útiles en la entrada de la cadena de emisión de un terminal emisor.
- La figura 4, un diagrama de bloques de las diferentes funciones sucesivamente implementadas por un terminal emisor.
- La figura 5, un esquema de los registros de desplazamiento de un código convolucional de rendimiento ½.
- La figura 6, una representación, para el ejemplo de código convolucional asociado a la figura 5, de la matriz generadora de dicho código.
- La figura 7, una ilustración de la condición necesaria y suficiente para imponer, en la salida del código convolucional definido en las figuras 5 y 6, los valores de una secuencia de bits consecutivos.
- La figura 8, una ilustración de las relaciones de paridad de un código punzonado.

A continuación en la descripción, la expresión “datos útiles”, “bits útiles”, “información útil” se emplea para designar los datos binarios que hay que transmitir entre la aplicación ejecutada por un emisor y la aplicación correspondiente ejecutada por un receptor por oposición a los datos binarios presentes en las tramas transmitidas pero que no están destinados a la aplicación ejecutada por el receptor sino que se emplean con una finalidad de señalización, sincronización o cualquier otra función necesaria para el correcto funcionamiento del sistema de comunicación.

La figura 1a ilustra, en dos esquemas, el principio de la sincronización entre un receptor y un emisor compatibles de un mismo sistema de telecomunicación.

En la parte superior de la figura 1a se representa un sistema de comunicación inalámbrico en forma de un emisor EM que comunica con un receptor REC por onda de radio. La transformación de los datos binarios que hay que emitir en señal S de radio puede especificarse mediante un estándar o una norma de telecomunicaciones. Esta especificación define, en particular, la inserción, en el interior de la señal que hay que emitir, de secuencias SINC de sincronización. Dichas secuencias están constituidas por símbolos conocidos de los equipos del sistema y posicionadas de forma periódica o según un motivo temporal también conocido a la vez del emisor EM y del receptor REC los cuales implementan el mismo estándar de telecomunicaciones.

En la parte inferior de la figura 1a se representa, en un gráfico amplitud/tiempo, el resultado 10 obtenido al efectuar una correlación de la secuencia de sincronización conocida con la señal recibida. El pico de amplitud del resultado de correlación da una estimación de la posición temporal de la secuencia de sincronización en la señal recibida. Al efectuar dicha operación de correlación, el receptor REC puede sincronizarse temporalmente con el emisor EM detectando, por ejemplo, el inicio, la parte central o el final de una trama, indicado por la presencia de dicha secuencia, en el interior de la señal emitida. Un objetivo de la invención es neutralizar el sistema EM, REC de comunicaciones perturbando la sincronización de los equipos emisores y receptores.

La figura 1b ilustra, en dos esquemas, el procedimiento de perturbación según la invención.

En la parte superior de la figura 1b se representa un conjunto de emisores EM_B1, EM_B2,…, EM_BN (siendo N = 3 en el ejemplo ilustrativo pero no limitativo de la figura 1b, identificándose los emisores como EM_B1, EM_B2, EM_B3), compatibles del mismo estándar de telecomunicaciones que el emisor EM y el receptor REC que se desea

neutralizar. Estos emisores EM_B1, EM_B2,…, EM_BN (siendo todavía N = 3 en el ejemplo de la figura 1b) se modifican, según la invención, con el fin de generar una señal S_I de perturbación destinada al receptor REC. Estos constituyen un sistema de perturbación cooperativo también llamado red de aleatorizadores cooperativos. La señal S_I de perturbación es del mismo tipo que la señal S transmitida por un emisor EM del sistema de comunicaciones que hay que neutralizar con la sola diferencia de que consta al menos de una secuencia $SINC_F$ de sincronización ficticia. En la figura 1b se representa la señal S_I de perturbación constituida por la superposición de las señales de perturbación respectivamente emitidas por N emisores adaptados EM_B1, EM_B2,…, EM_BN. Cada una de las N señales de perturbación consta al menos de una secuencia de sincronización ficticia en las características estacionarias $SINC_{F1}$, $SINC_{F2}$,…, $SINC_{FN}$, situadas en la trama según unas recurrencias adaptadas a la ventana temporal de los receptores de destino. Estas secuencias son, por lo tanto, a la vez plausibles para el receptor de destino y suficientemente frecuentes para estar regularmente presentes en las ventanas de tratamiento de dichos receptores. De manera ventajosa, las secuencias ficticias pueden constar de los mismos símbolos que la secuencia SINC de sincronización real, pero transmitirse en instantes temporales y/o en frecuencias diferentes, y con unas recurrencias de trama múltiplos de la del motivo SINC de sincronización real. En una variante de realización, cada una de las N señales de perturbación procedentes de los emisores EM_B1, EM_B2,…, EM_BN consta de las N secuencias $SINC_{F1}$, $SINC_{F2}$,…, $SINC_{FN}$ de sincronización ficticias posicionadas en unos instantes temporales diferentes en la trama emitida. Esta variante presenta la ventaja de aumentar el rendimiento global de la señal de perturbación frente a unas redes de comunicaciones que utilizan una combinatoria no trivial de secuencias SINC de sincronizaciones.

En otra variante de realización, cada uno de los N emisores EM_B1, EM_B2,…, EM_BN de perturbación emite respectivamente un número significativo de ejemplares de cada una de las secuencias de sincronización ficticias (respectivamente k_1 secuencias $SINC_{F1}$, k_2 secuencias $SINC_{F2}$,…, k_N secuencias $SINC_{FN}$) posicionadas en instantes temporales diferentes en cada trama emitida. Esta variante presenta aquí también la ventaja de aumentar el rendimiento global de la señal de perturbación frente a unos receptores de destino cuya sincronización precisa no la conocen los aleatorizadores EM_B1, EM_B2,…, EM_BN.

La transmisión de señales de perturbación que comprenden unas secuencias de sincronización ficticias posicionadas en unas ubicaciones aleatorias en las tramas emitidas tiene como efecto hacer más difícil, e incluso imposible, la sincronización del receptor REC que no sabría discriminar el motivo de sincronización real de los motivos de sincronización ficticios.

Este efecto se ilustra en el gráfico representado en la parte inferior de la figura 1b que da el resultado de la operación de correlación, efectuada por el receptor REC, en presencia de las señales de perturbación emitidas. El pico de amplitud del resultado de correlación 10 que corresponde a la secuencia SINC de sincronización real está muy contaminado por una multitud de picos de amplitudes de resultados de correlación 11, 12, 13, 14 que corresponden a la presencia de secuencias de sincronización ficticias.

Uno de los objetivos de la invención es permitir la inserción de motivos de sincronización ficticios en la señal emitida por un terminal del sistema de comunicaciones que hay que aleatorizar sin modificar los elementos de la cadena de emisión del emisor sino al contrario actuando únicamente sobre los datos binarios útiles en la entrada de la cadena de emisión.

La figura 2 ilustra de forma esquemática la transformación experimentada por una secuencia de datos Du binarios útiles que hay que transmitir para obtener una secuencia S_T de símbolos modulados, listos para emitirse a través de una señal de radio. La transformación ejecutada corresponde a la función F de transferencia de la cadena de transmisión del emisor. La secuencia S_T de símbolos modulados está constituida, por una parte, por símbolos S_U útiles procedentes de la transformación de los datos binarios Du útiles y, por otra parte, por al menos una secuencia SINC de sincronización o por una secuencia equivalente compuesta por símbolos conocidos de todos los equipos del sistema de comunicación.

La figura 3 ilustra la implementación del procedimiento según la invención, en un terminal compatible del sistema de comunicación que hay que aleatorizar o en cualquier otro tipo de emisor capaz de producir unas señales compatibles del sistema de comunicación que hay que aleatorizar.

En una primera etapa, las secuencias $SINC_F$ de sincronización ficticias generadas se posicionan en una trama T_F ficticia vacía del mismo tamaño que una secuencia S_T de símbolos modulados real, en las posiciones temporales seleccionadas, diferentes de la posición temporal de una secuencia de sincronización real.

En una segunda etapa, se estiman el valor y la posición de los bits B_F ficticios que hay que insertar en el interior de la secuencia de datos que hay que emitir en la entrada de la cadena de emisión de modo que se obtenga, en la salida de la cadena de emisión, el valor y la posición temporal predefinidos de los símbolos de dichas secuencias $SINC_F$ ficticias. Esta operación puede realizarse calculando la función F^{-1} de transferencia inversa de la función F de transferencia implementada por la cadena de transmisión y a continuación aplicando la función F^{-1} de transferencia inversa en la trama T_F ficticia para obtener una trama D_F modulada que comprende los bits B_F ficticios.

Dichos bits B_F ficticios se insertan a continuación en la secuencia de datos Du real que hay que transmitir en la entrada de la cadena de transmisión punzonando las posiciones relativas de los bits ficticios de la trama D_F ficticia en una secuencia de datos real. La secuencia de símbolos modulados contenidos en la salida de la cadena de transmisión comprende a la vez los símbolos S_U de datos útiles, el motivo SINC de sincronización real y los motivos $SINC_F$ de sincronización ficticios.

Los bits ficticios pueden introducirse directamente en el tren de datos útiles que hay que transmitir y hacerlo sin modificar ni efectuar ninguna intrusión en la cadena de transmisión del equipo emisor.

Es posible, por ejemplo, utilizar unas formas de transmisión de datos en el contenido útil completamente programable (por ejemplo unos servicios de mensajería) y construir mensajes completamente deterministas que contienen los bits ficticios que producen en la salida los motivos $SINC_F$ ficticios buscados.

También es posible utilizar unos modos de transmisión que corresponden a unas aplicaciones, intervenir por ejemplo sobre un flujo útil proveniente de un codificador de origen de audio o de vídeo. Para ello, se interceptan, por ejemplo, los datos binarios de aplicación antes de su entrada en la cadena de transmisión al nivel de la capa física de un módem. Esta interceptación puede hacerse también al nivel de una capa intermedia, por ejemplo al nivel de la capa de red. En ambos casos debe tenerse en cuenta, según los procedimientos descritos más adelante, los bits de información útil cercanos que son aleatorios y variables.

La figura 4 representa un diagrama de bloques de las diferentes funciones sucesivamente implementadas por un emisor de un sistema de comunicación para emitir una señal que contiene unos datos que hay que transmitir. Se representan las principales funciones tradicionalmente implementadas, entendiéndose que el esquema de la figura 4 se da a título ilustrativo y no limitativo. En particular, pueden omitirse algunas funciones y puede modificarse el orden de algunas funciones. La función F de transferencia de la cadena de emisión es igual a la composición de las funciones de transferencia de cada bloque funcional independiente de la cadena, entendiéndose que los bloques están conectados en serie. La función F^{-1} de transferencia inversa es, cuando existe, igual a la composición, en el orden inverso, de las funciones de transferencia inversas de cada bloque. Dicho de otro modo, si f_1, f_2,…, f_N son las funciones de transferencia de cada bloque funcional de la cadena, entonces la función F de transferencia global es igual a $F= f_1 \text{ o } f_{2O...O} f_N$ y la función F^{-1} de transferencia inversa es igual a $F^{-1} = f_N^{-1} \text{ o } f_{N^{-1}\ O...O}\ f_1^{-1}$.

Para estimar la función F^{-1} de transferencia inversa global conviene, por lo tanto, determinar la función de transferencia inversa de cada bloque unitario.

La función F de transferencia directa de la cadena de emisión puede conocerse cuando la invención la implementa el diseñador del sistema de comunicaciones o cuando dicho sistema cumple con un estándar conocido. También puede estimarse probando el equipo emisor, por ejemplo inyectando unas señales de prueba en su entrada y analizando las señales obtenidas en la salida.

Las transformaciones aplicadas en la cadena de emisión en el tren binario son por lo general reversibles, es decir que es posible a partir del tren binario en la salida encontrar el tren binario de entrada.

Sin embargo, algunas funciones implementadas por la cadena de emisión de un sistema de comunicaciones pueden no ser siempre sobreyectivas. Dicho de otro modo, puede ocurrir que un tren TBC binario codificado, del que se querrían forzar los valores, en la salida de un módulo de la cadena de emisión no corresponda a ninguna serie TBU de bits útiles en la entrada de dicho módulo. Por ejemplo, las operaciones de codificación de canal o de entramado transforman un tren binario útil de longitud Lu en un tren binario codificado de longitud Lc. A causa de las operaciones de entramado, necesarias para garantizar la sincronización del receptor, y de codificación correctora de error, necesarias para compensar el efecto del canal de propagación, se tiene en la práctica siempre Lc > Lu. Esto significa que entre las 2^{Lc} secuencias de Lc bits codificados, solo pueden obtenerse 2^{Lu} secuencias mediante codificación. La codificación nunca es por lo tanto sobreyectiva.

En dicho caso, no es posible determinar la función F^{-1} de transferencia inversa de la cadena de emisión global, sino solo la F^{-1} inversa de la cadena de emisión en la imagen restringida F({TBU}) del conjunto {TBU} de los trenes de bits utilizados en la entrada de la cadena de emisión.

Se busca, por lo tanto, determinar en qué medida es posible forzar el valor de algunos de los bits de la señal codificada. En particular, se busca determinar el número de bits cuyo valor puede imponerse y en qué medida es posible seleccionar no solo el valor sino también la posición de estos bits. En el caso de un codificador de canal, se busca imponer el valor de una serie de bits codificados consecutivos de manera que se obtengan unos motivos codificados que se asemejan a unos motivos de sincronización.

La implementación práctica consiste en analizar de forma sucesiva las diferentes transformaciones del tren binario comenzando por la transformación que interviene al final en la cadena de emisión. Para cada transformación, se determinan las entradas que deben aplicarse para obtener en la salida el tren binario codificado esperado.

En este sentido, se analizan caso por caso a continuación en la descripción las transformaciones elementales del tren binario y su invertibilidad.

La cadena 400 de emisión representada en la figura 4 consta de una aplicación 401 capaz de generar o transformar una secuencia de datos binarios que hay que emitir. Los datos que hay que emitir pueden ser unos datos de texto, de audio, de vídeo o cualquier otra información. La aplicación 401 también puede constar de una función de codificación de origen, por ejemplo un codificador de audio, de imagen o de vídeo capaz de suprimir o reducir la redundancia de información o de reducir el ruido que afecta a la secuencia. La aplicación 401 genera en la salida una secuencia T binaria útil para transmitir. La invención se implementa de manera ventajosa en la salida de la aplicación 401 modificando la secuencia T binaria útil para insertar unos bits ficticios de modo que se obtenga en la salida de la cadena de emisión una secuencia F(T)(t) de símbolos modulados que hay que emitir que comprenden al menos un motivo de sincronización ficticia.

La cadena 400 de emisión también puede constar de un módulo 402 de codificación correctora.

El objetivo de una función de codificación correctora es transformar la secuencia binaria de datos útiles recibida en la salida de la aplicación 401 en una secuencia binaria protegida de modo que el impacto de los errores causados por el canal de transmisión sea el menor posible. Para hacer la secuencia binaria de datos útiles más robusta frente a las imperfecciones del canal de transmisión, la función de codificación correctora añade redundancia a esta secuencia binaria.

La determinación de la función de transferencia inversa de un módulo de codificación correctora es equivalente a la búsqueda de la secuencia binaria que hay que producir en la entrada del codificador de corrección para, en la salida, obtener una secuencia codificada en la que se imponen el valor y la posición de un número predeterminado de bits.

Existen diferentes tipos de códigos correctores entre los cuales están los códigos lineales en bloques, los códigos convolucionales o también los turbo-códigos y los códigos de baja densidad LDPC (siglas del inglés *Low Density Parity Check*).

Por consiguiente, la determinación de la función de transferencia inversa de un codificador de corrección se describe para diferentes tipos de códigos correctores, los códigos lineales en bloques, los códigos convolucionales así como los turbo-códigos y los códigos LDPC.

Códigos lineales en bloques

Un código corrector lineal en bloques de rendimiento k/n transforma una secuencia binaria que comprende k símbolos en una secuencia binaria protegida que comprende n símbolos siendo n estrictamente superior a k. Dicho código introduce, por lo tanto, n-k símbolos de redundancia. Los símbolos pueden ser unos bits o estar constituidos por varios bits concatenados. La operación de codificación en bloques es una transformación bi-unívoca de una palabra del mensaje $i = (i_0, \ldots, i_{k-1})$ en una palabra de código $c = (c_0, \ldots, c_{n-1})$ definida por el siguiente sistema de ecuaciones lineales (donde "+" designa la suma módulo 2, "." designa la multiplicación módulo 2) y g_{en} son unos coeficientes de valor en el cuerpo de Galois GF(2), ordenados en una matriz de tamaño nxk:

$$c_n = \sum_{e=0}^{k} g_{en} . i_e \ , \text{ para } \ 0 \leq i \leq n\text{-}1$$

Entre las 2^n secuencias binarias que comprenden n bits que existen, solo pueden generarse 2^k. La operación de codificación correctora limita, por la tanto, la posibilidad de generar cualquier secuencia binaria requerida.

La codificación en bloques consiste en efectuar el producto de un vector de información en la entrada de k bits por una matriz binaria, de rango completo, de tamaño k*n, llamada matriz generadora, para obtener un vector codificado de n bits. A menudo el código se dice sistemático a la izquierda, respectivamente a la derecha, cuando los k primeros, respectivamente los k últimos, bits del vector codificado de n bits corresponden a los k bits del vector de información en la entrada. La operación de codificación se puede ilustrar mediante la siguiente relación, en la que $i_0, \ldots i_{k-1}$ son los bits de la secuencia útil en la entrada, $c_0, \ldots c_{n-1}$ son los bits de la secuencia codificada y $m_{i,j}$ son los coeficientes de la matriz generadora del código.

$$\begin{bmatrix} c_0 & \cdots & c_{n-1} \end{bmatrix} = \begin{bmatrix} i_0 & \cdots & i_{k-1} \end{bmatrix} \cdot \begin{bmatrix} m_{0,0} & m_{0,1} & m_{0,2} & \cdots & m_{0,n-1} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ m_{k-1,0} & m_{k-1,1} & m_{k-1,2} & \cdots & m_{k-1,n-1} \end{bmatrix}$$

Caso de códigos en bloques sistemáticos

En el caso de que el código sea sistemático a la izquierda, la secuencia codificada se escribe $[i_0 \ldots i_{k-1} c_k \ldots c_{n-1}]$. La transformada inversa de la operación de codificación consiste en analizar la palabra recibida para determinar si se trata de una palabra de código posible. Si no se trata de una palabra de código posible, es preciso sustituirla por la

palabra de código que se encuentra a distancia mínima de la palabra de código recibida. A continuación, como el código es sistemático, la información se obtiene suprimiendo los n-k últimos bits de la palabra. Dicho de otro modo, es posible imponer, mediante la elección de la secuencia de entrada, el valor en la salida de los k primeros bits de la palabra codificada. Los valores de los n-k bits restantes se deducen, por tanto, de los valores seleccionados para los k bits que se han forzado. Sucede exactamente lo mismo para un código en bloque sistemático a la derecha.

Caso de codificación en bloque procedente de códigos cíclicos

Los códigos en bloque utilizados habitualmente son unos códigos en bloque cíclicos o bien se derivan de códigos en bloque cíclicos mediante punzonado o acortamiento.
En el caso de un código en bloque cíclico, si $[c_0 \ldots c_{n-1}]$ es una palabra de código, cualquier permuta circular de la palabra $[c_i\ c_{i+1} \ldots c_{n-1}\ c_0 \ldots\ c_{i-1}]$ es también una palabra de código.
Al escribir de forma polinomial las palabras de código $c(x) = c_0+c_1 \cdot x+c_2 \cdot x^2+ \ldots +c_{n-1} \cdot x^{n-1}$, todas las palabras de código aparecen como múltiplos de un mismo polinomio $g(x) = g_0+g_1 \cdot x+g_2 \cdot x^2+ \ldots +g_{n-k} \cdot x^{n-k}$, de grado n-k, llamado polinomio generador del código.
Una palabra de información $[i_0 \ldots i_{k-1}]$ también se puede escribir de forma polinomial $i(x) = i_0+i_1 \cdot x+i_2 \cdot x^2+ \ldots +i_{k-1} \cdot x^{k-1}$.
Siempre es posible escribir la operación de codificación de forma sistemática a la derecha. Esta operación de codificación consiste en calcular la división de $i(x) \cdot x^{n-k}$ por $g(x)$. El resto de la división es $v(x)$ (de grado inferior o igual a n-k-1) y el cociente de la división es $k(x)$. Por lo tanto, tenemos $i(x) \cdot x^{n-k} = k(x) \cdot g(x)+v(x)$, y $c(x) = i(x) \cdot x^{n-k}+v(x) = k(x) \cdot g(x)$.
Como $v(x)$ es de grado inferior o igual a n-k-1, los valores de los coeficientes de $c(x)$ para los grados superiores o iguales a n-k son los coeficientes de $i(x)$ desplazados de n-k.

Por lo tanto, siempre es posible escribir los códigos en bloque cíclicos de forma sistemática a la derecha, y por lo tanto imponer el valor de los k últimos bits que son iguales a los bits de información. Como el código es cíclico y, por lo tanto, cualquier permuta circular de una palabra de código es también una palabra de código, esto significa que también es posible, siempre para los códigos cíclicos, imponer el valor de cualquier grupo de k bits consecutivos de una palabra de código.

Por otra parte, la dependencia entre los valores de los bits en la entrada del codificador y los bits en la salida del codificador es lineal. Los valores de los bits que deben forzarse en la entrada del codificador dependen de forma lineal de los valores de los demás bits en la entrada del codificador y de los valores de los bits forzados en la salida del codificador.

Caso general de los códigos en bloque

En el caso más general en el que el código no es sistemático ni procede de códigos cíclicos, una condición suficiente para poder imponer el valor de un grupo de d bits en la salida del codificador, siendo d inferior o igual a k, es que le conjunto de las posiciones p_1, $p_2, \ldots p_d$, de los bits en la secuencia codificada debe ser tal que la submatriz de la matriz generadora del código:

$$\begin{bmatrix} m_{0,p_1} & m_{0,p_2} & m_{0,p_3} & \cdots & m_{0,p_d} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ m_{k-1,p_1} & m_{k-1,p_2} & m_{k-1,p_3} & \cdots & m_{k-1,p_d} \end{bmatrix}$$

es de rango completo, es decir de rango igual a d.

En efecto, cuando se cumple esta condición, es posible determinar la secuencia $[i_0 \ \ldots \ i_{k-1}]$ en la entrada del codificador que permite fijar los valores de d bits o símbolos en la secuencia codificada resolviendo el siguiente sistema de ecuaciones:

$$[i_0 \quad \ldots \quad i_{k-1}] = [c_{p_1} \quad \ldots \quad c_{p_d}] \begin{bmatrix} m_{0,p_1} & m_{0,p_2} & m_{0,p_3} & \cdots & m_{0,p_d} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ m_{k-1,p_1} & m_{k-1,p_2} & m_{k-1,p_3} & \cdots & m_{k-1,p_d} \end{bmatrix}^{-1}$$

en el que $[C_{P1} \ldots C_{Pd}]$ son los bits o símbolos cuyo valor se fija en la secuencia codificada, designando los índices P_1, P_2, ... P_d las posiciones de los bits o símbolos en la secuencia de n bits o símbolos.

De este modo, para todos los códigos en bloque habituales, que son sistemáticos, o cíclicos, o se construyen a partir de códigos cíclicos, es decir para la mayoría de los códigos habituales, es posible imponer, mediante la elección de las entradas del codificador, cualquier grupo de k bits consecutivos entre los n bits del vector codificado. Por otra parte, en el caso de dos palabras de código sucesivas, al imponer los k últimos bits de la 1ª palabra de código y los k primeros bits de la 2ª palabra de código, es posible imponer el valor de un grupo de 2k bits sucesivos en una

secuencia binaria que comprende al menos dos palabras de códigos.

En el caso más general, la submatriz de la matriz generadora del código en bloque que corresponde a las posiciones de los bits o símbolos que hay que fijar es de rango completo. Sin embargo, en algunos casos, algunas submatrices de la matriz generadora pueden no ser de rango completo. Dicho caso se ilustra en un ejemplo no limitativo de un código de Hamming (7,4) cuya matriz generadora M(7,4) viene dado por:

$$M(7,4) = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

El polinomio generador de este código es $g(x) = 1+x+x^3$. Para este código es posible imponer los 4 últimos bits (m_0 a m_3) de la palabra de código ya que la codificación es sistemática a la derecha. Para obtener la palabra de código siguiente:

$[c_0\ c_1\ c_2\ m_0\ m_1\ m_2\ m_3]$, basta con codificar el vector de información $i(x) = [m_0\ m_1\ m_2\ m_3]$.

La operación de codificación se representa mediante la siguiente relación:

$c_{sis,derecha}(x) = i(x)\cdot x^{n-k} + v(x) = k_{sis,derecha}(x)\cdot g(x)$, en la que $k_{sis,derecha}(x)$ es el vector que hay que codificar y $c_{sis,derecha}(x)$ es la palabra de código obtenida.

También es posible imponer, por ejemplo, los 4 primeros bits de la palabra de código a los valores del vector de información $[m_0\ m_1\ m_2\ m_3]$. Para ello es preciso encontrar el vector de información que, una vez codificado, da la siguiente palabra de código:

$$k_{sis,izquierda}(x)\cdot g(x) = m_0 + m_1\cdot x + m_2\cdot x^2 + m_3\cdot x^3 + c'_4\cdot x^4 + c'_5\cdot x^5 + c'_6\cdot x^6 .$$

Para calcular este vector de información $k_{sis,izquierda}(x)$, se utiliza la propiedad por la que el código es invariable por permutación circular. Se pasa de una codificación sistemática a la derecha a una codificación sistemática a la izquierda mediante 4 permutas circulares hacia la derecha. Por lo tanto, la palabra de código $[c_0\ c_1\ c_2\ m_0\ m_1\ m_2\ m_3]$ se convierte en la palabra de código $[m_0\ m_1\ m_2\ m_3\ c_0\ c_1\ c_2]$ efectuando estas 4 permutas circulares. La palabra de código $[m_0\ m_1\ m_2\ m_3\ c_0\ c_1\ c_2]$ se obtiene codificando el vector de información $i'(x) = [m_3\ c_0\ c_1\ c_2]$ (ya que el código es sistemático a la derecha).

Por el contrario, hay que señalar que si se considera la segunda, la cuarta, la quinta y la sexta columna de la matriz generadora del código M(7,4), se obtiene la submatriz siguiente que no es de rango completo, siendo los coeficientes de su última línea iguales a 0:

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

En efecto, el rango de una matriz corresponde al número de columnas independientes de la matriz o de forma equivalente al número de líneas independientes de la matriz.
Por lo tanto, no es posible forzar los valores de estos 4 bits (el segundo, el cuarto, el quinto y el sexto) de la palabra codificada: si se fuerza el valor de 3 de estos bits, el valor del cuarto bit se deduce de los valores impuestos a los tres bits forzados.

Por lo tanto, no se puede, para cualquier código (n, k) de rendimiento k/n, forzar el valor de cualquier grupo de k bits. Por el contrario, para los códigos utilizados más habitualmente, es posible forzar el valor de cualquier grupo de $n-k$ bits consecutivos cuando la submatriz de la matriz generadora del código asociado a las posiciones de los bits que hay que fijar es de rango completo.

Se precisan aquí las dependencias entre los valores de los bits en la entrada del codificador y los valores de los bits en la salida del codificador.

- Si se desea imponer el valor de un número limitado l, inferior a k, de bits en la salida del codificador, basta con imponer el valor de l bits en la entrada del codificador.
- Por el contrario, el valor de estos bits depende no solo del valor del motivo generado en la salida del codificador sino también de los valores de los demás bits en la entrada del codificador) que se puede forzar, llegado el caso,

también en el marco de la implementación de la presente invención).

Retomando el ejemplo de la matriz de codificación:

$$M(7,4)=\begin{bmatrix}1&1&0&1&0&0&0\\0&1&1&0&1&0&0\\1&1&1&0&0&1&0\\1&0&1&0&0&0&1\end{bmatrix},$$

si se quiere forzar el valor de los 2 primeros bits codificados, c_0 y c_1, se puede seleccionar un vector de información $i(x) = [f_0\ f_1\ i_0\ i_1]$, en el que i_0 e i_1 son unos bits de información dejados libres y f_0 y f_1 unos bits forzados para obtener el motivo requerido. Los valores que hay que seleccionar para f_0 y f_1 con el fin de obtener los valores requeridos de c_0 y c_1 vienen dados por las relaciones:

$$f_0=c_0+i_1+i_2$$

$$f_1=c_0+c_1+i_2$$

Esto ilustra que los valores de los bits forzados en la entrada del codificador (f_0 y f_1) dependen de forma lineal de los valores de los bits del motivo en la salida del codificador (c_0 y c_1) y de los valores de los demás bits en la entrada del codificador.

Códigos convolucionales

Los códigos convolucionales constituyen la segunda gran familia de códigos correctores de errores. Mientras que los códigos lineales en bloques permiten cortar el mensaje en bloques de k símbolos, los códigos convolucionales aplican una ventana deslizante de $k^*(m+1)$ símbolos en el mensaje y producen una secuencia continua de símbolos codificados. En general, los símbolos son binarios (p. ej. de valor 0 o 1 en el cuerpo de Galois GF(2), "+" designa la suma en módulo 2 y "." designa la multiplicación en módulo 2). La mayoría de las veces los códigos convolucionales tienen como parámetro $k = 1$ y el rendimiento del código es, por lo tanto, de forma $1/n$. Si a_j es un símbolo de información, los símbolos de paridad b_j asociados a a_j se definen mediante la relación de convolución siguiente, en la que $g_{e,j,i}$ son los coeficientes de n^*k polinomios de grado m de coeficientes y valores en el cuerpo de Galois GF(2) (el código viene completamente definido por el conjunto de los coeficientes $g_{e,j,i}$, e = 0,..., m, j = 0...k-1, i = 0...n-1):

$$b_{p\cdot n+i}=\sum_{e=0}^{m}\sum_{j=0}^{k-1} g_{e,j,i}\cdot a_{p\cdot k+j-e\cdot k},\ \text{para i}\ =0...n-1,\forall p$$

Los n símbolos en la salida del codificador dependen linealmente de los $k^*(m+1)$ últimos símbolos en la entrada del codificador.

A partir de un código de rendimiento 1/n, se pueden construir unos códigos denominados "derivados", que corresponden a $k>1$, mediante punzonado (la mayoría de las veces $k = n-1$ después del punzonado).

De manera más general, cuando el rendimiento del código es igual a k/n, la codificación convolucional es una codificación periódica de periodo k bits en la señal binaria en la entrada. Para cada nuevo grupo de k bits, se calculan n bits codificados. Los n bits codificados son unas combinaciones de bits relativas a los (m+1) últimos grupos de k bits. m es la longitud de limitación del código.

Se ilustra a continuación, en un ejemplo no limitativo, la etapa del procedimiento según la invención que consiste en invertir la función de transferencia de un código convolucional, dicho de otro modo en determinar la secuencia de bits que hay que producir en la entrada para obtener en la salida una secuencia codificada en la que se fijan el valor y la posición de un número predeterminado de bits.

Ejemplo no limitativo de un código convolucional de rendimiento ½

Se considera un código binario convolucional habitual, cuyos registros se representan en la figura 5, de rendimiento 1/2 y de longitud m+1 = 7 definido por dos polinomios de grado 6 definidos en notación octal por (171, 133). Estos dos polinomios se escriben $G_1(X) = 1+X+X^2+X^3+X^6$ y $G_2(X) = 1+X^2+X^3+X^5+X^6$ y corresponden a las relaciones de recurrencia $b_{2n} = a_n+a_{n-1}+a_{n-2}+a_{n-3}+a_{n-6}$ y $b_{2n+1} = a_n+a_{n-2}+a_{n-3}+a_{n-5}+a_{n-6}$ en el cuerpo de Galois GF2 ("+" designa la suma en módulo 2). Los polinomios G_1 y G_2 se aplican a los bits de entrada para formar respectivamente los bits de salida de índice par y los bits de salida de índica impar entrelazados a continuación de dos en dos en forma de $b_{2n}b_{2n+1}$ para formar un flujo binario de tamaño igual a un múltiplo de 2. Se ilustra a continuación la posibilidad de seleccionar el tren binario en la entrada de manera que se generen los motivos requeridos después de la codificación.

Para este código, en cada periodo, para un bit producido en la entrada del codificador, se generan dos bits en la salida. Según el estado de los registros del codificador, hay que seleccionar los dos bits (b2n, b_{2n+1}) en la salida entre bien (0, 0) o (1,1), o bien (0,1) o (1,0). En efecto, el último bit que entra en el codificador se utiliza para el cálculo de cada una de las dos salidas del codificador: cambiando este bit, se cambian los valores de las dos salidas. Esto significa que, para un estado del registro dado, hay que seleccionar los dos bits posibles en la salida entre dos grupos complementarios. Es, por lo tanto, también posible seleccionar el bit en la entrada de manera que se fuerce el valor de uno de los dos bits de salida. Se puede, por lo tanto, forzar fácilmente con este código un bit de cada dos en la salida del codificador.

En la práctica, para los códigos más eficientes utilizados en un módem, los grupos de bits en salida, para un estado del codificador, se seleccionan de manera que estén a la distancia máxima unos de otros. Para estos códigos, el último bit que entra en el codificador se utiliza para el cálculo de cada una de las dos salidas del codificador. Se puede, por lo tanto, para todos los códigos de rendimiento 1/2 habituales, forzar el valor de un bit de cada dos en la salida del codificador.

Se demuestra ahora que también es posible seleccionar una serie de bits en la entrada del codificador de forma que se obtenga en la salida del codificador una secuencia codificada que comprende una serie de bits consecutivos de valor fijado. Consideramos de nuevo el ejemplo anterior del codificador $r = k/n = 1/2$, (171, 133) de longitud $m+1 = 7$. Para este codificador, la respuesta impulsional es decir la respuesta del codificador a una secuencia binaria de entrada que comprende un bit de valor 1 precedido y seguido de bits que tienen todos el valor 0, viene dada por la secuencia 11101111000111, de longitud 14 = 2m+2. La operación de codificación se puede escribir de forma matricial, representada en la figura 6, en la que las líneas de la matriz generadora del código corresponden a la respuesta impulsional del codificador, desplazadas 2 bits de una línea a otra (ya que $n = 2$), o de manera más general desplazadas n bits de una línea a otra cuando el código es de rendimiento 1/n.

Se obtiene entonces un formalismo idéntico al utilizado para los códigos lineales en bloques, es decir que la secuencia codificada se obtiene efectuando el producto matricial de la secuencia de información con la matriz generadora definida con anterioridad.

La misma regla anteriormente dictada relativa a los códigos lineales en bloques puede de este modo aplicarse a los códigos convolucionales, es decir que es posible imponer el valor y la posición de un conjunto de bits en la salida del codificador si y solo si la submatriz M que corresponde a las columnas de los bits de salida es de rango completo, lo que se ilustra en la figura 7.

Se ve, por lo tanto, que para este código convolucional, es posible forzar el valor de 14 bits consecutivos en la salida del codificador, a partir de las entradas. También se puede interpretar este resultado desde el punto de vista de las relaciones de paridad entre bits codificados. Estas relaciones -deterministas- caracterizan de manera biunívoca las dependencias verificadas por los grupos de bits codificados. Dicho de otro modo, los valores de los bits codificados que corresponden a una relación de paridad son interdependientes. Al no corresponder los valores de los bits codificados a una relación de paridad se pueden fijar de forma independiente unos de otros. El código anteriormente considerado tiene unas relaciones de paridad de longitud 14 y separadas 2 bits. Es, por lo tanto, posible seleccionar 14 bits codificados consecutivos que no corresponden a ninguna relación de paridad entera, lo que significa que sus valores se pueden fijar de manera independiente.

De manera más general, es posible imponer el valor y la posición de un conjunto de bits en la salida del codificador si y solo si la submatriz:

$$\begin{bmatrix} m_{0,p_1} & m_{0,p_2} & m_{0,p_3} & \cdots & m_{0,p_d} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ m_{k-1,p_1} & m_{k-1,p_2} & m_{k-1,p_3} & \cdots & m_{k-1,p_d} \end{bmatrix}$$

es de rango completo, siendo $m_{i,j}$ los coeficientes de la matriz generadora y p_1, p_2,…p_d el conjunto de las posiciones de los bits cuyo valor se fija en la secuencia codificada.

Por último, en el caso de un código de rendimiento ½, el número máximo de bits sucesivos del cual es posible fijar el valor es igual a la longitud de la respuesta impulsional del código, es decir 2m+2 en la que m es la longitud de la limitación del código.

Ejemplo no limitativo de un código convolucional punzonado construido a partir de un código convolucional de rendimiento ½

Para los códigos punzonados, que presentan una redundancia más baja, es posible forzar un mayor número de bits consecutivos en la salida del codificador. Por ejemplo, se puede punzonar el código definido por los polinomios anteriores (171, 133) para obtener un código de rendimiento 3/4. Las relaciones de paridad de este código tienen una longitud 26 (11111101011011001010011111) y están separadas 4 bits. Este ejemplo se ilustra en la figura 8 en

la que se representan los indicios 800 de los bits en la salida del codificador, no constando una porción P_{28} de 28 bits consecutivos de ninguna relación de paridad entera y de 3 relaciones de paridades R1, R2, R2 ligadas a dicho código, con una longitud igual a 26 bits. Las relaciones de paridad se verifican para todas las secuencias de 26 bits que arrancan en un índice desplazado 4 bits para cada nueva secuencia.

Por analogía directa con lo anterior, es por lo tanto posible seleccionar los valores considerados por un grupo de 4(6+1) = 28 bits.

Caso general de los códigos convolucionales habituales

Para el caso más general de un código convolucional de rendimiento (n-1)/n, y con una longitud de limitación *m*, las relaciones de paridad son por lo general de longitud $n \cdot m+2$ (en el caso del ejemplo anterior, teníamos $m = 6$ y $n = 4$ después del punzonado, las relaciones de paridad eran de longitud 26). En este caso, y por analogía con lo anterior, es posible seleccionar los bits en la entrada del codificador de manera que se fuerce el valor de $n \cdot (m+1)$ bits consecutivos en la salida de codificador.

Se ve, por lo tanto, que con los códigos convolucionales, es posible, como para los códigos en bloque forzar el valor de algunos bits codificados, eventualmente consecutivos y a veces en gran número. La condición para que esta operación sea posible es que la matriz formada por las columnas de la matriz de codificación que corresponde a los bits forzados sea de rango completo.

Es en particular posible forzar el valor de secuencias de bits consecutivos de longitud significativa, en especial cuando el rendimiento del código es cercano a 1 y cuando la memoria m del código es importante.

Sucede lo mismo para los códigos en bloque, la dependencia entre los valores de los bits en la entrada del codificador y los bits en la salida del codificador es lineal. Los valores de los bits que deben forzarse en la entrada del codificador dependen de forma lineal de los valores de los demás bits en la entrada del codificador y de los valores de los bits forzados en la salida del codificador.

Turbo-códigos de producto

La invención se aplica también a los códigos correctores del tipo turbo-códigos de producto.

Los turbo-códigos son unos códigos correctores que combinan al menos dos códigos simples entrelazando las entradas de manera que cada uno de los códigos simples vea una serie de información diferente, por una parte, y que la información específica de cada bit, bloque o mensaje se reparta en sus contiguos, por otra parte. Por ello, aunque una parte de los bits, de los bloques o de los mensajes se altera durante la transmisión, la información correspondiente existe todavía más o menos en estos bits, bloques o mensajes contiguos. El procedimiento de decodificación es iterativo y colaborativo entre cada código simple. Este hace intervenir una noción de confianza en cada bit, bloque o mensaje decodificado y se diferencia de la decisión final en sus valores (“decisión blanda” o “soft decision” en inglés). Cada uno de los decodificadores transmite a los demás la información procedente de su propia decodificación (denominada información extrínseca) que se multiplexa en la información en la entrada de los demás codificadores. Al bit, bloque o mensaje así transmitido lo decodifican una segunda vez los demás codificadores simples y la información correspondiente se retransmite hacia los demás codificadores, etc. (de ahí la denominación “turbo” que está ligada al procedimiento de descodificación y no al código propiamente dicho).

Los códigos simples que se pueden emplear son múltiples.
Es posible utilizar unos códigos convolucionales. Los códigos convolucionales recursivos y sistemáticos están en la práctica especialmente adaptados. Los códigos se pueden situar en serie o en paralelo. La gestión inteligente del entrelazamiento y la detección/corrección iterativa de los datos por cada código simple permite multiplicar el poder detector y corrector del procedimiento global limitando al mismo tiempo el número de iteraciones y la complejidad.

Otra estructura de codificación turbo corresponde a los códigos de producto. En la versión más simple que corresponde a unos códigos de producto en dos dimensiones, el código de producto de rendimiento $\frac{k_1 \cdot k_2}{n_1 \cdot n_2}$ se construye a partir de dos códigos elementales C_1 y C_2 de rendimiento $\frac{k_1}{n_1}$ y $\frac{k_2}{n_2}$. Los códigos elementales utilizados son unos códigos en bloque muy simples (tradicionalmente unos códigos de paridad, unos códigos de Hamming o bien unos códigos Hamming extendidos). Los $n_1.n_2$ bits sucesivos aparecen como una sucesión de n_2 palabras de código C_1. Si se considera el tren binario diezmado por un factor n_1, se obtienen las palabras del código C_2.

Los turbo-códigos de producto, construidos a partir de varios códigos en bloque se asemejan a los códigos en bloque cuando se trata de determinar si es posible generar el motivo requerido. En efecto, la operación de codificación se puede descomponer en operaciones de codificación y de entrelazamiento.

Vamos a ilustrar la descomposición de un código de producto en varias operaciones de codificación y de entrelazamiento a partir de un ejemplo simple y no limitativo. Consideramos un código de producto construido a partir de dos códigos C_1 y C_2 de rendimiento $\frac{k_1}{n_1}$ y $\frac{k_2}{n_2}$.

La operación de codificación de un bloque de $k_1 \cdot k_2$ bits se puede descomponer de la siguiente forma:

- Una codificación con el código C_1 del tren binario: la codificación transforma k_2 grupos de k_1 bits en k_2 grupos de n_1 bits.
- Un entrelazamiento línea/columna simple: los bits se escriben línea por línea en una matriz de tamaño k_2 líneas y n_1 columnas. Los bits se leen a continuación columna por columna.
- Una codificación con el código C_2 del tren binario: la codificación transforma n_1 grupos de k_2 bits en n_1 grupos de n_2 bits.
- Un entrelazamiento línea /columna simple: los bits se escriben línea por línea en una matriz de tamaño n_1 líneas y n_2 columnas. Los bits se leen a continuación columna por columna.

En la práctica, si se considera una porción limitada de bits consecutivos de un bloque $n_1.n_2$ bits codificados (una porción de longitud n.k' sustancialmente más pequeña que $n_1.k_2$), las limitaciones del código C_2 no afectan a esta porción ya que no existe ninguna relación de paridad ligada al código C_2 que esté completamente contenida en esta porción de bits. Por consiguiente, si se considera una porción de bits consecutivos de tamaño inferior a $n_1.k_2$, todo sucede como si solo hubiera el código C_1 cuando se trata de determinar los motivos que es posible forzar. Al igual que los códigos en bloque, es por lo tanto posible, en particular, generar unos motivos bits consecutivos de tamaño $2 \cdot k_1$.

Códigos LDPC (*Low Density Parity Check*)

Los códigos LDPC son unos códigos en bloque particulares que se construyen de manera que los bits de paridad se calculen haciendo intervenir una relación de paridad de baja densidad. Se trata, en la práctica, de códigos en bloque sistemáticos (pero no cíclicos) y el análisis hecho sobre los códigos en bloque se aplica por analogía con la descripción hecha con anterioridad.

Los códigos LDPC son por lo general de gran tamaño. Al igual que se ha establecido para los códigos en bloques, los códigos LDPC permiten por lo tanto, en particular, generar unos motivos seleccionados con series de bits consecutivos de gran longitud.

En resumen para cualquier código corrector para el cual la operación de codificación se pueda realizar multiplicando la secuencia de información por una matriz generadora para obtener la secuencia codificada, es posible fijar el valor y la posición de un conjunto de bits de la secuencia codificada imponiendo una secuencia binaria de entrada particular. Sin embargo, esta posibilidad existe únicamente si la submatriz, de la matriz generadora, definida por

$$\begin{bmatrix} m_{0,p_1} & m_{0,p_2} & m_{0,p_3} & \cdots & m_{0,p_d} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ m_{k-1,p_1} & m_{k-1,p_2} & m_{k-1,p_3} & \cdots & m_{k-1,p_d} \end{bmatrix}$$

es de rango completo, siendo $m_{i,j}$ los coeficientes de la matriz generadora y p_1, p_2,…p_d el conjunto de las posiciones de los bits cuyo valor se fija en la secuencia codificada. Una matriz de rango completo es una matriz en la que todas las columnas son independientes. Si este es el caso, basta con resolver el sistema de ecuaciones:

$$\begin{bmatrix} i_0 & \cdots & i_{k-1} \end{bmatrix} = \begin{bmatrix} c_{p_1} & \cdots & c_{p_d} \end{bmatrix} \begin{bmatrix} m_{0,p_1} & m_{0,p_2} & m_{0,p_3} & \cdots & m_{0,p_d} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ m_{k-1,p_1} & m_{k-1,p_2} & m_{k-1,p_3} & \cdots & m_{k-1,p_d} \end{bmatrix}^{-1}$$

en el que $[C_{P1}…C_{Pd}]$ son los bits o símbolos cuyo valor se fija en la secuencia codificada e i_1,… i_k son las incógnitas del sistema que hay que suministrar en la entrada del codificador.

La cadena 400 de emisión también puede constar de un módulo 403 de perturbación, también llamado mezclado.

El mezclado, o perturbación, se utiliza para hacer a la secuencia binaria que hay que emitir lo más aleatoria posible, con vistas a mejorar la sincronización de símbolo pero también para contribuir a la protección de los contenidos de los mensajes mezclados. Su objetivo es suprimir las largas sucesiones de bits iguales a 0 o a 1 que impiden una recuperación correcta del ritmo símbolo. Existen varios tipos de mezcladores entre los cuales tenemos los mezcladores síncronos que necesitan una referencia temporal previa o los mezcladores de sincronización

automática.

Esta transformación es invertible, la operación de perturbación consiste en transformar un grupo de L bits en otro grupo de L bits y la operación de inversión de la transformación del tren binario es una operación efectuada por un receptor. De este modo, la función de transferencia inversa de un mezclador o aleatorizador se puede deducir fácilmente de su función de transferencia directa.

Caso de un aleatorizador síncrono

En el caso de un aleatorizador síncrono, se añade una secuencia pseudoaleatoria en módulo 2 a la señal binaria que hay que aleatorizar. La serie binaria {..., e_k, e_{k+1}, ...} que representa la secuencia de perturbación es periódica, de periodo L largo. Las secuencias más habitualmente utilizadas para la perturbación son unas secuencias de longitud máxima, construidas a partir de un registro de desplazamiento. Estas secuencias se generan mediante un registro de desplazamiento en bucle según un polinomio primitivo $E(X) = 1 + c_1X + ...+ c_PX^P$ de grado P. Se obtiene una secuencia de periodo $L = 2^P-1$. Sin embargo, las secuencias utilizadas para la perturbación pueden estar truncadas. La k-ésima salida e_k del registro de desplazamiento verifica la relación de recurrencia $e_k = c_1. e_{k-1}+ ...+ c_P. e_{k-P}$, siendo $c_1,..., c_P$ unos coeficientes constantes en el cuerpo de Galois GF2 independientes de los datos de entrada y de los datos aleatorizados, designando "+" siempre la suma en módulo 2 en GF2. Para una serie binaria en la entrada escrita {..., b_k, b_{k+1}, ...}, la transformación F operada por un aleatorizador síncrono se puede escribir $F(\{..., b_k, b_{k+1}, ...\}) = \{..., b'_k, b'_{k+1}, ...\}$ siendo $b'_k = b_k+e_k$. La transformación F es perfectamente invertible. En efecto, esta transformación F es igual a su inversa $F^{-1} = F$ puesto que $b_k = b'_k+e_k$ et sabiendo que $e_k+e_k = 0$ en el cuerpo de Galois GF2. La operación F consiste en una nueva suma en módulo 2 del valor de los bits aleatorizados por las salidas del mismo registro de desplazamiento que en la emisión para producir una secuencia desaleatorizada de idéntica longitud. Con el fin de poder forzar al tren binario en la salida de un aleatorizador síncrono, basta con estar correctamente sincronizado con la operación de perturbación. Dicho de otro modo, las posiciones de los índices k o de inicio de periodo deben coincidir antes de aplicar la transformación $F^{-1} = F$ a los bits aleatorizados. En el caso de un aleatorizador síncrono, la dependencia entre el valor i de un bit en la entrada del aleatorizador y el valor c del bit en la salida de aleatorizador es afín. Según la posición del bit considerado, la dependencia es, bien c = i, o bien c = i+1.

Caso de un aleatorizador de sincronización automática

En el caso de un aleatorizador de sincronización automática, los estados del registro se rellenan con un número finito de datos aleatorizados. La salida del registro se suma en módulo 2 al bit de datos en la entrada para formar el nuevo bit aleatorizado.

La transformación F, operada por un aleatorizador de sincronización automática se define por un registro con desplazamiento de polinomio primitivo $E(X) = 1 + c_1.X + ...+ c_P.X^P$ de grado P de periodo $L = 2^P-1$ y se escribe para una serie binaria en la entrada escrita {..., b_k, b_{k+1}, ...} con la forma: $F(\{..., b_k, b_{k+1}, ...\} = \{..., b'_k, b'_{k+1}, ...\}$ siendo $b'_k = b_k + c_1. b'_{k-1} + ...+ c_P. b'_{k-P}$ designando "+" siempre la suma en módulo 2 en GF2). Esto permite efectuar el descifrado en la recepción de forma simple y sin tener necesidad de sincronizarse de forma previa. En la recepción, se inyectan los bits aleatorizados $b'_{k-1},...,b'_{k-P}$ en el mismo registro que en la emisión para reconstituir la salida $b_k = b'_k + c_1. b'_{k-1} + ...+ c_P. b'_{k-P}$.

La dificultad para forzar los bits en la salida proviene aquí de que la expresión de los estados del registro en función de los datos de entradas b_k (datos no aleatorizados) hace intervenir un número ilimitado de dichos datos de entrada. Dicho de otro modo, todas las entradas b_k desde la puesta en marcha del aleatorizador intervienen en el valor de los estados del registro.

Para poder forzar el valor de los datos aleatorizados es indispensable tener acceso de forma síncrona a los datos aleatorizados para poder adaptar los datos de entrada de forma dinámica. Esta condición es necesaria para la aplicación de la invención si la cadena de emisión consta de un aleatorizador de sincronización automática.

La cadena 400 de emisión también puede constar de un módulo 404 de entrelazamiento.

El entrelazamiento se utiliza mucho en los canales de transmisión para los cuales las ocurrencias de los errores se reagrupan por paquetes. Su función es repartir de la forma más uniforme posible dichos errores. En la recepción, los errores se sitúan, tras el desentrelazamiento, de tal forma que impacten unas palabras de código diferentes. Estos errores pueden por tanto considerarse como des-correlacionadas, y el poder corrector de los decodificadores permite minimizar su impacto. El entrelazamiento se muestra también como un medio para introducir diversidad temporal en la cadena de transmisión y contribuye de este modo a proteger a esta de los desvanecimientos, de las interferencias y de las eventuales aleatorizaciones. Numerosos entrelazadores están constituidos por una tabla, los bits en la entrada se ordenan por tanto por líneas en la tabla y los bits de salida se producen mediante la lectura de columna de la tabla.

La trasformación operada por un entrelazador también es una operación invertible. Su función de transferencia inversa puede deducirse de su función de transferencia directa. En efecto, se trata de una transformación que transforma un bloque de L bits en otro bloque de L bits, permutando simplemente el orden de los bits. Con el fin de obtener un bloque de bits requerido tras entrelazamiento $B'=b'_k, ...b'_{k+L-1}$, basta con sincronizarse en el bloque y con

aplicar en este tren B' binario la permuta inversa, lo que corresponde a la operación de desentrelazamiento efectuada de forma clásica por un receptor. Se obtiene, por tanto, un bloque de bits B = $b_k,...b_{k+L-1}$, que una vez entrelazado, es estrictamente igual al tren binario B.

La cadena 400 de emisión también puede constar de un módulo 405 de entramado. El entramado 405 permite al receptor sincronizarse con las transformaciones del tren binario como las operaciones de entrelazamiento o decodificación, y a continuación recuperar los datos estructurados en forma de una multiplexación de varios flujos o de datos en forma de palabras o de octetos. Para ello, una vez los datos estructurados en forma de tramas que corresponden a unos motivos de ordenación periódicos, la sincronización del receptor en las tramas se efectúa por medio de motivos periódicos sincronizados de las tramas. Cada trama está de este modo precedida y/o seguida, y/o contiene una palabra de sincronización específica utilizada para efectuar la sincronización del receptor con las tramas recibidas. También se pueden reagrupar varias tramas para formar una multi-trama o una hiper-trama. Los motivos de sincronización utilizados para el entramado se repiten en la secuencia emitida y se pueden detectar y aleatorizar como se ha explicado con anterioridad.

El entramado limita la posibilidad de forzar la señal modulada requerida puesto que algunos bits toman unos valores impuestos en un intervalo regular, y que es necesario conservar estos bits para el correcto funcionamiento del receptor conectado con el emisor. Esta transformación no es, por lo tanto sobreyectiva para unos bloques de datos cuyo tamaño en la salida sobrepasa el de un bloque de datos útiles por trama.
Sin embargo, el entramado solo ocupa una parte muy limitada (y temporalmente bien delimitada) del caudal total (ejemplo: motivo de inicio de trama utilizando solo algunos símbolos, etc.) y no impide generar la señal codificada requerida en el interior de una trama.

Este punto no es, por lo tanto, un escollo particular para la implementación de la invención.

La cadena 400 de emisión puede también constar de un módulo 406 de codificación binaria para señal.

La codificación binaria para señal se utiliza para adaptar la señal al canal de transmisión. Transforma el mensaje digital en una señal eléctrica de banda base o en una señal de baja frecuencia. Se pueden citar dos grandes clases de códigos binarios para señal, los códigos de transcodificación NRZ (*Non Remise à Zéro*, en español: sin retorno a cero) y los códigos alfabéticos.

La transformación operada por un codificador binario para señal es una operación invertible. Su función de transferencia inversa se puede deducir, por lo tanto, de su función de transferencia directa.

La cadena 400 de emisión consta, por último, de un modulador 407 que transforma en particular la secuencia binaria en una secuencia de símbolos modulados. Los símbolos se toman de un conjunto complejo llamado constelación. Un símbolo puede reagrupar varios bits. Se pueden citar, por ejemplo, las modulaciones de fase digital o PSK (*Phase Shift Keying*) o las modulaciones de amplitud digital o QAM (*Quadrature Amplitude Modulation*).

La transformación operada por un modulador es una operación invertible. Su función de transferencia inversa se puede deducir, por lo tanto, de su función de transferencia directa.

De acuerdo con lo anterior, las transformaciones aplicadas en la cadena de emisión en el tren binario son reversibles, es decir que es posible a partir del tren binario en la salida encontrar el tren binario de entrada. En la práctica, incluso se añade redundancia de manera que se pueda encontrar la señal binaria de origen en presencia de errores en el tren binario codificado.

Sin embargo, siempre de acuerdo con lo anterior, algunas funciones implementadas por la cadena de emisión de un sistema de comunicaciones pueden no ser siempre sobreyectivas ni invertibles. Puede suceder que un tren binario codificado TBC que se desearía forzar en la salida de módulo no corresponda a ninguna serie TBU de bits "útil" en la entrada de módulo. Es, en particular, el caso de una función de codificación correctora en bloque o convolucional que no cumple con los criterios enunciados con anterioridad o de cualquier otra operación de codificación no biyectiva. En dicho caso, no es posible determinar la función F^{-1} de transferencia inversa de la cadena de emisión global.

Para superar esta dificultad, se consideran dos variantes de realización de la invención.

Una primera variante consiste en buscar, en el conjunto F({TBU}) unas secuencias moduladas que es posible obtener en la salida de la cadena de emisión a partir del conjunto {TBU} de las secuencias binarias posibles en la entrada de la cadena de emisión, la secuencia T' binaria que minimiza la distancia entre la transformación modulada en la salida F(T') de la secuencia T' y la secuencia D modulada buscada que contiene al menos un motivo de sincronización ficticio posicionado en la situación deseada. La distancia considerada puede, por ejemplo, ser una distancia en el sentido de los mínimos cuadrados calculada integrando la diferencia entre una secuencia F(T) posible, perteneciendo T al conjunto {TBU}, y la secuencia D buscada en un intervalo temporal fijado. Se puede calcular un criterio posible por medio de la siguiente relación:

$$\forall T \in \{TBU\};\ C(T,D) = \left\| F(T) - D \right\|_{L^2} = \int \left| F(T)(t) - D(t) \right|^2 dt$$

A continuación se busca la secuencia de entrada T' ∈ {TBU} de la cadena de emisión que minimiza el criterio C:

$$T' = \underset{T \in \{TBU\}}{\text{Argmin}} \left[C(T, D) \right]$$

Sin embargo, la implementación práctica de esta variante debe tener en cuenta dos limitaciones principales.

Por una parte, por razones de complejidad, no es posible en la práctica obtener la imagen F{TBU} de las señales moduladas en la salida de la cadena de emisión que corresponde al conjunto de todos los trenes binarios útiles posibles (dicho de otro modo, que corresponde a todo el conjunto {TBU}), sino solo un subconjunto restringido de esta imagen.

Por otra parte, se puede simplificar la implementación de la invención basándose en los soportes temporales ligados principalmente a las operaciones de entrelazamiento, de codificación y de entramado. En efecto, no resulta útil en la práctica para la implementación de la invención considerar unos trenes T binarios útiles en el conjunto {TBU} cuyas señales moduladas F(T) en la salida (tras la codificación y entrelazamiento en particular) se encontrarían dispersadas en unos intervalos de tiempo excesivamente largos, o en un excesivo número de tramas en la salida, o cuya distribución de las posiciones de los símbolos tendría demasiadas lagunas.

En la práctica, por lo tanto, esta variante de la invención puede implementarse restringiendo el conjunto {TBU} a un subconjunto {TBU'} que solo comprende unas secuencias T de bits útiles cuyas longitudes y posiciones tras el paso por la cadena de emisión (y en particular tras el paso por los módulos de codificación y de entrelazamiento) corresponden a unos motivos $SINC'_F$ ficticios, llegado el caso inadecuados con respecto a la perturbación de la red de comunicación, pero compatibles con el estándar de implementación del sistema de telecomunicación en términos de posiciones, de recurrencia y de periodicidad de trama y, por lo tanto, fácil de generar y de insertar en las tramas de datos útiles.

La estructura tiempo/frecuencia y/o la elección de las posiciones, longitudes y recurrencias de las señales $SINC'_F$ ficticias que se acercan en el sentido del criterio mencionado más arriba a las modificaciones $SINC_F$ ficticias requeridas en la salida, se fijan basándose en las periodicidades de tramas.

El conjunto restringido {TBU'} se pre-determina a continuación invirtiendo para los motivos $SINC'_F$, de forma analítica o por simulación, los módulos de la cadena de emisión y en particular los módulos de entrelazamiento y los módulos de codificación.

Esto permite, en particular, simplificar la implementación de la invención concentrando la búsqueda de las secuencias T binarias en la imagen inversa (en particular mediante las transformaciones de codificación y de entrelazamiento) de un conjunto restringido de motivos $SINC'_F$ ficticios en la salida, que se acercan correctamente a la secuencia $SINC_F$ requerida en el sentido del criterio anteriormente mencionado, y que corresponden además a la duración de una trama en la salida de cadena de emisión (y/o a la duración de una trama de información útil en la entrada de dicha cadena de emisión). Se periodizan las secuencias T' binarias así obtenidas por trama de información útil para generar en la salida unos motivos ficticios también periódicos y de periodo indexado en el de la trama de la señal en la salida.

Por tanto, se puede realizar de una sola vez el cálculo de las posiciones de los bits en la secuencia T' binaria para obtener un motivo ficticio en la salida de la cadena de emisión.

Si se construye la secuencia T' binaria de manera completamente artificial utilizando un modo de mensajería que permite una programación completa del contenido de los mensajes, también se puede realizar el cálculo de los valores de los bits de la secuencia T' binaria de una sola vez.

Si se construye la secuencia T' binaria a partir de una aplicación de tipo codificador de audio o de vídeo que produce un contenido variable y aleatorio, también se puede realizar de una sola vez el cálculo de las relaciones (deterministas) que dan los valores de los bits de la secuencia T' binaria (en función del motivo ficticio $SINC'_F$ buscado y de los bits de información cercanos).

En la práctica, para generar dichas secuencias T binarias se debe por lo tanto forzar el valor de algunos bits útiles, denominados bits ficticios, (situados en la entrada de la codificación de canal) en unas posiciones precisas. La transformación aplicada consiste en determinar, en primer lugar, una partición de la trama de los datos útiles invirtiendo los módulos de entrelazamiento de codificación: esto determina unas posiciones muy precisas, en las que se seleccionan de forma artificial los valores de los bits ficticios de manera que se generen las secuencias ficticias tras el entrelazamiento y codificación. Finalmente, en esta variante simplificada de implementación de la invención:

- las posiciones que toman estos bits ficticios forzados se determinan partiendo de la posición y de los valores de las secuencias $SINC'_F$ ficticias y subiendo las transformaciones del tren binario una por una;
- los valores que toman estos bits ficticios o las relaciones que dan los valores de los bits ficticios según los bits de información contiguos se determinan también subiendo las transformaciones del tren binario una por una.

Aunque afecta a todas las etapas de la cadena de emisión, las etapas clave abarcan la inversión de las operaciones de entrelazamiento y de codificación correctora.

Manteniéndose en la escala de una trama de señal en la salida de la cadena de emisión, las posiciones, valores o relaciones que dan los valores de los bits ficticios en el flujo de información útil son fijos. Se pueden calcular de una sola vez y a continuación aplicarse en cada trama sucesiva.

Una segunda variante de realización de la invención consiste en aplicar la invención a un subconjunto situado aguas abajo de la cadena de emisión constituida por bloques en serie cuyas funciones de transferencia son todas invertibles. En la práctica, se identifica la porción de longitud máxima de la cadena de transmisión para la cual puede realizarse de forma satisfactoria la inversión del tren binario. Para ello se sube desde la señal en la salida del modulador hacia la secuencia de datos útiles que hay que emitir. El tren binario destinado a producir unas señales moduladas D(t), que contiene una secuencia de sincronización ficticia, solo se inyecta al nivel de la entrada de este subconjunto, dicho de otro modo en la salida de la primera función no invertible de la cadena de emisión en el orden secuencial inverso de las funcione de emisión (que van del modulador -final de la cadena- al codificador corrector -inicio de la cadena). Esta segunda variante de la invención se aplica, por ejemplo, si la cadena de emisión consta de un dispositivo de cifrado o criptografía cuya función presenta por naturaleza una inversa no determinable.

El procedimiento según la invención se puede implementar a partir de elementos de software. Dicho software lo puede ejecutar el equipo emisor de modo que se modifique la secuencia binaria útil que hay que emitir que se produce, por ejemplo, mediante una aplicación. También lo puede ejecutar un ordenador conectado a dicho emisor con el objetivo de parametrizarlo.

El procedimiento según la invención puede estar disponible como producto programa de ordenador en un soporte legible por un ordenador. El soporte puede ser electrónico, magnético, óptico, electromagnético o ser un soporte de difusión de tipo infrarrojo. Dichos soportes son, por ejemplo, unas memorias con semiconductor (*Random Access Memory* RAM, *Read-Only Memory* ROM), unas cintas, unos disquetes o discos magnéticos u ópticos (*Compact Disk - Read Only Memory* (CD-ROM), *Compact Disk - Read/Write* (CD-R/W) *and DVD*).

REIVINDICACIONES

1. Procedimiento de perturbación de un sistema (EM, REC) de comunicación que consiste en generar, en una señal (S) destinada a emitirse mediante al menos un emisor (EM_B1, EM_B2,…, EM_B3) compatible de dicho sistema (EM, REC) de comunicación, comprendiendo dicha señal (S) unos datos (D_U, T) generados por una aplicación y destinados a trasmitirse a un receptor, al menos una secuencia ($SINC_F$) de sincronización ficticia, comprendiendo dicho procedimiento las siguientes etapas:

- definir dicha secuencia ($SINC_F$) de sincronización ficticia y su posición en tiempo y/o en frecuencia en el interior de la señal (S) que hay que emitir;
- estimar el valor y la posición de los bits ficticios (B_F) que hay que insertar en el interior de la secuencia (D_U, T) de datos que hay que emitir en la entrada de la cadena de emisión o de una subparte de la cadena de emisión de modo que se obtengan, en la secuencia producida en la salida de la cadena de emisión, el valor y la posición temporal predefinidas de los símbolos de dicha secuencia ($SINC_F$) de sincronización ficticia;
- insertar en el interior de dicha secuencia (D_U, T) de datos los bits (B_F) ficticios en las posiciones obtenidas.

2. Procedimiento de perturbación según la reivindicación 1 en el que, cuando la función F de transferencia de la cadena de emisión es invertible, el valor y la posición de los bits (B_F) ficticios se estiman determinando la función F^{-1} de transferencia inversa de dicha función F de transferencia y aplicando dicha función F^{-1} de transferencia inversa a dicha secuencia ($SINC_F$) de sincronización ficticia.

3. Procedimiento de perturbación según la reivindicación 2 en el que la función F^{-1} de transferencia inversa de dicha cadena de emisión se determina efectuando la composición, en el orden inverso, de las funciones de transferencia inversas de los diferentes bloques que componen dicha cadena.

4. Procedimiento de perturbación según la reivindicación 3 en el que la cadena de emisión consta al menos de un código (402) corrector de rendimiento k/n para el cual la inversión de su función de transferencia se realiza resolviendo el siguiente sistema de ecuaciones:

$$[i_0 \quad \dots \quad i_{k-1}] = [c_{p_1} \quad \dots \quad c_{p_d}] \begin{bmatrix} m_{0,p_1} & m_{0,p_2} & m_{0,p_3} & \dots & m_{0,p_d} \\ \dots & \dots & \dots & \dots & \dots \\ m_{k-1,p_1} & m_{k-1,p_2} & m_{k-1,p_3} & \dots & m_{k-1,p_d} \end{bmatrix}^{-1}$$

en la que [C_{P1}…C_{Pd}] son los símbolos cuyo valor es fijado en la secuencia codificada, [i_0,… i_{k-1}] es la secuencia que hay que producir en la entrada de dicho código, y $m_{i,pj}$ son los coeficientes de la matriz generadora de dicho código, siendo p_1, p_2,…p_d el conjunto de las posiciones de los símbolos cuyo valor se fija en la secuencia codificada, siendo d un entero igual como máximo al número n de símbolos de la secuencia codificada.

5. Procedimiento de perturbación según la reivindicación 4 en el que dicho código corrector es un código lineal en bloque o un código convolucional, o un turbo-código, o un código LDPC de baja densidad.

6. Procedimiento de perturbación según una de las reivindicaciones 3 a 5 en el que la cadena de emisión consta además de un mezclador (403) y/o de un entrelazador (404) y/o de un módulo de entramado (405) y/o de un codificador (406) binario para señal y/o de un modulador (407).

7. Procedimiento de perturbación según la reivindicación 1 en el que, cuando la función F de trasferencia de la cadena de emisión es no suryectiva, el valor y la posición de los bits ficticios se estiman buscando la secuencia T' de entrada de la cadena de emisión que minimiza un criterio de distancia entre la secuencia F(T')(t) obtenida en la salida de la cadena de emisión cuando dicha secuencia T' se produce efectivamente en su entrada y dicha secuencia ($SINC_F$) de sincronización ficticia.

8. Procedimiento de perturbación según la reivindicación 7 en el que dicho criterio de distancia se considera igual a la integral, en una duración dada, de la norma al cuadrado de la diferencia entre la secuencia F(T')(t) obtenida en la salida de la cadena de emisión y dicha secuencia ($SINC_F$) de sincronización ficticia.

9. Procedimiento de perturbación según una de las reivindicaciones 7 u 8 en el que, la búsqueda de la secuencia T' de entrada de la cadena de emisión que minimiza dicho criterio de distancia se efectúa en un subconjunto del conjunto de las secuencias binarias posibles en la entrada de la cadena de emisión.

10. Procedimiento de perturbación según una de las reivindicaciones anteriores en el que dicha secuencia (D_U, T) de datos la produce una aplicación (401), entre las cuales están un codificador de audio, de imagen o de vídeo.

11. Procedimiento de perturbación según la reivindicación 1 en el que, cuando la función F de transferencia de la cadena de emisión es no sobreyectiva, los bits (B_F) ficticios se insertan en la secuencia producida en la entrada de una subparte de la cadena de emisión cuya función de transferencia es sobreyectiva y cuya salida es común a la salida de la cadena de emisión.

12. Procedimiento de perturbación según la reivindicación 11 en el que dicha cadena de emisión consta de un dispositivo de cifrado y dicha subparte de la cadena de emisión excluye este dispositivo.

13. Procedimiento de perturbación según una de las reivindicaciones anteriores en el que los valores de los símbolos de dicha secuencia ($SINC_F$) de sincronización ficticia son idénticos a los de los símbolos de una secuencia (SINC) de sincronización que consta de dicha señal para sincronizar entre sí un receptor y un emisor compatibles de dicho sistema de comunicación.

14. Procedimiento de perturbación según una de las reivindicaciones anteriores en el que la posición en tiempo y/o en frecuencia de los símbolos de dicha secuencia ($SINC_F$) de sincronización ficticia son diferentes de las de los símbolos de una secuencia (SINC) de sincronización de la que consta dicha señal para sincronizar entre sí un receptor y un emisor compatibles de dicho sistema de comunicación.

15. Dispositivo (EM_B1, EM_B2,…, EM_B3) de emisión de una señal que consta de una cadena de emisión para transformar una secuencia (D_U, T) de datos a emitir en una señal (S) a emitir y un medio adaptado para implementar las etapas del procedimiento según una de las reivindicaciones 1 a 14.

16. Sistema de perturbación cooperativo que comprende una multitud de dispositivos (EM_B1, EM_B2,…, EM_B3) de emisión según la reivindicación 15 y para los cuales las posiciones en tiempo y/o en frecuencia de dichas secuencias ($SINC_F$) de sincronización ficticias insertadas en la señal emitida por cada dispositivo de emisión son diferentes entre sí.

17. Sistema de perturbación cooperativo que comprende una multitud de dispositivos (EM_B1, EM_B2,…, EM_B3) de emisión según la reivindicación 15 y para los cuales cada uno de dichos dispositivos (EM_B1, EM_B2,…, EM_B3) de emisión emite una señal (S) que comprende una multitud de secuencias ($SINC_F$) de sincronización ficticias cuyos valores y/o posiciones en tiempo y/o posiciones en frecuencia son diferentes entre sí.

18. Programa de ordenador que consta de instrucciones para la ejecución de las etapas del procedimiento según una cualquiera de las reivindicaciones 1 a 14, cuando al programa lo ejecuta un procesador.

19. Soporte de grabación legible por un procesador en el que está grabado un programa que consta de las instrucciones para la ejecución de las etapas del procedimiento según una cualquiera de las reivindicaciones 1 a 14, cuando al programa lo ejecuta un procesador.

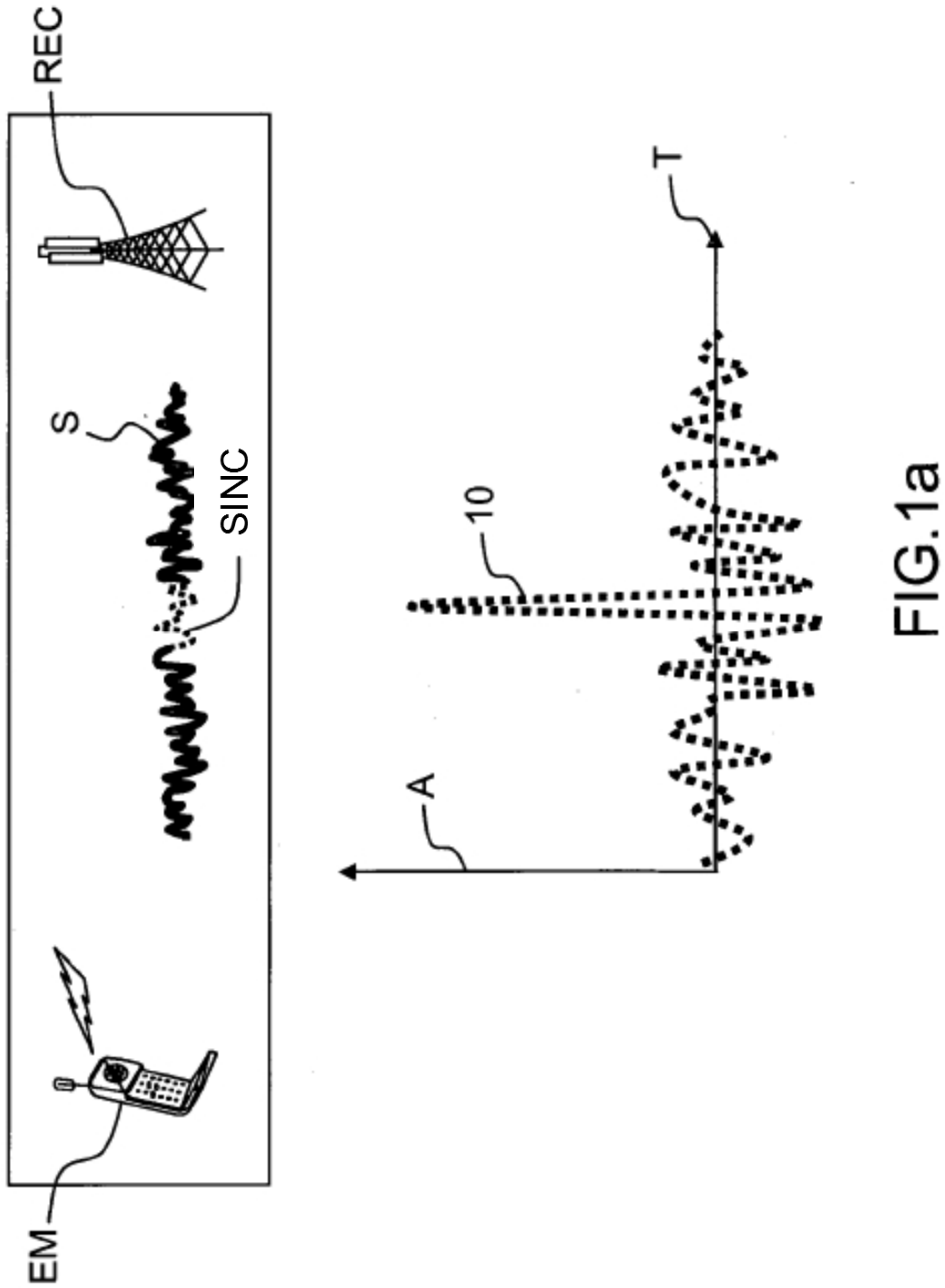

FIG.1a

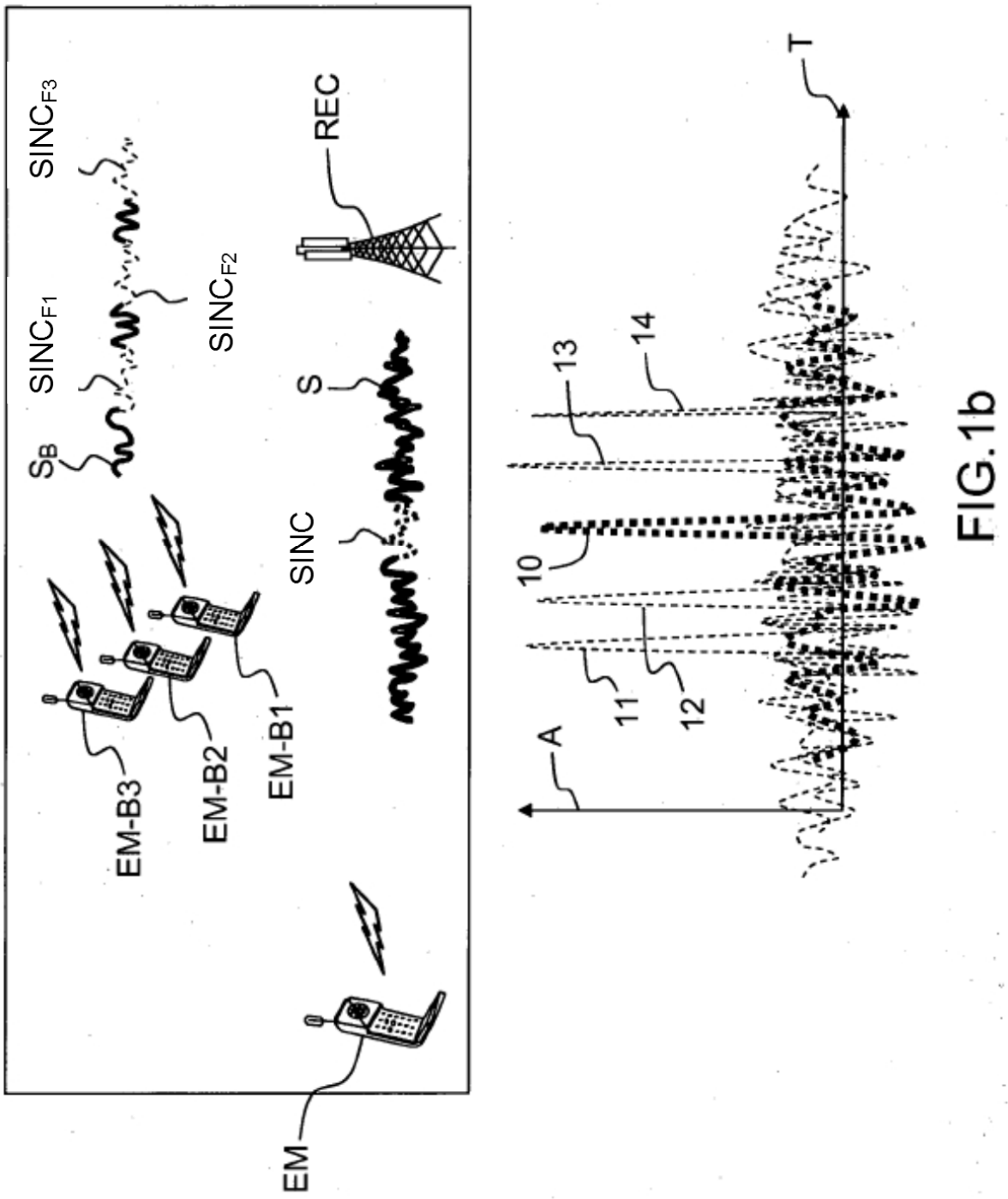
EM
EM-B3
EM-B2
EM-B1
SINC
S
REC
$SINC_{F3}$
$SINC_{F1}$
S_B
$SINC_{F2}$
A
10
11
12
13
14
T

FIG.1b

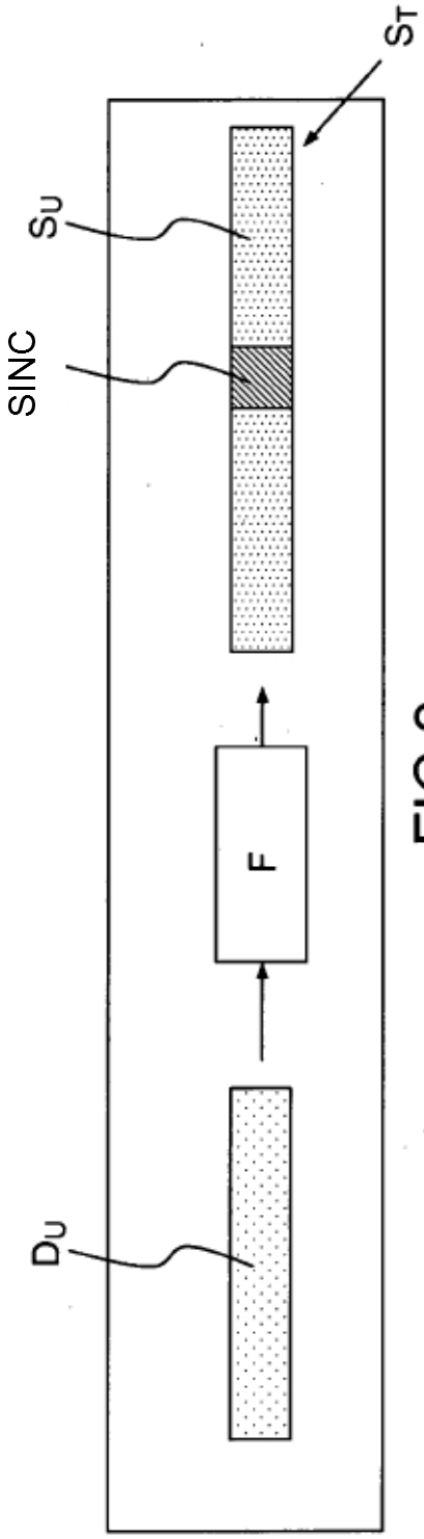

FIG.2

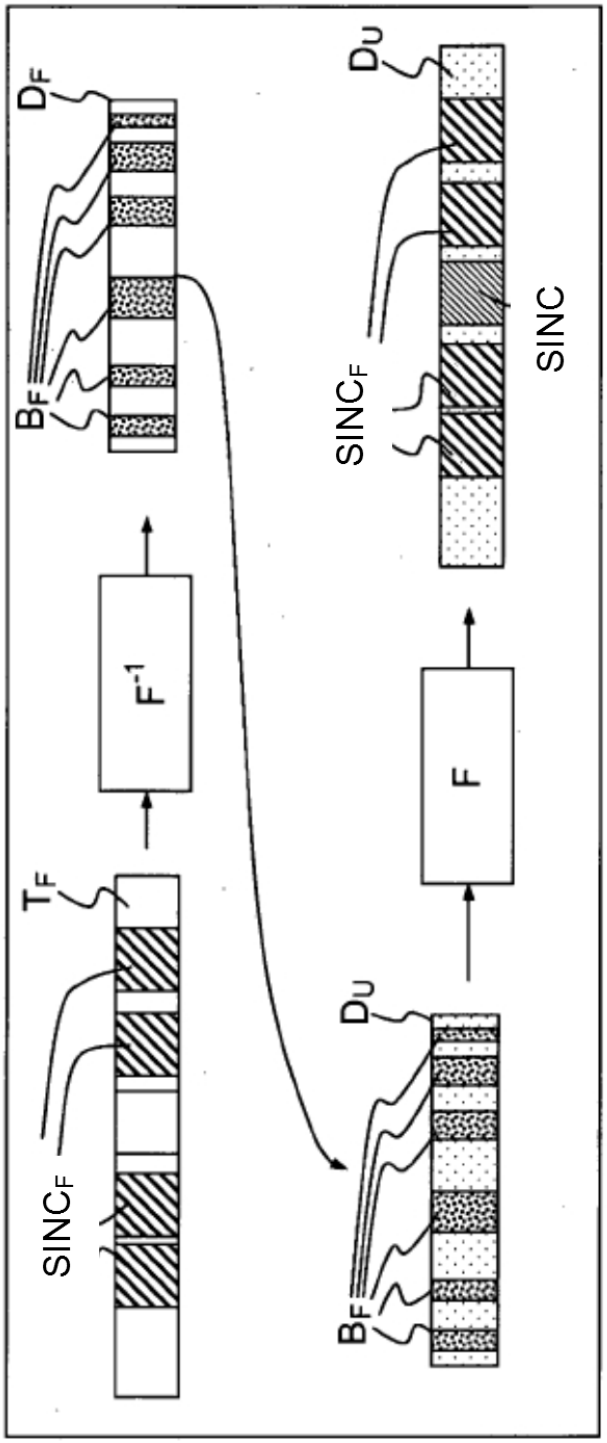

FIG.3

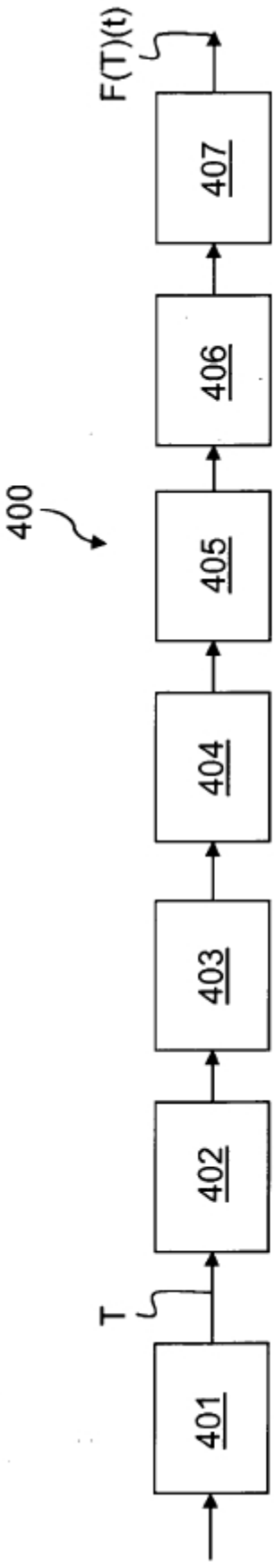
400
401
402
403
404
405
406
407
T
F(T)(t)

FIG.4

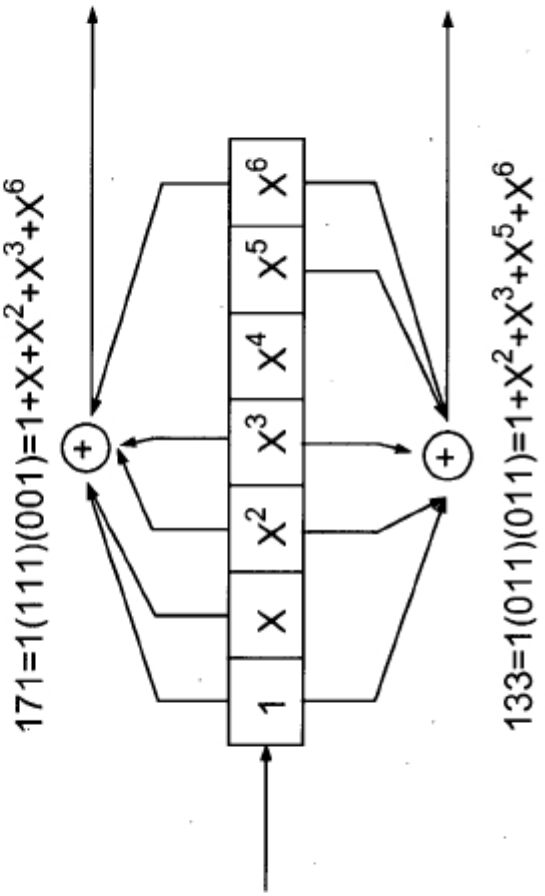
171=1(111)(001)=1+X+X2+X3+X6
133=1(011)(011)=1+X2+X3+X5+X6
1
X
X2
X3
X4
X5
X6
+
+

FIG.5

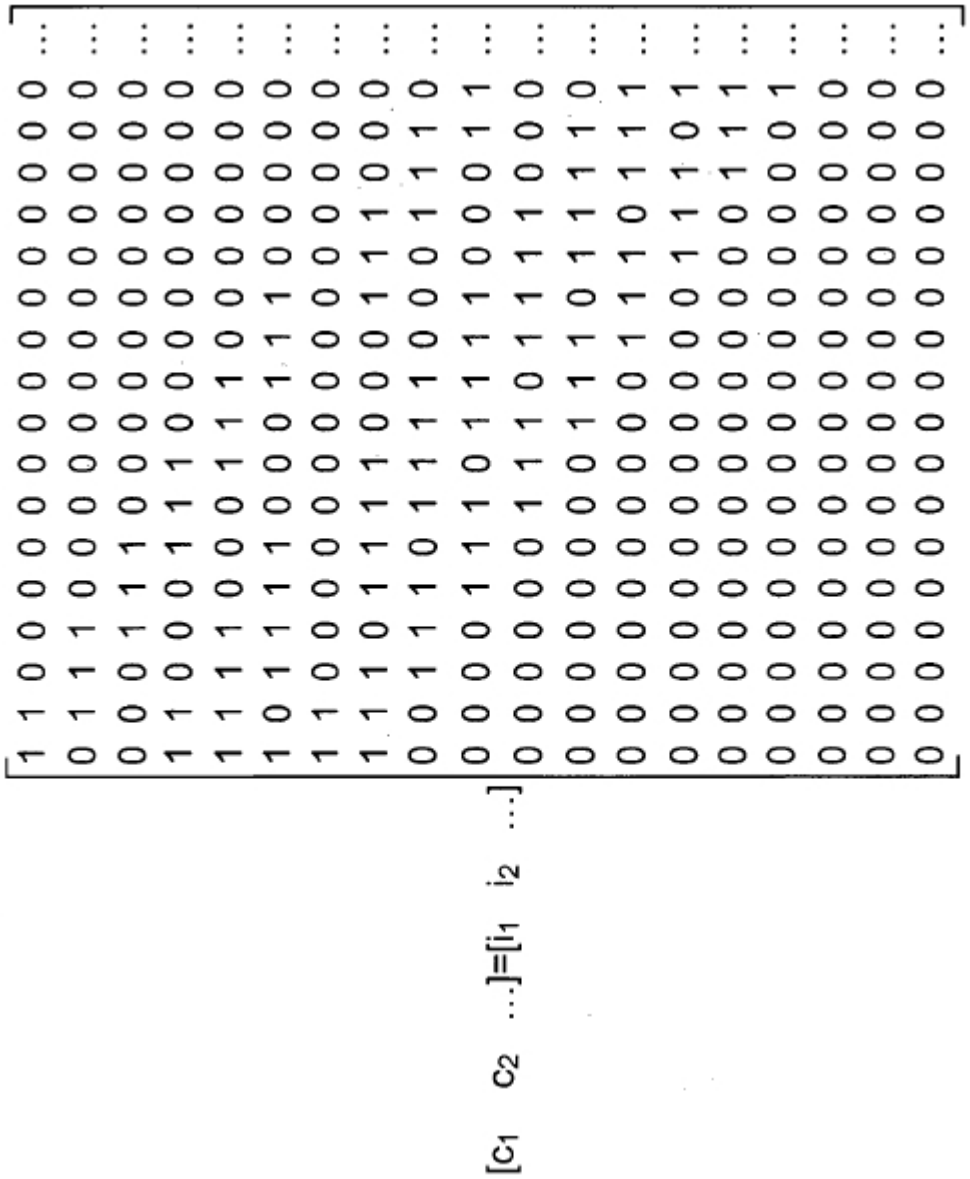

FIG.6

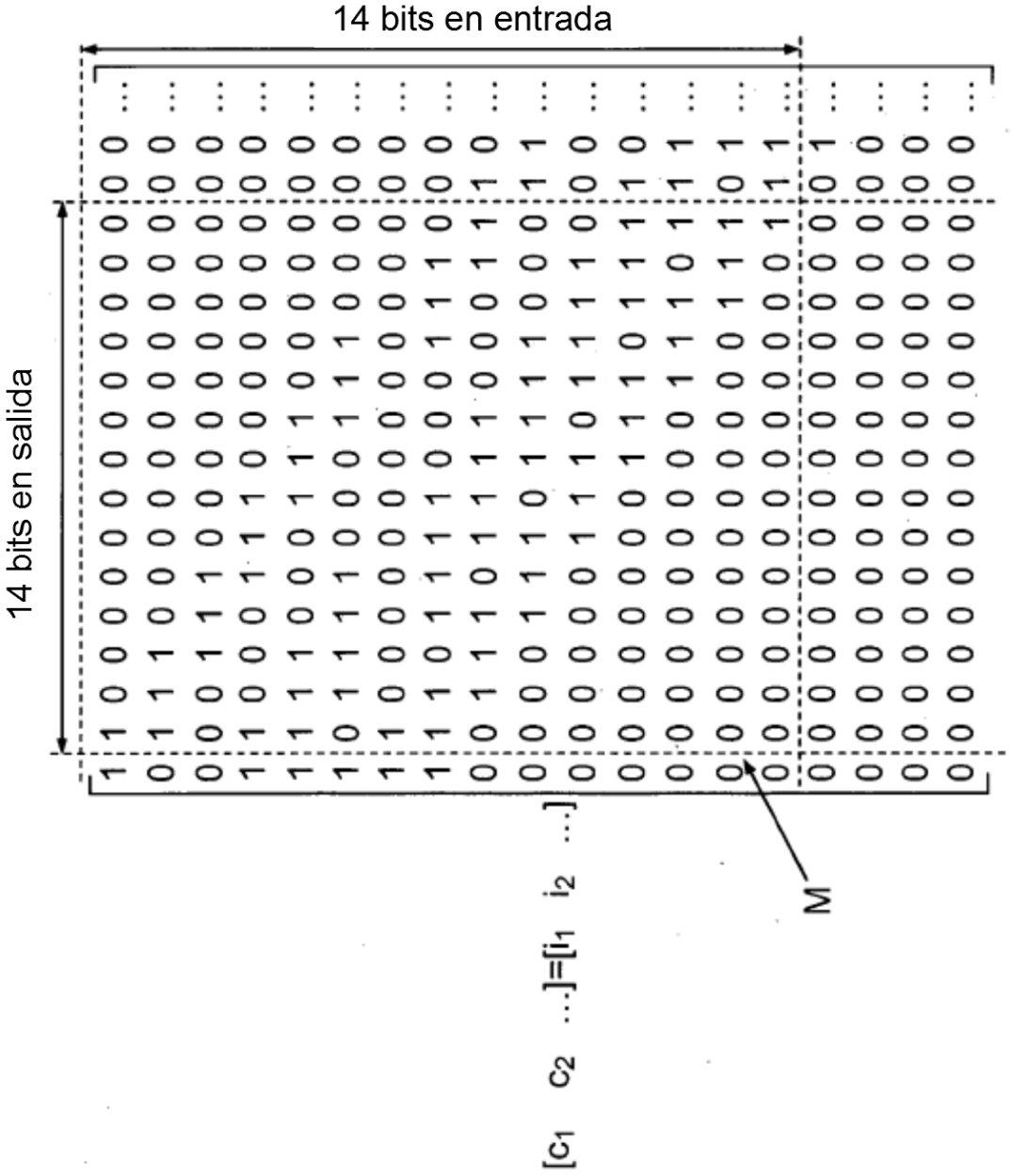

FIG.7

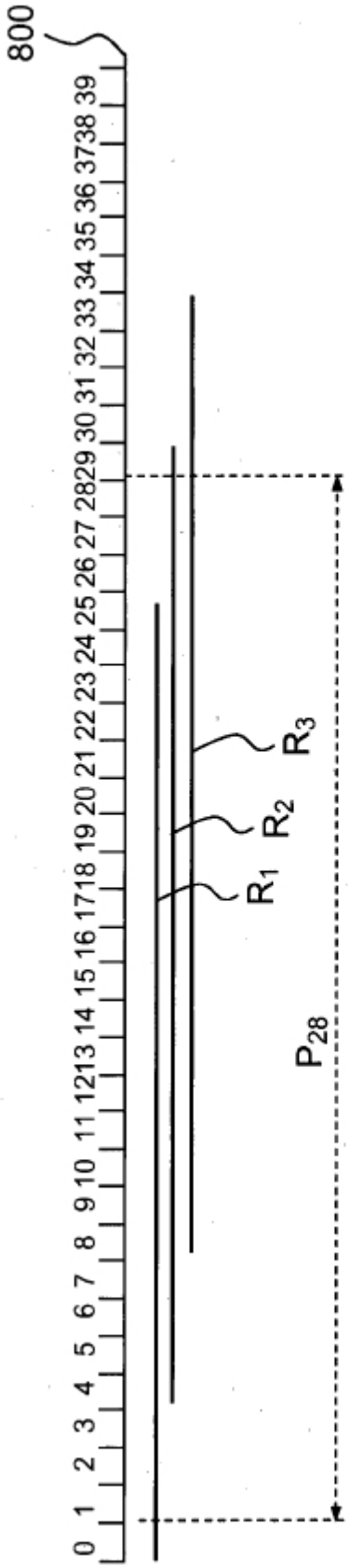
800
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39
R1
R2
R3
P28

FIG.8