

3

CODIFICACION DE CANAL

Introducción.

La finalidad de la codificación de canal es la detección y corrección de errores producidos en el canal de comunicación o en medios de grabación, como consecuencia del ruido y distorsión introducidos, tanto por el medio de propagación, como por las no linealidades en el propio sistema de transmisión. El tema es muy amplio y su tratamiento a fondo queda fuera del contexto de esta obra, por lo que aquí, se tratarán únicamente los principales aspectos relacionados con la transmisión digital de señales de televisión y en la bibliografía al final del capítulo, se dan algunas referencias para consulta de quienes deseen profundizar en estos temas.

3.1 Generalidades.

En los sistemas digitales de comunicaciones hay dos causas principales que influyen en el deterioro de la señal recibida. La primera es el ruido introducido por el propio canal de comunicaciones, en que los mecanismos de propagación juegan un papel muy importante. La segunda es el ruido de cuantificación, consecuencia del proceso de codificación, que se introduce inevitablemente en el transmisor y que se transporta por todo el sistema hasta la salida del receptor. El ruido del canal produce errores de transmisión, que hacen que la señal reconstruida por el receptor no sea la misma que la señal transmitida. La fidelidad de la transmisión de información se mide en términos de la *tasa de errores* o *probabilidad de error* es decir, la probabilidad de que el símbolo a la salida del receptor o reproductor sea diferente al símbolo transmitido o grabado. Tanto los sistemas de transmisión como de grabación y reproducción están sujetos a errores.

En la transmisión y grabación de señales digitales de audio, vídeo o datos en general, igual que en cualquier sistema digital de comunicaciones, el caudal binario recibido o reproducido debe ser, en la medida posible, igual que el transmitido o grabado, por lo que la información debe protegerse al máximo contra las degradaciones que inevitablemente introduce el medio de transmisión o los circuitos de procesado de la señal. Los sistemas de comunicaciones por cable y satélite son menos hostiles que los de radiodifusión terrestre, ya que los primeros utilizan un medio de transmisión muy estable: cable o fibra óptica, en que el principal problema es la atenuación, fácilmente predecible y cuyos efectos pueden compensarse con relativa facilidad. En el caso de comunicaciones por satélite también el comportamiento del medio de propagación, aunque sujeto a variaciones por meteoros atmosféricos en ciertas bandas, es bastante predecible y por consecuencia, sus efectos también pueden compensarse.

Por el contrario, los sistemas de radiodifusión terrestre presentan los problemas más complejos a causa de los diversos mecanismos que intervienen en la propagación y que contribuyen considerablemente a la degradación de la señal. En los sistemas terrestres los efectos multicamino desempeñan un papel importante en la atenuación y retardo de la señal recibida y los desvanecimientos de la señal causados por la variabilidad del entorno de propagación producen tanto atenuación variable, como dispersión temporal y frecuencial de la señal, que se traduce en destrucción de la información transmitida o en interferencia entre símbolos con el consecuente aumento en la tasa de errores.

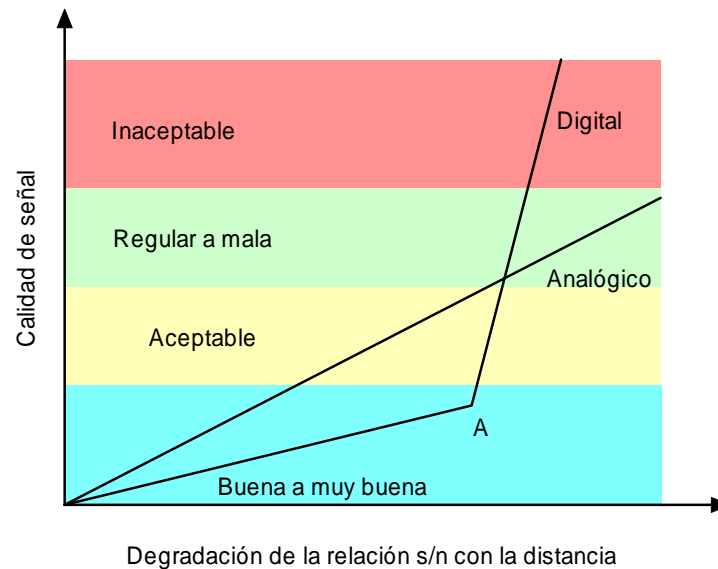


Fig. 3.1. Degradación de la señal en sistemas analógicos y digitales, en función de la distancia.

En los sistemas analógicos la degradación de la señal es suave; es decir, la relación señal a ruido disminuye lentamente con la distancia y la señal, aunque con ruido, puede recibirse en condiciones aceptables en un rango considerable de distancias. En los sistemas digitales, por el contrario, la señal deja de recibirse cuando la tasa de errores aumenta por encima del valor que es capaz de manejar el receptor. La degradación en estas condiciones es muy brusca y se pasa rápidamente de una zona de recepción satisfactoria a una de recepción nula. Esta situación se ilustra gráficamente en la figura 3.1. A partir de la distancia correspondiente al punto A en la figura, el número de errores en la transmisión digital es suficientemente grande como para degradar completamente la señal, en tanto que en el sistema analógico la recepción es posible aún con ruido. Como consecuencia de lo anterior, es necesario proteger la información transmitida de forma tal que, aunque una parte de ella se pierda o destruya durante el proceso de transmisión, pueda ser recuperada o reconstruida en el receptor, de modo que el observador no perciba esa pérdida. En un sistema de televisión digital es literalmente imposible la corrección de errores mediante la retransmisión de la información ya que el procesado de la información ocurre en tiempo real. Por ello el sistema debe implementarse de forma tal, que aunque parte de la información se destruya, el receptor la reconstruya por sí solo, al menos aproximadamente. La codificación de canal estudia las técnicas y procedimientos para llevar a cabo la detección y corrección de errores.

Conviene señalar que la codificación de canal no tiene que ver con la codificación de fuente. El codificador de canal tiene como entrada una señal digital procedente del codificador de fuente, en este caso, el codificador MPEG. El codificador de canal no "sabe" si la señal es de audio, vídeo o de otro tipo, para él se trata únicamente de una secuencia de bits cuya integridad debe proteger de alguna manera para que pueda ser reproducida fielmente en el receptor. **El papel del codificador de canal en el contexto del sistema digital de televisión se ilustra esquemáticamente en el diagrama de bloques de la figura 6.1 (capítulo 6).**

3.2 Codificación de la forma de onda - Introducción a la modulación digital.

La codificación de canal puede enfocarse de dos formas¹: codificación de la forma de onda, o bien codificación mediante secuencias estructuradas o de redundancia estructurada. La primera tiene por objeto transformar la forma de onda de la señal a fin de que el proceso de detección sea más

¹ Sklar, B. *Digital Communications: Fundamentals and Applications*. Prentice-Hall International, Inc. 1988

inmune a los errores de transmisión. La segunda, transforma las secuencias de datos en "secuencias mejores" agregando redundancia, es decir más bits a la información de entrada para que el receptor pueda detectar y corregir errores en la información transmitida. La detección de errores, por sí sola, no es suficiente para proporcionar la protección necesaria y es necesario implementar algún medio para que, una vez detectado el error pueda estimarse y reconstruirse la información perdida. Así, en tanto que la codificación de fuente tiene por función reducir al máximo el caudal binario preservando el contenido de información, la codificación de canal tiene que agregar información al código de fuente para hacer posible la detección y corrección de errores.

La codificación de forma de onda tiene por objeto transformar un conjunto de pulsos en otro conjunto mejorado, de modo que cada una de las formas de onda así codificadas sea lo menos parecida posible a cualquier otra del conjunto. En otras palabras, se busca que los coeficientes de correlación cruzada entre todos los pares de señales del conjunto sea el menor posible. De hecho, la codificación de forma de onda se asocia directamente con la *modulación digital*.

A diferencia de la transmisión en banda base, para transmitir una señal digital en un canal en banda de paso es necesario modular una portadora, generalmente senoidal, con los datos de entrada. Dichos datos pueden representar prácticamente cualquier cosa, desde la información producida por una computadora hasta una señal de vídeo o audio. El medio de transmisión puede ser una línea física, tal como un cable telefónico, una línea coaxial o una fibra óptica, o bien el aire o en general, el espacio. Las características del medio pueden ser constantes o variables como en el caso de comunicaciones móviles y, en cualquier caso, el proceso de modulación conlleva la variación de la amplitud, frecuencia o fase de una portadora de acuerdo a los datos de entrada. En el caso de comunicaciones digitales esto da lugar a tres tipos de modulación: variación de amplitud (ASK o *Amplitude Shift Keying*), de frecuencia (FSK o *Frequency Shift Keying*) y de fase (PSK o *Phase Shift Keying*), que constituyen casos especiales de modulación de amplitud, frecuencia y fase² donde en comunicaciones digitales, la señal moduladora consiste de un tren de pulsos o bien de una versión de éstos, codificada en niveles múltiples³.

Si la modulación se realiza sobre una portadora senoidal, la codificación multinivel se aprovecha por el modulador para producir cambios discretos en la amplitud (ASK), frecuencia (FSK) o fase de la portadora (PSK). Quizá lo anterior resulte más claro si se piensa que una secuencia de bits puede codificarse en grupos de dos bits. Sea, por ejemplo, la secuencia

101100100111

que puede codificarse en la forma:

10 11 00 10 01 11

Supóngase ahora que se utiliza modulación de amplitud y que a cada posible grupo de dos bits se asigna una amplitud de voltaje a una señal senoidal, por ejemplo:

00	1 V
01	2 V
10	3 V
11	4 V

La señal senoidal resultante variará su amplitud en la forma mostrada en la figura 3.2.

En este ejemplo, el voltaje de la portadora puede tener hasta cuatro diferentes niveles, que dependen de la combinación de bits a la entrada del modulador. Si la secuencia binaria se agrupara

² Haykin, S. *Digital Communications*. John Wiley & Sons. 1988.

³ El término *nivel múltiple* o *multinivel* se emplea aquí en lugar de la designación en inglés *M-ary*.

en símbolos de tres bits, se tendrían ocho niveles de amplitud y así sucesivamente. En este ejemplo se ha ilustrado en términos muy simples la modulación ASK multinivel. Algo semejante ocurre si la modulación es en frecuencia (MFSK) o en fase (MPSK), en que M indica el número de frecuencias o fases diferentes producidas por cada combinación de símbolos. El conjunto de bits que, en el ejemplo anterior producen un determinado nivel de la señal senoidal, se designa como *símbolo*. Para este caso, cada símbolo está formado por dos bits.

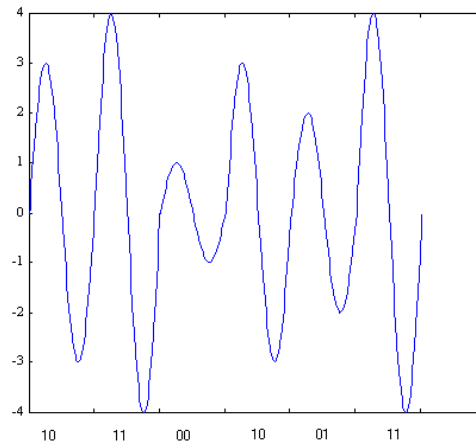


Fig. 3.2. Modulación ASK con dos bits por Hz.

De la discusión anterior se pueden extraer algunas conclusiones. La codificación de forma de onda da lugar al agrupamiento de varios bits consecutivos de una secuencia, en un solo símbolo. El símbolo resultante puede modular a una portadora senoidal ya sea en amplitud frecuencia o fase variando cada uno de ellos en forma discreta. Esta característica permite reducir el ancho de banda respecto al que sería necesario si no se hubiera codificado la forma de onda es decir, si solamente se transmitiera un bit por símbolo. La *eficiencia espectral* en este último caso sería de un bit por cada ciclo de la portadora (1 bit/Hz), en tanto que en el caso del ejemplo, la eficiencia es de 2 bits/Hz. Este aspecto es de gran importancia en canales limitados en banda en los que es necesario aprovechar al máximo el ancho de banda disponible. Así, si el caudal binario original es, por ejemplo, de 10 Mbit/s cuya transmisión requiere un ancho de banda del orden de 10 MHz, utilizando codificación de forma de onda de dos niveles, el ancho de banda requerido se reduce, teóricamente, a la mitad.

Es importante tener en cuenta que la señal ASK anterior, no es la que se transmite por el canal de radiofrecuencia, de modo que la *modulación de canal* no debe confundirse con la *modulación de RF*. Por ejemplo, si la señal anterior modula a una portadora de RF en AM completa, con un índice de modulación de 0.5 y para 1 V de amplitud de la portadora, la señal resultante sería como la que se ilustra en la figura 3.3.

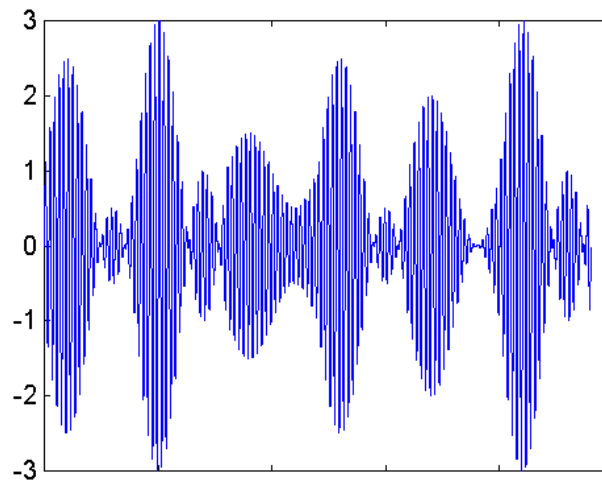


Fig.3.3. Portadora de RF modulada por una señal ASK.

El tratamiento detallado de la teoría y características de cada uno de los tipos de modulación digital queda fuera del alcance de este texto⁴.

3.3 Detección de errores

Una ventaja importante de los sistemas digitales sobre los analógicos, es su capacidad, aunque limitada, para reconocer y corregir errores causados por ruido u otras interferencias en el sistema. Si el número de errores excede los límites de capacidad de corrección, se emplea otra técnica designada como *ocultación* (concealment) de errores y que se aplica para reducir sus efectos visibles. Esta técnica es de particular importancia en las reproductoras de vídeo y en los sistemas de transmisión.

Los primeros sistemas de cómputo electrónico emplearon métodos relativamente simples de detección de errores; el principal, basado en el cómputo de *paridad* de los símbolos de información, que consiste en agregar uno o más bits a cada símbolo o palabra de código de forma tal que el número total de bits de un símbolo sea par o impar. Supóngase por ejemplo, un código octal en que cada símbolo se representa mediante tres bits en la forma siguiente:

0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Supóngase ahora que en la transmisión del símbolo correspondiente al 2 (010) se produce un error durante la transmisión y en lugar de recibir el código 010 se recibe 011 (3). El receptor no tiene posibilidad de "saber" que el símbolo recibido es erróneo y la información a la salida del receptor será falsa.

⁴ Véase bibliografía al final del capítulo para referencias en que se trata este tema con mayor profundidad.

Supóngase ahora que en el transmisor se agrega un bit adicional (paridad) a cada símbolo, de modo que el número total de bits en cada símbolo sea siempre par. El código transmitido será ahora:

0	0000
1	0011
2	0101
3	0110
4	1001
5	1010
6	1100
7	1111

El bit de paridad se indica en la posición menos significativa (a la derecha) en la tabla anterior.

En el receptor los bits de información, los tres más significativos en este ejemplo, se separan del bit de paridad y con los bits de información el receptor vuelve a calcular la paridad que corresponde a cada símbolo recibido. La paridad así calculada se compara con el bit de paridad correspondiente al símbolo recibido, y si los bits de paridad recibidos y los generados en el receptor no coinciden, será indicación de que ha habido un error en la transmisión. Sin embargo, en estas condiciones, el receptor no puede saber si el error está en los bits de información recibidos o en el bit de paridad recibido y tampoco tiene posibilidad alguna de estimar cuál debió ser el símbolo correcto ni de corregirlo. Tampoco tiene posibilidad de saber si el error alcanza sólo a un bit o a más en un mismo símbolo. El proceso anterior se resume en el diagrama de bloques de la figura 3.4.

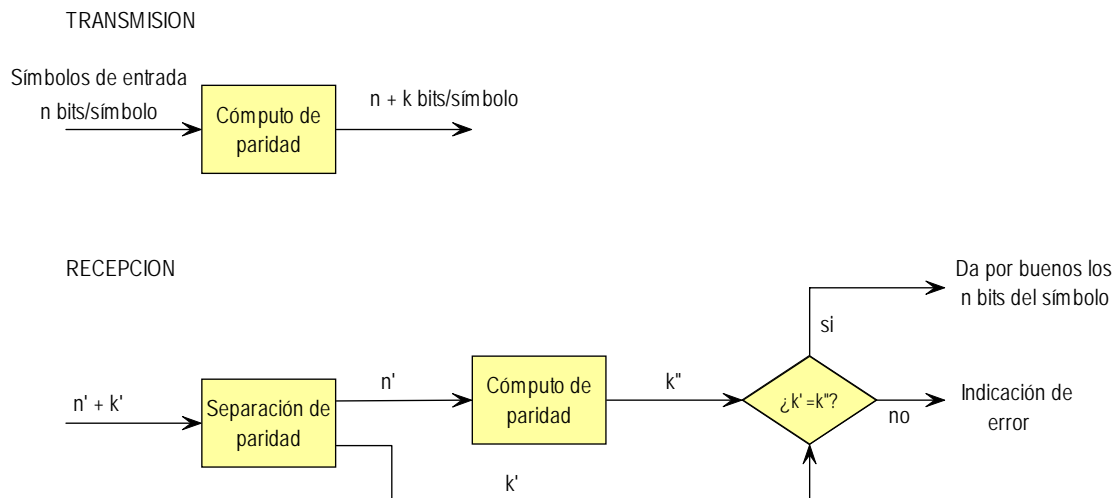


Fig. 3.4. Empleo de paridad en la detección de errores

En general, en el transmisor, por cada símbolo de n bits de longitud, se computan k bits de paridad y se transmiten como un símbolo de longitud $n+k$ bits. En el receptor, se recibe un símbolo $n'+k'$, de la misma longitud que el transmitido. Es decir n' representa ahora un símbolo de la misma longitud que n , pero cuyos bits pueden haber sido corrompidos durante la transmisión. Lo mismo ocurre con los bits de la paridad recibida, k' . La primera acción del receptor, una vez que se ha demodulado la señal, es separar los n' bits de información de los k' bits de paridad. Con los n' bits de información, el receptor calcula la paridad que correspondería a ese símbolo recibido, k'' , y la compara con los bits de paridad recibidos, k' . Si $k'=k''$, el receptor asume que $n'=n$ y da por válido el símbolo recibido. En caso contrario genera una señal de error que será utilizada para tomar las acciones necesarias, por ejemplo solicitar la retransmisión del símbolo o intentar corregirlo en el propio receptor.

Como puede apreciarse, este método de detección de errores es muy simple y considerablemente limitado, ya que no permite identificar si el error está en los bits de información o de la propia paridad y, por otra parte, no permite inferir la naturaleza del error ni corregirlo. Aún así, su propia simplicidad hace que continúe siendo un método atractivo, fácil de implementar y de bajo costo, suficiente en muchas aplicaciones, pero no en transmisión de información de televisión. El ejemplo anterior proporciona una idea básica del principio empleado en la detección de errores y el concepto de paridad se suele emplear extensamente para este fin, si bien con procedimientos más complejos en que se utiliza tanto la *paridad transversal* del ejemplo como la *paridad longitudinal* en que se calcula tanto la paridad por símbolo como la paridad de los bits en la misma posición relativa de una secuencia de símbolos, a fin de dar mayor fiabilidad a la detección de errores.

En los sistemas que emplean paridad, la detección por sí sola no suele permitir identificar el bit en error en un símbolo, por lo que la corrección de errores puede hacerse considerando los bloques de datos como matrices bidimensionales (longitud del bloque \times longitud de un byte o de un símbolo). Por ejemplo un bloque de 8 bytes de longitud, puede considerarse como una matriz de 8×8 bits. Asignando un bit de paridad a cada fila y a cada columna de la matriz, pueden localizarse errores de un bit en la intersección de la fila y columna en que se detecta el error y el bit dañado puede corregirse invirtiendo su valor. Esta forma de corrección de errores se designa como código cruzado (crossword) o de producto.

El esquema anterior de corrección es adecuado únicamente para errores de un solo bit por bloque y se han desarrollado sistemas más sofisticados que pueden corregir más errores. Dos de los códigos más utilizados en televisión digital son el código de Hamming y el de Reed Solomon.

3.4 Código de Hamming de protección contra errores

Este código fue inventado por Richard Hamming en 1949 y es un código binario capaz de *corregir*⁵ errores de un bit por símbolo y de detectar hasta dos errores de bit por símbolo, pero no de corregirlos. Es un código que se ha utilizado extensamente en las memorias RAM de computadoras y resulta una buena elección cuando los errores son aleatorios. Si los errores ocurren en ráfagas, como suele ocurrir en la transmisión terrestre de televisión digital, es necesario utilizar otros códigos más complejos como los códigos no binarios de bloque, o los códigos convolucionales.

El código de Hamming utiliza bits adicionales de paridad para la detección, y si es posible, corrección de errores. El número de bits de paridad, o verificación de errores está dado por la regla de Hamming y es función del número de bits de información transmitidos. Esta regla se expresa mediante la desigualdad siguiente:

$$d + p + 1 \leq 2^p$$

donde d es el número de bits de información o datos y p el número de bits de paridad. Los bits de paridad se agregan a los de información, constituyendo así una palabra o símbolo de Hamming. El tamaño, c , de esta palabra es, simplemente, $d + p$ y el código se designa como (c,d) . El ejemplo de la sección 3.3, constituye un caso particular en que $p = 1$ y, aunque se emplea en algunos casos debido a su simplicidad, aquí resulta de utilidad únicamente para ilustrar el concepto de paridad. En el código de Hamming, por lo general se emplean tres o más bits de paridad, como se ilustra con el ejemplo a continuación para un código $(7,4)$, es decir, con siete bits en total, de los cuales cuatro son de información y los tres restantes de paridad. El código del ejemplo es de paridad par. Un código que cumple la regla anterior de Hamming se designa como *código perfecto* del que el código $(7,4)$ es un ejemplo.

⁵ Es importante no confundir la *detección* de errores con su *corrección*. Algunos códigos sólo pueden detectar errores, pero no corregirlos, otros son capaces de detectar errores, pero no de corregirlos todos. Este es el caso más general de los códigos FEC.

Las palabras o símbolos de Hamming se generan multiplicando los bits de información por una matriz generadora G , en aritmética de módulo 2 y los símbolos resultantes constituyen vectores de código (c_1, c_2, \dots, c_n), que consisten de los bits de información originales y de los bits de paridad calculados. La matriz generadora utilizada en la construcción de los códigos de Hamming consiste de una matriz de identidad (I) y una matriz generadora de paridad (A), de modo que:

$$G = [I : A]$$

por ejemplo:

$$G = \left[\begin{array}{cccc|ccc} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{array} \right]$$

Supóngase que el símbolo a transmitir, compuesto por cuatro bits sin paridad es $[1 \ 0 \ 0 \ 1]$. La multiplicación del vector de 4 bits (d_1, d_2, d_3, d_4) por G , da como resultado un vector de símbolos de 7 bits de la forma ($d_1, d_2, d_3, d_4, p_1, p_2, p_3$): $c = [1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1]$.

Es evidente que la partición A de G es la responsable de la generación de los bits de paridad. Cada columna de A representa un cálculo de paridad computado sobre un subconjunto de d . La regla de Hamming requiere que $p = 3$ para un código (7,4), por consiguiente, A debe contener tres columnas que produzcan tres bits de paridad. Si las columnas de A se eligen de forma que cada columna sea única, (p_1, p_2, p_3) representa la paridad calculada de tres subconjuntos distintos de d . La validación del símbolo o palabra de código recibida, r , se realiza multiplicando esta palabra por una matriz de verificación de paridad de la forma

$$H = [A^T \ | \ I]$$

donde A^T es la matriz transpuesta de A , es decir

$$H = \left[\begin{array}{cccc|ccc} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{array} \right]$$

Multiplicando la matriz H por el símbolo recibido (información + paridad) se calcula un vector s , designado como *síndrome*⁶

$$s = H r$$

Si el símbolo recibido es igual al transmitido, el valor de s es cero y no es necesaria la corrección del error.

Supóngase ahora que durante la transmisión ocurre un error en un bit y que el símbolo recibido es, por ejemplo, $r = [1 \ 0 \ 1 \ 1 \ 0 \ 0 \ 1]$. Al efectuar la verificación de paridad en el receptor, el vector del síndrome calculado resulta $[1 \ 0 \ 1]$, que corresponde a la tercera fila de la porción A de la matriz generadora G . Esto identifica como erróneo al tercer bit de r y la corrección se lleva a cabo simplemente invirtiendo el tercer bit de r .

⁶ Síndrome: Por analogía con el término médico utilizado para designar un patrón de síntomas indicativo de una enfermedad.

Si éste es cero, la información recibida es correcta. Si s contiene elementos distintos de cero, el bit erróneo puede determinarse analizando que comprobaciones de paridad fallaron, siempre y cuando sólo haya un bit erróneo. Por ejemplo, si $r = [1\ 0\ 1\ 1\ 0\ 0\ 1]$, $s = [1\ 0\ 1]$ y este síndrome es igual a la tercera columna de H , lo que corresponde al tercer bit de r , el bit erróneo.

Los bits de datos de un bloque se transmiten sin alteración y van seguidos de un número específico de bits de paridad. Esto se conoce como *código sistemático* y, en el código Hamming, la matriz para la generación de los bits de paridad se diseña de manera que el patrón de los bits de paridad en error o sin él, proporciona la dirección de memoria de los bits de datos erróneos. En la práctica, Los códigos de Hamming son fáciles de implementar en hardware y pueden detectar y corregir con facilidad, errores de un bit por bloque. Eventualmente pueden detectarse errores adicionales, pero no pueden corregirse. El ejemplo anterior ilustra de forma simple el funcionamiento del código Hamming, si bien se emplean también códigos no sistemáticos en que los bits de paridad se intercalan entre los de información.

Una forma especial del código Hamming es el código de verificación de redundancia cíclica⁷ (CRC) y se utiliza con frecuencia en transmisión de datos. En televisión, para la transmisión digital de audio y vídeo se utiliza el código Reed-Solomon (R-S), mucho más potente que el de Hamming y, en algunos casos, como en radiodifusión terrestre, se utiliza el código R-S concatenado a otro código convolucional.

3.5 Efectos de los errores en la información de vídeo

El efecto visual de un error en un bit depende de la posición significativa de ese bit. por ejemplo, en un sistema PCM con cuantificación a 8 bits, un error en el bit menos significativo (LSB) del símbolo representa sólo una parte en 256, por lo que el efecto será apenas notable. Sin embargo, un error en el bit más significativo (MSB), puede producir un deterioro importante de la señal. Este rango de sensibilidades es objeto de consideración en algunos sistemas de protección contra errores, aunque actualmente no es ya un tema tan importante, puesto que los sistemas modernos de protección contra errores son capaces corregir casi todo los errores de un bit. Además, todos los sistemas que utilizan compresión pueden hacer que todos los bits sean igualmente significativos, con lo que un error de un solo bit puede ser completamente visible. De hecho, la corrección de errores es muy importante en los sistemas que comprimen la información.

3.6 Tipos de control de errores

Puesto que los bits sólo tienen dos estados, la detección de un error de bit puede corregirse fácilmente, simplemente invirtiendo el valor del bit. La ocultación de errores, que se aplica principalmente en reproductores de vídeo, es lo que podría designarse como un procedimiento de último recurso cuando no es posible corregir el error, por ejemplo, cuando el error afecta a un número considerable de bits. El principio en que se basa la ocultación de errores es la interpolación, tanto espacial como temporal, de bits adyacentes. Sin embargo, la aplicación adecuada de la ocultación de errores requiere del conocimiento de la existencia del error, por lo que siempre es necesario un medio para poder detectarlo.

⁷ Un código cíclico es aquél en el que cualquier permutación circular de una palabra de código produce también una palabra de código.

El hecho de agregar más bits a un símbolo para su transmisión constituye una forma de hacer redundante la información y hay dos formas básicas de aplicar la redundancia en el control de errores. La primera consiste en detectar los errores y retransmitir los símbolos erróneos. Así, en el ejemplo anterior, al detectarse un error en el receptor se ignora la información recibida sin intentar corregirla y se solicita al transmisor el reenvío de la información hasta lograr una recepción sin errores. Tal es la práctica común por ejemplo, en transacciones bancarias. Es claro que en estas condiciones se requiere comunicación bidireccional continua (*full-dúplex*) entre transmisor y receptor.

El segundo tipo de control de errores es el que se designa como *corrección de errores hacia adelante* o FEC (*forward error correction*), en el que la comunicación es unidireccional del transmisor al receptor. En este caso, los bits de paridad agregados a la información en el transmisor, se configuran de forma tal que no sólo permiten la detección de errores, sino su corrección sin necesidad de requerir retransmisión. Tal capacidad de corrección es limitada y, en general, no es posible corregir todos los errores y recuperar la información original transmitida. Cuando no es posible corregir el error, el receptor emplea alguna técnica para enmascararlo, por ejemplo interpolando valores entre la información anterior y la posterior al error. Estas técnicas de enmascaramiento u ocultación de errores no sólo se emplean en la transmisión digital; desde hace bastantes años se aplican en las máquinas reproductoras de vídeo para enmascarar la información perdida como consecuencia de deficiencias microscópicas en el material ferromagnético de las cintas de vídeo⁸.

La detección y control o corrección de errores requiere, por tanto, de métodos de codificación en el transmisor para la verificación de paridad, designadas como *secuencias estructuradas* porque constituyen métodos de inserción de redundancia estructurada en los datos de la fuente de forma tal que pueda detectarse la ocurrencia de errores y, en algunos casos, corregirlos. La codificación de secuencias estructuradas se subdivide en dos categorías: *codificación de bloques* y *codificación convolucional*.

Ante de iniciar la discusión de estos métodos, conviene hacer notar que la entrada al codificador de canal son símbolos de un *alfabeto*⁹ de ciertas características, que representa la señal de salida del codificador de fuente y cuya salida son símbolos de otro alfabeto, en el que se ha incluido la redundancia necesaria para el manejo de los errores. Es claro que la salida del codificador de canal tendrá, en general, más bits por símbolo que su entrada, por lo que el ancho de banda requerido o la velocidad de transmisión, será mayor. Este tipo de codificación se designa como *codificación de bloques* y es importante notar que, en las condiciones anteriores el codificador de canal no expande el alfabeto, únicamente le agrega redundancia, es decir, utiliza más bits de los estrictamente necesarios para codificar cada símbolo. La redundancia permite la detección y corrección de errores y el precio que se paga es un caudal binario mayor y, por consecuencia, mayor ancho de banda que el requerido para la transmisión del alfabeto original. Por otra parte, es posible expandir el alfabeto mediante un tipo de codificación designado como *codificación convolucional*, sin aumentar el ancho de banda, con lo que también es posible corregir errores empleando técnicas distintas a las de la codificación de bloques. Los sistemas de transmisión de televisión utilizan ambos tipos de codificación, de bloques y convolucional combinadas o concatenadas, con lo que aumenta la eficiencia de codificación y se consigue mayor robustez contra errores.

⁸ Esta técnica se designa como *compensación de "dropouts"*, en que la información perdida se recupera interpolando los niveles de señal inmediatos a los puntos deficientes (*dropouts*) en la cinta.

⁹ Por *alfabeto* se entiende aquí una secuencia ordenada de símbolos como los ocho del código octal, utilizados en el ejemplo de la Sección 9.3.

3.7 Decodificaciones dura y suave.

Asumiendo que la entrada al receptor es una señal en alguna banda de RF, ya sea que se trate de transmisión radioeléctrica o por cable, las primeras etapas del receptor corresponden, en términos generales, a las de un receptor convencional, es decir, amplificador frontal, mezclador, filtros y amplificadores de frecuencia intermedia y, finalmente, el detector o demodulador. La salida del detector, que debe suministrarse al decodificador, es la que es de interés en esta discusión y da lugar a dos tipos de decodificación, designados como *dura* o *firme* (*hard*) y *suave* (*soft*). La primera se asocia usualmente a los códigos de bloque y la segunda a los convolucionales. En términos muy simples, puede decirse que la decodificación dura se da cuando el demodulador entrega al decodificador de canal la información tal como la detecta sin intentar efectuar ninguna corrección, dejando esa tarea al decodificador de fuente. En la decodificación suave, por el contrario, el demodulador de canal detecta el error y hace una estimación del símbolo más probable, pasándolo luego al decodificador de fuente.

3.8 Codificación de canal en DVB

En la figura 3.5, se muestra el esquema que sigue la codificación de canal en DVB-T y que es similar la utilizada en DTV.

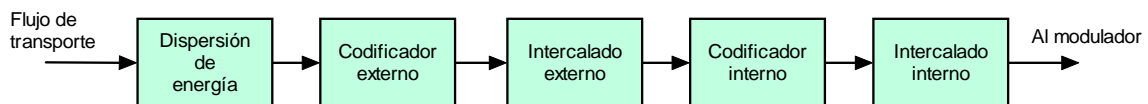


Fig. 3.5. Codificador de canal en DVB-T

Dispersión de energía.

La entrada al codificador de canal es el flujo de transporte resultante del multiplexor de transporte, constituido por paquetes de 188 bytes. La primera operación que se realiza en el codificador de canal es la dispersión de energía o aleatorización¹⁰, en que la secuencia de entrada se convierte a una secuencia pseudoaleatoria, cuyo espectro es similar al del ruido blanco. Esta operación distribuye los bits de manera que no haya grupos grandes de bits uno o de bits cero juntos. En transmisión digital por cable esta función también es necesaria para reducir la componente de corriente continua de la señal. El aleatorizador utilizado en DVB tiene la forma mostrada en la figura 3.6.

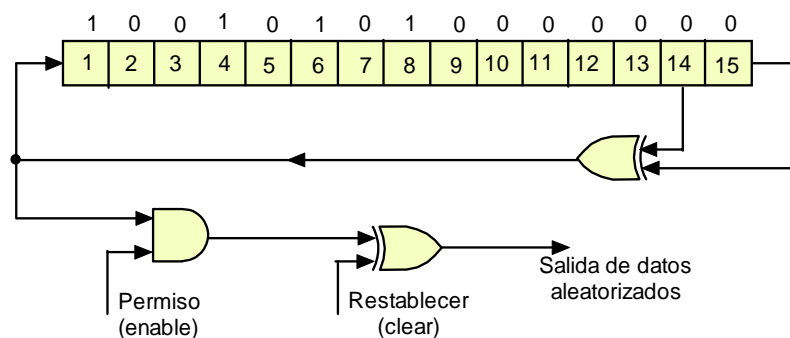


Fig. 3.6. Aleatorizador

¹⁰ Scrambling

El aleatorizador esta constituido por un registro de desplazamiento de 15 posiciones y se inicia por la secuencia indicada en la parte superior de la figura, al principio de cada ocho paquetes de transporte¹¹. Para iniciar la secuencia de desaleatorización en el decodificador de canal del receptor, se invierten los bits del byte de sincronismo de los paquetes MPEG-2. Los datos de entrada se aplican a la entrada de permiso (enable) de la puerta Y. Por otra parte, el proceso de aleatorización debe estar activo permanentemente para evitar la transmisión de la portadora sin modulación. El espectro resultante de la dispersión de energía es similar al mostrado en la figura 1.12 (Cap. 1).

Codificador externo.

En DVB-T se emplean dos códigos concatenados de protección contra errores, uno de bloque designado como *externo* y otro convolucional, designado como *interno*. El código externo es Reed-Solomon (R-S), que se trata en la sección 3.10, de tipo (204,188), con 188 bytes de datos y 16 de paridad, capaz de corregir hasta ocho bytes erróneos. El sistema DVB emplea un código R-S (208,188), 20 bytes de paridad y capaz de corregir hasta 10 bytes erróneos.

Intercalado¹² externo.

El intercalado tiene por objeto alterar el orden de la secuencia de datos para dispersar los errores en ráfaga. En el entorno hostil de transmisión terrestre, los efectos multicamino deterioran la señal dañando a grupos de símbolos. Dado que el código de protección contra errores proporciona una capacidad de corrección limitada, en el sentido de que es capaz de corregir sólo hasta ocho bytes en un paquete, pero *no contiguos*, el intercalado tiene por objeto desordenar la secuencia dispersando los símbolos adyacentes en la secuencia original. Una forma de explicar este proceso es suponiendo una secuencia de símbolos como la siguiente:

ABCDEFGHIJKLMNOP

Supóngase que ocurre un error en ráfaga y se dañan los símbolos contiguos DEFG. La secuencia recibida sera:

ABCXXXXHIJKLMNOP

En que las X representan a los símbolos dañados. El decodificador del receptor no será capaz de corregir ese error. Ahora bien, si en el proceso de intercalado en el codificador se reordena la secuencia anterior en forma de una matriz que se escribe fila a fila con la información de entrada, por ejemplo:

```
A B C D
E F G H
I J K L
M N O P
```

El “intercalador” entrega ahora la información, pero leyendo la matriz columna a columna, es decir:

```
A E I M
B F J N
C G K O
D H L P
```

¹¹ Seseña, J. “The DVB satellite, cable and SMATV systems. EBU Technical Review. Winter 1995.

¹² El término en inglés es *interleaving* que aquí se ha traducido como intercalado. Quizá un término más adecuado sería el de “barajado”. El significado preciso aquí es *la alteración del orden de la secuencia*.

Así, la secuencia de símbolos entregada al codificador interno será:

AEIMBFJNCGKODHLP

La entrada al decodificador del receptor será la secuencia anterior dañada por el error en ráfaga:

AEIXXXNCGKODHLP

En el decodificador esta secuencia se reordena a la forma de la secuencia en el codificador, antes del intercalado, con lo que se tiene:

AXCDEXGHIXKLXNOP

Como se puede apreciar, los símbolos erróneos en la secuencia recibida y reordenada están ahora aislados y la probabilidad de corregirlos es considerablemente mayor. En DVB el intercalado se basa en un procedimiento conocido como *algoritmo convolucional de Forney*, que permite dispersar los errores cada 12 símbolos ($I = 12$).

Codificador interno.

La señal recibida no sólo está sujeta a deterioro por los efectos multicamino. También se corrompe por el ruido gaussiano aditivo, de modo que para aumentar la robustez de la protección contra errores se agrega un codificador interno que, además de brindar protección adicional, permite emplear una relación flexible de código para los diferentes estándares DVB. Se trata de un codificador convolucional y se aplica la técnica de “perforación¹³” del código (véase la sección 3.14 de este capítulo). Las relaciones o tasas de código que se contemplan pueden ser 1/2, 2/3, 3/4, 5/6 y 7/8. En el decodificador convolucional se emplea el *algoritmo de Viterbi*.

Intercalado interno.

Por la naturaleza del proceso de codificación convolucional, pueden aparecer errores en ráfaga a la salida del codificador, de modo que para eliminarlos se aplica otro intercalado a su salida.

Como puede inferirse de lo anterior, el proceso de codificación de canal es complejo, pero a pesar de su complejidad es capaz de corregir los errores de manera óptima, repartiendo la tarea entre el código de bloque y el convolucional, además de los procesos de intercalado y, en cierta medida, de la aleatorización que, además de dispersar la energía, contribuye a reducir los errores.

3.9 Códigos de bloque¹⁴.

Los códigos de bloque son códigos de longitud fija en que cada bloque está constituido por un número fijo de símbolos de información a los que se agrega una cantidad fija de símbolos de paridad, mediante los cuales es posible detectar errores y, eventualmente corregirlos. Por ejemplo, un código de bloque especificado como (208,188) indica que hay 188 símbolos de información y 20 de paridad. Los bits de paridad agregan redundancia a la información con el fin de detectar y corregir errores y dan como resultado un aumento en el ancho de banda y, por consecuencia, reducción en la eficiencia de transmisión. Este es el precio que hay que pagar a cambio de disminuir la tasa de errores. Entre los códigos de bloque bien conocidos están los de Hamming, Golay, BCH y, en particular para el caso de televisión, el código Reed-Solomon, que aunque no es

¹³ *Puncturing*

¹⁴ Para un tratamiento más amplio del tema consúltese por ejemplo el texto de Bernard Sklar, *Digital Communications. Fundamentals and Applications*. Prentice Hall International, Inc. 1988 y Simon Haykin, *Communication Systems*, 4th. Edition. John Wiley & Sons, Inc. 2001.

el más eficiente, es para el que se han desarrollado algoritmos más rápidos. El tratamiento detallado de la teoría de los códigos de bloque es extensa y excede también el alcance de esta obra. En el Apéndice a este capítulo, se presenta brevemente la teoría de generación de los códigos de bloque, así como de la detección de errores.

3.10 Códigos Reed-Solomon

La codificación Reed-Solomon se utiliza no sólo en transmisión digital de televisión. se emplea también en CD de audio, CD-ROM y grabadoras digitales de televisión. Es un esquema de codificación de bloque que puede corregir ráfagas de errores hasta un cierto límite determinado por la cantidad de redundancia¹⁵ con que se diseñe el código. Aunque el procesado computacional de estos códigos es sumamente complejo, ha sido implementado en circuitos integrados en gran escala (VLSI) y en la actualidad se incluye en el hardware de sistemas a un costo relativamente bajo.

Los códigos Reed-Solomon (R-S) se hicieron públicos en la literatura técnica en 1960 y desde entonces han sido utilizados en numerosas aplicaciones, que van desde las comunicaciones interplanetarias, hasta los reproductores de CD, telefonía móvil y televisión digital. Estos códigos pertenecen a la familia de códigos de bloque, en que el codificador procesa un bloque de símbolos sin codificar, a los que agrega redundancia para producir otro bloque, de mayor longitud, de símbolos codificados, a diferencia de la *codificación convolucional*, en que el codificador no trabaja sobre bloques de símbolos de longitud fija, sino sobre el flujo continuo de estos. La codificación convolucional puede interpretarse como la convolución de los símbolos del mensaje con la respuesta impulsiva del codificador.

Los códigos R-S son códigos no binarios, sistemáticos, cíclicos y lineales que operan sobre símbolos consistentes de varios bits y tienen buenas propiedades para la corrección de errores en grupo o ráfaga ya que la corrección se realiza a nivel de símbolo, no de bit . Una longitud común de símbolo para los códigos no binarios es 8 bits (un byte), pero de hecho, pueden tenerse símbolos de cualquier longitud. Trabajando con símbolos en la decodificación se posibilita la corrección de símbolos con ráfagas de ocho bits erróneos tan fácilmente como de símbolos con un solo bit erróneo. Se dice que un código es sistemático cuando las palabras codificadas contienen los símbolos del mensaje sin alteración. En el codificador se aplica una función matemática reversible para generar los símbolos de redundancia o paridad y la palabra de código se forma agregando los símbolos de paridad a los del mensaje. La implementación del código se simplifica cuando éste es sistemático, es decir, los símbolos del mensaje se mantienen en su forma original y los de paridad se añaden al final de la secuencia de aquellos, como se ilustra en la figura 3.7.

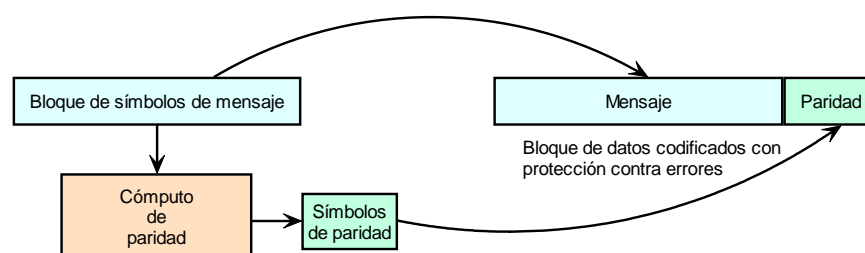


Fig. 3.7. Esquema general de la codificación de bloque.

Los códigos R-S son cíclicos. Esto significa que si se desplaza circularmente una palabra válida del código, se produce otra palabra válida. Este tipo de códigos es popular ya que hay técnicas de decodificación muy eficientes para ellos. Finalmente, los códigos R-S también son lineales, lo que

¹⁵ En inglés, los bits de redundancia que se agregan a los datos se designan frecuentemente como *overhead*.

significa que si se suman dos palabras válidas cualesquiera del código, se tiene como resultado otra palabra válida.

Si n es el número total de símbolos y k el número de símbolos del mensaje original, el número de símbolos de paridad agregados es de $n - k$. A la relación entre el número total de símbolos y los símbolos del mensaje original se designa como *relación de código* o *relación* o *tasa de codificación*¹⁶ y se designa como $R = n/k$.

Cada paquete en el sistema de transporte tiene una carga de información de 188 bytes, entre los que se encuentra no sólo la información de vídeo o audio comprimida por el codificador de fuente, sino también los encabezados de enlace y adaptación. La codificación de bloque agrega a estos paquetes 16 bytes de paridad para su transmisión, de modo que el código puede describirse como R-S (204,188).

3.11 Código R-S como un código cruzado o de producto

Si se define un macrobloque de datos que contiene un cierto número de bloques R-S se consigue una capacidad de corrección de errores mucho más poderosa que con la simple implementación R-S bloque a bloque. Esto se utiliza frecuentemente con los datos de un cuadro de imagen completo o con los datos de una pista de cinta magnética en una reproductora de vídeo. La codificación R-S se aplica a los datos de entrada y luego, todo el macrobloque de datos se almacena en memoria en forma de matriz bidimensional. Los datos se extraen de la memoria leyendo la matriz de forma perpendicular a la forma en que se almacenó y se aplica un segundo proceso de codificación R-S. Por ejemplo, los datos pudieron almacenarse en la matriz fila a fila y luego extraerse columna a columna. La primera codificación R-S, es decir, sobre los datos de entrada, se designa como *externa*¹⁷ y la segunda codificación, como *interna*. En el receptor, la decodificación interna se realiza primero, se efectúa luego la transformación de la matriz y se realiza la decodificación externa para recuperar los datos originales. Este esquema permite la corrección de ráfagas grandes de errores, tales como las que ocurren a causa de las manchas (*dropouts*¹⁸) en las cintas magnéticas.

En la codificación R-S, un bloque de datos está compuesto por símbolos que pueden ser bytes (8 bits) y, a partir de los datos se calculan un cierto número de símbolos de protección contra errores. Estos símbolos se transmiten junto con los datos para ser utilizados en el decodificador R-S del receptor. Cuanto mayor sea el número de símbolos de protección, respecto al tamaño del bloque de datos, más potente será la corrección. Los bloques se especifican mediante dos números, por ejemplo (208,192), que significan que hay 192 símbolos de datos por bloque, con 16 símbolos de protección contra errores, de modo que la longitud total del bloque es de 208 símbolos.

Aunque la codificación R-S ofrece una protección muy robusta contra errores y puede manejar errores múltiples en cada bloque de datos, en algunos medios de transmisión, como en el caso de transmisión terrestre en que se tienen efectos multicamino severos, pueden ocurrir ráfagas de errores superiores a los que es capaz de manejar el código R-S. Este problema puede mitigarse alterando el orden de la secuencia de datos entre bloques (*interleaving*¹⁹), de modo que los datos de cada bloque, incluyendo los símbolos de protección, no se transmiten en su secuencia original, sino que son “barajados” después de ser codificados, de modo que los errores se dispersan en varios símbolos. El receptor conoce la secuencia de alteración y recupera el orden original antes de la decodificación R-S. Lo anterior da lugar a que, en caso de ocurrir un error que afecte a varios símbolos consecutivos, al volver a reordenarse ese error se habrá dispersado en varios símbolos,

¹⁶ El término en inglés es *code rate*.

¹⁷ No confundir con la codificación externa e interna en el código R-S.

¹⁸ El término *dropout*, traducido aquí como *mancha*, se refiere a defectos microscópicos en el material magnético de una cinta de vídeo que impiden la grabación adecuada y dan lugar a ráfagas de errores.

¹⁹ En general, aquí se empleará el término *intercalado* o *barajado* para expresar la alteración del orden original de la secuencia.

afectando sólo a unos pocos bits de cada símbolo recuperado, lo que hace posible la corrección de errores de este tipo.

El proceso de intercalado decorrelaciona los errores de ráfaga. Mediante la alteración del orden de una secuencia de código de longitud determinada, el código puede corregir una cantidad de errores que de otra forma, requeriría de un código de mucho mayor longitud que sin intercalado.

El codificador opera sobre los símbolos de los bloques del mensaje a codificar, que pueden considerarse como un vector fila, formado por una secuencia de un número fijo de bits, mediante una matriz generadora del código. La salida del codificador es una matriz que contiene los símbolos del mensaje original y los símbolos de paridad. La matriz generadora también es conocida por el decodificador. Para una matriz generadora dada existe una matriz de dimensiones tales que sus filas son ortogonales a las de la primera y el producto de la matriz generadora por la matriz transpuesta de la segunda es cero. Esta segunda matriz recibe el nombre de *matriz de verificación de paridad*. En el apéndice a este capítulo se presenta un tratamiento algo más amplio de este proceso.

En el decodificador se realiza la verificación de paridad sobre los datos del mensaje recibido mediante la operación descrita antes, es decir multiplicando la matriz del mensaje por la matriz de verificación de paridad. El resultado de esta multiplicación se designa como *síndrome* (véase sección 3.4). Si el síndrome es cero, el mensaje recibido es válido. Si es diferente de cero, la matriz de síndrome permite localizar la posición de los símbolos erróneos y corregirlos utilizando una tabla de búsqueda residente en el decodificador. Entre las operaciones requeridas en el proceso de corrección de errores se emplean el algoritmo de Euclides y el algoritmo de búsqueda de Chien-Forney, que no se tratan aquí.

El hecho de alterar la secuencia de datos tanto a la entrada como a la salida del codificador, hace que esta codificación sea sumamente robusta frente a errores. Para dar una idea aproximada de la potencia de este tipo de codificación, supóngase por ejemplo que la tasa de errores de un caudal binario sin codificación contra errores es, digamos de 10^{-3} , es decir, un bit en error por cada 1000 bits. Con codificación de errores la tasa de errores puede reducirse a menos de 10^{-6} , menos de un error por cada millón de bits.. Estas cifras son únicamente indicativas y, en la práctica, utilizando codificación concatenada, se consiguen tasas de error del orden de 10^{-9} o menores.

3.14 Codificación convolucional

En el entorno relativamente hostil de la radiodifusión terrestre de televisión, en que los efectos multicamino pueden causar deterioros frecuentes a la información digital, el empleo de sólo una etapa de codificación de bloque como protección contra errores no es suficiente, ya que aún a pesar de la considerable reducción de la tasa de errores, la enorme cantidad de información transmitida daría lugar a que los errores no corregidos en la recepción fueran suficientes como para ser perceptibles en la imagen con relativa frecuencia, por ejemplo con intervalos de varios segundos entre sí, de modo que los artefactos en la imagen serían suficientes como para hacer la visualización molesta o desagradable al espectador.

Aunque en algunos sistemas, como los de cable, bajo ciertas condiciones puede ser necesaria sólo una codificación modesta contra errores, los sistemas de radiodifusión terrestre de televisión constituyen lo que podría designarse como un caso límite que requiere una extraordinaria robustez que difícilmente se encuentra en otros sistemas de comunicaciones. Para aumentar aún más la protección que brinda la codificación de bloques podría agregarse una segunda etapa, similar a la primera y a continuación de ésta. Sin embargo, esto traería como consecuencia mayor complejidad al sistema y un aumento del ancho de banda de transmisión que no es deseable. En televisión digital se ha optado por agregar una segunda etapa de codificación contra errores, que opera sobre la información ya codificada bajo Reed Solomon y a la que se designa como *código interno*. Este

segundo nivel de codificación de canal no se basa en codificación de bloques, sino que emplea un principio diferente, designado como *codificación convolucional*. Las dos etapas de codificación actúan de forma concatenada como se muestra en la figura 3.5. La primera o externa es la codificación de bloques R-S, la segunda es convolucional.

En la codificación de bloque se expande el alfabeto al agregar redundancia y, por tanto, aumenta el ancho de banda. Una propiedad importante de la codificación convolucional es que expande el alfabeto, pero no aumenta el ancho de banda. La codificación de bloque opera sobre bloques completos de información y es, por tanto, discreta. La convolucional opera bit a bit, o sobre bloques pequeños y puede considerarse continua.

En las secciones anteriores se trataron los códigos de bloque como herramientas para la detección y corrección de errores. Los códigos lineales de bloque se describen mediante dos números enteros, (n, k) y un polinomio o matriz generadora del código; n es el número total de bits del mensaje codificado y k , el número de bits del mensaje original, antes de la codificación. Cada palabra codificada está determinada de forma única por la palabra del mensaje original a la entrada del codificador. La detección de errores se basa en la redundancia añadida a la información del mensaje original ($n - k$ bits) y la relación k/n , designada como *tasa de código*, es la información por bit codificado y da una medida de la redundancia añadida.

La codificación convolucional, que también permite la detección y codificación de errores, se basa en principios muy diferentes a los de los códigos de bloque. En éstos, el proceso de detección y corrección de errores se realiza sobre bloques de datos de longitud fija, en tanto que la codificación convolucional opera de forma continua sobre la secuencia de datos de entrada al codificador, a nivel de bit, o de bloques pequeños de datos. Una característica adicional que distingue a la codificación convolucional de la de bloque, es que ésta opera únicamente sobre la información presente a la entrada del codificador, es decir, no tiene en cuenta la información pasada y, por tanto, no tiene memoria, en tanto que la codificación convolucional opera tanto sobre la información actual, presente a la entrada del codificador, como sobre la información pasada y se trata, por consecuencia, de un proceso con memoria. En la codificación de bloque, cada bloque es independiente de los demás bloques que forman la secuencia de datos, en la codificación convolucional, los símbolos a la salida del codificador son interdependientes.

Los códigos convolucionales se especifican mediante tres parámetros, (n, k, m) donde n es el número de bits a la salida del codificador, k el número de bits de información a la entrada de éste y m , el número de registros de memoria. La relación k/n es la relación o tasa de código, tiene el mismo significado que para los códigos de bloque y proporciona una medida de la eficiencia de codificación. En la práctica, k y n suelen variar de 1 a 8, m de 2 a 10 y k/n de 1/8 a 7/8. En algunas aplicaciones de comunicaciones con vehículos espaciales a distancias muy grandes de la tierra se han utilizado tasas de código muy bajas, del orden de 1/100 o menores.

Es común que los fabricantes de circuitos integrados especifiquen los parámetros del código convolucional como (n, k, L) en que L se designa como *longitud de constricción* y está dada por

$$L = k(m - 1)$$

L representa el número de bits en la memoria del codificador, previos al bit actual de entrada, que afectan a la generación del código.

Los elementos básicos de un codificador convolucional son un registro de desplazamiento, constituido por m elementos de memoria (flip-flops) y n generadores de señal que, en este caso, no son otra cosa que sumadores en módulo²⁰ 2 (puertas OR exclusivas). En la figura 3.8 se ilustra un decodificador convolucional simple.

²⁰ La suma en módulo 2 sigue la siguiente regla: $0 + 0 = 0$; $0 + 1 = 1 + 0 = 1$; $1 + 1 = 0$.

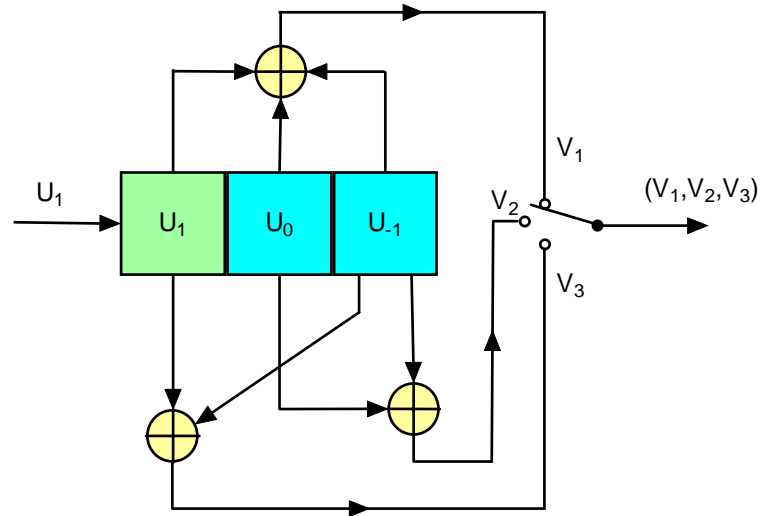


Fig. 3.8. Estructura de un codificador convolucional simple.

El codificador de la figura 3.8, utilizado como ejemplo, está constituido por un registro de desplazamiento con tres elementos de memoria y tres sumadores en módulo 2 o generadores de función, en los que se combinan los bits contenidos en las memorias (flip-flops). El código de salida se obtiene conmutando secuencialmente las salidas de los generadores de función durante el período de cada bit de entrada, es decir, por cada bit de entrada se producen, en este caso 3 bits de salida.

El primer flip-flop contiene al bit de entrada \$u_1\$ y los dos siguientes, almacenan los bits anteriores \$u_0\$ y \$u_{-1}\$. La selección de cuáles bits se suman para producir cada uno de los bits de salida, se designa como *polinomio generador* del código y, para el ejemplo:

$$v_1 = (u_1 + u_0 + u_{-1})_{\text{mod}2}$$

$$v_2 = (u_0 + u_{-1})_{\text{mod}2}$$

$$v_3 = (u_1 + u_{-1})$$

El código anterior se designa como (3,1,3) ya que \$n = 3\$, \$k = 1\$ y \$m = 3\$. La relación de código es, por consecuencia 1/3. Es decir, por cada bit de entrada se generan tres bits de salida.

Los polinomios generadores dan al código la calidad de la protección y ocurre que dos códigos de igual designación, por ejemplo (3,1,3), pueden tener propiedades completamente diferentes dependiendo de los polinomios generadores elegidos para cada uno. Así, para un código de orden \$m\$ hay múltiples polinomios posibles y no todos producen secuencias “buenas” desde el punto de vista de protección contra errores²¹.

La salida del codificador es una secuencia que depende del bit de entrada y del contenido de los flip-flops, es decir, de los bits de entrada previos. Por ejemplo el codificador (2,1,4) de la figura 3.9, genera dos bits de salida por cada bit de entrada y utiliza cuatro flip-flops o registros.

²¹ Peterson, W.W. and Weldon, Jr., E. J. *Error Correcting Codes*, 2nd Ed. The MIT Press, Cambridge, MA, 1972. Esta obra contiene una lista completa de estos polinomios. Los polinomios “buenos” se obtienen a partir de esta lista, por lo general mediante simulación.

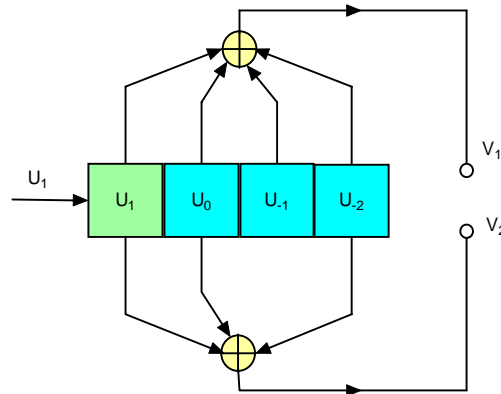


Fig. 3.9. Codificador convolucional (2,1,4)

La longitud de constricción es $L = 3$. lo que significa que hay tres bits anteriores al actual (u_1) almacenados en el registro de desplazamiento (u_0, u_1, u_2) de modo que hay ocho posibles combinaciones diferentes de estos tres bits que determinan la secuencia codificada de salida (v_1, v_2). El número de combinaciones de los bits previos al actual se designa como *estados del código* y están dados por:

$$\text{Número de estados} = 2^L$$

La tasa de código es, evidentemente, $1/2$. Para el caso particular de $k = 1$, las relaciones de código $1/2, 1/3, 1/4, 1/5$ y $1/7$, se designan con frecuencia como *códigos madre* y se pueden combinar varios de éstos para formar códigos que se designan como *perforados*²² con lo que se obtienen relaciones de código distintas a $1/n$. Así, dos códigos de relación $1/2$ se pueden implementar como un código de relación $2/3$, en que se tienen dos bits de entrada y tres de salida. La eficiencia de este código es mayor que la del código $1/2$, pero sus propiedades de corrección de errores se reducen algo²³.

Diagrama de árbol.

Partiendo de un estado inicial del registro de desplazamiento, por ejemplo (0,0,0,0), la salida es (0,0) y es posible obtener el código de salida para cada nuevo bit de entrada. Por ejemplo si el bit de entrada es cero, se ve que la salida sigue manteniéndose en (0,0). Si el bit de entrada es un 1, el estado del registro será (1,0,0,0) y la salida será ahora (1,1). Si el siguiente bit de entrada es un 0, el contenido del registro es (0,1,0,0) y la salida (1,1). Si por el contrario, el bit de entrada es un 1, el registro contendrá ahora (1,1,0,0) y la salida será (0,0). El análisis continuado en esta forma da lugar a un diagrama de árbol en que el número de ramas aumenta cada vez que entra un nuevo bit al registro. Para el caso del codificador (2,1,4) de la figura 3.9, el diagrama de árbol tiene la forma mostrada en la figura 3.10.

²² Punctured. Otra posible designación en español es *picados*.

²³ Un tratamiento, tanto de la codificación convolucional, como de los códigos perforados, queda fuera del propósito de este trabajo, en que lo que se pretende es dar una visión razonablemente suficiente para la comprensión del tema, en el contexto de la transmisión digital de televisión.

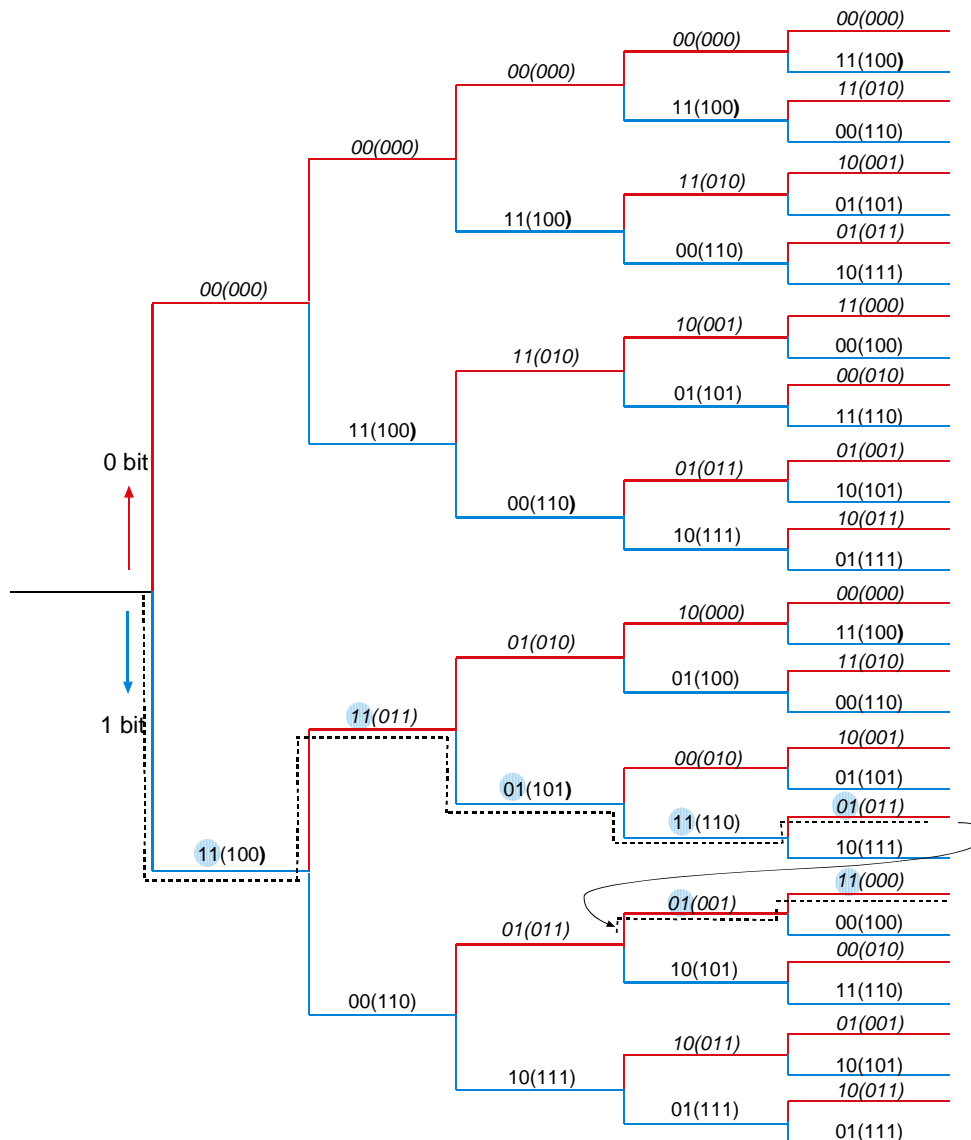


Fig. 3.10. Diagrama de árbol para el codificador de la figura 3.9

En el diagrama se avanza una rama hacia la derecha cada vez que entra un nuevo bit al registro de desplazamiento. Si el bit es 0, se avanza hacia arriba, si es un 1, hacia abajo. En cada una de las ramas se indican los valores del código de salida y, entre paréntesis, el estado de los tres bits previos al bit de entrada.

Diagrama trellis²⁴

Aunque la operación de codificación puede describirse adecuadamente mediante el diagrama de árbol, resulta poco práctico para analizar secuencias aún relativamente cortas, ya que el número de ramas crece como función de 2^L , donde L es la longitud de constricción. Si se observa la figura 3.10, se aprecia que, para el ejemplo mostrado, la estructura se repite a partir del cuarto intervalo de tiempo (t_4), es decir, después de la tercera ramificación, la estructura se repite después de L ramificaciones, donde L es la longitud de constricción. Haciendo uso de esta propiedad, es posible una representación más adecuada que el diagrama de árbol. Esta representación se designa como

²⁴ El término *trellis* significa celosía, similar a la utilizada para soportar plantas trepadoras. El uso de esta palabra se ha generalizado y no debe confundirse con un nombre propio.

diagrama trellis, de ahí que a veces la codificación convolucional se designa también como *codificación trellis*.

En la representación gráfica de la codificación convolucional suele preferirse el diagrama trellis, en cierta forma similar al de árbol, pero más sencillo y fácil de interpretar que éste. Este diagrama se muestra en la figura 3.11 y, en la dirección horizontal se tienen los instantes de tiempo discreto o de reloj, en tanto que en el eje vertical se representan todos los posibles estados de los bits del registro de desplazamiento sin incluir el bit actual o de entrada. Partiendo de un estado determinado hay dos posibles salidas según el bit de entrada sea 0 o 1. En el diagrama estos posibles estados se indican mediante líneas continuas o punteadas, las primeras si el bit de entrada es cero y las segundas si es 1.

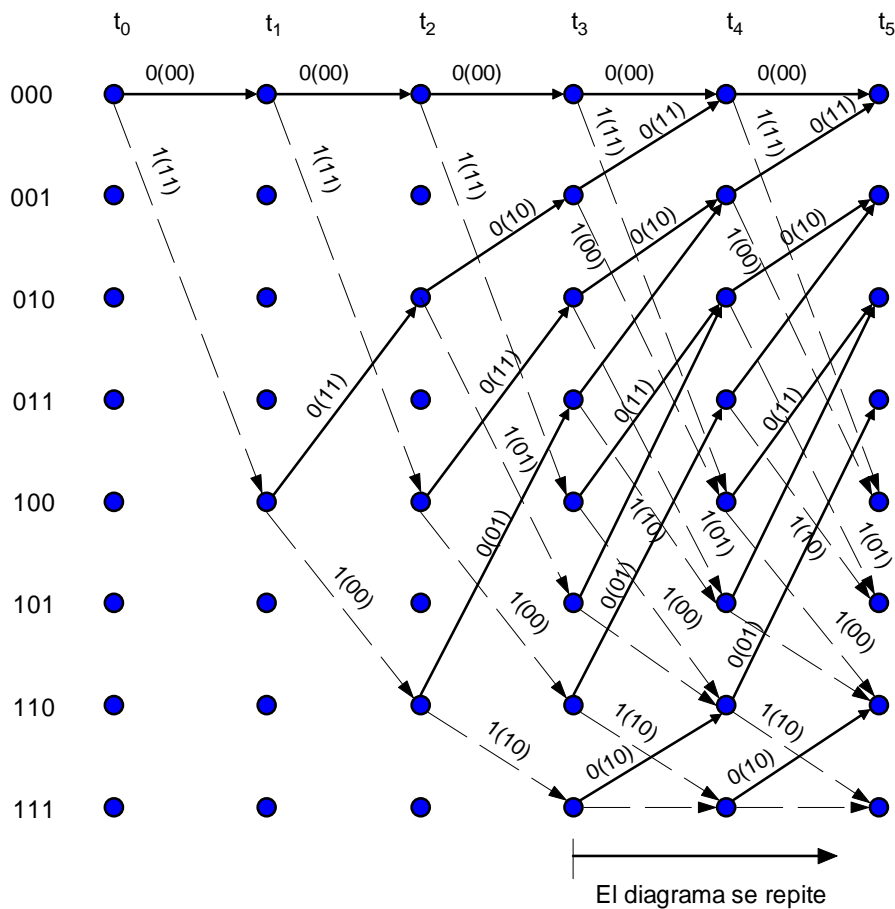


Fig. 3.11. Diagrama trellis para el codificador (2,1,4) de la figura 3.9.

En el diagrama trellis anterior, se tienen a la izquierda los estados correspondientes a los bits u_0 , u_1 y u_2 del registro de desplazamiento de la figura 3.9 y, en la parte superior, los instantes en que ocurren las transiciones al haber un bit presente a la entrada. Los cambios de estado se indican, con líneas continuas, cuando el bit a la entrada es un cero y, con líneas punteadas cuando el bit de entrada es un uno. En cada una de estas ramas se indica el bit de entrada y, entre paréntesis, el código resultante de salida

De este diagrama pueden obtenerse algunas propiedades de importancia para la corrección de errores en la decodificación.. Así, si por ejemplo, en el instante t_0 el código de salida fue 00, en el instante t_1 el código de salida será 00, si el bit a la entrada es un 0, o bien 11 si el bit a la entrada es

un 1, pero no podrá ser 01 ni 10. Es decir, de un estado 00, el codificador sólo puede pasar al estado 00 o 01. Si en el proceso de decodificación después de un estado 00 se tiene un estado 01 o 10, será indicación de que ha habido un error en la transmisión y será necesario corregirlo. El decodificador no puede “saber” si el símbolo correcto deber ser 00 o 01, por lo que tendrá que hacer una decisión con respecto al bit de entrada. Si el decodificador realiza una decisión binaria, es decir, asigna un valor de 0 o de 1 al bit de entrada en que se ha producido el error, tal decisión se designa como *dura*. Si, por el contrario, el decodificador en lugar de hacer una decisión binaria, realiza una estimación más precisa de la probabilidad de que el bit correcto de entrada sea 1 o 0, utilizando mayor número de niveles de acuerdo a la probabilidad de ocurrencia de uno u otro bit, se dice entonces que el decodificador ha realizado una decisión *suave*. Para ello, en la decodificación se utiliza el *algoritmo de Viterbi*²⁵, que fue inventado por Viterbi en 1967 y que, esencialmente, realiza una decodificación designada como de *máxima similitud*. La base de la decodificación de Viterbi radica en que, si dos trayectorias cualesquiera en el diagrama trellis dan lugar a un mismo estado de salida, siempre puede eliminarse una de ellas en la búsqueda de la trayectoria óptima. La trayectoria que se mantiene se designa como trayectoria superviviente y equivale a elegir el símbolo con la métrica de máxima similitud o la mínima distancia métrica. El tratamiento detallado de este algoritmo queda fuera del contexto de este texto y, en la bibliografía, se ofrecen algunas referencias que puede consultar aquél interesado en conocer más a fondo este tema.

3.15 Modulación con codificación trellis

Los códigos de bloque, según se mencionó, consiguen la detección y corrección de errores agregando redundancia a la información original, es decir, expandiendo el alfabeto, lo que da lugar a expansión del ancho de banda necesario para transmisión. En el caso de canales limitados en banda, no es posible expandir el ancho de banda y, por ello la codificación para control de errores no resultó muy popular para este tipo de canales, por ejemplo, los canales telefónicos. Sin embargo, en las últimas décadas se han desarrollado sistemas que combinan modulación y codificación contra errores, con los que es posible obtener ganancia de codificación sin aumentar el ancho de banda. Tal es el caso de la modulación con codificación trellis²⁶. En principio parecería que al utilizar la codificación contra errores, en que necesariamente se expande el alfabeto, se viola el principio de relación entre potencia, ancho de banda y probabilidad de error. Esto no ocurre, ya que se juega aquí con otro compromiso de realización, en que se consigue la ganancia de codificación, sin aumentar el ancho de banda, a expensas de la complejidad del decodificador.

La modulación con codificación trellis combina un conjunto de señales con niveles múltiples de amplitud y fase y un esquema de codificación trellis. Este esquema de amplitudes y fases multinivel es lo que permite agregar redundancia, o en otras palabras, expandir el alfabeto, *sin aumentar el ancho de banda*.²⁷

²⁵ Viterbi, A.J. and Omura, J.K. *Principles of Digital Communication and Coding*. McGraw-Hill Kogakusha, Ltd. 1979.

²⁶ Trellis-coded modulation o TCM

²⁷ Ungerboeck, G. “Channel Coding with Multilevel/Phase Signals”. *IEEE Transactions on Information Theory*. Vol IT28. Jan. 1982. pp. 55-67.

APENDICE AL CAPITULO 3

1. Generación de códigos de bloque.

La entrada al codificador de bloque procede del codificador de fuente y es por tanto, la señal comprimida de televisión es decir, el caudal MPEG, ya sea de audio o vídeo. El mensaje a codificar puede considerarse como un vector fila, formado por una secuencia de k bits:

$$\mathbf{m} = m_1, m_2, \dots, m_k \quad (9.1)$$

con estos bits de mensaje se genera un *vector de código*, \mathbf{U} , multiplicando el vector de mensaje \mathbf{m} por una *matriz generadora* \mathbf{G} :

$$\mathbf{U} = \mathbf{m}\mathbf{G} \quad (9.2)$$

Esta matriz generadora, \mathbf{G} aplicada en el codificador, también es conocida por el decodificador y está constituida por k vectores \mathbf{V} , linealmente independientes:

$$\mathbf{G} = \begin{bmatrix} \mathbf{V}_1 \\ \mathbf{V}_2 \\ \dots \\ \mathbf{V}_k \end{bmatrix} = \begin{bmatrix} v_{11} & v_{12} & \dots & v_{1n} \\ v_{21} & v_{22} & \dots & v_{2n} \\ \dots & \dots & \dots & \dots \\ v_{k1} & v_{k2} & \dots & v_{kn} \end{bmatrix} \quad (9.3)$$

donde las variables v_{ij} toman únicamente los valores 1 o 0.

La operación matricial (9.2) se efectúa aquí, de acuerdo a la regla siguiente:

$$c_{ij} = \sum_k^n a_{ik} b_{kj}$$

para $i = 1, \dots, l$ y $j = 1, \dots, m$.

Ejemplo

Sea $\mathbf{m} = 1\ 0\ 1$, y

$$\mathbf{G} = \begin{bmatrix} 11010 \\ 01101 \\ 10100 \end{bmatrix}$$

con lo que el vector de código \mathbf{U} resulta:

$$\begin{aligned} \mathbf{U} = \mathbf{m}\mathbf{G} &= [1\ 0\ 1] \begin{bmatrix} 11010 \\ 01101 \\ 10100 \end{bmatrix} = 1 \cdot \mathbf{V}_1 + 0 \cdot \mathbf{V}_2 + 1 \cdot \mathbf{V}_3 \\ &= 11010 + 00000 + 10100 \end{aligned}$$

La suma anterior se realiza en módulo 2, con lo que el vector de código para el vector de mensaje dado es:

$$U = 01110$$

Nótese que el vector resultante de la suma de dos o más vectores de la matriz generadora, no produce como resultado ningún vector de ésta. El vector de código, U , es una combinación lineal de los vectores fila de la matriz generadora y su longitud, en general, es mayor que la del vector mensaje.

Los códigos de bloque en que parte del vector generado U coincide con los k bits del mensaje, se designan como *códigos de bloque lineales sistemáticos* y, para ellos, la matriz generadora tiene la forma:

$$G = [P : I_k] = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1,(n-k)} & 1 & 0 & \cdots & 0 \\ p_{21} & p_{22} & \cdots & p_{2,(n-k)} & 0 & 1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ p_{k1} & p_{k2} & \cdots & p_{k,(n-k)} & 0 & 0 & \cdots & 1 \end{bmatrix} \quad (9.4)$$

Donde la porción izquierda de la matriz es la submatriz de paridad con $p_{ij} = 0$ o 1 y la derecha es una submatriz unidad, de dimensión $k \times k$, que no es necesario almacenar en el decodificador. El vector de código, en estas condiciones, se obtiene como:

$$U = mG = [m_1, m_2, \cdots, m_k] \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1,(n-k)} & 1 & 0 & \cdots & 0 \\ p_{21} & p_{22} & \cdots & p_{2,(n-k)} & 0 & 1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ p_{k1} & p_{k2} & \cdots & p_{k,(n-k)} & 0 & 0 & \cdots & 1 \end{bmatrix} \quad (9.5)$$

$$= p_1, p_2, \cdots, p_{n-k}, m_1, m_2, \cdots, m_k$$

Donde los bits p son bits de paridad y los m , de mensaje y,

$$p_1 = m_1 p_{11} + m_2 p_{21} + \cdots + m_k p_{k1}$$

$$p_2 = m_1 p_{12} + m_2 p_{22} + \cdots + m_k p_{k2}$$

etc.

2. Matriz de verificación de paridad

Para cada matriz ($k \times n$) generadora, G , existe una matriz H de dimensión $(n - k) \times n$, tal que las filas de G son ortogonales a las de H , con lo que:

$$GH^T = 0 \quad (9.6)$$

Donde H^T es la matriz transpuesta de H y 0 es una matriz de dimensión $(n - k) \times n$ toda ceros.

La matriz H se expresa como:

$$H = [I_{n-k} : P^T] \quad (9.7)$$

y,

$$\begin{aligned}
 \mathbf{H}^T &= \begin{bmatrix} \mathbf{I}_{n-k} \\ \dots \\ \mathbf{P} \end{bmatrix} \\
 &= \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ p_{11} & p_{12} & \dots & p_{1,(n-k)} \\ \vdots & \vdots & & \vdots \\ p_{k1} & p_{k2} & \dots & p_{k,(n-k)} \end{bmatrix}
 \end{aligned} \tag{9.8}$$

y se cumple, por la definición de \mathbf{U} que

$$\mathbf{UH}^T = 0 \tag{9.9}$$

3. Prueba del síndrome

Supóngase que el vector transmitido es \mathbf{U} y el recibido, $\mathbf{r} = r_1, r_2, \dots, r_n$, que puede expresarse como:

$$\mathbf{r} = \mathbf{U} + \mathbf{e} \tag{9.10}$$

donde $\mathbf{e} = e_1, e_2, \dots, e_n$ es el vector de error introducido por el canal. Se define ahora el *síndrome* como:

$$\mathbf{S} = \mathbf{rH}^T \tag{9.11}$$

El síndrome así definido, es el resultado de la verificación de paridad realizada sobre el vector \mathbf{r} para determinar si es o no un vector o símbolo válido del código. Si \mathbf{r} es un vector válido, entonces $\mathbf{S} = 0$ y, si \mathbf{r} contiene errores detectables, $\mathbf{S} \neq 0$. Además, si los errores son corregibles, el valor del síndrome indica el patrón del error y, por consecuencia permite recuperar el vector transmitido \mathbf{U} sin errores.

De (9.9) y (9.10) se ve que:

$$\begin{aligned}
 \mathbf{S} &= (\mathbf{U} + \mathbf{e})\mathbf{H}^T \\
 &= \mathbf{UH}^T + \mathbf{eH}^T \\
 &= \mathbf{eH}^T
 \end{aligned} \tag{9.12}$$

Para cumplir lo anterior, la matriz de verificación de paridad no debe tener ninguna columna con todo ceros y, además, cada columna debe ser única.

De lo anterior se ve que la prueba del síndrome realizada, ya sea sobre el vector recibido o sobre el patrón de error produce el mismo síndrome y, como la correspondencia entre síndromes y patrones de error es uno a uno, es posible corregir tales errores. El procedimiento de corrección, en términos generales, consiste primero en subdividir los patrones de error en subconjuntos designados como *cosets*. Estos patrones de error, así como la matriz de verificación de paridad son conocidos por el decodificador, es decir, están almacenados en éste.

Con el vector recibido \mathbf{r} se calcula el síndrome, $\mathbf{S} = \mathbf{r}\mathbf{H}^T$. A continuación, se localiza el coset para el cual el síndrome es $\mathbf{e}_j = \mathbf{r}\mathbf{H}^T$ y se asume que ése patrón de error \mathbf{e}_j es el causado por el canal. El vector corregido será $\mathbf{U} = \mathbf{r} + \mathbf{e}_j$.

Evidentemente, en situaciones prácticas, un código de bloque no puede corregir todos los posibles errores, ya que las necesidades de memoria y la cantidad de operaciones necesarias lo haría prácticamente imposible.

Bibliografía Adicional

- [1] - Anderson J. B. and Mohan, S. *Source and Channel Coding: An Algorithmic Approach*. Kluwer Academic Press. Boston, 1991.
- [2] - Berlekamp, E. R. *Algebraic Coding Theory*. Mc.Graw-Hill Book Co. New York, 1968.
- [3] - Biglieri, E. Divsalar, D. McLane, P.J. and Simon, M.K. *Introduction to Trellis-Coded Modulation with Applications*. Macmillan, New York, 1991.
- [4] - Clark, Jr. G.C. and Cain, J. B. *Error-correction Coding for Digital Communications*. Plenum Publishers, New York, 1981.
- [5] - Hamming, R. W. "Error-detecting and error-correcting codes". *Bell Syst. Tech Jour.* Vol. 26, . pp. 147-160. 1950.
- [6] - Jayant, N.S. and Noll, P. *Digital Coding of waveforms: Principles and applications to speech and video*. Prentice Hall, Englewood Cliffs, N.J. 1984.
- [7] - Lin, S. and Costello Jr., D.J. *Error Control Coding: Fundamentals and Applications*. Prentice Hall, Inc., Englewood Cliffs, N.J. 1983.
- [8] - Michelson, A.M. and Levesque, A.H. *Error control Techniques for Digital Communications*. John Wiley and Sons, Inc. New York, 1985.
- [9] - Ninomiya, Y. "Error management in digital terrestrial television broadcasting". *SMPTE Jour.* Vol 103, N° 9, Sept. 1994, pp. 595-607.
- [10] - Purser, M. *Introduction to error-correcting codes*. Artech House, Boston, 1995.
- [11] - Schlegel, C. *Trellis Coding*. IEEE Press, Piscataway, N.J. 1997.
- [12] - Strachan, D. and Conrod, R. "EDH-Error detection and handling in digital television". *SMPTE Jour.* Vol. 104, N° 10, Oct. 1995. pp 662-665.
- [13] - Watkinson, J. *The Art of Digital Video*. Focal Press. Oxford. UK. 1991.