Capítulo 8

Técnicas de codificación de canal

- Modelado de canales de comunicaciones
- Teorema de codificación del canal
- Codificación para comunicaciones confiables
- Codificación lineal en bloques
- Codificación convolucional
- Codificación para canales con restricciones en frecuencia
- Aplicaciones prácticas de codificación

El propósito de cualquier sistema de comunicaciones es transmitir información desde una *fuente* a un *destino* mediante un *canal de comunicaciones*. Un ingeniero de comunicaciones usualmente tiene muy poco control sobre estos tres componentes. Su rol es diseñar transmisores y receptores que envian la salida de la fuente sobre el canal al destino con alta fidelidad (baja distorsión, que puede ponderarse mediente probabilidad de error u otra medida de distorsión, como discutido en el Capítulo 4). El modelo matemático para fuentes de información junto con una medida cuantitativa de información fueron discutidos en el Capítulo 4. En este capítulo se estudiará la otra componente importante de un sistema de comunicaciones, o sea, el canal de comunicaciones. Se introducirá también el concepto de codificación para la protección de mensajes contra errores en el canal.

8.1 Modelado de canales de comunicaciones

Como discutido en el Capítulo 3, un canal de comunicaciones es cualquier medio sobre el cual puede transmitirse información o en el cual la información puede almacenarse. Los ejemplos de canales descriptos en ese Capítulo incluyen el cable coaxil, propagación ionosférica, espacio libre, fibra óptica, discos magnéticos y ópticos. Lo común en todos ellos es que aceptan señales como entrada y proveen señales como salida en un instante posterior (en el caso de almacenamiento) o en otro lugar (en el caso de transmisión). En consecuencia cada canal de comunicaciones está caracterizado por una relación entre su entrada y su salida.

Existen diferentes factores que hacen que la salida de un sistema de comunicaciones sea diferente de la entrada. Entre esos factores están la atenuación, las no linealidades, las limitaciones de ancho de banda, la propagación multicamino y el ruido. Todos esos factores contribuyen a una relación entrada-salida usualmente compleja en un canal de comunicaciones. Debido a la presencia de desvanecimiento y ruido, la relación entrada-salida en un canal de comunicaciones es generalmente una relación estocástica.

Los canales encontrados en la práctica aceptan generalmente una señal como entrada (contínua en el tiempo) y producen una señal a la salida. Dado que el ancho de banda de cualquier canal práctico está limitado, mediante el uso del teorema del muestreo este tipo de canales es equivalente a un canal discreto en el tiempo. En un canal discreto, tanto la entrada como la salida son señales discretas. En una clasificación general existen en nuestro caso tres tipos de canales de interés: de entrada discreta - salida discreta, de entrada discreta - salida contínua, de entrada contínua - salida contínua.

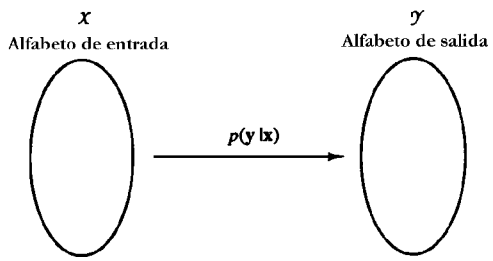

Figura 8.1: Un canal discreto.

En un canal específico, si los valores que pueden tomar la entrada y la salida son finitos, el canal se denomina *canal discreto*. Un ejemplo de un canal discreto es un canal de entrada binaria - salida binaria. En general un canal discreto se define por el alfabeto de entrada $\mathcal{X}$, el alfabeto de salida $\mathcal{Y}$, las probabilidades de ocurrencia asociadas $p(\boldsymbol{x})$ y $p(\boldsymbol{y})$, y $p(\boldsymbol{y}|\boldsymbol{x})$, y la probabilidad condicional de la secuencia de salida $\boldsymbol{y}$ dada una secuencia de entrada $\boldsymbol{x}$. Una representación esquemática de un canal discreto se muestra en la figura 8.1. En general, la salida y_i depende no solo de la entrada en el mismo instante x_i sino también de las entradas previas (en canales con ISI, ver el Capítulo 6), o aún de entradas previas y futuras (en canales de almacenamiento). De esta forma, un canal puede tener *memoria*. Sin embargo, si un canal discreto no tiene memoria se denomina *canal discreto sin memoria* y para tal canal, para cualquier $\boldsymbol{y} \in \mathcal{Y}^m$ y $\boldsymbol{x} \in \mathcal{X}^n$, se tiene que

$$p(\boldsymbol{y}|\boldsymbol{x}) = \prod_{i=1}^{n} p(y_i|x_i)$$

Todos los canales que se discutirá aquí son sin memoria. Un caso especial de un canal discreto sin memoria es el *canal binario simétrico* (BSC) ilustrado en la figura 8.2. En un canal binario simétrico, $\epsilon = p(0|1) = p(1|0)$ se denomina la *probabilidad de cruzamiento*.

Ejemplo: Es posible suponer un canal AWGN con señalización binaria antipodal. Como ya se discutió en el Capítulo 5, la probabilidad de error para ese tipo de canal, que un 1 sea detectado en lugar de un 0 transmitido o visceversa, está dada por

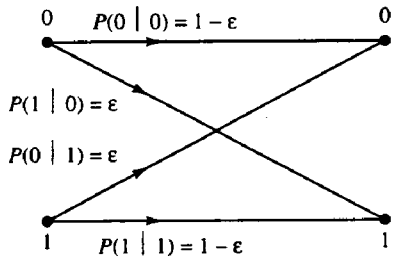

Figura 8.2: El canal binario simétrico.

$$\epsilon = p(1|0) = p(0|1) = Q\left[\sqrt{\frac{2\mathcal{E}_b}{N_0}}\right]$$

donde N_0 es la densidad espectral de potencia del ruido y $\mathcal{E}_b$ es la energía de cada una de las señales antipodales representando un 0 o un 1. Este canal discreto es un ejemplo de un canal binario discreto.

Ejemplo: (entrada discreta - salida contínua) Cualquiera de las constelaciones del Capítulo anterior definen una entrada discreta que asociada a un canal AWGN (con varianza acotada), define a su vez salidas contínuas.

El canal de alfabeto contínuo más importante (entrada contínua - salida contínua) es el canal AWGN con una restricción en la potencia de la entrada. En este canal tanto $\mathcal{X}$ como $\mathcal{Y}$ son conjuntos de números reales, cuya relación entrada salida está por

$$Y = X + Z$$

donde Z es el ruido de canal, supuesto Gaussiano, con media cero y varianza igual a P_N. Se supone además que la entrada a este canal satisface una limitación de potencia de forma que, para n grande, los bloques de entrada de longitud n satisfacen

$$\frac{1}{n}\sum_{i=1}^{n} x_i^2 \leq P$$

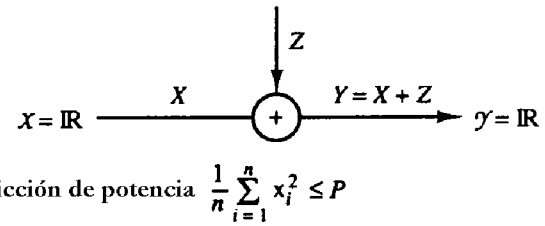

Figura 8.3: Canal de ruido aditivo blanco Gaussiano con restricción de potencia.

donde P es un valor fijo que limita la potencia. Este modelo de canal se ilustra en la figura 8.3.

8.2 Teorema de codificación del canal

Se vió en el Capítulo 4 que $H(X)$ define un límite fundamental sobre la velocidad a la cual puede codificarse una fuente discreta sin errores en su reconstrucción, y $R(D)$ define una velocidad fundamental para la

reproducción de la fuente con distorsión menor o igual a D. Un **límite fundamental** similar existe también para la transmisión de información sobre canales de comunicaciones.

Obviamente, el principal objetivo cuando se transmite información sobre cualquier canal de comunicaciones es la *confiabilidad*, la cual es una medida de la probabilidad de recepción correcta en el receptor. Debido a la presencia de ruido, a primera vista parece que la probabilidad de error será siempre mayor que cero (algún número positivo pequeño). Sin embargo, un resultado fundamental de la teoría de información es que una transmisión confiable (o sea, aquella con probabilidad de error por debajo de un valor especificado) es posible aún sobre canales ruidosos siempre que la velocidad de transmisión sea menor que cierto número denominado la *capacidad del canal*. Este resultado notable, demostrado por Shannon (1948), se conoce como el *teorema de codificación de canal ruidoso*. Lo que dice el teorema de codificación de canal es que *la limitación básica que introduce el ruido en un canal de comunicaciones no es sobre la confiabilidad de las comunicaciones sino sobre la velocidad de la misma.*

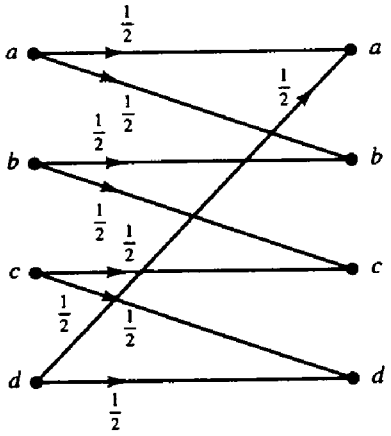

Figura 8.4: Ejemplo de un canal discreto.

La figura 8.4 muestra un canal discreto sin memoria con cuatro entradas y salidas. Si el receptor recibe a, no puede reconocer si a o d fueron transmitidos; si recibe b, no puede discriminar si a o d fueron transmitidos, etc; En consecuencia, siempre existe una probabilidad de error. Pero si el transmisor y el receptor concuerdan en que el transmisor use solo las letras a y c, entonces no existirá ambigüedad.

En este caso, si el receptor recibe a o b, ya sabe que fue transmitido a; y si recibe c o d, ya sabe que fue tranmisitdo c. Esto significa que los dos símbolos a y c pueden transmitirse por el canal sin error, o sea, es posible evitar los errores utilizando solo un subconjunto de las posibles entradas al canal. Logicamente, utilizar un subconjunto menor de las entradas posibles reduce el número de entradas posibles, pero este es un precio que se paga por una comunicación confiable. Esta es la esencia del teorema de codificación de canal ruidoso, o sea, utilizar solo aquellas entradas que correspondan a salidas posibles disjuntas. Las entradas elegidas deberían estar en algún sentido *lejos* tal que de sus *imágenes* a la salida del canal no se solapen (o tengan un solapamiento despreciable).

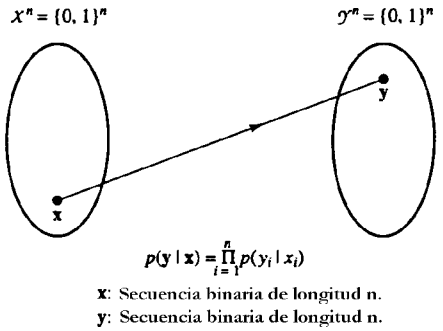

Figura 8.5: n-ésima extensión de un canal binario simétrico.

Volviendo al canal binario simétrico y tratando de aplicar la misma idea, es posible observar que no existe forma de no solapar salidas. En realidad, esta es la situación en muchos canales. Para usar los resultados de la argumentación anterior al canal binario simétrico es necesario aplicarla no al canal mismo sino a la *extensión del canal*. La n-ésima extensión de un canal con alfabetos de entrada $\mathcal{X}$ y salida $\mathcal{Y}$ y probabilidad condicional es $p(x|y)$, es un canal con alfabetos de entrada $\mathcal{X}^n$ y salida $\mathcal{Y}^n$ y probabilidad condicional $p(\boldsymbol{x}|\boldsymbol{y}) = \prod_{i=1}^{n} p(x_i|y_i)$. La n-ésima extensión de un canal binário simétrico toma bloques binarios de longitud n de entrada y salida. Este canal se ilustra en la figura 8.5. Por la ley de los grandes números (ver Capítulo 2), para n suficientemente grande, si se transmite una secuencia binaria de longitud n sobre el canal, la salida no concordará con la entrada con alta probabilidad en $n\epsilon$ posiciones. El número de secuencias posibles que no concuerdan con una secuencia de longitud n en $n\epsilon$ posiciones está dada por las *combinaciones de n, tomadas de a* $n\epsilon$, o sea

$$\begin{pmatrix} n \\ n\epsilon \end{pmatrix} = \frac{n!}{(n-n\epsilon)!\,(n\epsilon)!}$$

tal que, usando la aproximación $n! \cong n^n e^{-n}\sqrt{2\pi n}$, conduce a

$$\begin{pmatrix} n \\ n\epsilon \end{pmatrix} \cong 2^{nH_b(\epsilon)}$$

donde $H_b(\epsilon) = -\epsilon \log \epsilon - (1-\epsilon)\log(1-\epsilon)$ es la función entropía binaria, como definida en el Capítulo 4. Esto significa que para cualquier bloque de entrada existen rigurosamente $2^{nH_b(\epsilon)}$ salidas correspondientes altamente probables. Por otro lado, el número total de secuencias de salida altamente probables es $2^{nH(Y)}$. El número máximo de secuencias de entrada que produce secuencias de salida casi sin solapamiento es entonces casi igual a

$$M = \frac{2^{nH(Y)}}{2^{nH_b(\epsilon)}} = 2^{n(H(Y)-H_b(\epsilon))}$$

y la velocidad de transmisión por cada uso del canal será

$$R = \frac{\log M}{n} = H(Y) - H_b(\epsilon)$$

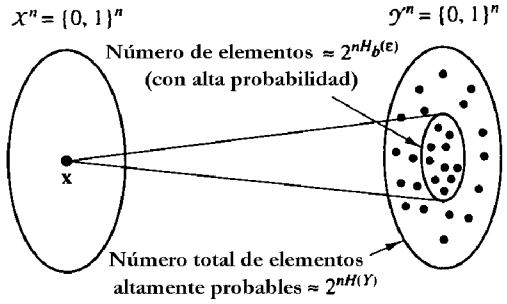

Figura 8.6: Representación esquemática de un canal binario simétrico.

La figura 8.6 muestra una representación esquemática de este caso.

En esa relación, $R = H(Y) - H_b(\epsilon)$ depende del canal y no es posible controlarla. Sin embargo, la distribución de probabilidad de la variable aleatória Y depende de la distribución de la entrada $p(x)$ y de las propiedades del canal caracterizadas por ϵ. Para maximizar la velocidad de transmisión sobre el canal, es necesario elegir $p(x)$ para maximizar $H(Y)$. Si X se elige como una variable uniformemente distribuida, Y resultará del mismo tipo y de esta forma $H(Y)$ será maximizada. El valor máximo de $H(Y)$ es 1, de forma que se obtiene

$$R = 1 - H_b(\epsilon)$$

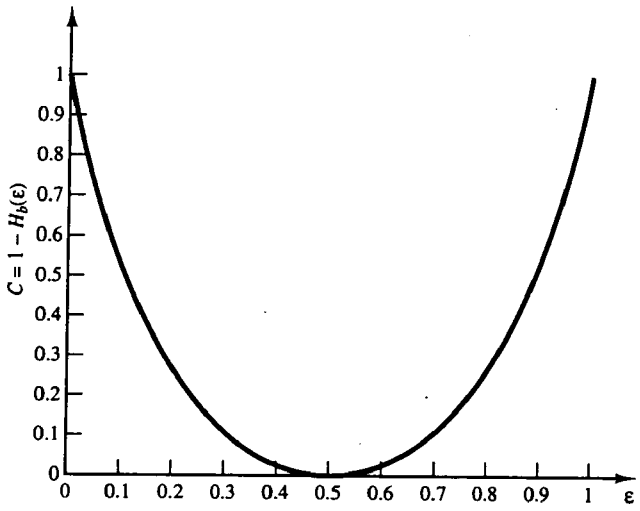

Figura 8.7: Capacidad de un canal binario simétrico.

Es posible demostrar que esta velocidad de transmisión es la máxima en la cual puede lograrse una transmisión confiable en un canal binario simétrico. Por *transmisión confiable* se entiende que la probabilidad de error puede hacerse tender a cero cuando la longitud del bloque n tiende a infinito. Un gráfico de la capacidad del canal en este caso se muestra en la figura 8.7. Es interesante notar que para los casos $\epsilon = 0$ y $\epsilon = 1$ resulta que $C = 1$. Esto significa que un canal que siempre cambia la entrada es tan bueno como un canal sin errores. El peor caso, por supuesto, ocurre cuando el canal cambia la entrada con $\epsilon = 0.5$.

La velocidad máxima a la cual es posible comunicarse sobre un canal discreto sin memoria, manteniendo una probabilidad de error tan baja como deseado con una longitud de bloque creciente, se denomina *capacidad del canal*, C. El teorema de codificación de canal ruidoso determina la capacidad de un canal discreto sin memoria en general.

> **Teorema. Codificación de canal ruidoso**. La capacidad de un canal discreto sin memoria está dada por
>
> $$C = \max_{p(x)} I(X;Y)$$
>
> donde $I(X;Y)$ es la información mutua entre la entrada al canal X y la salida Y, definida previamente en el Capítulo 4. Si la velocidad de transmisión R es menor que C, entonces para cualquier $\epsilon > 0$ existe un código de longitud n (para n suficientemente grande) cuya probabilidad de error es menor que ϵ. Si $R > C$, la probabilidad de error de cualquier código con cualquier longitud de bloque no puede hacerse arbitrariamente pequeña.

Este teorema es uno de los límites esenciales de teoría de información e introduce un límite fundamental sobre la posibilidad de comunicaciones confiables sobre un canal con ruido. De acuerdo a este teorema, independientemente de las otras propiedades, cualquier canal de comunicaciones está caracterizado por un número, denominado capacidad, que determina cuanta información puede transmitirse sobre él. De esta forma, para comparar dos canales, desde el punto de vista de transmisión de la información, es suficiente comparar sus capacidades.

Ejemplo: Se determinará la capacidad del canal ilustrado en la figura 8.8.

Es necesario hallar la distribución que maximiza $I(X;Y)$. Se tiene que

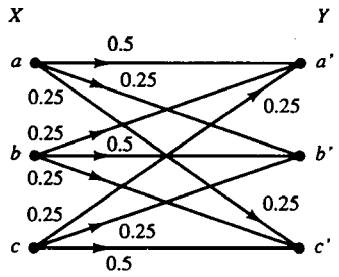

Figura 8.8: Canal discreto sin memoria.

$$I(X;Y) = H(Y) - H(Y|X)$$

pero

$$H(Y|X) \quad = \quad p(X=a)H(Y|X=a) + p(X=b)H(Y|X=b) + p(X=c)H(Y|X=c)$$

De la relación entrada - salida del canal puede obtenerse que en caso que $X = a$, $X = b$ ó $X = c$, entonces Y es una variable aleatória ternária con probabilidades 0.25, 0.25 y 0.5. En consecuencia

$$H(Y|X=a) = H(Y|X=b) = H(Y|X=c) = 1.5$$

entonces $H(Y|X) = 1.5$, de donde $I(X;Y) = H(Y) - 1.5$. Para maximizar $I(X;Y)$ basta maximizar $H(Y)$, lo cual se consigue cuando Y es una variable aleatória equiprobable. No está claro si existe una variable aleatória que resulte en una distribución uniforme a la salida. Sin embargo, en este caso especial una distribución uniforme de entrada resulta en una distribución uniforme de salida (ver Apéndice del Capítulo 4) y para esta distribución

$$H(Y) = \log 3 = 1.585$$

Esto significa que la capacidad del canal está dada por

$$C = 1.585 - 1.5 = 0.085 \,\text{bits/uso del canal}$$

8.2.1 Capacidad de un canal Gaussiano

Un canal Gaussiano discreto con restricción en la potencia de entrada está caracterizado por la relación entrada - salida dada por

$$Y = X + Z$$

donde Z es una variable aleatória con varianza P_N, y para n suficientemente grande, la restricción de potencia de entrada tiene la forma

$$\frac{1}{n}\sum_{i=1}^{n} x_i^2 \leq P$$

lo cual se aplica para cualquier secuencia de longitud n. Teniendo en cuenta bloques de longitud n de la entrada, la salida y el ruido, se tiene que

$$\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{z}$$

Para n grande, y por la ley de los grandes números se tiene que

$$\frac{1}{n}\sum_{i=1}^{n} z_i^2 = \frac{1}{n}\sum_{i=1}^{n} |y_i - x_i|^2 \leq P_N$$

ó

$$|\boldsymbol{y} - \boldsymbol{x}|^2 \leq nP_N$$

Esto significa que con probabilidad tendiendo a uno (cuando n aumenta), $\boldsymbol{y}$ estará localizada en una esfera n-dimensional de radio $\sqrt{nP_N}$ y centrada en $\boldsymbol{x}$. Por otro lado, debido a la restricción de potencia de la entrada y la independencia de la entrada y el ruido, la potencia de salida es la suma de la potencia de la entrada y el ruido, o sea

$$\frac{1}{n}\sum_{i=1}^{n} y_i^2 \leq P + P_N$$

ó

$$|\boldsymbol{y}|^2 \leq n(P + P_N)$$

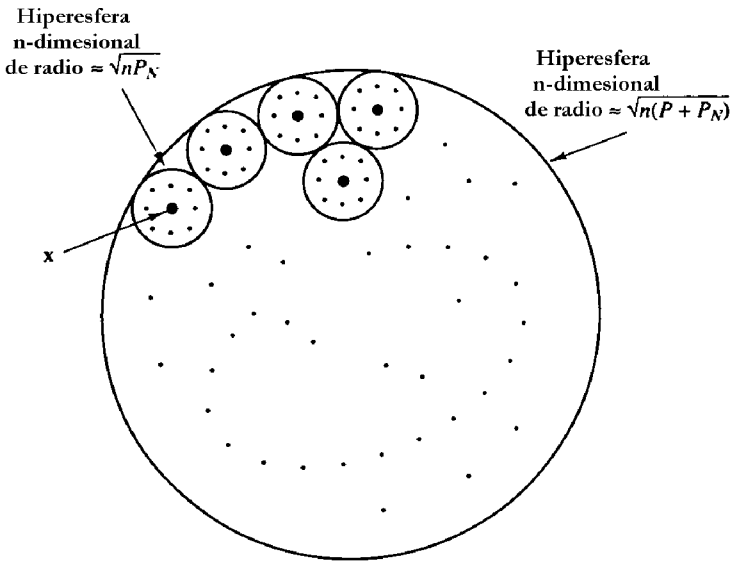

Figura 8.9: Las secuencias de salida de un canal Gaussiano con restricción de potencia.

Esto implíca que las secuencias de salida (nuevamente, en forma asintótica y con alta probabilidad) estarán dentro de una esfera n-dimensional de radio $\sqrt{n(P + P_N)}$ y centrada en el orígen. La figura 8.9 ilustra las secuencias en el espacio de salida. La pregunta ahora es: ¿Cuántas secuencias $\boldsymbol{x}$ pueden transmitirse por este canal tal que las esferas correspondientes a esas secuencias no se solapen en el espacio de salida? Logicamente si esta condición es satisfecha las secuencias de salida pueden decodificarse en forma confiable. Una pregunta equivalente es ¿Cuántas esferas de radio $\sqrt{nP_N}$ caben en una esfera de radio $\sqrt{n(P_N + P)}$? La respuesta es basicamente la relación de los volumenes de las dos esferas. Si el volúmen de una esfera

n-dimensional es $V_n = K_n R^n$, donde R es el radio y K_n es independiente de R, es posible concluir que el número de mensajes que pueden transmitirse en forma confiable sobre este canal es

$$M = \frac{K_n(n(P_N + P))^{\frac{n}{2}}}{K_n(nP_N)^{\frac{n}{2}}} = \left(\frac{P_N + P}{P_N}\right)^{\frac{n}{2}} = \left(1 + \frac{P}{P_N}\right)^{\frac{n}{2}}$$

En consecuencia, la capacidad de un canal discreto AWGN, con restricción en la potencia de entrada P, estará dada por

$$C = \frac{1}{n}\log M = \frac{1}{n}\frac{n}{2}\log\left(1 + \frac{P}{P_N}\right) = \frac{1}{2}\log\left(1 + \frac{P}{P_N}\right) \tag{8.1}$$

Si se hubiera partido de un canal AWGN contínuo con densidad espectral $N_0/2$, restricción de potencia P y ancho de banda W, es posible muestrear a la velocidad de Nyquist y obtener un canal discreto. La potencia por muestra será P y la potencia de ruido por muestra será

$$P_N = \int_{-W}^{+W} \frac{N_0}{2} df = WN_0$$

Sustituyendo este resultado en (8.1) se obtiene

$$C = \frac{1}{2}\log\left(1 + \frac{P}{WN_0}\right) \quad \text{bits/uso del canal}$$

Si se multiplica este resultado por el número de transmisiones por segundo, $2W$, obtenemos la capacidad del canal en bits/seg, dada por

$$C = W\log\left(1 + \frac{P}{WN_0}\right) \quad \text{bits/seg}$$

Esta es la conocida fórmula de Shannon para la capacidad de un canal AWGN.

Ejemplo Es posible obtener la capacidad para un canal telefónico con ancho de banda $W = 3000$ kHz y SNR de 30 dB. Esta SNR es equivalente a 1000. Entonces, usando la fórmula de Shannon se obtiene

$$C = 3000\log(1 + 1000) \cong 30000 \quad \text{bits/seg}$$

8.3 Cotas en comunicaciones

De la sección previa, la capacidad de un canal AWGN está dada por

$$C = W\log\left(1 + \frac{P}{WN_0}\right)$$

De este resultado es posible concluir que los factores que determinan la capacidad del canal son el ancho de banda W, la densidad espectral de potencia de ruido N_0, y la potencia de la señal P. Existirá un compromiso entre P y W en el sentido que uno compensa al otro.

Incrementar la potencia de la señal de entrada obviamente incrementa la capacidad del canal, dado que cuando se tiene disponible más potencia, es posible elegir un mayor número de niveles de entrada de forma que estén más apartados y en consecuencia sea posible más bits de información por transmisión. Sin

embargo, el incremento en la capacidad con el aumento de la potencia es logarítmico y suave. Esto se debe a que si se transmite con cierto número de niveles de entrada separados en Δ, para permitir cierto nivel de inmunidad contra el ruido y deseamos incrementar el número de niveles, es necesario introducir nuevos niveles con amplitudes mayores que la de los niveles existentes y esto requiere gran cantidad de potencia. A pesar de esto, la capacidad del canal podría incrementarse a cualquier valor incrementando la potencia de entrada.

El efecto del ancho de banda del canal, sin embargo, es totalmente diferente. Incrementar W tiene dos efectos contrapuestos. Por un lado, sobre un canal de ancho de banda mayor es posible transmitir más muestras por segundo y de esta forma incrementar la velocidad de transmisión. Por otro lado, un ancho de banda mayor significa mayor potencia de ruido en el receptor y esto degrada el desempeño. Esto puede verse en los dos W que aparecen en la relación que describe la capacidad del canal. Para ver el efecto de incrementar el ancho de banda, es posible hacer tender el ancho de banda a infinito y usar la regla de L'Hôpital para obtener

$$\lim_{W\to\infty} C = \frac{P}{N_0}\log e = 1.44\frac{P}{N_0}$$

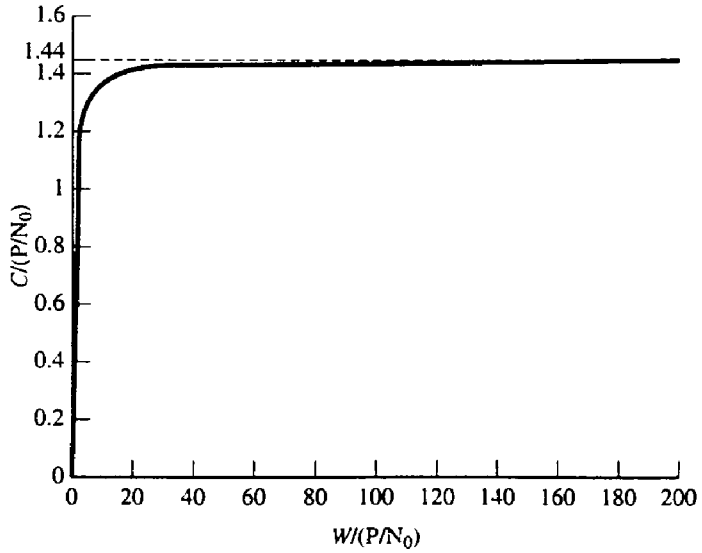

Figura 8.10: Gráfico de la capacidad del canal versus el ancho de banda.

Esto significa que, al contrário del caso de la potencia, incrementar el ancho de banda solamente no permite incrementar la capacidad a cualquier valor. La figura 8.10 muestra C en función de W.

En cualquier sistema de comunicaciones práctico se tiene que $R < C$. Si se emplea un canal AWGN se tiene que

$$R < W\log\left(1+\frac{P}{WN_0}\right)$$

Dividiendo ambos lados por W y definiendo $r = R/W$, la velocidad de transmisión normalizada, se obtiene

$$r < \log\left(1+\frac{P}{WN_0}\right)$$

Si $\mathcal{E}_b$ es la energía por bit, entonces $\mathcal{E}_b = P/R$, tal que sustituyendo en la relación previa se obtiene

$$r < \log\left(1+r\frac{\mathcal{E}_b}{N_0}\right)$$

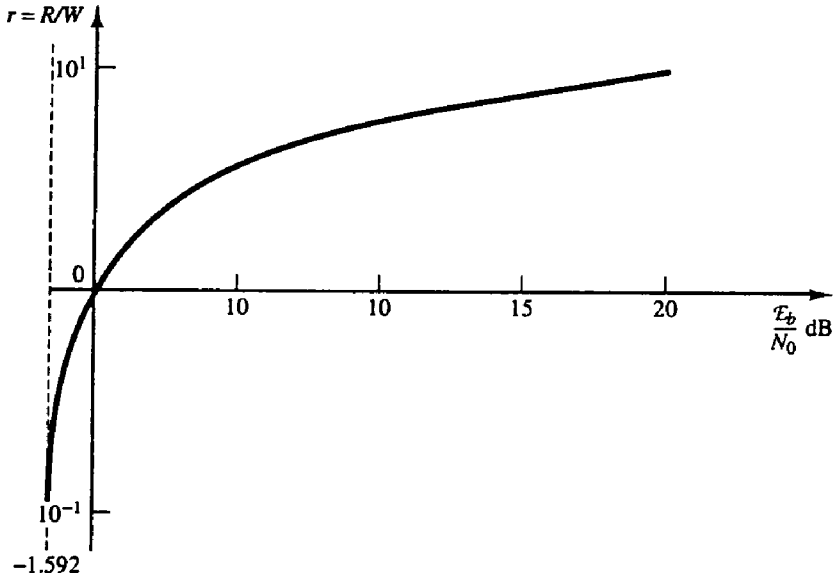

Figura 8.11: Eficiencia espectral versus SNR/bit en un sistema óptimo.

Esta relación se grafica en la figura 8.11. La curva definida por $r = \log\left(1 + r\frac{\mathcal{E}_b}{N_0}\right)$ divide el plano en dos regiones. En la región por debajo de la curva es posible la comunicación confiable, en la región superior las comunicaciones confiables no son posibles. El desempeño de cualquier sistema de comunicaciones puede describirse mediante un punto en este plano y cuanto más cerca esté este punto a la curva, más próximo será el desempeño del sistema al óptimo. De esta curva es posible ver que

$$\frac{\mathcal{E}_b}{N_0} = \ln 2 = 0.693 \quad \text{(-1.592 dB)}$$

es un mínimo absoluto para las comunicaciones confiables. En otras palabras, para una comunicación confiable se debe cumplir

$$\frac{\mathcal{E}_b}{N_0} > 0.693$$

En la figura 8.11, cuando $r \ll 1$, se tiene un caso donde el ancho de banda es grande y la principal preocupación debería ser la limitación de potencia. Este caso se denomina usualmente el *caso de potencia limitada*. Los esquemas de señalización de alta dimensionalidad, tal como los ortogonales, biortogonales y simplex se utilizan frecuentemente en esos casos. El caso donde $r \gg 1$ sucede cuando el ancho de banda del canal es pequeño, y en consecuencia se denomina el *caso de ancho de banda limitado*. Los esquemas de señalización de baja dimensionalidad con constelaciones densas se implementan en esos casos.

En el Capítulo 4 se introdujeron los limites fundamentales que existen en la codificación de fuentes de información. Esos límites fundamentales se expresan en términos de la entropía de la fuente, la función velocidad - distorsión asociada con la fuente, y la medida de distorsión correspondiente. La entropía determina un límite inferior sobre la velocidad de codificación que permite reproducir la fuente sin error, y la función velocidad - distorsión determina una cota inferior sobre la velocidad de codificación permitida para reproducir la fuente con distorsión D. Si deseamos transmitir una fuente U confiablemente mediante un canal con capacidad C, es necesario que

$$H(U) < C \tag{8.2}$$

Si se desea una transmisión con distorsión máxima D, entonces la condición es

$$R(D) < C \tag{8.3}$$

Estas dos relaciones definen los limites fundamentales para la transmisión de información.

Ejemplo. Una fuente Gaussiana con media cero y densidad espectral de potencia uniforme, $S_X(f) = u(\frac{f}{20000})$ ($u(f)$ es el escalón unitario), se transmite por un canal telefónico como el descripto en el último ejemplo de la sección anterior. ¿Cual es la distorsión mínima alcanzable si se utiliza un critério de distorsión de media cuadrática?
El ancho de banda de la fuente es 10000 Hz, de forma que puede muestrearse a 20000 muestras/seg. La potencia de la fuente es

$$P_X = \int_{-\infty}^{+\infty} S_X(f)df = 20000$$

La varianza de cada muestra está dada por $\sigma^2 = P_X = 20000$, y la función velocidad - distorsión para $D < 20000$ está dada por

$$R(D) = \frac{1}{2}\log\frac{\sigma^2}{D} = \frac{1}{2}\log\frac{20000}{D} \quad \text{bits/muestra}$$

lo cual es equivalente a

$$R(D) = 10000\ \log\frac{20000}{D} \quad \text{bits/seg}$$

Dado que la capacidad del canal telefónico era 30000 bits/seg, es posible obtener la menor distorsión posible resolviendo

$$30000 = 10000\ \log\frac{20000}{D}$$

de donde se obtiene $D = 2500$.

En general, puede mostrarse que la función velocidad - distorsión para una fuente Gaussiana con densidad espectral de potencia

$$S_X(f) = \begin{cases} A, & |f| < W_s \\ 0, & \text{para todo otro } f \end{cases}$$

y con medida de distorsión

$$d(u(t), v(t)) = \lim_{T\to\infty}\frac{1}{T}\int_{-T/2}^{T/2}[u(t) - v(t)]^2 dt$$

está dada por

$$R(D) = \begin{cases} W_s\log(\frac{2AW_s}{D}), & |D| < 2AW_s \\ 0, & |D| < 2AW_s \end{cases}$$

Si esta fuente se transmite mediante un canal AWGN con ancho de banda W_c, potencia P y densidad espectral de ruido N_0, la distorsión mínima alcanzable se obtiene resolviendo la ecuación

$$W_c\log(1 + \frac{P}{N_0 W_c}) = W_s\log(\frac{2AW_s}{D})$$

De esta ecuación se obtiene

$$D = 2AW_s(1 + \frac{P}{N_0 W_c})^{-\frac{W_c}{W_s}}$$

Puede concluirse que en un sistema óptimo, la distorsión disminuye exponencialmente cuando W_c/W_s aumenta. La relación W_c/W_s se denomina el *factor de expansión de ancho de banda.* Es posible obtener también la SQNR dada por

$$SQNR = \frac{2AW_s}{D} = (1 + \frac{P}{N_0 W_c})^{\frac{W_c}{W_s}}$$

La conclusión importante es que en un sistema óptimo, la relación señal a ruido (distorsión) incrementa exponencialmente con el factor de expansión de ancho de banda. Con este critério es posible comparar el desempeño de varios sistemas de transmisión.

8.3.1 Aplicación a la transmisión de información analógica con PCM

Se vió en el Capítulo 4 que PCM es uno de los esquemas más comunmente utilizados para la transmisión de señales analógicas. Se analizará aquí el desempeño de un sistema PCM cuando se transmite sobre un canal ruidoso y se comparará los resultados con el sistema óptimo.

Como ya se discutió en ese Capítulo, la distorsión o sea, el ruido de cuantización en PCM está dado por

$$E[\tilde{X}^2] = \frac{x_{max}^2}{3 \times 4^{\nu}}$$

donde x_{max} es la amplitud máxima de entrada y ν es el número de bits por muestra. Si el ancho de banda de la fuente es W_s, entonces la frecuencia de muestreo en el sistema óptimo es $f_s = 2W_s$, y la velocidad de transmisión es $R = 2\nu W_s$. El ancho de banda mínimo de canal que puede acomodar esta velocidad (del Capítulo 6) es $W_c = R/2 = \nu W_s$. En consecuencia, el ruido de cuantización puede expresarse como

$$E[\tilde{X}^2] = \frac{x_{max}^2}{3 \times 4^{\frac{W_c}{W_s}}}$$

Si la salida de un sistema PCM se transmite sobre un canal ruidoso con probabilidad de error P_b, algunos bits serán recibidos con error y otros tipos de distorsión debido a la transmisión. Esta distorsión será independiente de la distorsión de cuantización; debido a que el error de cuantización tiene media cero, la distorsión total será la suma de esas distorsiones. Para simplificar el cálculo de la distorsión de transmisión, es posible suponer que P_b es suficientemente pequeña tal que en un bloque de longitud ν existe a lo sumo un error. Este error único puede ocurrir en cualquiera de las posiciones ν del bloque, y en consecuencia su contribución a la distorsión total variará en forma acorde. Si ocurre un error en el bit menos significativo, esto resulta en un cambio de Δ en el nivel de salida; si el error ocurre en el bit próximo, el cambio será será de 2Δ; y si el error ocurre en el bit más significativo, el cambio de nivel es igual a $2^{\nu-1}\Delta$ en la salida. Suponiendo que esos son los únicos errores posibles, la expresión para la distorsión de transmisión será

$$E[D_T^2] = \sum_{i=1}^{\nu} P_b(2^{i-1}\Delta)^2 = P_b\Delta^2\frac{4^{\nu}-1}{3}$$

y la distorsión total será

$$D_{total} = E[\tilde{X}^2] + E[D_T^2] = E\frac{x_{max}^2}{3 \times 4^{\frac{W_c}{W_s}}} + P_b\Delta^2\frac{4^{W_c/W_s}-1}{3}$$

Es posible notar que $\Delta = \frac{2x_{max}}{2^{\nu}} = \frac{x_{max}}{2^{\nu-1}}$. Sustituyendo Δ, y asumiendo que $4^{\nu} \gg 1$, la expresión para D_{total} se simplifica de la siguiente forma

$$D_{total} = \frac{x_{max}^2}{3}(4^{-W_c/W_s} + 4P_b)$$

La relación señal a ruido (distorsión) en el receptor será entonces

$$SNR = \frac{3\overline{X}^2}{x_{max}^2(4^{-W_c/W_s} + 4P_b)}$$

El valor de P_b depende obviamente del esquema de modulación utilizado para transmitir las salidas del sistema PCM. Si se emplea señalización binaria antipodal, entonces $P_b = Q[\sqrt{2\mathcal{E}_b/N_0}]$, y si se utiliza señalización ortogonal binaria con detección coherente, $P_b = Q[\sqrt{\mathcal{E}_b/N_0}]$.
De la expresión para la SNR es posible concluir que para P_b pequeño, la SNR crece casi exponencialmente con W_c/W_s, el factor de expansión de ancho de banda y, en este sentido, un sistema PCM utiliza el ancho de banda eficientemente. La modulación FM también introduce un compromiso para manejar la inmunidad frente al ruido. Sin embargo, la SNR de un sistema FM es una función cuadrática del factor de expansión de ancho de banda y en consecuencia un sistema FM no es tan eficiente en términos de ancho de banda como un sistema PCM.

8.4 Codificación para comunicaciones confiables

En los Capítulos 6 y 7 se mostró que en los esquemas de modulación banda base y por portadora la probabilidad de error es una función de la distancia entre los puntos de señal en la constelación. En realidad, para señales binarias equiprobables la probabilidad de error puede expresarse como $P_e = Q[d_{12}/\sqrt{2N_0}]$, donde d_{12} es la distancia Euclidiana entre $s_1(t)$ y $s_2(t)$, dada por

$$d_{12}^2 = \int_{-\infty}^{+\infty} [s_1(t) - s_2(t)]^2 dt$$

Por otro lado se vió que, para el caso binario con demodulación coherente, la señalización funciona mejor y tiene una distancia Euclidiana $d_{12} = 2\sqrt{\mathcal{E}_b}$. En consecuencia, $P_e = Q[\sqrt{2\mathcal{E}_b/N_0}]$. De esta relación es posible concluir que para disminuir la probabilidad de error es necesario incrementar la energía de la señal. Esto puede lograrse de dos formas, incrementando la potencia transmitida o incrementando la duración de la transmisión. No siempre es posible incrementar la potencia transmitida dado que cada transmisor tiene una limitación en su potencia promedio. Incrementar la duración de la transmisión, a su vez, disminuye la velocidad de transmisión, y en consecuencia parece ser el único camino para hacer disminuir la probabilidad de error. En realidad, este era el punto de vista de los ingenieros de diseño de sistemas de comunicaciones en las épocas previas a los resultados de Shannon.

También se vió en el Capítulo 6 que el empleo de señales ortogonales permite lograr una transmisión confiable a una velocidad no nula, siempre que $\mathcal{E}_b/N_0 > 2\ln 2$. Aquí se mostrará que $\mathcal{E}_b/N_0 > \ln 2$ es suficiente para lograr una transmisión confiable. Notar que esta es la misma condición que garantiza una transmisión confiable sobre un canal AWGN cuando el ancho de banda tiene a infinito.

8.4.1 Una cota precisa sobre la probabilidad de error de señales ortogonales

Para obtener una cota precisa de la probabilidad de error para señales ortogonales si comparada con la obtenida en el Capítulo 5, se utilizará otra técnica. El problema con la cota de unión del Capítulo 5 es que no es precisa para SNR pequeñas. Para compensar esto usaremos dos cotas, una para SNR grandes, la cual es esencialemte la cota de unión, y otra para SNR pequeñas.

En el Capítulo 5 se mostró que la probabilidad de error para la señalización ortogonal está dada por

$$P_M = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \left\{1 - (Q[u])^{M-1}\right\} e^{-(u-\sqrt{2\mathcal{E}_2/N_0})^2/2} du$$

Definiendo $a = \sqrt{2\mathcal{E}_2/N_0}$ y $f(u) = \frac{1}{\sqrt{2\pi}} e^{-u^2/2}$, se tiene que

$$P_M = \int_{-\infty}^{\infty} \left\{1 - (Q\,[u])^{M-1}\right\} f(u-a)du \tag{8.4}$$

Usando la desigualdad $nx + (1-x)^n \geq 1$, válida para $0 \leq x \leq 1$ y $n \geq 1$[1], se obtiene

$$(1 - Q[u])^{M-1} \geq 1 - (M-1)Q[u]$$

y en consecuencia

$$1 - (1 - Q[u])^{M-1} \leq (M-1)Q[u] < MQ[u] < Me^{-u^2/2}$$

Esta cota es basicamente la cota de unión, que se utilizará para SNR grandes (u grandes). Para SNR pequeñas se utilizará la cota obvia

$$1 - (1 - Q[u])^{M-1} \leq 1$$

Sustituyendo esta cota en (8.4) se obtiene

$$P_M < \int_{-\infty}^{y_0} f(u-a)du + M\int_{y_0}^{\infty} e^{-u^2/2} f(u-a)du = P_1 + MP_2 \tag{8.5}$$

donde y_0 es cualquier número positivo. Para obtener la cota más precisa se diferencia esta última ecuación con respecto a y_0 y se iguala la derivada a cero. De esta forma se obtiene

$$f(y_0 - a) - Me^{-y_0^2/2} f(y_0 - a) = 0$$

ó $e^{-y_0^2/2} = M$, de lo cual se obtiene $y_0 = \sqrt{2\ln M} = \sqrt{2\ln 2 \log_2 M} = \sqrt{2k\ln 2}$. El próximo paso es hallar cotas sobre P_1 y P_2. Para P_1 se tiene que

$$P_1 = \int_{-\infty}^{y_0} f(u-a)du = \int_{-\infty}^{y_0-a} f(x)dx = \int_{a-y_0}^{\infty} y_0 - af(x)dx = Q[a - y_0]$$

y utilizando las cotas sobre Q se obtiene

$$P_1 \leq \frac{1}{2} e^{-(a-y_0)^2/2} < e^{-(a-y_0)^2/2} \quad \text{para } y_0 \leq a$$

Para acotar P_2 se tiene que

$$\begin{aligned} P_2 &= \int_{y_0}^{\infty} f(u-a)du = \int_{y_0}^{\infty} e^{-u^2/2} \frac{1}{\sqrt{2\pi}} e^{-(u-a)^2/2} du \\ &= e^{-a^2/4} \int_{y_0}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-(u-a/2)^2/2} du = e^{-a^2/4} \int_{\sqrt{2}(y_0 - a/2)}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-x^2/2} dx \\ &= \frac{1}{\sqrt{2}} e^{-a^2/4} Q\left[\sqrt{2}(y_0 - a/2)\right] \end{aligned}$$

y usando que para todo x $Q[x] \leq 1$, y que para $x \geq 0$, $Q[x] \leq \frac{1}{2}e^{-x^2/2}$, se obtiene

[1] Para ver esto, definir $g(x) = nx + (1-x)^n$, luego $g(0) = 1$ y $g'(x) = n - n(1-x)^{n-1}$. Pero para $0 \leq x \leq 1$ se tiene que $0 \leq 1 - x \leq 1$, entonces $(1-x)^{n-1} \leq 1$ para todo $n \geq 1$. Esto significa que para $0 \leq x \leq 1$, $g'(x) \geq 0$ y de esta forma $g(x)$ es una función creciente en este intervalo. Esto a su vez significa que $g(x) \leq 1$ para $0 \leq x \leq 1$ y $n \geq 1$.

$$P_2 \leq \begin{cases} e^{-a^2/4}e^{-(y_0-a/2)^2}, & \text{para } y_0 \geq a/2 \\ e^{-a^2/4}, & \text{para } y_0 \leq a/2 \end{cases}$$

Sustituyendo las cotas obtenidas para P_1 y P_2 en (8.5) y usando $M = e^{y_0^2/2}$ se obtiene

$$P_M < \begin{cases} e^{-(a-y_0)^2/2} + e^{y_0^2/2}e^{-a^2/4}, & \text{para } 0 \leq y_0 \leq a/2 \\ e^{-(a-y_0)^2/2} + e^{y_0^2/2}e^{-a^2/4-(y_0-a/2)^2}, & \text{para } a/2 \leq y_0 \leq a \end{cases} \tag{8.6}$$

Para $0 \leq y_0 \leq a/2$ se tiene que

$$e^{-(a-y_0)^2/2} + e^{y_0^2/2}e^{-a^2/4} = e^{y_0^2/2-a^2/4}[1 + e^{-(y_0-a/2)^2}] \leq 2e^{y_0^2/2-a^2/4}$$

y para $a/2 \leq y_0 \leq a$ los dos exponentes en (8.6) son iguales. De esta forma la cota puede simplificarse a

$$P_M < \begin{cases} 2e^{y_0^2/2-a^2/4}, & \text{para } 0 \leq y_0 \leq a/2 \\ 2e^{-(a-y_0)^2/2}, & \text{para } a/2 \leq y_0 \leq a \end{cases}$$

Luego, usando las definiciones de y_0 y a se obtiene el siguiente resultado final

$$P_M < \begin{cases} 2e^{\frac{k}{2}(\mathcal{E}_b/N_0 - 2\ln 2)}, & \text{para } 0 \leq \ln M \leq \mathcal{E}_s/4N_0 \\ 2e^{-k(\sqrt{calE_b/N_0} - \sqrt{\ln 2})^2}, & \text{para } \mathcal{E}_s/4N_0 \leq \ln M \leq \mathcal{E}_s/N_0 \end{cases}$$

La primera cota coincide con la obtenida en el Capítulo 5. La segunda cota es mejor que la cota de unión para valores grandes de M (valores pequeños de $\mathcal{E}_b/N_0$). Esta cota muestra que la condición para comunicaciones confiables es $\mathcal{E}_b/N_0 > \ln 2$ antes que $\mathcal{E}_b/N_0 > 2\ln 2$ obtenida en el Capítulo 5.
También es posible expresar esas cotas en términos de la velocidad de transmisión y la capacidad del canal. Usando las relaciones $k = RT$, $\mathcal{E}_b = P/R$, $R = \frac{1}{T}\log M$ y $C_\infty = \frac{P}{N_0}\ln 2$, se obtiene

$$P_M < \begin{cases} 2 \times 2^{-T(\frac{1}{2}C_\infty - R)}, & \text{para } 0 \leq R \leq \frac{1}{4}C_\infty \\ 2 \times 2^{-T(\sqrt{C_\infty} - \sqrt{R})^2}, & \text{para } \frac{1}{4}C_\infty \leq R \leq C_\infty \end{cases}$$

o equivalentemente

$$P_M < 2 \times 2^{-TE^*(R)}$$

donde

$$E^*(R) = \begin{cases} \frac{1}{2}C_\infty - R), & \text{para } 0 \leq R \leq \frac{1}{4}C_\infty \\ \left(\sqrt{C_\infty} - \sqrt{R}\right)^2, & \text{para } \frac{1}{4}C_\infty \leq R \leq C_\infty \end{cases}$$

La función $E^*(R)$ se denomina la *función confiabilidad del canal* y se grafica en la figura 8.12.

8.4.2 Objetivo de la codificación de canal

De la discusión anterior es posible concluir que la señalización ortogonal permite lograr la capacidad a velocidad no arbitrariamente pequeñas. Sin embargo, el precio pagado por ello es muy alto. Para lograr este desempeño es necesario hacer $k \to \infty$, ó $T \to \infty$. Por otro lado el número de señales ortogonales está dado por $M = 2^k = 2^{RT}$. Si es necesario, por ejemplo, emplear señalización PPM, esto significa que la duración T tiene que dividirse en 2^{RT} partes cada una de duración $T/2$. Esto a su vez significa que el ancho de los pulsos se aproxima a cero exponencialmente, y de esta manera el ancho de banda requerido para

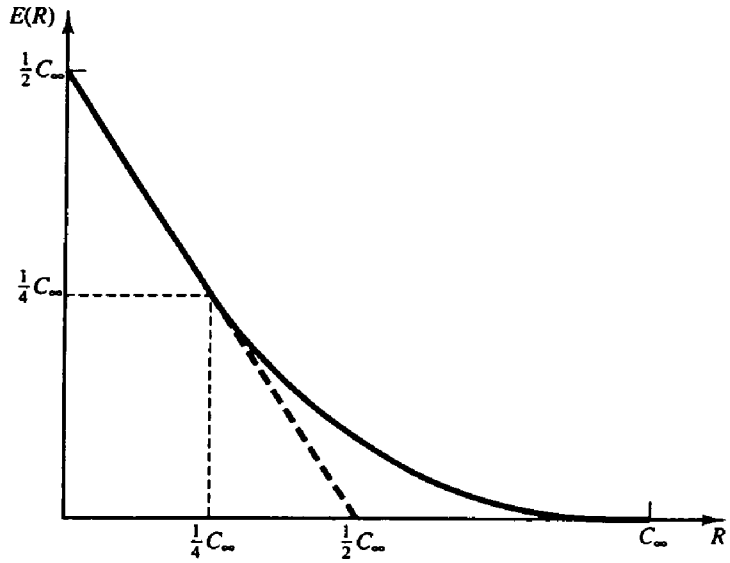

Figura 8.12: Función confiabilidad para un canal AWGN con ancho de banda infinito.

transmitirlos aumente también exponencialmente[2]. Ahora la pregunta es, ¿existe una forma de transmitir mensajes sobre un canal ruidoso a una velocidad finita sin el incremento exponencial en el ancho de banda? La respuesta es positiva y la herramienta para lograrlo es la codificación. El siguiente ejemplo clarifica este punto.

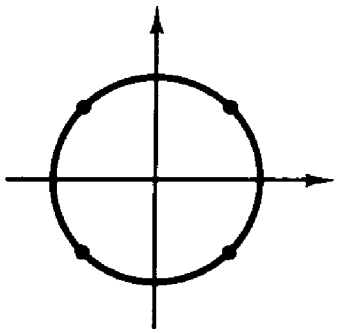

Figura 8.13: Constelación para un esquema PSK $M = 4$.

Ejemplo. En un sistema de comunicaciones digitales, la potencia del transmisor es P y la velocidad de la fuente es R. El sistema emplea señalización PSK con $M = 4$ donde un par de bits de información se mapea en cualquiera de las cuatro señales de la constelación ilustrada en la figura 8.13. Puede verse facilmente que $\mathcal{E}_b = P/R$, y la mínima distancia Euclidiana entre dos señales cualquiera está dada por

$$d_{min}^2 = 4\mathcal{E}_b = 4\frac{P}{R}$$

Es posible suponer ahora que en lugar de transmitir con una señal PSK con $M = 4$ (la cual es bidimensional), se emplean *tres* señales ortogonales para transmitir los mismos *dos* bits. Por ejemplo, es posible suponer que

[2]Este incremento exponencial en el requerimiento de ancho de banda es una característica de los sistemas de señalización ortogonal y no una propiedad particular de las señales PPM. Puede mostrarse que el número de señales ortogonales que están "casi" limitadas a un ancho de banda W y de una duración "casi" limitada T es $2WT$.

las señales ortogonales son $\psi(t)$, $\psi(t-T)$ y $\psi(t-2T)$, donde $\psi(t)$ es igual a cero fuera del intervalo $[0,T]$ y $\int_0^T \psi^2(t) = 1$. De esta forma, es posible suponer que en lugar de transmitir las señales PSK $M=4$, se transmiten las siguientes señales

$$\begin{aligned} s_1(t) &= \sqrt{\mathcal{E}}(+\psi(t) + \psi(t-T) + \psi(t-2T)) \\ s_2(t) &= \sqrt{\mathcal{E}}(+\psi(t) - \psi(t-T) - \psi(t-2T)) \\ s_3(t) &= \sqrt{\mathcal{E}}(-\psi(t) - \psi(t-T) + \psi(t-2T)) \\ s_4(t) &= \sqrt{\mathcal{E}}(-\psi(t) + \psi(t-T) - \psi(t-2T)) \end{aligned}$$

o equivalentemente, en notación vectorial

$$\begin{aligned} \boldsymbol{s}_1 &= \sqrt{\mathcal{E}}(+1,\ +1,\ +1) \\ \boldsymbol{s}_2 &= \sqrt{\mathcal{E}}(+1,\ -1,\ -1) \\ \boldsymbol{s}_3 &= \sqrt{\mathcal{E}}(-1,\ -1,\ +1) \\ \boldsymbol{s}_4 &= \sqrt{\mathcal{E}}(-1,\ +1,\ -1) \end{aligned}$$

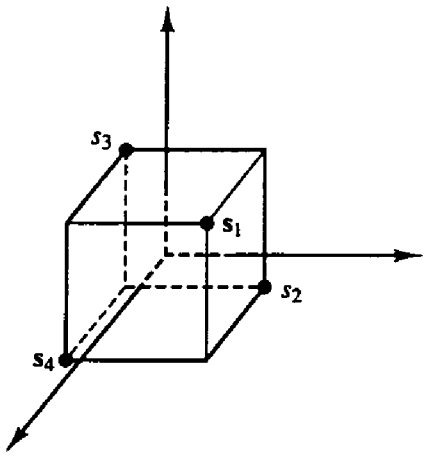

Figura 8.14: Palabras de código en los vértices de un cubo.

La constelación correspondiente en el espacio tridimensional se muestra en la figura 8.14. Puede verse que con esta elección de las palabras de código, cada palabra difiere de cualquier otra al menos en dos componentes. En consecuencia, la distancia Euclidiana entre dos señales cualquiera estará dada por

$$d_{ij}^2 = |\boldsymbol{s}_i - \boldsymbol{s}_j|^2 = 8\mathcal{E} \quad \text{para } i \neq j$$

Para hallar $\mathcal{E}$ en términos de P es posible notar que en un intervalo de tiempo T se transmiten dos bits. De esa forma, $2\mathcal{E}_b = 3\mathcal{E}$, y $\mathcal{E} = \frac{2}{3}\mathcal{E}_b = \frac{2}{3}\frac{P}{R}$. Así,

$$d_{ij}^2 = \frac{16}{3}\frac{P}{R} \quad \text{para } i \neq j$$

Comparando esto con la distancia mínima de la señal PSK con $M=4$ se observa que la distancia mínima se ha incrementado por un factor de

$$\frac{d_{ij}^2}{d_{4PSK}^2} = \frac{\frac{16}{3}\frac{P}{R}}{4\frac{P}{R}} = \frac{4}{3}$$

Dado que la probabilidad de error es una función decreciente de la distancia Euclidiana mínima, se ha reducido la probabilidad de error mediante el empleo de este nuevo esquema. En realidad, puede decirse que la reducción en la probabilidad de error obtenida es equivalente a la reducción en la probabilidad de error debido al incremento en la potencia en un factor de 4/3. Esto a su vez es equivalente a 1.25 dB de ganancia de potencia. Esta ganancia de potencia no ha sido obtenida, logicamente, gratis. Puede verse que con este esquema de codificación en una duración en tiempo de $2/R$, la cual corresponde a la duración para transmitir dos bits, es necesario transmitir tres señales. En consecuencia, el ancho de esas señales se ha reducido por un factor de 2/3, y el ancho de banda requerido para transmitirlas se ha incrementado por un factor de 3/2. Otro problema con el esquema propuesto es que es obviamente más elaborado y requiere un esquema de decodificación más complejo.

El ejemplo anterior describe basicamente lo que efectúa la codificación. La codificación resulta en una menor probabilidad de error (lo que es equivalente a una SNR efectiva mayor) al precio de incrementar el ancho de banda y la complejidad del sistema. Debe mencionarse aqui que, a pesar que el empleo de codificación incrementa el ancho de banda[3], este incremento no es exponencial, como era el caso con la señalización ortogonal.

En un esquema general de señalización con señales codificadas, secuencias de longitud $k = RT$ de salida de la fuente se mapean en secuencias de longitud n de la forma

$$\boldsymbol{s}_i = \sqrt{\mathcal{E}}\underbrace{(\pm 1,\ \pm 1, \cdots,\ \pm 1)}_{n}$$

Estos puntos se ubican en los vértices de un hipercubo de lado igual a $2\sqrt{\mathcal{E}}$. La relación $R_c = k/n$, define la *taza de codificación*. Existe un total de 2^n vértices en un hipercubo n-dimensional, de los cuales es necesario elegir $M = 2^k$ palabras de código. Obviamente, es necesario seleccionar los 2^k vértices de forma tal que queden apartados entre sí tanto cuanto sea posible. Esto hace la distancia Euclidiana entre ellos grande y, de esta forma, reduce la probabilidad de error.

Es posible suponer la elección de 2^k vértices del hipercubo como las palabras del código y cada palabra del código difiere de otra en al menos d^H_{min} componentes. Este parámetro se denomina *distancia mínima de Hanning del código* y será definido más detalladamente en las próximas secciones. La relación entre distancia Euclidiana y distancia de Hamming es muy simple. Si las secuencias $\boldsymbol{s}_i$ y $\boldsymbol{s}_j$ difieren en d^H_{ij} posiciones, entones su distancia Euclidiana d^E_{ij} está relacionada con d^H_{ij} por

$$\left(d^E_{ij}\right)^2 = \sum_{\substack{1 \le l \le n \\ l\,:\, s^l_i \neq s^l_j}} \left(\pm 2\sqrt{\mathcal{E}}\right)^2 = 4 d^H_{ij}\mathcal{E}$$

Esto significa que la distancia Euclidiana mínima puede expresarse en términos de la distancia de Hamming mínima como

$$(d^E_{min})^2 = 4 d^H_{min}\mathcal{E}$$

Si suponemos ahora que se transmite $\boldsymbol{s}_i$ y se utiliza la cota de unión se tiene que

$$p_{ei} \le MQ\left[\sqrt{\frac{4d^H_{min}\mathcal{E}}{2N_0}}\right] \le \frac{M}{2} e^{-d^H_{min}\mathcal{E}/N_0}$$

donde en el último paso se usó la cota sobre la función Q introducida en el Capítulo 2. Teniendo en cuenta ahora que el contenido de energía de cada palabra de código es $n\mathcal{E}$ y este es igual a PT, se tiene que

[3] Existen esquemas de modulación - codificación que incrementan la distancia Euclidiana entre palabras de código pero no incrementan el ancho de banda. Esos esquemas se discutirán más adelante en este Capítulo.

$$\mathcal{E} = \frac{PT}{n} = \frac{RT}{n}\mathcal{E}_b = \frac{k}{n}\mathcal{E}_b = R_c\mathcal{E}_b$$

donde se usó la relación $\mathcal{E}_b = P/R$. De esta forma,

$$p_e \leq \frac{M}{2} e^{-d_{min}^H R_c \mathcal{E}_b/N_0}$$

(el subíndice i no se incluye porque la cota es independiente de i). Si no se hubiera empleado codificación, o sea, si se hubieran empleado todos los vértices del hipercubo k-dimensional en lugar de los 2^k vértices de un hipercubo n dimensional, se tendría la siguiente cota de unión de probabilidad de error

$$p_e \leq MQ\left[\sqrt{\frac{2\mathcal{E}_b}{2N_0}}\right] \leq \frac{M}{2} e^{-\mathcal{E}_b/N_0}$$

Comparando estas dos cotas es posible concluir que la codificación resultó en una ganancia de potencia equivalente a

$$G_{codif} = d_{min}^H R_c$$

la cual se denomina *ganancia de codificación*. Es posible ver que la ganancia de codificación es una función de dos parámetros de la codificación, la distancia de Hanning mínima y la taza de codificación. Notar que, en general, $R_c < 1$ y $d_{min}^H \geq 1$, y de esta forma la ganancia de codificación puede ser mayor o menor que 1 [4]. Es importante tener en cuenta que existen muchos códigos que proveen buena ganancia de codificación. La relación que define la ganancia de codificación enfatiza nuevamente que para n y k dados, el mejor código es aquel que provee la mayor distancia de Hamming mínima.

Para estudiar los requerimientos de ancho de banda de codificación, es posible observar que, cuando no se utiliza codificación, el ancho de los pulsos empleados para trasmitir un bit está dado por

$$T_b = \frac{1}{R}$$

Después de usar codificación, se deben transmitir ahora n pulsos en la misma duración en la que se transmitían k pulsos, lo cual significa que la duración de cada pulso se reduce en un factor de $k/n = R_c$. En consecuencia, *la relación de expansión de ancho de banda* está dada por

$$B = \frac{W_{con\ codif}}{W_{sin\ codif}} = \frac{1}{R_c} = \frac{n}{k}$$

De esta forma, el ancho de banda ha aumentado linealmente. Puede demostrarse que para un canal AWGN existe una secuencia de códigos con parámetros (n_i, k_i) con una taza fija ($\frac{k_i}{n_i} = R_i$ independiente de i), que satisface

$$R_c = \frac{k}{n} < \frac{1}{2}\log\left(1 + \frac{P}{N_0 W}\right)$$

donde $\frac{1}{2}\log\left(1 + \frac{P}{N_0 W}\right)$ es la capacidad del canal en bits por transmisión[5], para la cual la probabilidad de error tiende a cero cuando n_i se vuelve cada vez mayor. Logicamente, para tal esquema el ancho de banda se expande en un factor modesto y no crece exponencialemtne, como en el caso de la señalización ortogonal.

En este Capítulo se estudiarán dos tipos principales de codificación, *codificación en bloques* y *codificación convolucional*. La codificación por bloques es la que se ha discutido hasta el momento. En cada bloque de

[4] A pesar que en un diseño de códigos malo es posible tener $d_{min}^H = 0$, se ignorarán estos casos.

[5] La capacidad en bits por segundo es $W\log\left(1 + \frac{P}{N_0 W}\right)$.

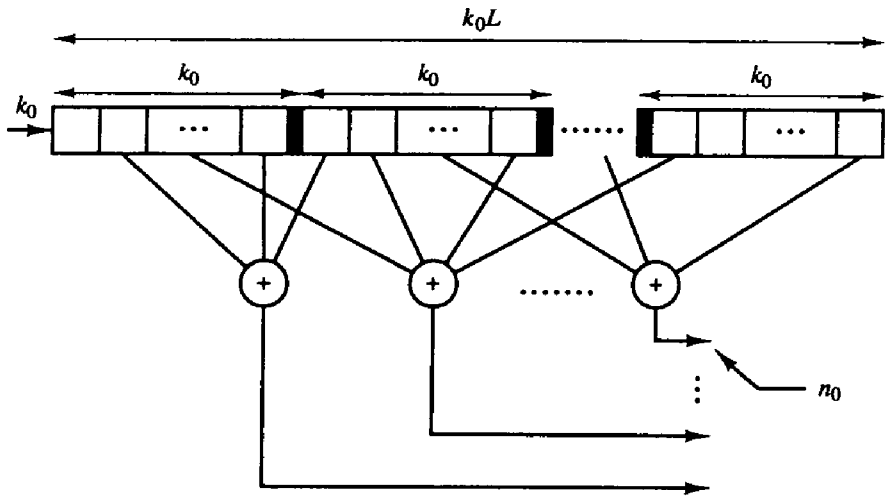

Figura 8.15: Codificador convolucional.

código la secuencia de información se divide en bloques de longitud k, y cada bloque se mapea a la entrada al canal de longitud n. Este mapeo es independiente de los bloques previos, o sea, no existe memoria de un bloque a otro. En la codificación convolucional existe un registro de desplazamiento de longitud k_0L como mostrado en la figura 8.15. Los bits de información entran al registro de desplazamiento de a k_0 bits a la vez, y luego se transmiten por el canal n_0 bits, combinaciones lineales de los varios bits del registro de desplazamiento. Esos n_0 bits dependen solamente de los bits k_0 recientes que han entrado al registro de desplazamiento y también de los $(L-1)k_0$ contenidos previos del registro de desplazamiento que constituyen su *estado*. La cantidad

$$m = Lk_0$$

se define como la *restricción de longitud* de la codificación convolucional y el número de estados del código convolucional es igual a 2^{m-k_0}. La taza de la codificación convolucional se define como

$$R_c = \frac{k_0}{n_0}$$

La principal diferencia entre codificación en bloques y codificación convolucional es la existencia de memoria en esta última.

8.5 Codificación lineal en bloques

Un código de bloques (n,k)[6] está completamente definido por $M = 2^k$ secuencias binarias de longitud n llamadas *palabras de código*. Un código $\mathcal{C}$ consiste en M palabras de código $\boldsymbol{c}_i$ para $1 \leq i \leq 2^k$,

$$\mathcal{C} = \{\boldsymbol{c}_1, \boldsymbol{c}_2, \cdots, \boldsymbol{c}_M\}$$

donde cada $\boldsymbol{c}_i$ es una secuencia de longitud n con componentes iguales a 0 o 1.

Definición 1. Un código de bloques es *lineal* si cualquier combinación de dos palabras de código es también otra palabra de código. En el caso binario esto requiere que si $\boldsymbol{c}_i$ y $\boldsymbol{c}_j$ son palabras de código, entonces $\boldsymbol{c}_i \oplus \boldsymbol{c}_j$ es también una palabra de código, donde $\oplus$ es la suma módulo 2.

Con esta definición es fácil ver que cualquier código de bloques lineal es un subespacio k dimensional de un espacio n dimensional. Es obvio también que toda secuencia $\mathbf{0}$ es una palabra de código de cualquier

[6]Desde este punto se discutirán códigos binarios a menos que se especifique lo contrário.

código de bloques lineal dado que puede escribirse como $\boldsymbol{c}_i \oplus \boldsymbol{c}_j$ para cualquier palabra de código $\boldsymbol{c}_j$. Notar que, de acuerdo a esta definición, la linealidad de un código depende solamente de las palabras de código y no de la forma en que se mapean las secuencias de información en las palabras de código. Sin embargo, es natural suponer que si la secuencia de información x_1 (de longitud k) se mapea en una palabra de código $\boldsymbol{c}_1$ (de longitud n) y la secuencia de información x_2 se mapea en $\boldsymbol{c}_2$, entonces $x_1 \oplus x_2$ se mapeará en $\boldsymbol{c}_1 \oplus \boldsymbol{c}_2$. Apartir de aquí se supondrá que todos los códigos lineales que se estudiarán poseen esta propiedad especial.

Ejemplo Un código $(5, 2)$ está definido por

$$\mathcal{C} = \{00000,\ 10100,\ 01111.\,11011\}$$

Es muy fácil verificar que este código es lineal. Si el mapeo entre secuencias de información y palabras de código está dado por

$$\begin{aligned} 00 &\rightarrow 00000 \\ 01 &\rightarrow 10100 \\ 10 &\rightarrow 01111 \\ 11 &\rightarrow 11011 \end{aligned}$$

entonces, la propiedad especial mencionada es satisfecha. Si es mapeo está dado por

$$\begin{aligned} 00 &\rightarrow 10100 \\ 01 &\rightarrow 01111 \\ 10 &\rightarrow 00000 \\ 11 &\rightarrow 11011 \end{aligned}$$

entonces la propiedad especial no se satisface. En ambos casos el código es lineal.

Definición 2. La *distancia de Hamming* entre dos palabras de código $\boldsymbol{c}_i$ y $\boldsymbol{c}_j$ es el número de componentes en las cuales difieren las dos palabras de código y se notará por $d(\boldsymbol{c}_i, \boldsymbol{c}_j)$[7].

Definición 3. El *peso de Hamming*, o simplemente el *peso* de una palabra de código $\boldsymbol{c}_i$ es el número de componentes no cero de la palabra de código, notada por $w(\boldsymbol{c}_i)$.

Definición 4. La distancia mínima de un código es la distancia de Hamming mínima entre dos palabras de código diferentes, o sea

$$d_{min} = \min_{i \neq j, \boldsymbol{c}_i, \boldsymbol{c}_j} d(\boldsymbol{c}_i, \boldsymbol{c}_j)$$

Definición 5. El peso mínimo de un código es el mínimo de los pesos de las palabras de código excepto la palabra de código todo ceros, o sea

$$w_{min} = \min_{\boldsymbol{c}_i \neq \boldsymbol{0}} w(\boldsymbol{c}_i)$$

[7] Desde ahora la distancia de Hamming será d y la distancia Euclidiana d^E.

Teorema. En cualquier código lineal, $d_{min} = w_{min}$.
Prueba: Si $\boldsymbol{c}$ es una palabra de código, entonces $w(\boldsymbol{c}) = d(\boldsymbol{c}, \boldsymbol{0})$. También, si $\boldsymbol{c}_i$ y $\boldsymbol{c}_j$ son palabras de código, también lo es $\boldsymbol{c} = \boldsymbol{c}_i \oplus \boldsymbol{c}_j$ y, además, $d(\boldsymbol{c}_i, \boldsymbol{c}_i) = w(\boldsymbol{c})$. Esto implica que en un código lineal, a cualquier peso de una palabra de código existe una distancia de Hamming entre dos palabras de código, y para cualquier distancia de Hamming existe un peso de una palabra de código. En particular, esto muestra que $d_{min} = w_{min}$.

Matrices generadora y de paridad. En un código de bloques lineal (n, k) consideraremos las palabras de código correspondientes a las secuencias de información $\boldsymbol{e}_1 = (1000\cdots0)$, $\boldsymbol{e}_2 = (0100\cdots0)$, $\cdots$, $\boldsymbol{e}_k = (0000\cdots1)$, dadas por $\boldsymbol{g}_1, \boldsymbol{g}_2, \cdots, \boldsymbol{g}_k$, respectivamente, donde cada una de las secuencias $\boldsymbol{g}_i$ es una secuencia binaria de longitud n. Ahora, cualquier información $\boldsymbol{x} = (x_1, x_2, \cdots, x_k)$ puede escribirse como

$$\boldsymbol{x} = \sum_{i=1}^{k} x_i \boldsymbol{e}_i$$

y en consecuencia la palabra de código correspondiente será

$$\boldsymbol{c} = \sum_{i=1}^{k} x_i \boldsymbol{g}_i$$

Si se define la *matriz generadora* de este código como

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{g}_1 \\ \boldsymbol{g}_2 \\ \vdots \\ \boldsymbol{g}_k \end{bmatrix} = \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1n} \\ g_{21} & g_{22} & \cdots & g_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ g_{k1} & g_{k2} & \cdots & g_{kn} \end{bmatrix}$$

entonces es posible escribir

$$\boldsymbol{c} = \boldsymbol{x}\boldsymbol{G}$$

Esto muestra que cualquier combinación lineal de las filas de la matriz generadora es una palabra de código. La matriz generadora para cualquier código de bloques lineal es una matriz de $n \times k$ de rango k (debido a que, por definición, la dimensión del subespacio es k). La matriz generadora de un código describe completamente el código. Cuando se dispone de la matriz generadora, la estructura del codificador es muy simple.

Ejemplo. Hallar la matriz generadora para el primer código del ejemplo anterior.

Es necesario hallar las palabras de código correspondientes a las secuencias de información (01) y (10), o sea, (10100) y (01111), respectivamente. Entonces

$$\boldsymbol{G} = \begin{bmatrix} 10100 \\ 01111 \end{bmatrix}$$

Puede verse que, para la secuencia de información (x_1, x_2), la palabra de código está dada por

$$(c_1, c_2, c_3, c_4, c_5) = (x_1, x_2)\boldsymbol{G}$$

ó $c_1 = x_1$, $c_2 = x_2$, $c_3 = x_1 \oplus x_2$, $c_4 = x_2$, $c_5 = x_2$.

El código anterior tiene la propiedad que la palabra de código correspondiente a cada secuencia de información comienza con una réplica de la secuencia de información seguida por bits extra. Este tipo de código se denomina *código sistemático*, y los bits adicionales que siguen a la secuencia de información en la palabra de código se denominan *bits de chequeo de paridad*. Una condición necesaria y suficiente para que un código sea sistemático es que la matriz generadora tenga la forma

$$\boldsymbol{G} = [\boldsymbol{I}_k \mid \boldsymbol{P}]$$

donde $\boldsymbol{I}_k$ es la matriz identidad de $k \times k$ y $\boldsymbol{P}$ es una matriz binaria de $k \times (n-k)$. En un código sistemático se tiene que

$$c_i = \begin{cases} x_i & 1 \leq i \leq k \\ \sum_{j=1}^{k} p_{ji} x_j & k+1 \leq i \leq n \end{cases}$$

donde todas las sumas son módulo 2.

Por definición, un código de bloques lineal $\mathcal{C}$ es un subespacio lineal k dimensional del espacio n dimensional. De álgebra lineal se sabe que si se toman todas las secuencias de longitud n que son ortogonales a todos los vectores de este subespacio k dimensional, el resultado será un subespacio $(N-k)$ dimensional denominado *complemento ortogonal* del subespacio k dimensional. Este subespacio $(n-k)$ dimensional define naturalmente un código lineal $(n, n-k)$, conocido como el *dual* del código $\mathcal{C}$ original (n, k), cuya notación es $\mathcal{C}^\top$. Obviamente las palabras de código del código original $\mathcal{C}$ y las del código dual $\mathcal{C}^\top$ son ortogonales. En particular, si escribimos la matriz generadora del código dual con $\boldsymbol{H}$, una matriz de $(n-k) \times k$, entonces cualquier palabra de código del código original es ortogonal a todas las filas de $\boldsymbol{H}$, o sea

$$\boldsymbol{c}\boldsymbol{H}^t = \boldsymbol{0} \quad \text{para todo } \boldsymbol{c} \in \mathcal{C}$$

La matriz $\boldsymbol{H}$, que representa la matriz generadora del código dual $\mathcal{C}^\top$ se denomina *matriz de chequeo de paridad* del código original $\mathcal{C}$. Dado que todas las filas de la matriz generadora son palabras de código, es posible concluir que

$$\boldsymbol{G}\boldsymbol{H}^t = \boldsymbol{0}$$

En el caso especial de un código sistemático ($\boldsymbol{G} = [\boldsymbol{I}_k \mid \boldsymbol{P}]$), la matriz de chequeo de paridad tiene la forma siguiente

$$\boldsymbol{H} = [-\boldsymbol{P}^t \mid \boldsymbol{I}_{n-k}]$$

Ejemplo. Hallar la matriz de chequeo de paridad para el ejemplo anterior.

En ese caso

$$\boldsymbol{G} = \begin{bmatrix} 10100 \\ 01111 \end{bmatrix} \quad \boldsymbol{I} = \begin{bmatrix} 10 \\ 01 \end{bmatrix} \quad \boldsymbol{P} = \begin{bmatrix} 100 \\ 111 \end{bmatrix}$$

Notando que en el caso binario $-\boldsymbol{P}^t = \boldsymbol{P}^t$, se concluye que

$$\boldsymbol{P}^t = \begin{bmatrix} 11 \\ 01 \\ 01 \end{bmatrix} \quad \boldsymbol{H} = \left[\begin{array}{cc|c} 1 & 1 & 100 \\ 0 & 1 & 010 \\ 0 & 1 & 001 \end{array}\right]$$

Códigos de Hamming. Los códigos de Hamming son una clase de códigos de bloques lineales con $n = 2^m - 1$, $k = 2^m - m - 1$ y $d_{min} = 3$, para algún entero $m \geq 2$. Como se verá posteriormente, con esta distancia

mínima esos códigos son capaces de proveer la capacidad de corrección de error para errores simples. La matriz de chequeo de paridad para esos códigos tiene una estructura muy simple. Consiste en todas las secuencias binarias de longitud m excepto la secuencia todo cero. La taza de esos códigos está dada por

$$R_c = \frac{2^m - m - 1}{2^m - 1}$$

la cual para valores grandes de m es próxima a 1. En consecuencia, los códigos Hamming son códigos de alta taza con una distancia mínima relativamente pequeña ($d_{min} = 3$). Se verá más adelante que la distancia mínima de un código está proximamente relacionada con la capacidad de corrección de errores. En consecuencia, los códigos de Hamming tiene una capacidad de corrección de errores limitada.

Ejemplo. Hallar la matriz de chequeo de paridad y la matriz generadora de un código de Hamming $(7, 4)$ en la forma sistemática.

En este caso $m = 3$, y en consecuencia $\boldsymbol{H}$ está formada por todas las secuencias binarias de longitud 3 excepto la secuencia 000. La matriz de chequeo de paridad en la forma sistemática será

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

y la matriz generadora será

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

8.5.1 Decodificación y desempeño de códigos lineales en bloques

El principal propósito del uso de la codificación en sistemas de comunicaciones es incrementar la distancia Euclidiana entre las señales transmitidas y, de esa forma, reducir la probabilidad de error para una potencia trasmitida especificada. Esto fue mostrado mediante un ejemplo en la sección previa. Teniendo en cuenta la figura 8.14, es posible ver que el objetivo se logra mediante la elección de palabras de código en los vértices del cubo tan apartadas como sea posible. Esto significa que una buena medida para comparar el desempeño de dos códigos es la distancia de Hamming entre las palabras de código. Analizar todas las distancias entre cualquier par de palabras es difícil y, en algunos casos, imposible. En consecuencia, la comparación entre diferentes códigos usualmente se basa en la distancia mínima del código, la que para códigos de bloques lineales es el peso mínimo. De esto se concluye que para cualquier n y k, un código con mayor d_{min} (ó w_{min}) se desempeña usualmente mejor cuando comparado con un código con menor distancia mínima.

Decodificación con decisión suave (soft decision)

En los Capítulos 5 y 6 se vió que el esquema de detección de señales óptimo en un canal AWGN es la detección basada en la minimización de la distancia Euclidiana entre la señal recibida y la señal trasmitida. Esto significa que después de recibir la salida del canal y pasarla a través de los filtros acoplados, se elige la señal de mensaje que está más próxima a la señal recibida en distancia Euclidiana. Al utilizar señales codificadas la situación es la misma. Suponiendo que se emplea PSK binaria para la transmisión de un mensaje codificado, una palabra de código $\boldsymbol{c}_i = (c_{i1}, c_{i2}, \cdots, c_{in})$ se mapea en una secuencia $s_i(t) = \sum_{k=1}^{n} \psi_{ik}(t - (k-1)T)$, donde

$$\psi_{ik}(t) = \begin{cases} \psi(t) & \text{si } c_{ik} = 1 \\ -\psi(t) & \text{si } c_{ik} = 0 \end{cases}$$

y $\psi(t)$ es una señal de duración T y energía $\mathcal{E}$, la cual es cero fuera del intervalo $[0, T]$. La distancia Euclidiana entre dos señales arbitrárias es

$$\left(d_{ij}^E\right)^2 = \sum_{1 \le k \le n,\, k:c_{ik} \neq c_{jk}} \left(\pm 2\sqrt{\mathcal{E}}\right)^2 = 4 d_{ij}^H \mathcal{E}$$

Esto determina una relación simple entre la distancias Euclidiana y de Hamming cuando se emplea el esquema PSK binario de señalización (o cualquier esquema antipodal de señalización). Para la señalización ortogonal, donde $\psi_1(t)$ y $\psi_2(t)$ son ortogonales, la relación equivalente es

$$\left(d_{ij}^E\right)^2 = \sum_{1 \le k \le n,\, k:c_{ik} \neq c_{jk}} \int_0^T [\psi_1(t) - \psi_2(t)]^2 dt = 2 d_{ij}^H \mathcal{E}$$

Luego, usando la relación general

$$p_e \;=\; Q\left[\frac{d^E}{\sqrt{2N_0}}\right]$$

se obtiene

$$p(j \text{ recibido } |i \text{ enviado}) = \begin{cases} Q\left[\sqrt{\frac{d_{ij}\mathcal{E}}{N_0}}\right] & \text{para señalización ortogonal} \\ Q\left[\sqrt{\frac{2d_{ij}\mathcal{E}}{N_0}}\right] & \text{para señalización antipodal} \end{cases}$$

Dado que $d_{ij} \ge d_{min}$ y como $Q[x]$ es una función decreciente de x, se concluye que

$$p(j \text{ recibido } |i \text{ enviado}) \le \begin{cases} Q\left[\sqrt{\frac{d_{min}\mathcal{E}}{N_0}}\right] & \text{para señalización ortogonal} \\ Q\left[\sqrt{\frac{2d_{min}\mathcal{E}}{N_0}}\right] & \text{para señalización antipodal} \end{cases}$$

Utilizando ahora la cota de unión del Capítulo 5 se obtiene

$$p(\text{ error } |i \text{ enviado}) \le \begin{cases} (M-1)Q\left[\sqrt{\frac{d_{min}\mathcal{E}}{N_0}}\right] & \text{para señalización ortogonal} \\ (M-1)Q\left[\sqrt{\frac{2d_{min}\mathcal{E}}{N_0}}\right] & \text{para señalización antipodal} \end{cases}$$

y suponiendo mensajes equiprobables, finalmente se concluye que

$$p_e \le \begin{cases} (M-1)Q\left[\sqrt{\frac{d_{min}\mathcal{E}}{N_0}}\right] & \text{para señalización ortogonal} \\ (M-1)Q\left[\sqrt{\frac{2d_{min}\mathcal{E}}{N_0}}\right] & \text{para señalización antipodal} \end{cases}$$

Estas son cotas sobre la probabilidad de error de un sistema de comunicaciones con codificación cuando se emplea demodulación óptima. Demodulación óptima significa pasar la señal recibida $r(t)$ a través de un banco de filtros acoplados para obtener el vector recibido $\boldsymbol{r}$ y luego hallar el punto de la constelación más próximo a $\boldsymbol{r}$ en el sentido de la distancia Euclidiana. Este tipo de decodificación, que involucra hallar la distancia Euclidiana mínima se denomina *decodificación de decisión suave (soft decision)* y requiere su cálculo con números reales.

Ejemplo Comparar el desempeño de un sistema de transmisión de datos no codificado con el desempeño de un sistema codificado que usa un código de Hamming $(7, 4)$, dado en el último ejemplo, cuando se aplican a la transmisión de una fuente binaria con velocidad $R = 10^4$ bits/seg. El canal se supone AWGN, la potencia recibida es de $1\,\mu$W y la densidad espectral de potencia es $\frac{N_0}{2} = 10^{-11}$ W/Hz. El esquema de modulación es PSK binaria.

1. Si no se emplea codificación se tiene que

$$p_b = Q\left[\sqrt{\frac{2\mathcal{E}_b}{N_0}}\right] = Q\left[\sqrt{\frac{2P}{RN_0}}\right]$$

Pero $\frac{2P}{RN_0} = \frac{10^{-6}}{10^4 \times 10^{-11}} = 10$ y de esta forma

$$p_b = Q[\sqrt{10}] = Q[3.16] \cong 7.86 \times 10^{-4}$$

La probabilidad de error para cuatro bits será

$$p_{\text{error en 4 bits}} = 1 - (1 - p_b)^4 \cong 3.1 \times 10^{-3}$$

2. Si se emplea codificación, se tiene que $d_{min} = 3$ y

$$\frac{\mathcal{E}}{N_0} = R_c \frac{\mathcal{E}_c}{N_0} = R_c \frac{P}{RN_0} = \frac{4}{7} \times 5 = \frac{20}{7}$$

Entonces, la probabilidad de error de mensaje estará dada por

$$p_e \leq 15Q\left[\sqrt{\frac{2d_{min}}{N_0}}\right] = 15Q\left[\sqrt{3\frac{40}{7}}\right] = 15Q[4.14] \cong 2.6 \times 10^{-4}$$

Puede verse que utilizar este código simple disminuye la probabilidad de error por un factor de 12. El precio pagado, obviamente, fue el incremento en el ancho de banda requerido para la transmisión de los mensajes. La relación de expansión de ancho de banda está dada por

$$\frac{W_{codif}}{W_{no\ codif}} = \frac{1}{R_c} = \frac{7}{4} = 1.75$$

Decodificación con decisión fuerte (hard decision). Un esquema de decodificación más simple y frecuente es realizar *decisiones binarias fuertes* (hard decision) sobre las componentes del vector recibido $\boldsymbol{r}$, y luego hallar la palabra de código más próxima en el sentido de Hamming. El ejemplo siguiente clarifica la distinción entre *soft y hard decision*.

Ejemplo: Un código $(3, 1)$ consiste de dos palabras de código 000 y 111, Las palabras de código se transmiten usando modulación PSK binaria con $\mathcal{E} = 1$. El vector recibido (las salidas muestreadas de los filtros acoplados) es $\boldsymbol{r} = (0.5, 0.5, -3)$. Si se utiliza decisión suave se debe comparar la distancia Euclidiana entre $\boldsymbol{r}$ y las dos constelaciones de puntos $(1, 1, 1)$ y $(-1, -1, -1)$ y elegir la menor. Se tiene que $(d^E(\boldsymbol{r}, (1, 1, 1))^2 = 0.5^2 + 0.5^2 + 4^2 = 16.6$ y $(d^E(\boldsymbol{r}, (-1, -1, -1))^2 = 1.5^2 + 1.5^2 + (-2)^2 = 8.5$ y entonces un decodificador suave decidiría $\boldsymbol{r}$ como $(-1, -1, -1)$ o equivalentemente $(0, 0, 0)$. Sin embargo, si se empleara decisión fuerte, $\boldsymbol{r}$ se detecta primero componente a componente como $+1$ o 0. Esto requiere una comparación de las componentes de $\boldsymbol{r}$ con el umbral 0. El vector resultante $\boldsymbol{y}$ es entonces $\boldsymbol{y} = (1, 1, 0)$. Luego, es necesario comparar $\boldsymbol{y}$ con $(1, 1, 1)$ y $(0, 0, 0)$ y hallar el más próximo en el sentido de la distancia de Hamming. El resultado es obviamente $(1, 1, 1)$. Como puede concluirse de este ejemplo los resultados de la decodificación suave o fuerte pueden ser muy diferentes. Obviamente, la detección óptima es usando soft decisión y a que este método permite lograr una menor probabilidad de error.

$\boldsymbol{c}_1$	$\boldsymbol{c}_2$	$\boldsymbol{c}_3$	$\cdots$	$\boldsymbol{c}_M$
$\boldsymbol{e}_1$	$\boldsymbol{e}_1 \oplus \boldsymbol{c}_2$	$\boldsymbol{e}_1 \oplus \boldsymbol{c}_3$	$\cdots$	$\boldsymbol{e}_1 \oplus \boldsymbol{c}_M$
$\boldsymbol{e}_2$	$\boldsymbol{e}_2 \oplus \boldsymbol{c}_2$	$\boldsymbol{e}_2 \oplus \boldsymbol{c}_3$	$\cdots$	$\boldsymbol{e}_2 \oplus \boldsymbol{c}_M$
$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$
$\boldsymbol{e}_2^{(n-k)} - 1$	$\boldsymbol{e}_2^{(n-k)} - 1 \oplus \boldsymbol{c}_2$	$\boldsymbol{e}_2^{(n-k)} - 1 \oplus \boldsymbol{c}_3$	$\cdots$	$\boldsymbol{e}_2^{(n-k)} - 1 \oplus \boldsymbol{c}_M$

Tabla 8.1: El arreglo estándar.

Existen tres pasos básicos involucrados en la decodificación fuerte. Primero, se realiza la demodulación pasando la señal recibida a través de los filtros acoplados y se muestrea la salida para obtener el vector $\boldsymbol{r}$. Segundo, se compara las componentes de $\boldsymbol{r}$ con el umbral y se cuantiza cada componente a uno de los dos niveles para obtener el vector $\boldsymbol{y}$. Finalmente, se realiza la decodificación hallando la palabra de código más cercana a $\boldsymbol{y}$ en el sentido de Hamming. En esta sección se discutirá una forma sistemática de realizar el proceso de decisión fuerte.

Se definirá primero la noción de *arreglo estándar*. Las palabras del código en cuestión serán $\boldsymbol{c}_1$, $\boldsymbol{c}_2$, $\cdots$, $\boldsymbol{c}_M$, cada una de longitud n, $M = 2^k$ y además $\boldsymbol{c}_1$ es la palabra todo ceros. Un arreglo estándar es un arreglo de $2^{n-k} \times 2^k$ cuyos elementos son secuencias binarias de longitud n y se genera escribiendo todas las palabras de código en una fila comenzando con la palabra todos ceros. Esto constituye la primera fila del arreglo estándar. Para escribir la segunda fila, se buscan todas las secuencias binarias de longitud n que no están en la primera fila del arreglo. Se elige de esas palabras de código la que tenga el peso mínimo y se llamará $\boldsymbol{e}_1$. Se la escribirá en $\boldsymbol{c}_1$ (notar que $\boldsymbol{c}_1 \oplus \boldsymbol{e}_1 = \boldsymbol{e}_1$, dado que $c_1 = (0, 0, \cdots, 0)$), y se escribe $\boldsymbol{e}_1 \oplus \boldsymbol{c}_i$ en $\boldsymbol{c}_i$ para $2 \leq i \leq M$. La tercera fila del arreglo se completa en forma similar. De las secuencias de longitud n que no han sido utilizadas en las primeras dos filas se elige la de menor peso y se la denomina $\boldsymbol{e}_2$. Luego, los elementos de la tercera fila comienzan con $\boldsymbol{c}_i \oplus \boldsymbol{e}_2$. Este proceso continúa hasta que no existan secuencias de longitud n para comenzar una nueva fila. La tabla 8.1 muestra el arreglo estándar generado como explicado. Cada fila del arreglo estándar se denomna *conjunto complementario* (coset), y el primer elemento de cada coset ($\boldsymbol{e}_l$ en general) se denomina el *conjunto complementário guía* (coset leader). El arreglo estándar tiene las siguientes propiedades importantes.

Teorema. Todos los elementos del arreglo estándar son diferentes.
Prueba. Suponiendo que dos elementos del arreglo estándar son iguales, esto puede suceder de dos formas,

1. Los dos elementos iguales pertenecen al mismo coset. En este caso se tiene $\boldsymbol{e}_l \oplus \boldsymbol{c}_i = \boldsymbol{e}_l \oplus \boldsymbol{c}_j$, de lo cual se concluye que $\boldsymbol{c}_i = \boldsymbol{c}_j$ es imposible.
2. Los dos elementos iguales pertenecen a diferentes cosets. Aquí se tiene que $\boldsymbol{e}_l \oplus \boldsymbol{c}_i = \boldsymbol{e}_l \oplus \boldsymbol{c}_j$ para $l \neq k$, lo cual significa que $\boldsymbol{e}_l = \boldsymbol{e}_k \oplus (\boldsymbol{c}_i \oplus \boldsymbol{c}_j)$. Por la linealidad del código $\boldsymbol{c}_i \oplus \boldsymbol{c}_j$ es también una palabra de código; se la denominará $\boldsymbol{c}_m$. Esto significa que $\boldsymbol{e}_l = \boldsymbol{e}_k \oplus \boldsymbol{c}_m$ y en consecuencia $\boldsymbol{e}_l$ y $\boldsymbol{e}_k$ pertenecen al mismo coset, lo cual es imposible por suposición.

De este teorema es posible concluir que los arreglos estándar contienen exactamente 2^{n-k} filas.

Teorema. Si $\boldsymbol{y}_1$ y $\boldsymbol{y}_2$ son elementos del mismo coset, $\boldsymbol{y}_1\boldsymbol{H}^t = \boldsymbol{y}_2\boldsymbol{H}^t$.
Prueba. Es suficiente notar que, dado que $\boldsymbol{y}_1$ y $\boldsymbol{y}_2$ no están en el mismo coset, $\boldsymbol{y}_1 = \boldsymbol{e}_l \oplus \boldsymbol{c}_i$ y $\boldsymbol{y}_2 = \boldsymbol{e}_l \oplus \boldsymbol{c}_2$, entonces

$$\boldsymbol{y}_1\boldsymbol{H}^t = (\boldsymbol{e}_l \oplus \boldsymbol{c}_i)\boldsymbol{H}^t = \boldsymbol{e}_l\boldsymbol{H}^t + \boldsymbol{0} = (\boldsymbol{e}_l \oplus \boldsymbol{c}_j)\boldsymbol{H}^t = \boldsymbol{y}_2\boldsymbol{H}^t$$

De este teorema se concluye que cada coset del arreglo estándar puede identificarse univocamente mediante el producto $\boldsymbol{e}_l\boldsymbol{H}^t$. En general, para cualquier secuencia binaria $\boldsymbol{y}$ de longitud n, es posible definir el *síndrome* $\boldsymbol{s}$ como

$$\boldsymbol{s} = \boldsymbol{y}\boldsymbol{H}^t$$

si $\boldsymbol{y} = \boldsymbol{e}_l \oplus \boldsymbol{c}_i$, o sea, $\boldsymbol{y}$ pertenece al coset $(l+1)$-ésimo, entonces obviamente $\boldsymbol{s} = \boldsymbol{e}_l \boldsymbol{H}^t$. El síndrome es una secuencia binaria de longitud $n-k$ y correspondiente a cada coset existe un único síndrome. Obviamente, el síndrome correspondiente al primer coset, el cual consiste en las palabras del código, es $\boldsymbol{s} = \boldsymbol{0}$.

Ejemplo. Hallar el arreglo estándar para un código $(5, 2)$ con palabras 00000, 01011, 10101, 11110, Hallar también los síndromes correspondientes a cada coset.
La matriz generadora del código es

$$\boldsymbol{G} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

y la matriz de chequeo de paridad correspondiente a $\boldsymbol{G}$ es

$$\boldsymbol{H} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Usando la construcción explicada antes se obtiene el siguiente arreglo estándar

00000	01011	10101	11110	síndrome = 000
10000	11011	00101	01110	síndrome = 101
01000	00011	11101	10110	síndrome = 011
00100	01111	10001	11010	síndrome = 100
00010	01001	10111	11100	síndrome = 010
00001	01010	10100	11111	síndrome = 001
11000	10011	01101	00110	síndrome = 110
10010	11001	00111	01100	síndrome = 111

Suponiendo que el vector recibido $\boldsymbol{r}$ se ha comparado componente a componente con un umbral y el vector binario resultante es $\boldsymbol{y}$, es necesario hallar la palabra de código que está a distancia mínima de $\boldsymbol{y}$. Primero se busca en que coset se encuenta $\boldsymbol{y}$. Para hacer esto, se halla el síndrome de $\boldsymbol{y}$, calculando $\boldsymbol{s} = \boldsymbol{y}\boldsymbol{H}^t$. Luego de hallar $\boldsymbol{s}$, en el arreglo estándar se halla el coset correspondiente a $\boldsymbol{s}$. Suponiendo que el coset guía correspondiente a este coset es $\boldsymbol{e}_l$, y dado que $\boldsymbol{y}$ pertenece a este coset, será de la forma $\boldsymbol{e}_l \oplus \boldsymbol{c}_i$ para algún i. La distancia de Hamming de $\boldsymbol{y}$ a cualquier palabra de código $\boldsymbol{c}_j$ es entonces

$$d(\boldsymbol{y}, \boldsymbol{c}_j) = w(\boldsymbol{y} \oplus c_j) = w(\boldsymbol{e}_l \oplus \boldsymbol{c}_i \oplus \boldsymbol{c}_j)$$

Dado que el código es lineal, $\boldsymbol{c}_i \oplus \boldsymbol{c}_k = \boldsymbol{c}_k$ para algún $1 \leq k \leq M$. Esto significa que

$$d(\boldsymbol{y}, \boldsymbol{c}_j) = w(\boldsymbol{c}_k \oplus \boldsymbol{e}_l)$$

pero $\boldsymbol{c}_k \oplus \boldsymbol{e}_l$ pertenecen al mismo coset que $\boldsymbol{y}$. En consecuencia, para minimizar $d(\boldsymbol{y}, \boldsymbol{c}_j)$ es necesario hallar el elemento de peso mínimo en el coset al que pertenece $\boldsymbol{y}$. Por construcción del arreglo estándar, este elemento es el coset guía, o sea, se elige $\boldsymbol{c}_k = \boldsymbol{0}$ y entonces $\boldsymbol{c}_j = \boldsymbol{c}_i$. Esto significa que $\boldsymbol{y}$ se decodifica en $\boldsymbol{c}_i$ hallando

$$\boldsymbol{c}_i = \boldsymbol{y} \oplus \boldsymbol{e}_l$$

De esta forma, el procedimiento para la decodificación de decisión fuerte (hard decisión) puede resumirse como sigue

1. Hallar $\boldsymbol{r}$, la representación vectorial de la señal recibida.

2. Comparar cada componente de $\boldsymbol{r}$ con el umbral óptimo y hacer una decisión binaria sobre él para obtener el vector binario $\boldsymbol{y}$.
3. Hallar el síndrome $\boldsymbol{s} = \boldsymbol{y}\boldsymbol{H}^t$ de $\boldsymbol{y}$.
4. Hallar el coset correspondiente a $\boldsymbol{s}$ utilizando el arreglo estándar.
5. Hallar el coset guía correspondiente $\boldsymbol{e}$ y decodificarlo como $\boldsymbol{c} = \boldsymbol{y} \oplus \boldsymbol{e}$.

Debido a que en este esquema de decodificación la diferencia entre el vector $\boldsymbol{y}$ y el vector decodificado $\boldsymbol{c}$ es $\boldsymbol{e}$, este se denomina frecuentemente *el patrón de error*. Esto significa que *los cosets guía constituyen el conjunto de todos los patrones de error corregibles.*

Para obtener cotas de error en la decodificación fuerte, es posible notar que, dado que se realiza una decisión sobre cada bit individual, la probabilidad de error para cada bit para la señalización antipodal es

$$p_b = Q\left[\sqrt{\frac{2\mathcal{E}_b}{N_0}}\right]$$

y para señalización ortogonal es

$$p_b = Q\left[\sqrt{\frac{\mathcal{E}_b}{N_0}}\right]$$

La descripción del canal, entre la palabra de código de entrada $\boldsymbol{c}$ y la salida del limitador $\boldsymbol{y}$, es de entrada binaria - salida binaria y puede modelarse mediante un canal binario simétrico con probabilidad de cruzamiento p_b. Dado que el código es lineal, la distancia entre dos palabras de código $\boldsymbol{c}_i$ y $\boldsymbol{c}_j$ es igual a la distancia entre la palabra de código todo ceros $\mathbf{0}$ y la palabra de código $\boldsymbol{c}_i \oplus \boldsymbol{c}_j = \boldsymbol{c}_k$. Sin pérdida de generalidad es posible suponer que se transmite la palabra de código todo cero. Si se transmite $\mathbf{0}$, entonces la probabilidad de error, por la cota de unión, no puede exceder $(M-1)$ veces la probabilidad de decodificar la palabra de código que está más próxima de $\mathbf{0}$ en el sentido de la distancia de Hamming. Para esta palabra de código $\boldsymbol{c}$, que está a una distancia d_{min} de $\mathbf{0}$, se tiene

$$p(c|\mathbf{0}\text{ enviado}) = \begin{cases} \sum_{i=(d_{min}+1)/2}^{d_{min}} \binom{d_{min}}{i} p_b^i (1-p_b)^{d_{min}-i} & d_{min}\text{ impar} \\ \sum_{i=(d_{min}+2)/2}^{d_{min}} \binom{d_{min}}{i} p_b^i (1-p_b)^{d_{min}-i} + \binom{d_{min}}{i} p_b^{\frac{d_{min}}{2}} (1-p_b)^{\frac{d_{min}}{2}} & d_{min}\text{ par} \end{cases}$$

o en general,

$$p(c|\mathbf{0}\text{ enviado}) \leq \sum_{i=[d_{min}/2]+1}^{d_{min}} \binom{d_{min}}{i} p_b^i (1-p_b)^{d_{min}-i}$$

De esta forma,

$$p(c|\mathbf{0}\text{ enviado}) \leq (M-1) \sum_{i=[d_{min}/2]+1}^{d_{min}} \binom{d_{min}}{i} p_b^i (1-p_b)^{d_{min}-i}$$

Esto determina una cota superior sobre la probabilidad de error de un código de bloques lineal utilizando decodificación con decisión fuerte (hard decisión). Como se vió aqui, en ambos tipos de decodificación d_{min} juega un papel principal en la cota de la probabilidad de error. Esto significa que, para un (n, k) especificado, es deseable tener códigos con d_{min} grande.

Puede mostrarse que la diferencia entre el desempeño de los dos tipos de decodificaciones es rigurosamente 2 dB para un canal AWGN. O sea, la probabilidad de error de un esquema con decodificación de desición suave es comparable a un esquema de decisión dura cuya potencia es 2 dB más alta que para el esquema de

decisión suave. Puede mostrarse también que si en lugar de la cuantización de cada componente de $\boldsymbol{r}$ con dos niveles, se emplea un cuantizador de ocho niveles (tres bits por componente), la diferencia de desempeño con decisión suave (precisión infinita) se reduce a 0.1 dB. Este esquema de cuantización multinivel, que resulta de un compromiso entre las decisiones suave y fuerte, también se denomina en la literatura como decisión suave.

Ejemplo. Si en el ejemplo de comparación del desempeño de PSK con un canal AWGN se utiliza decodificación con decisión fuerte, ¿ como cambiarán los resultados?
Aquí $p_b = Q[\frac{10}{7}] = Q[2.39] = 0.0084$ y $d_{min} = 3$. En consecuencia,

$$p_e \leq \left(\begin{array}{c} 7 \\ 2 \end{array} \right) p_b^2(1-p_b)^2 + \left(\begin{array}{c} 7 \\ 3 \end{array} \right) p_b^3(1-p_b)^3 + \cdots + p_b^2 \cong 21 p_b^2 \cong 1.5 \times 10^{-3}$$

En este caso la codificación ha disminuido la probabilidad de error en un factor de 2, comparado con 12 del caso de decisión suave.

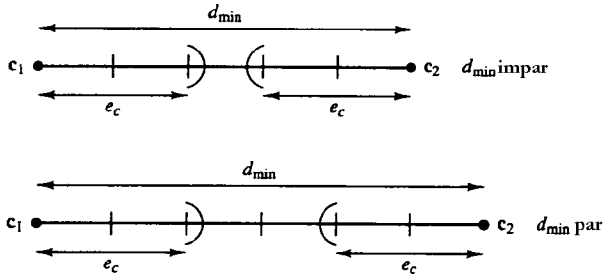

Figura 8.16: Relación entre e_c y d_{min}.

Detección de errores versus corrección de errores. Es posible considerar un código de bloques lineal $\mathcal{C}$ con distancia mínima d_{min}. Entonces, si se transmite $\boldsymbol{c}$ y se utiliza decodificación con decisión fuerte, cualquier palabra de código será decodificada correctamente si el receptor $\boldsymbol{y}$ está más próximo a $\boldsymbol{c}$ que cualquier otra palabra de código. Esta situación se muestra en la figura 8.16. Como mostrado, alrededor de cada palabra de código hay una "esfera de Hamming" de radio e_c, donde e_c es el número de errores corregibles. Siempre que esas esferas sean disjuntas, el código es capaz de corregir e_c errores. La condición para que las esferas no se solapen es la siguiente

$$\begin{cases} 2e_c + 1 \leq d_{min} & \text{para } d_{min} \text{ impar} \\ 2e_c + 2 \leq d_{min} & \text{para } d_{min} \text{ par} \end{cases}$$

ó

$$e_c \leq \begin{cases} \frac{d_{min}-1}{2} & \text{para } d_{min} \text{ impar} \\ \frac{d_{min}-2}{2} & \text{para } d_{min} \text{ par} \end{cases}$$

lo cual puede resumirse como

$$e_c \leq \left[\frac{d_{min} - 1}{2} \right]$$

En algunos casos son de interés procedimientos de decodificación que puedan *detectar* errores antes que corregirlos. Por ejemplo, en un sistema de comunicaciones donde está disponible un enlace de realimentación del receptor al transmisor, puede ser deseable detectar si ha ocurrido un error y, si fue así, solicitar al

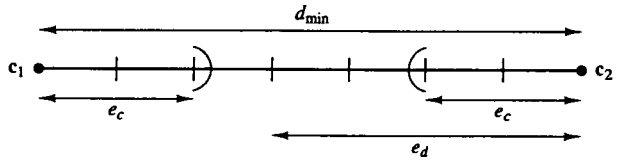

Figura 8.17: Relación entre e_c, e_d y d_{min}.

transmisor mediante el canal de realimentación que retransmita el mensaje. Si la capacidad de detección del error del código es e_d, entonces obviamente $e_d \leq d_{min} - 1$, dado que si ocurren $d_{min} - 1$ errores o menos, la palabra de código será convertida en una secuencia no perteneciente al código y de esta forma se detectará un error. Si es deseable la corrección de errores y la detección de errores, entonces existirá un compromiso entre esos requerimientos. La figura 8.17 demuestra esto, ya que de allí se puede concluir que

$$e_c + e_d \leq d_{min} - 1 \tag{8.7}$$

con la condición adicional $e_c \leq e_d$.

8.5.2 Códigos cíclicos

Los códigos cíclicos son un subconjunto de los códigos de bloques lineales para los cuales existen codificadores y decodificadores facilmente implementables. En esta sección se estudiará la estructura de los códigos cíclicos.

Definición. Un *código cíclico* es un código de bloques lineal con la condición adicional que si $\boldsymbol{c}$ es una palabra de código, un desplazamiento cíclico (un desplazamiento cíclico de una palabra de código $\boldsymbol{c} = (c_1, c_2, \cdots, c_{n-1}, c_n)$ se define como $\boldsymbol{c}^{(1)} = (c_2, c_3, \cdots, c_{n-1}, c_n, c_1)$), también es una palabra de código.

Ejemplo. El código $\{000, 110, 101, 011\}$ es un código cíclico dado que, como puede verificarse facilmente, es lineal, y un desplazamiento cíclico de cualquier palabra de código es también una palabra de código. El código $\{000, 010, 101, 111\}$ no es cíclico porque, a pesar de ser lineal, un desplazamiento cíclico de 101 no es una palabra de código.

Estructura de los códigos cíclicos

Para estudiar las propiedades de los códigos cíclicos es más simple representar cada palabra de código como un polinomio denominado el *polinomio de la palabra de código*. El polinomio de la palabra de código correspondiente a $\boldsymbol{c} = (c_1, c_2, \cdots, c_{n-1}, c_n)$ se define como

$$c(p) = \sum_{i=1}^{n} c_i p^{n-i} = c_1 p^{n-1} + c_2 p^{n-2} + \cdots + c_{n-1} p + c_n$$

El polinomio de $\boldsymbol{c}^{(1)} = (c_2, \cdots, c_{n-1}, c_n, c_1)$, el desplazamiento cíclico de $\boldsymbol{c}$, es

$$c^{(1)}(p) = c_2 p^{n-1} + c_3 p^{n-2} + \cdots + c_n p + c_1$$

lo cual puede escribirse como

$$c^{(1)}(p) = pc(p) + c_1(p^n + 1)$$

Notar que en el campo binario la suma y la resta son equivalentes, por lo que esto se reduce a

$$pc(p) = c^{(1)}(p) + c_1(p^n + 1)$$

ó

$$c^{(1)}(p) = pc(p) \mod (p^n + 1)$$

Si se desplaza $\boldsymbol{c}^{(1)}$ una vez más el resultado también será una palabra de código y su polinomio asociado será

$$c^{(2)}(p) = pc^{(1)}(p) \mod (p^n + 1) = p^2 c(p) \mod (p^n + 1)$$

donde $a(p) = b(p) \mod d(p)$ significa que $a(p)$ y $b(p)$ tiene el mismo resto cuando se dividen por $d(p)$. En general, para i desplazamientos se tiene el siguiente polinomio

$$c^{(i)}(p) = p^i c(p) \mod (p^n + 1)$$

Y para $i = n$ se obtiene

$$c^{(n)}(p) = p^n c(p) \mod (p^n + 1) = (p^n + 1)c(p) + c(p) \pmod{(p^n + 1)} = c(p)$$

El resultado anterior es evidente ya que desplazar cualquier palabra de código n veces la deja inalterada.

El teorema siguiente, que se introducirá sin prueba (Lin y Costello, 1983) es fundamental para un mejor estudio de los códigos cíclicos.

Teorema. En cualquier código cíclico (n, k) todos los polinomios son múltiplos de un polinomio de grado $n - k$ de la forma

$$g(p) = p^{n-k} + g_2 p^{n-k-1} + \cdots + g_{n-k}p + 1$$

denominado *polinomio generador*, donde $g(p)$ divide a $(p^n + 1)$. Además, para cualquier secuencia de información $\boldsymbol{x} = (x_1, x_2, x_3, \cdots, x_{k-1}, x_k)$ se tiene el *polinomio de secuencia de información* $X(p)$, definido por

$$X(p) = x_1 p^{k-1} + x_2 p^{k-2} + \cdots + x_{k-1}p + x_k$$

y cualquier polinomio de palabra de código correspondiente a $\boldsymbol{x}$ está dado por $c(p) = X(p)g(p)$.

El hecho que cualquier polinomio de palabra de código es el producto del polinomio generador y la secuencia de información implica que $\boldsymbol{c} = (c_1, c_2, \cdots, c_n)$ es la convolución discreta de las secuencias $\boldsymbol{x}$ y $\boldsymbol{g} = (1, g_2, g_3, \cdots, g_{n-k}, 1)$. Esto es muy importante para el diseño de los codificadores cíclicos.

Ejemplo. Para generar un código cíclico $(7, 4)$ es necesario un polinomio generador de grado $n - k = 3$. Este polinomio debe dividir a $p^7 + 1$. Dado que

$$p^7 + 1 = (p + 1)(p^3 + p^2 + 1)(p^3 + p + 1)$$

los únicos polinomios de tercer grado que dividen a $p^7 + 1$ son $p^3 + p^2 + 1$ y $p^3 + p + 1$. Es posible elegir cualquiera de esos polinomios para generar un código cíclico. Si elegimos $g(p) = p^3 + p^2 + 1$, y lo multiplicamos por todos los $X(p)$ de la forma

$$X(p) = x_1 p^3 + x_2 p^2 + x_3 p + x_4$$

donde x_i es cero o uno, se generan 16 polinomios de palabras de código de los cuales es posible obtener las palabras de código. El polinomio $X(p)$ es el *polinomio de mensaje* correspondiente a la secuencia binaria a ser codificada en general.

$$c(p) = X(p)g(p)$$

La tabla siguiente muestra las secuencias binarias y las palabras de código correspondientes:

Entrada	$X(p)$	$c(p) = X(p)g(p)$	Palabra de código
0000	0	0	0000000
0001	1	$p^3 + p^2 + p$	0001101
0010	p	$p^4 + p^3 + p$	0011010
0011	p^2	$p^5 + p^4 + p^2$	0110100
0100	p^3	$p^6 + p^5 + p^3$	1101000
0101	$p + 1$	$p^4 + p^2 + p + 1$	0010111
0110	$p^2 + p$	$p^5 + p^3 + p^2 + p$	0101110
1100	$p^3 + p^2$	$p^6 + p^4 + p^3 + p^2$	1011100
1001	$p^3 + 1$	$p^6 + p^5 + p^2 + 1$	1100101
0101	$p^2 + 1$	$p^5 + p^4 + p^3 + 1$	0111001
1010	$p^3 + p$	$p^6 + p^5 + p^4 + p$	1110010
0111	$p^2 + p + 1$	$p^5 + p + 1$	0100011
1110	$p^3 + p^2 + p$	$p^6 + p^2 + p$	1000110
1101	$p^3 + p^2 + 1$	$p^6 + p^4 + 1$	1010001
1011	$p^3 + p + 1$	$p^6 + p^5 + p^4 + p^3 + p^2 + p + 1$	1111111
1111	$p^3 + p^2 + p + 1$	$p^6 + p^3 + p + 1$	1001011

Como puede observarse de la tabla, todos los desplazamientos cíclicos de una palabra de código son palabras de código en si mismas.

La matriz generadora. Dado que la matriz generadora $\boldsymbol{G}$ para un código de bloques lineal no es única, para cualquier código cíclico con un polinomio generador especificado, existen varias matrices generadoras. Aquí se describe un método para obtener una matriz generadora en la forma sistemática. Una matriz generadora está en la forma sistemática si puede escribirse como sigue

$$\boldsymbol{G} = [\boldsymbol{I}_k | \boldsymbol{P}]$$

Por otro lado, todas las filas de una matriz generadora son palabras de código. Esto significa que los polinomios correspondientes a diferentes filas de la matriz generadora son múltiplos del polinomio generador $g(x)$. Por otro lado, la fila i-ésima de la matriz $\boldsymbol{G}$ es

$$\boldsymbol{g}_i = (0, 0, \cdots, 0, 1, 0, p_{i,1}, p_{i,2}, \cdots, p_{i,n-k}) \quad 1 \leq i \leq k$$

donde las primeras k componentes de $\boldsymbol{g}_i$ son todos ceros excepto la i-ésima componente, la cual es 1 y $p_{i,j}$ es el (i,j)-ésimo elemento de la matriz $\boldsymbol{P}$. De esta forma, el polinomio correspondiente a $\boldsymbol{g}_i$ será

$$g_i(p) = p^{n-i} + p_{i,1}p^{n-k-1} + p_{i,2}p^{n-k-2} + \cdots + p_{i,n-k}$$

Dado que $g_i(p)$ es el polinomio de palabra de código correspondiente a la palabra de código $\boldsymbol{g}_i$, debe ser un múltiplo de $g(p)$, el polinomio generador del código cíclico bajo estudio. En consecuencia se tiene que

$$g_i(p) = p^{n-i} + p_{i,1}p^{n-k-1} + p_{i,2}p^{n-k-2} + \cdots + p_{i,n-k} = X(p)g(p)$$

Ahora, dado que $g(p)$ es de grado $n-k$, se concluye que

$$p_{i,1}p^{n-k-1} + p_{i,2}p^{n-k-2} + \cdots + p_{i,n-k} = p^{n-i} \pmod{g(p)} \quad 1 \leq i \leq k$$

Esta relación muestra que teniendo $g(p)$ es posible hallar los $p_{i,j}$ para todo $1 \leq i \leq k$ y $1 \leq j \leq n-k$. En consecuencia, $\boldsymbol{P}$ puede obtenerse de $g(p)$ y así obtener $\boldsymbol{G}$, la matriz generadora.

Ejemplo. Hallar la matriz generadora en la forma sistemática para un código cíclico $(7,4)$ generado por $g(p) = p^3 + p^2 + 1$.
Es posible observar que

$$\begin{aligned} p^6 \, mod(p^3+p^2+1) &= p^2+p \\ p^5 \, mod(p^3+p^2+1) &= p+1 \\ p^4 \, mod(p^3+p^2+1) &= p^2+p+1 \\ p^3 \, mod(p^3+p^2+1) &= p^2+1 \end{aligned}$$

Entonces, teniendo los $p_{i,j}$ es posible obtener la matriz $\boldsymbol{G}$ como sigue

$$\boldsymbol{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

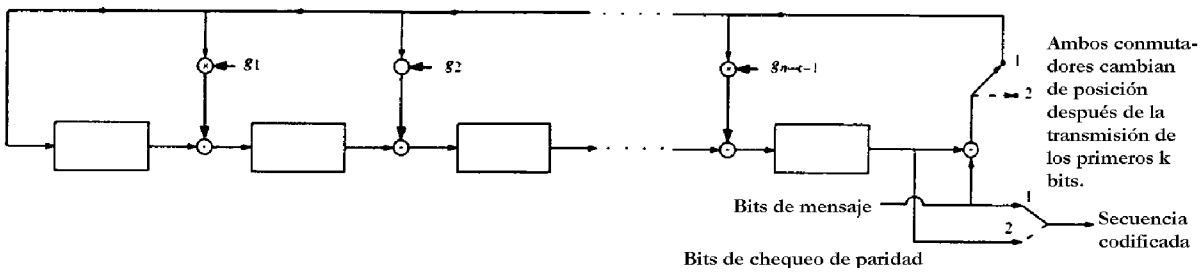

Figura 8.18: Implementación de un codificador cíclico.

Codificación de los códigos cíclicos. Comparados con la clase general de los códigos de bloques lineales, los códigos cíclicos tienen una estructura constructiva que permite una implementación de los codificadores más simple. Los codificadores cíclicos pueden implementarse mediante registros de desplazamiento. En esta seccón se examinarán algunas estructuras básicas para la codificación de códigos cíclicos. Un codificador cíclico facilmente implementable se basa en la observación que cualquier polinomio de palabra de código puede obtenerse multiplicando el polinomio generador $g(p)$ por la secuencia de entrada $X(p)$, o equivalentemente, convolucionando las secuencias $\boldsymbol{x}$ y $\boldsymbol{p}$. Esto puede hacerse facilmente mediante una estructura semejante a la de un filtro digital como ilustrado en la figura 8.18.

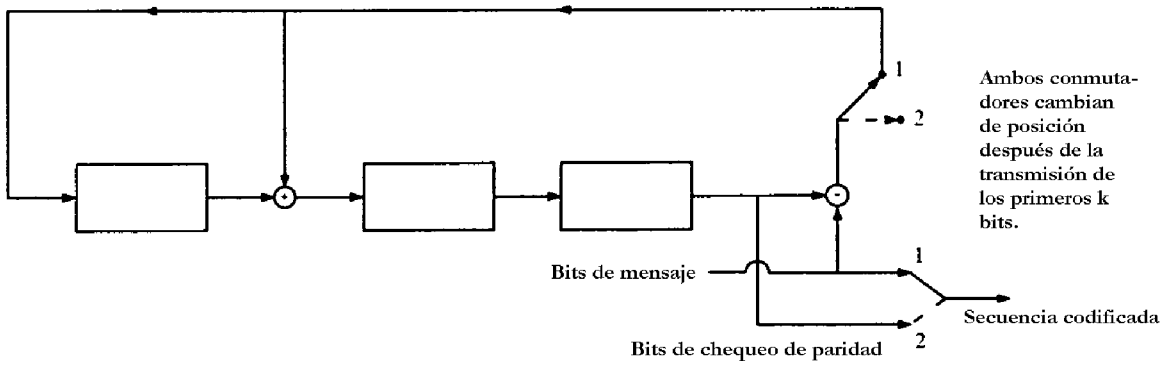

Figura 8.19: Estructura del codificador para un código cíclico de Hamming $(7, 4)$.

Ejemplo. La estructura del código cíclico $(7, 4)$ se ilustra en la figura 8.19.

Códigos BCH. Los códigos Bose, Chaudhuri y Hocquenghem (BCH) son un subconjunto de los códigos cíclicos que, en general, pueden diseñarse para la corrección de t errores. Esta versatilidad de diseño y la existencia de un algoritmo eficiente de decodificación para esos códigos (el algoritmo de Berlekamp-Massey) hace a esos códigos particularmente atractivos.

Para ser más específicos, para cualquier m y t, existe un código BCH con parámetros

$$\begin{aligned} n &= 2^m - 1 \\ n - k &\leq mt \\ d_{min} &= 2t + 1 \end{aligned}$$

Dado que m y t son arbitrários, el diseñador del sistema de comunicaciones tiene un gran número de selecciones en esta familia de códigos. Los códigos BCH están muy bien tabulados. La tabla 8.2 muestra los coeficientes de los polinomios generadores para códigos BCH de longitudes $7 \leq n \leq 255$. Los coeficientes de $g(p)$ se dan en forma octal. Así los coeficientes del polinomio generador de un código $(15, 5)$ son 2467, lo que en forma binaria corresponede a 10, 100, 110, 111. En consecuencia, el polinomio generador para este código es $g(p) = p^{10} + p^8 + p^5 + p^4 + p^2 + p + 1$.

Figura 8.20: Codificador Reed - Solomon.

Códigos Reed-Solomon. Los códigos Reed-Solomon son un subconjunto de los códigos BCH y pertenecen entonces a la família de los códigos cíclicos. Los códigos Reed-Solomon son códigos no binarios, o sea, en una palabra de código $\boldsymbol{c} = (c_1, c_2, \cdots, c_N)$, los elementos c_i, $1 \leq i \leq N$ son mienbros de un alfabeto q-ário. En la mayoría de los casos prácticos q se elige como una potencia de 2, $q = 2^k$. En tal caso, k bits de información se mapean en un único elemento del alfabeto q-ário y luego, utilizando un código Reed-Solomon (N, K), K símbolos q-ários se mapean en N símbolos q-ários y luego son transmitidos por el canal. La figura 8.20 muestra este procedimiento.

n	k	t	$g(p)$
7	4	1	13
15	11	1	23
	7	2	721
	5	3	2467
31	26	1	45
	21	2	3551
	16	3	107657
	11	5	5423325
	6	7	313365047
63	57	1	103
	51	2	12471
	45	3	1701317
	39	4	166623567
	36	5	1033500423
	30	6	157464165547
	24	7	17323260404441
	18	10	1363026512351725
	16	11	6331141367235453
	10	13	472622305527250155
	7	15	5231045543503271737
127	120	1	211
	113	2	41567
	106	3	11554743
	99	4	3447023271
	92	5	624730022327
	85	6	130704476322273
	78	7	26230002166130115
	71	9	6255010713253127753
	64	10	12065340255707731000 45
	57	11	335265252505705053517721
	50	13	544465125233140124215014 21
	43	14	177217722136512275212205743 43
	36	15	314607466652207504476457572 1735
	29	21	403114461367670603667530141176155
	22	23	123376070404722522435445626637647043
	15	27	220570424456045547705230137622176043 53
	8	31	704726405275103065147622427156773313021 7
255	247	1	435
	239	2	267543
	231	3	156720665
	223	4	75626641375
	215	5	23157564726421
	207	6	16176560567636227
	199	7	7633031270420722341
	191	8	2663470176115333714567

Tabla 8.2: Coeficientes de los polinomios generadores de algunnos códigos BCH.

n	k	t	$g(p)$
	187	9	52755313540001322236351
	179	10	22624710717340432416300455
	171	11	154162142123423560770616306 37
	163	12	75004155100756025515747245146 01
	155	13	37575130054076650157225064646776 33
	147	14	16421301735371655253041653054410117 11
	139	15	46140173206017556157072273024745356744 5
	131	18	21571333147151015126125027744214202416547 1
	123	19	12061405224206600371721032651614122627250626 7
	115	21	60526665572100247263636404600276352556313472737
	107	22	22205772322066256312417300235347420176574750154441
	99	23	10656667253473174222741416201574332252411076432303431
	91	25	67502650303274441727236317247325110755507627207243445 61
	87	26	11013676341474323643523163430717204620672254527331172131 7
	79	27	66700035637657500020270344207366174621015326711766541342355
	71	29	24024710520644321515554172112331163205444250362557643221706035
	63	30	10754475055163544325315217357707003666111726455267613656702543301
	55	31	73154252035011001330152753060320543254143267550105570444260354736 17
	47	42	25335420170626465630330413774062331751233341454460450056602455254317 3
	45	43	15202056055234161131101346376423740156367002447076237033202157025051541
	37	45	51363302550670074141774472454375304207357061743234323476443547374030440 03
	29	47	30257155366730714655270640123613771153422423242011741140602547574104035650 37
	21	55	12562152570603326560017731536076121032273414056530745425211531216144665134737 25
	13	59	46417320050525645444265737142500660043306774454765614031746772135702613446050054 7
	9	63	15726025217472463201031043255355134614162367212044074545112766115547705561677551605 7

Tabla 8.3: Coeficientes de los polinomios generadores de algunnos códigos BCH (continuación).

Un código Reed-Solomon es un código BCH definido por los siguientes parámetros

$$\begin{aligned} N &= q-1 = 2^k - 1 \\ K &= 1, 2, 3, \cdots, N-1 \\ D_{min} &= N - K + 1 \\ R_c &= \frac{K}{N} \end{aligned}$$

Este código es capaz de corregir hasta

$$t = \left[\frac{D_{min} - 1}{2} \right]$$

errores de símbolo.

Los códigos Reed-Solomon tienen particularmente buenas propiedades de distancia, son totalmente adecuados para la utilización junto con modulación q-ária, y son útiles en situaciones donde los errores tienden a aparecen en *ráfagas* antes que aleatoriamente. Esta última propiedad es una consecuencia de que las ráfagas causan solo algunos errores de símbolo en un código Reed-Solomon, lo cual puede corregirse facilmente.

Los códigos Reed-Solomon también pueden concatenarse con un código binario para proveer altos niveles de protección contra errores. El código binario utilizado en concatenación con el código Reed-Solomon puede ser un código en bloques o un código convolucional. El codificador binario y el decodificador binario se ubican a la derecha antes del modulador y después del demodulador y se denominan par codificador-decodificador *interno*. Si el código interno es de tipo (n, k), la combinación del codificador interno, el modulador digital, la señal del canal, el demodulador digital y el decodificador interno, pueden considerarse como un canal cuya entrada y salida son bloques binarios de longitud k, o equivalentemente, elementos de un alfabeto q-ário donde $q = 2^k$. Entonces puede utilizarse el código Reed-Solomon (usualmente denominado el código *externo*) sobre este canal de entrada q-ária y salida q-ária para proveer mayor protección contra errores. Puede verse facilmente que si las tazas de los códigos interno y externo son r_c y R_c, respectivamente, la taza del código concatenado será

$$R = R_c r_c$$

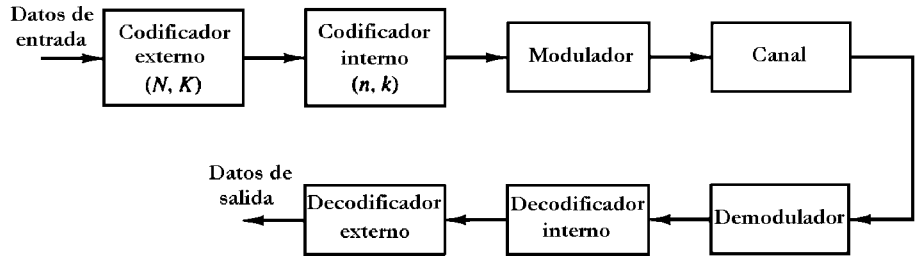

Figura 8.21: Diagrama en bloques de un sistema de comunicaciones con codificación concatenada.

También, la distancia mínima del código concatenado es el producto de las distancias mínimas de los códigos interno y externo. La figura 8.21 muestra un diagrama de bloques de un sistema de comunicaciones digitales que emplea el esquema de codificación concatenada.

8.6 Codificación convolucional

Los códigos convolucionales son diferentes de los códigos en bloques por la existencia de memoria en el esquema de codificación. En los códigos de bloques, cada bloque de k bits de entrada se mapea en un bloque de longitud de salida de n bits mediante un procedimiento definido por el código (por ejemplo, mediante $\boldsymbol{G}$ o $g(p)$) e independientemente de las entradas previas al codificador. La taza de ese código está dada por

$$R_c = \frac{k}{n}$$

En los códigos convolucionales cada bloque de k bits se mapea nuevamente en un bloque de n bits para ser transmitido por un canal, pero esos n bits no están solo determinados por los k bits de información sino también por los bits de información previos. Esta dependencia de los bits de información previos hace que el codificador sea una máquina de estados finita.

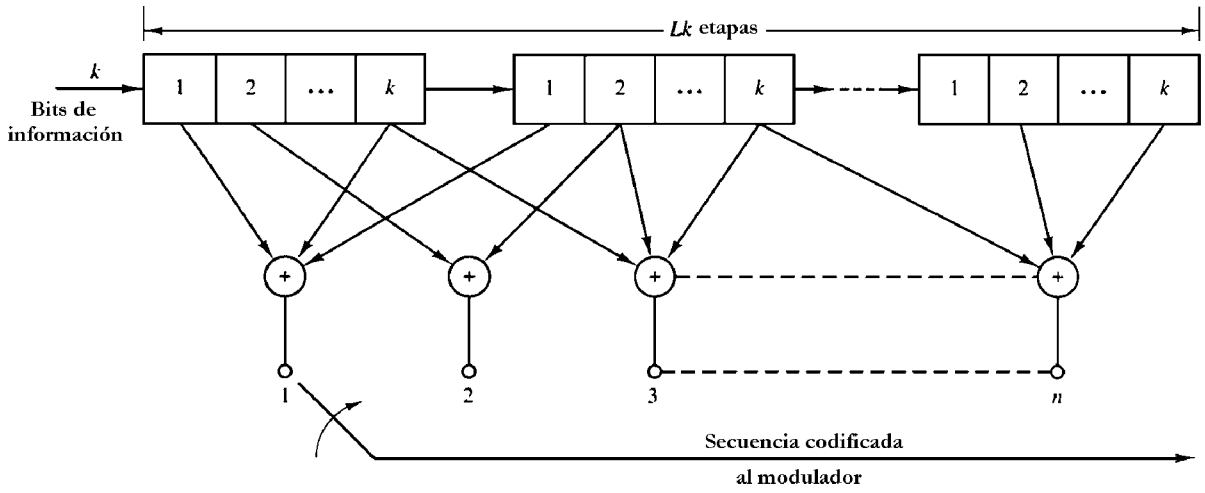

Figura 8.22: Diagrama en bloques de un codificador convolucional.

Para ser más específico, el diagrama de bloques de un codificador convolucional se muestra en la figura 8.22. El codificador convolucional consiste en un registro de desplazamiento con kL etapas donde L se denomina la *longitud restringida* del código. En cada instante de tiempo entran al registro de desplazamiento k bits de información y el contenido de las últimas k etapas del registro de desplazamiento se elimina. Después de que k bits han entrado al resgistro de desplazamiento, se calculan n combinaciones lineales de los contenidos del registro de desplazamiento, como mostrado en la figura, y se utilizan para codificar la señal. De este procedimiento de codificación es obvio que las n salidas del codificador dependen no solo de los k bits más recientes que han entrado al codificador sino también del contenido de las primeras $(L-1)k$ etapas del registro de desplazamiento antes que los k bits hubieran llegado. En consecuencia, el registro de desplazamiento es una máquina de estados finita con $2^{(L-1)k}$ estados. Dado que para k bits de entrada se tienen n bits de salida, la taza del código es

$$R_c = \frac{k}{n}$$

Ejemplo: En la figura 8.23 se muestra un codificador convolucional. En este codificador: $k = 1$, $n = 2$ y $L = 3$. En consecuencia, la taza del código es $\frac{1}{2}$ y el número de estados es $2^{(L-1)k} = 4$. Una forma de describir este código (diferente a dibujar el codificador) es especificar como dependen los dos bits de salida del codificador de los contenidos del registro de desplazamiento. Esto se hace usualmente especificando n vectores $\boldsymbol{g}_1$, $\boldsymbol{g}_2$, $\cdots$, $\boldsymbol{g}_n$, conocidos como *secuencias generadoras* del código convolucional. La i-ésima componente $(1 \leq i \leq kL)$ de $\boldsymbol{g}_j$, $1 \leq j \leq n$, es uno si la i-ésima etapa del registro de desplazamiento está conectada al combinador correspondiente al j-ésimo bit de salida, y cero en otro caso. En este ejemplo, las secuencias generadoras están dadas por

$$\boldsymbol{g}_1 = [1\,0\,1] \qquad \boldsymbol{g}_2 = [1\,1\,1]$$

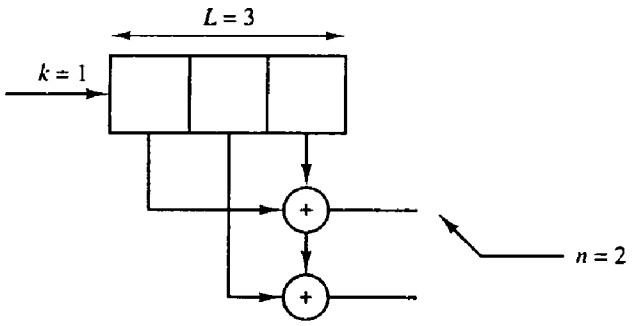

Figura 8.23: Codificador convolucional de taza $R_c = 1/2$.

8.6.1 Propiedades básicas

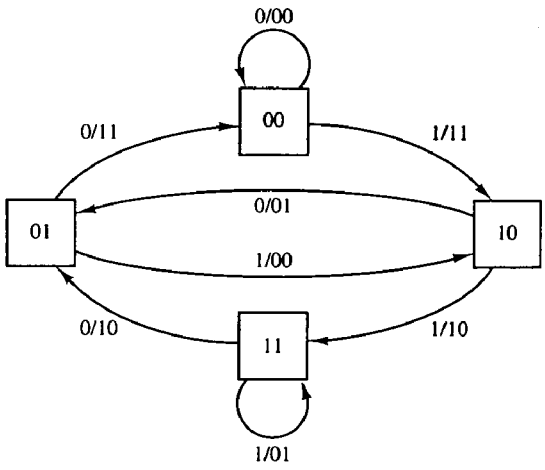

Figura 8.24: Diagrama de transición de estados para el codificador de taza 1/2.

Dado que un codificador convolucional tiene memoria finita, puede representarse facilmente mediante un *diagrama de transición de estados*. En este diagrama de transición, cada estado del codificador convolucional se representa mediante un bloque y las transiciones entre estados son líneas que conectan cada bloque. Sobre cada línea están especificadas tanto las entradas que causan la transición como las salidas correspondientes. El número de líneas que emergen de cada estado es en consecuencia igual al número de entradas posibles al codificador de ese estado, la cual es igual a 2^k. El número de líneas que van hacia cada estado es igual al número de estados del cual es posible una transición hacia ese estado. Este es igual al número de combinaciones de bits posibles que dejan el codificador cuando k bits entran al codificador. Ese número es nuevamente 2^k. La figura 8.24 muestra el diagrama de transición de estados para el código convolucional de la figura 8.23.

Un segundo, y más popular, método para describir códigos convolucionales es especificar su *diagrama trellis*. El diagrama trellis es una forma de mostrar la transición entre varios estados cuando el tiempo evoluciona. El diagrama trellis se obtiene especificando todos los estados sobre un eje vertical y repitiendo este eje vertical a lo largo del eje de tiempos. Luego, cada transición de un estado a otro se especifica por una línea conectando los dos estados sobre dos ejes verticales correspondientes a los dos instantes de tiempo. En cierto sentido el diagrama trellis no es nada más que una repetición del diagrama de transición de estados a lo largo del eje de tiempos. Como en el caso del diagrama de transición de estados, se tiene 2^k ramas de la trellis dejando cada estado y 2^k ramas volviendo a cada estado. En el caso en que $k = 1$, es común notar la rama correspondiente a la entrada 0 al codificador mediante una línea llena y la rama correspondiente a la entrada 1 al codificador con una línea a trazos. La figura 8.25 muestra el diagrama trellis para el código

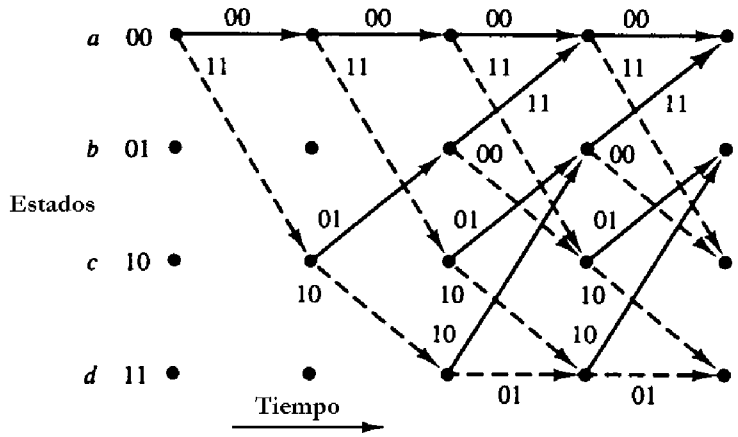

Figura 8.25: Trellis para el codificador de taza 1/2.

descripto mediante el codificador de la figura 8.23.

Codificación. El procedimiento de codificación en un código convolucional es muy simple. Se supone que el codificador se carga con ceros antes que el primer bit de información entre. Los bits de información entran al codificador de a k bits por vez y los n bits de salida correspondientes se transmiten por el canal. Este procedimiento continúa hasta que el último grupo de k bits se carga en el codificador y los n bits de salida correspondientes se envian por el canal. Se supondrá, por simplicidad, que después del último conjunto de k bits entra al codificador otro conjunto de $k(L-1)$ bits todos ceros y las n salidas correspondientes se transmiten por el canal. Esto hace que el codificador quede listo para la próxima transmisión.

Ejemplo ¿Cual es la secuencia codificada correspondientes a la secuencia de información $\boldsymbol{x} = (1101011)$ en el código convolucional mostrado en la figura 8.23?

Es suficiente notar que el codificador está en el estado 0 antes de la transmisión, y también después de que el último bit de información sea transmitido. Esto significa que la secuencia transmitida es $\boldsymbol{x}_1 = (11010110)$. usando esta secuencia de transmisión, se obtiene la siguiente secuencia codificada $\boldsymbol{c} = (11101000010010101 1)$.

Función transferencia. Para cada código convolucional, la función transferencia provee información sobre los diferentes caminos a través de la trellis que comienzan en el estado todo ceros y vuelven a ese estado por primera vez. De acuerdo a la convención de codificación descripta antes, cualquier palabra de código de un codificador convolucional corresponde a un camino a través de la trellis que comienza en el estado todo ceros y vuelve al estado todo ceros. Como se verá posteriormente, la función transferencia de un código convolucional juega un papel principal para obtener una cota para la probabilidad de error del código. Para obtener la función transferencia de un código convolucional, se divide el estado cero en dos estados, uno que especifica el estado de comienzo y el otro el primer retorno al estado todo ceros. Todos los otros estados se especifican como estados intermedios.

Se define una función de la forma $D^{\alpha}N^{\beta}J$, correspondiente a cada rama que conecta dos estados, donde α es el número de unos en la secuencia de bit de salida de cada rama y β es el número de unos en la secuencia de bits de entrada para esa rama.

La *función transferencia del código convolucional* es entonces la función transferencia del diagrama de flujo entre el estado todo ceros de inicio y el estado todo ceros final y será una función de tres parámetros D, N, J, especificada por $T(D, N, J)$. Cada elemento de $T(D, N, J)$ corresponde a un camino de la trellis, comenzando del estado todo ceros y terminando en el estado todo ceros. El exponente de J indica el número de ramas abarcadas por ese camino, el exponente de D muestra el número de unos en la palabra de código correspondientes a ese camino (o, equivalentemente, la distancia de Hamming de la palabra de código a la palabra de código todo ceros), y finalmente el exponente de N indica el número de unos en la secuencia de

información de entrada.

Dado que $T(D, N, J)$ indica las propiedades de todos los caminos a través de la trellis comenzando del estado todo ceros y volviendo a él *por primera vez*, entonces al obtenerla, cualquier lazo en el estado todo ceros debe ignorarse. Para obtener la función transferencia del código convolucional es posible utilizar las reglas para obtener la función transferencia de un diagrama de flujo.

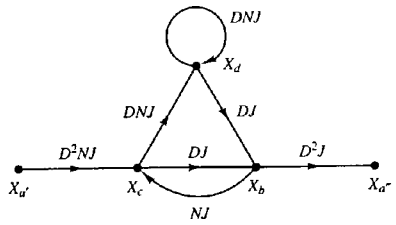

Figura 8.26: Diagrama de flujo para hallar la función transferencia.

Ejemplo Hallar la función transferencia del código convolucional de la figura 8.23.

La figura 8.26 ilustra el diagrama utilizado para obtener la función transferencia de este código. El código tiene un total de cuatro estados especificados por los contenidos de las dos primeras etapas del registro de desplazamiento. Se especificarán esos estados mediante las siguientes letras

$$\begin{aligned} 00 &\rightarrow a \\ 01 &\rightarrow b \\ 10 &\rightarrow c \\ 11 &\rightarrow d \end{aligned}$$

Como puede verse en la figura, el estado a se divide en dos estados a' y a'', o sea los estados de comienzo y de retorno. Usando relaciones de diagramas de flujo es posible escribir

$$\begin{aligned} X_c &= X_{a'}D^2NJ + NJX_b \\ X_b &= DJX_d + DJX_c \\ X_d &= DNJX_c + DNJX_d \\ X_{a''} &= D^2NJX_b \end{aligned}$$

Eliminando X_b, X_c y X_d se obtiene que

$$T(D, N, J) = \frac{X_{a''}}{X_{a'}} = \frac{D^5NJ^3}{1 - DNJ - DNJ^2}$$

Expandiendo $T(D, N, J)$ en la forma de un polinomio se obtiene

$$T(D, N, J) = D^5NJ^3 + D^6N^2J^4 + D^7N^3J^5 + \cdots$$

Esto indica que existe exactamente un camino a través de la trellis comenzando del estado todo ceros y volviendo al estado todo ceros por primera vez, que abarca tres ramas, correspondiente a la secuencia de información de entrada que contiene un 1 (y de esta forma, dos ceros) y la palabra de código para este camino tiene un peso de Hamming igual a 5. Este camino se indica con líneas llenas en la figura 8.27. Este camino es similar al de la palabra de código de peso mínimo de códigos de bloques. En realidad, este camino corresponde a la palabra de código que está a "distancia mínima" de la palabra de código todo ceros. Esta

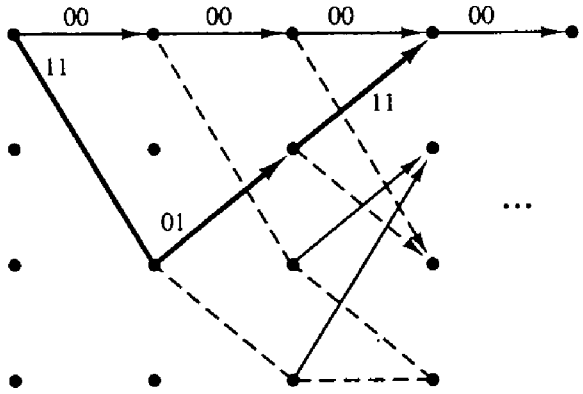

Figura 8.27: El camino correspondiente a D^5NJ^3 en el código representado en la figura 8.23.

distancia mínima, que es igual a la potencia mínima de D en la expansión de $T(D, N, J)$, se denomina la *distancia libre del código*, d_{free}. La distancia libre del código anterior es igual a 5. La forma general de la función transferencia es entonces

$$T(D, N, J) = \sum_{d=d_{free}}^{\infty} a_d D^d N^{f(d)} J^{g(d)}$$

Una forma más corta de la función transferencia, que provee solo información de los pesos de las palabras de código, puede obtenerse de $T(D, N, J)$ haciendo $N = J = 1$. Esta forma más corta está dada por

$$T_1(D) = \sum_{d=d_{free}}^{\infty} a_d D^d$$

y se utilizará más tarde para obtener una cota sobre la probabilidad de error de códigos convolucionales.

Ejemplo. Para el código de la figura 8.23 se tiene que

$$\begin{aligned} T(D, N, J) &= \left. \frac{D^5 NJ^3}{1 - DNJ - DNJ^2} \right|_{N=J=1} = \frac{D^5}{1 - 2D} \\ &= D^5 + 2D^6 + 4D^7 + \cdots = \sum_{i=0}^{\infty} 2^i D^{5+i} \end{aligned}$$

Códigos catastróficos. Un código convolucional mapea una secuencia larga de bits de entrada en una palabra de código a ser transmitida por el canal. El propósito de la codificación es proveer altos niveles de protección contra el ruido de canal. Obviamente, un código que mapea secuencias de información que no están apartadas en palabras de código no es un buen código, porque esas dos palabras de código pueden ser confundidas facilmente y el resultado será un mayor número de errores de bit en el flujo de información. Un caso límite de esta propiedad indeseable sucede cuando dos secuencias de información que son diferentes en un número infinito de posiciones se mapean en palabras de código que difieren solo en un número finito de posiciones. En tal caso, dado que las palabras de código difieren en un número finito de bits, existe siempre la probabilidad que ellos sean decodificados erroneamente, y esto a su vez resulte en un número infinito

de errores al detectar la secuencia de información de entrada. Los códigos que exiben esta propiedad se denominan *códigos catastróficos* y deberían ser evitados en la práctica.

Como ejemplo de un código catastrófico, es posible considerar el código $(2, 1)$ descripto por

$$\boldsymbol{g}_1 = [1\ 1\ 0] \quad \boldsymbol{g}_2 = [0\ 1\ 1]$$

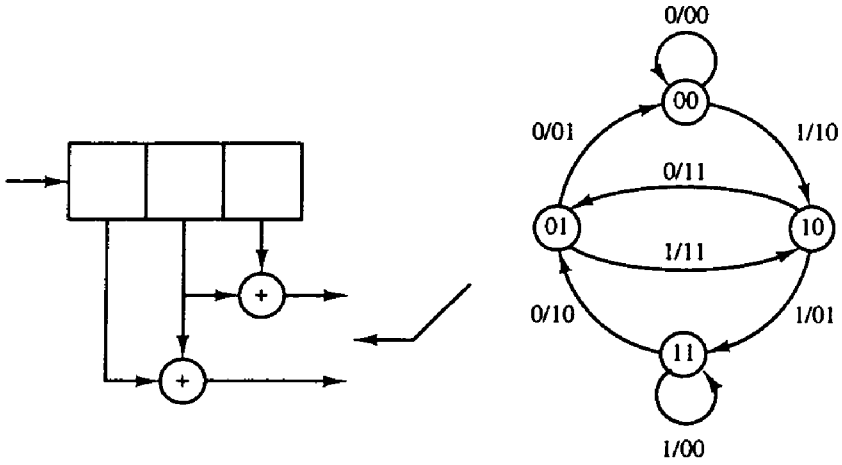

Figura 8.28: Codificador y diagrama de transición de estados para un código catastrófico.

El codificador y el diagrama de transición de estados para este código se muestran en la figura 8.28. Como puede verse en el diagrama, existe un lazo en el estado "11" que corresponde a la entrada "1" al codificador y las salidas correspondientes consisten en todo ceros. En consecuencia, si la secuencia de información de entrada consiste en todos unos, la salida correspondiente será

$$\boldsymbol{c} = (1001000 \cdots 0001001)$$

Si comparamos esta palabra de código con la palabra de código correspondiente a la secuencia de información todo ceros

$$\boldsymbol{c}_0 = (0000000 \cdots 0000000)$$

es posible observar que, a pesar que las secuencias de información son diferentes en un gran número de posiciones, las secuencias de salida correspondientes están muy próximas (la distancia de Hamming es solo 4) y, en consecuencia, para un gran número de transmisiones, esos bits pueden estar en error muy facilmente. La existencia de tales lazos correspondientes a k entradas, que no son todo ceros y para los cuales las n salidas son todos ceros, muestra que un código es catastrófico y debería, en consecuencia, evitarse.

8.6.2 Decodificación óptima de la codificación convolucional: El algoritmo de Viterbi

En la discusión de los diferentes esquemas de decodificación para códigos de bloques, se vió que existe la posibilidad de realizar decodificación con decisión suave o decisión fuerte. En la decodificación con decisión suave, el vector de las salidas de los filtros acoplados $\boldsymbol{r}$ se compara con los diferentes puntos de la constelación del sistema de modulación codificado y se elige aquel más próximo a él en distancia Euclidiana. En el caso de decisión fuerte $\boldsymbol{r}$ se convierte primero en una secuencia binaria $\boldsymbol{y}$ se hacen decisiones sobre las componentes individuales de $\boldsymbol{r}$, y luego se elige la palabra de código que está más próxima de $\boldsymbol{y}$ en el sentido de la distancia de Hamming. Puede verse que en las dos soluciones una tarea fundamental es *hallar un camino a través de la trellis que esté a una distancia mínima de la secuencia dada.* Esos problemas fundamentales aparecen en varias areas de las comunicaciones y otras disciplinas de la ingeniería eléctrica. Particularmente,

el mismo problema se encuentra en la estimación de secuencias de máxima verosimilitud (MLSD) cuando se transmite sobre canales de ancho de banda limitado con ISI (Capítulo 6), demodulación de esquemas de CPM (Capítulo 7), reconocimiento de voz, algunos esquemas de clasificación de patrones, etc. Todos esos problemas son esencialmente el mismo y pueden denominarse como *algoritmos de búsqueda óptima sobre trellis*. El conocido algoritmo de Viterbi, descripto previamente, provee una solución satisfactória a todos esos problemas.

En la decodificación con decisión fuerte de un código convolucional, se desea elegir el camino a través de la trellis cuya palabra de código, $\boldsymbol{c}$, está a distancia de Hamming mínima de la secuencia recibida cuantizada $\boldsymbol{y}$. En la decodificación con decisión fuerte el canal es binario y sin memoria (como el canal no tiene memoria se concluye que el ruido del canal es blanco). Dado que el camino deseado comienza del estado todo ceros y vuelve al estado todo ceros, se supondrá que este camino abarca un total de m ramas, y dado que cada rama corresponde a n bits de la salida del codificador, el número total de bits en $\boldsymbol{c}$ y $\boldsymbol{y}$ es mn. Las secuencias de bits correspondientes a la i-ésima rama serán $\boldsymbol{c}_i$ y $\boldsymbol{y}_i$, respectivamente, donde $1 \leq i \leq m$ y cada $\boldsymbol{c}_i$ y $\boldsymbol{y}_i$ es de longitud n. La distancia de Hamming entre $\boldsymbol{c}$ y $\boldsymbol{y}$ es en consecuencia

$$d(\boldsymbol{c}, \boldsymbol{y}) = \sum_{i=1}^{m} d(\boldsymbol{c}_i, \boldsymbol{y}_i) \tag{8.8}$$

En la decodificación de decisión suave se tiene una situación similar con tres diferencias,

1 En lugar de $\boldsymbol{y}$ se trabaja directamente con el vector $\boldsymbol{r}$, el vector de salida del demodulador digital (ya sea del tipo filtro acoplado o correlador).

2 En lugar de la secuencia binaria $\boldsymbol{c}$ se trabaja con la secuencia correspondiente $\boldsymbol{c}'$ con (para señales antipodales)

$$c'_{ij} = \begin{cases} \sqrt{\mathcal{E}} & si\ c_{ij} = 1 \\ -\sqrt{\mathcal{E}} & si\ c_{ij} = 0 \end{cases} \quad \text{para } 1 \leq i \leq m \text{ y } 1 \leq j \leq n.$$

3 En lugar de la distancia de Hamming se utiliza la distancia Euclidiana. Esta es una consecuencia de que el canal bajo estudio es AWGN.

De lo anterior se tiene que

$$d_E^2(\boldsymbol{c}', \boldsymbol{r}) = \sum_{i=1}^{m} d_E^2(\boldsymbol{c}'_i, \boldsymbol{r}_i) \tag{8.9}$$

De (8.8) y (8.9) puede verse que la forma genérica del problema que es necesario resolver es: dado un vector $\boldsymbol{a}$ hallar un camino a través de la trellis, que comienza en el estado todo ceros y termina en el estado todo ceros, tal que se minimize alguna medida de distancia entre $\boldsymbol{a}$ y una secuencia $\boldsymbol{b}$ correspondiente al camino deseado. El aspecto importante que hace a este problema fácil de resolver es que la distancia entre $\boldsymbol{a}$ y $\boldsymbol{b}$, en ambos casos de interés, puede escribirse como la suma de distancias correspondientes a las ramas individuales del camino.

Es posible suponer ahora que se trabaja con un código convolucional con $k = 1$. Esto significa que existen solo dos ramas entrando a cada estado de la trellis. Si el camino óptimo en un cierto punto pasa a través del estado S, existen dos caminos que conectan los estados previos S_1 y S_2 a ese estado (ver la figura 8.29). Si se desea ver cual de esas dos ramas es un buen candidato para minimizar la distancia total, es necesario sumar las métricas totales (mínimas) en los estados S_1 y S_2 a las métricas de las ramas que conectan esos dos estados al estado S. Luego, obviamente, la rama que tiene la acumulación de métrica mínima hasta el estado S es candidata a ser considerada para todos los estados posteriores a S. Esta rama se denomina *superviviente* en el estado S y la otra rama simplemente no es una candidata adecuada y se descarta. Después de determinar el superviviente en el estado S, se guarda también la métrica hasta ese estado y se pasa al estado siguiente. Este procedimiento continúa hasta alcanzar el estado todo cero al final de la trellis. Para casos donde $k > 1$, la única diferencia es que en cada etapa es necesario elegir un camino superviviente entre 2^k ramas que conducen al estado S.

El procedimiento anterior puede resumirse en el siguiente algoritmo conocido como *algoritmo de Viterbi*,

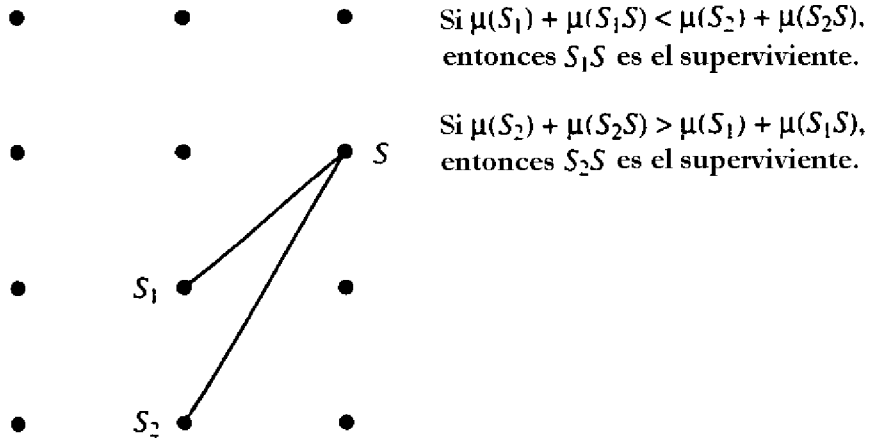

Figura 8.29: El algoritmo de Viterbi.

1 Dividir la secuencia recibida en m secuencias cada una de longitud n.

2 Dibujar la trellis de profundidad m para el código bajo estudio. Para las últimas $L-1$ etapas de la trellis dibujar solo los caminos correspondientes a las secuencias de entrada todo ceros (esto se hace porque se conoce que la secuencia de entrada ha sido rellenada con $k(L-1)$ ceros).

3 Hacer $l = 1$ y la métrica del estado todo ceros inicial igual a cero.

4 Hallar la distancia de la subsecuencia l-ésima de la secuencia recibida a todas las ramas conectando la etapa l-ésima a la etapa $(l+1)$-ésima de la trellis.

5 Sumar esas distancias a las métricas de los estados de la l-ésima etapa para obtener las métricas candidatas para los estados de la $(l+1)$-ésima etapa. Para cada estado de la $(l+1)$-ésima etapa exiten 2^k métricas candidatas cada una correspondiente a una rama terminando en ese estado.

6 Para cada estado en la estapa $(l+1)$-ésima, elegir la menor métrica candidata e identificar la rama correspondiente a este valor mínimo como la *superviviente*, y asignar el mínimo de las métricas candidatas como las métricas de los estados de la $(l+1)$-ésima etapa.

7 Si $l = m$, ir al próximo paso, sino incrementar l en 1 e ir al paso 4.

8 Comenzando con el estado todo ceros en la $(m+1)$-ésima etapa, ir hacia atrás a través de la trellis siguiendo las ramas supervivientes hastas alcanzar el estado todo ceros inicial. Este camino es el camino óptimo y la secuencia de bits de entrada correspondiente a él es la secuencia de información decodificada en sentido de máxima verosimilitud. Para obtener una mejor selección de la secuencia de bits de entrada, remover los últimos $k(L-1)$ ceros de esta secuencia.

Como puede verse del algoritmo, el retardo de decodificación y la cantidad de memoria requerida para una secuencia de información larga es inaceptable. La decodificación no puede comenzar hasta que la secuencia completa (que en caso de codificadores convolucionales puede ser muy larga) sea recibida, y el total de los caminos supervivientes sea almacenado. En la práctica se desea una solución subóptima que no cause esos problemas. Una aproximación posible, denominada *truncamiento de la memoria de caminos*, es que el decodificador en cada etapa busque solo δ etapas hacia atrás en la trellis y no hasta el comienzo de la trellis. Con esta aproximación en la $(\delta+1)$-ésima etapa, el decodificador hace una decisión sobre los bits de entrada correspondientes a la primera etapa de la trellis (los primeros k bits) y los futuros bits recibidos no cambian esta decisión. Esto significa que el retardo de decodificación será de $k\delta$ bits y solo se requiere mantener los caminos supervivientes correspondientes a las últimas δ etapas. A través de simulaciones computacionales es posible verificar que, si $\delta \geq 5L$, la degradación del desempeño debido al truncamiento de la memoria de los caminos es despreciable.

Ejemplo. Es posible suponer que en decodificación con decisión fuerte la secuencia cuantizada recibida es

$$\boldsymbol{y} = (01101111010001)$$

Este código convolucional se muestra en la figura 8.23. Hallar la secuencia de información de máxima verosimilitud y el número de errores.

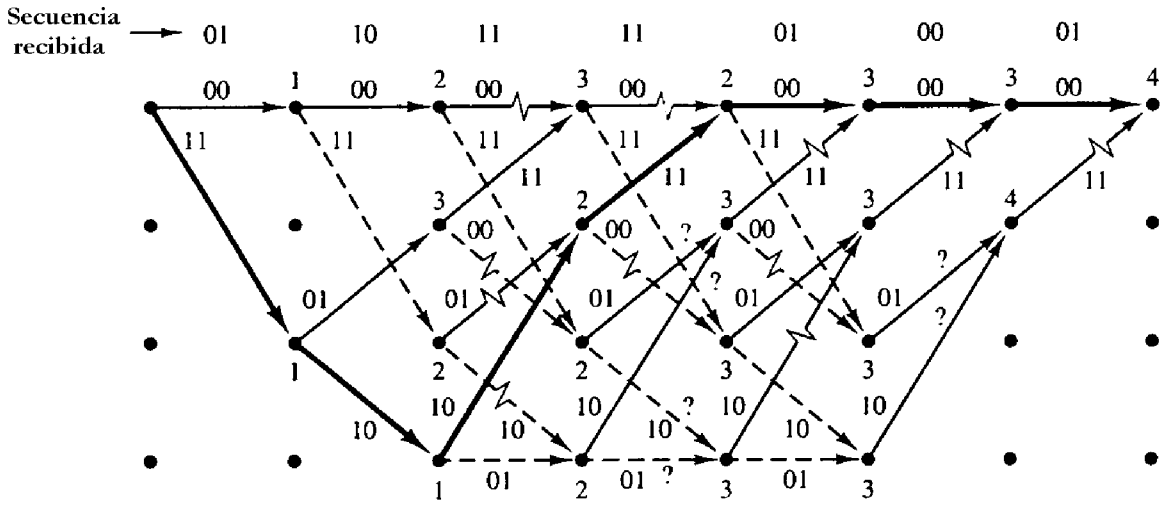

Figura 8.30: Diagrama trellis para la decodificación de Viterbi de la secuencia (01101111010001).

Se trata de un código $(2, 1)$ con $L = 3$. La longitud de la secuencia recibida $\boldsymbol{y}$ es 14. Esto significa que $m = 7$ y es necesario dibujar una trellis de profundidad 7. Notar también que debido a que la secuencia de información está rellenada con $k(L-1)$ ceros para las dos etapas finales de la trellis, se dibujan solamente las ramas correspondientes a las entradas todo cero. Esto significa también que la longitud real de la secuencia de entrada es 5, la cual, después de rellenada con dos ceros se ha incrementado a 7. La trellis para este caso se muestra en la figura 8.30.

La secuencia recibida dividida $\boldsymbol{y}$ también se muestra en esta figura. Notar que al dibujar las dos últimas etapas de la trellis se ha considerado solamente las entrada cero al codificador (notar que en las etapas finales no existen líneas a trazos correspondientes a entradas 1). Ahora la métrica del estado inicial todo cero se inicializa y se calculan las métricas de la próxima etapa.

En este paso hay una sola rama entrante a cada estado, en consecuencia no hay comparación, y las métricas (que son las distancias de Hamming entre esa parte de la secuencia recibida y las ramas de la trellis) se suman a la métrica de la etapa previa.

En la próxima etapa no existe comparación tampoco.

En la tercera etapa, por primera vez se tienen dos ramas entrando a cada estado. Esto significa que es necesario hacer una comparación y deben elegirse las supervivientes. De las dos ramas que entran en cada estado, la métrica que corresponde al menor total acumulado permanece como una superviviente y las otras ramas se descartan.

Si en cualquier etapa resultan dos caminos con la misma métrica, cada una de ellas puede ser una superviviente. Tales casos se han marcado con "?" en el diagrama trellis.

El procedimiento continúa hasta el estado todo ceros final de la trellis y luego comenzando desde ese estado es necesario moverse a lo largo de los caminos supervivientes hacia el estado todo ceros inicial. Este camino, la línea gruesa en el diagrama de la trellis, es el camino óptimo. La secuencia de bits de entrada correspondiente a este camino es 1100000, donde los últimos dos ceros no son bits de información sino que fueron adicionados para retornar al codificador al estado todo ceros. En consecuencia, la secuencia de información es 11000. La palabra de código correspondiente para el camino seleccionado es 11101011000000, la cual está a una distancia de Hamming de 4 de la secuencia recibida. Todos los otros caminos a través de la trellis corresponden a palabras de código que están a mayor distancia de Hamming de la secuencia recibida.

Para el caso de la decodificación con decisión suave el procedimiento es similar pero con las distancias de Hamming sustituidas por distancias Euclidianas al cuadrado.

Otros algoritmos para decodificación. El algoritmo de Viterbi provee decodificación de máxima verosimilitud para códigos convolucionales. Sin embargo, como se ha discutido, la complejidad del algoritmo es proporcional al número de estados de la trellis. Esto significa que la complejidad del algoritmo aumenta exponencialmente con la longitud restringida de los códigos convolucionales. En consecuencia, el algoritmo de Viterbi puede aplicarse solo sobre códigos con bajas longitudes restringidas, usualmente 10 o menor. Para longitud restringidas mayores se han propuesto otros esquemas de decodificación subóptimos ,tales como: la decodificación secuencial de Wozencraft (1957), el algoritmo de Fano (1963), el algoritmo de pila (Zigangirov, 1966 y Jelinek, 1969), el algoritmo de decodificación con realimentación (heller, 1975) y la decodificación con lógica de mayoría (Massey, 1963).

8.6.3 Cotas sobre la probabilidad de error de códigos convolucionales

La búsqueda de cotas sobre la probabilidad de error de los códigos convolucionales es diferente de la realizada en el caso de códigos de bloques, debido a que se trabaja con secuencias de longitud muy grande y, como la distancia libre de esos códigos es usualmente muy pequeña, ocurrirán eventualmente algunos errores. El número de errores es una variable aleatória que depende de las características del canal (SNR en la decodificación con decisión suave y probabilidad de cruzamiento en la decodificación con detección fuerte) y la longitud de la secuencia de entrada. Cuanto más larga sea la secuencia de entrada mayor será la probabilidad de errores. En consecuencia, tiene sentido normalizar el número de errores de bit en relación a la longitud de la secuencia de entrada. Una medida que se adopta usualmente para comparar el desempeño de códigos convolucionales es el número esperado de bits recibidos en error por bit de entrada.

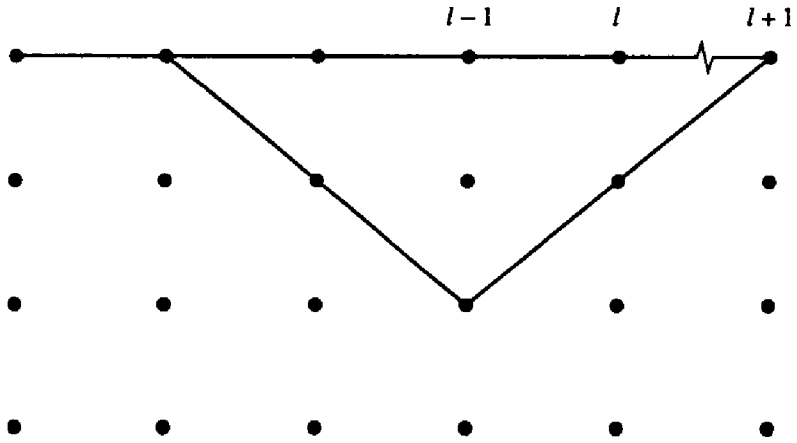

Figura 8.31: Camino correspondiente al primer evento de error.

Para hallar una cota sobre el número promedio de bits en error por cada bit de entrada es necesario obtener primero una cota sobre el número promedio de bits en error por cada secuencia de entrada de longitud k. Para determinar esto, se asumirá que se transmite la secuencia todo ceros y, hasta la etapa l de la decodificación, no hubo errores. Ahora, k bits de información entran al codificador y resultan en el desplazamiento a una nueva etapa en la trellis. El interés apunta a encontrar una cota sobre el número esperado de errores que pueden ocurrir debido a este bloque de entrada de longitud k. Como se asume que hasta la etapa l no han existido errores, entonces hasta esa etapa el camino todo ceros a través de la trellis tiene la métrica mínima. Luego, moviendose a la próxima etapa $(l+1)$, es posible que otro camino a través de la trellis tenga una métrica menor que la del camino todo ceros y en consecuencia cause errores. Si esto sucede, se debe tener un camino a través de la trellis que converge con el camino todo ceros, por primera vez en la $(l+1)$-ésima etapa y que tiene una métrica menor que el camino todo ceros. Tal evento se denomina el *primer evento de error* y la probabilidad correspondiente se denomina *probabilidad del primer evento de error*. Esta situación se muestra en la figura 8.31.

El primer paso sería acotar la probabilidad del primer evento de error. Notaremos por $P_2(d)$ la probabilidad que tiene un camino a través de la trellis, que está a una distancia de Hamming d del camino todo ceros, de ser un superviviente en la etapa $(l+1)$. Como $d > d_{free}$, es posible acotar la probabilidad del primer evento de error por

$$P_e \leq \sum_{d=d_{free}}^{\infty} a_d P_2(d)$$

donde en el lado derecho se han incluído todos los caminos a través de la trellis que abarcan, junto con el camino todo ceros, hasta la $(l+1)$-ésima etapa. El valor de $P_2(d)$ depende del empleo de decodificación suave o decodificación fuerte.

Para el caso de decodificación suave, y si se utiliza señalización antipodal (PSK), se tiene que

$$P_2(d) = Q\left[\frac{d^E}{\sqrt{2N_0}}\right] = Q\left[\sqrt{\frac{2\mathcal{E}_d}{N_0}}\right] = Q\left[\sqrt{2R_c d\frac{\mathcal{E}_b}{N_0}}\right]$$

y en consecuencia

$$P_e \leq \sum_{d=d_{free}}^{\infty} a_d Q\left[\sqrt{2R_c d\frac{\mathcal{E}_b}{N_0}}\right]$$

Usando la cota superior sobre la función Q se tiene que

$$Q\left[\sqrt{2R_c d\frac{\mathcal{E}_b}{N_0}}\right] \leq \frac{1}{2}e^{-R_c d\mathcal{E}_b/N_0}$$

Teniendo en cuenta ahora que $e^{-R_c d\mathcal{E}_b/N_0} = D^d\big|_{D=e^{-R_c\mathcal{E}_b/N_0}}$, finalmente se obtiene que

$$P_e \leq \frac{1}{2}\sum_{d=d_{free}}^{\infty} a_d D^d\Big|_{D=e^{-R_c\mathcal{E}_b/N_0}} = \frac{1}{2}T_1(D)|_{D=e^{-R_c\mathcal{E}_b/N_0}}$$

Esta es una cota sobre la probabilidad del primer evento de error. Para encontrar una cota sobre el número promedio de bits en error para k bits de entrada, $\overline{P}_b(k)$, es posible notar que cada camino a través de la trellis hace que un cierto número de bits de entrada se decodifiquen incorrectamente. Para un término general de la forma $D^d N^{f(d)} J^{g(d)}$ en la expansión de $T(D,N,J)$, existe un total de $f(d)$ bits de entrada no cero. Esto significa que el número promedio de bits de entrada en error puede obtenerse multiplicando la probabilidad de elegir cada camino por el número total de errores de entrada que podrían resultar si ese camino fuese elegido. De esta forma, el número promedio de bits en error, en el caso de decodificación con decisión suave, puede acotarse mediante

$$\begin{aligned}\overline{P}_b(k) &\leq \sum_{d=d_{free}}^{\infty} f(d)P_2(d) = \sum_{d=d_{free}}^{\infty} a_d f(d) Q\left[\sqrt{2R_c d\frac{\mathcal{E}_b}{N_0}}\right] \\ &\leq \frac{1}{2}\sum_{d=d_{free}}^{\infty} a_d f(d) e^{-R_c d\mathcal{E}_b/N_0} \qquad (8.10)\end{aligned}$$

Si se define

$$T_2(D,N) = T(D,N,J)|_{J=1} = \sum_{d=d_{free}}^{\infty} a_d D^d N^{f(d)}$$

se tiene que

$$\frac{\partial T_2(D,N)}{\partial N} = \sum_{d=d_{free}}^{\infty} a_d f(d) D^d \tag{8.11}$$

Luego, usando (8.10) y (8.11) se obtiene

$$\overline{P}_b(k) \leq \frac{1}{2} \left. \frac{\partial T_2(D,N)}{\partial N} \right|_{N=1, D=e^{-R_c \mathcal{E}_b/N_0}}$$

Para obtener el número promedio de bits en error por cada bit de entrada, es necesario dividir esta cantidad por k. De esta forma, el resultado final será

$$\overline{P}_b \leq \frac{1}{2k} \left. \frac{\partial T_2(D,N)}{\partial N} \right|_{N=1, D=e^{-R_c \mathcal{E}_b/N_0}}$$

Para el caso de decodificación con decisión fuerte, el procedimiento básico sigue los pasos anteriores. La única diferencia es la cota sobre $P_2(d)$. Puede demostrarse que en este caso $P_2(d)$ puede acotarse mediante

$$P_2(d) \leq [4p(1-p)]^{d/2}$$

Usando este resultado es directo mostrar que para el caso de decodificación con decisión fuerte, la probabilidad de error tiene una cota superior dada por

$$\overline{P}_b \leq \frac{1}{k} \left. \frac{\partial T_2(D,N)}{\partial N} \right|_{N=1, D=\sqrt{4p(1-p)}}$$

Una comparación de las decodificaciones con decisión fuerte y suave para códigos convolucionales muestra que, como en el caso de los códigos lineales de bloques, el segundo tipo supera al primero por un márgen de 2 dB para canales AWGN.

8.6.4 Códigos convolucionales con buenas propiedades de distancia

Del análisis llevado a cabo es obvio que d_{free} juega un papel principal en el desempeño de los códigos convolucionales. Para n y k especificados, la distancia libre de un código convolucional depende de la longitud restringida del código. La búsqueda de códigos convolucionales con buenas propiedades de distancia ha sido llevada a cabo extensivamente en la literatura. Las tablas 8.4 y 8.5 resúmen los resultados de simulaciones computacionales llevadas a cabo para códigos convolucionales de tazas 1/2 y 1/3. En esas tablas, para cada longitud restringida, se tabula el código convolucional que logra la distancia libre más grande. Para este código, los generadores $\boldsymbol{g}_i$ se escriben en forma octal. También se incluye en esas tablas la distancia libre resultante.

8.7 Codificación para canales con restricciones en ancho de banda (*)

En las dos clases principales de códigos estudiados hasta aquí, o sea códigos convolucionales y de bloques, la mejora en el desempeño del sistema de comunicaciones se logra mediante la expansión del ancho de banda. En ambos casos la distancia Euclidiana entre las señales codificadas transmitidas se incrementa mediante el uso de codificación, y al mismo tiempo el ancho de banda se incrementa en un factor $n/k = 1/R_c$. Esos tipos de códigos tienen gran aplicación en casos donde existe suficiente ancho de banda y el diseñador no tiene restricciones muy fuertes. Ejemplos de tales casos son los sistemas de comunicaciones espaciales y el almacenamiento en medios de alta densidad. Sin embargo, en muchas aplicaciones prácticas se trabaja con canales de comunicaciones con restricciones de ancho de banda estrictas, y la expansión de ancho de banda

Longitud Restringida L	Generadores	(base octal)	d_{free}
3	5	7	5
4	15	17	6
5	23	35	7
6	53	75	8
7	133	171	10
8	247	371	10
9	561	753	12
10	1167	1545	12
11	2335	3661	14
12	4335	5723	15
13	10533	17661	16
14	21675	27123	16

Tabla 8.4: Códigos de distancia libre máxima, taza 1/2.

Longitud Restringida L	Generadores	(base	octal)	d_{free}
3	5	7	7	8
4	13	15	17	10
5	25	33	37	12
6	47	53	75	13
7	133	145	175	15
8	225	331	367	16
9	557	663	711	18
10	1117	1365	1633	20
11	2353	2671	3175	22
12	4767	5723	6265	24
13	10533	10675	17661	24
14	21645	35661	37133	26

Tabla 8.5: Códigos de distancia libre máxima, taza 1/3.

debida a la codificación puede no ser aceptable. Por ejemplo, en la transmisión de datos digitales sobre canales telefónicos (diseño de modems) se trabaja con un canal que tiene un ancho de banda restringido y la sobrecarga debida a la codificación impone una restricción fundamental en la velocidad de transmisión. En esta sección se discutirá un esquema de modulación y codificación conjunto denominado *modulación codificada en trellis* (TCM) que es particularmente útil para canales con restricciones de ancho de banda.

8.7.1 Combinación de codificación y modulación (*)

El uso de códigos convolucionales o de bloques introduce redundancia que a su vez causa un incremento en la distancia Euclidiana entre señales codificadas. Por otro lado, si se utiliza modulación PSK, la dimensionalidad de la señal transmitida se incrementará de k dimensiones por transmisión a n dimensiones por transmisión. Este incremento en dimensionalidad resulta en un incremento en ancho de banda dado que la dimensionalidad y el ancho de banda son proporcionales. Si se desea obtener los beneficios de la codificación y al mismo tiempo no se desea incrementar el ancho de banda, es necesario utilizar un esquema de modulación diferente de PSK, o sea, un esquema que sea más eficiente en términos de ancho de banda. Esto implíca utilizar un esquema de modulación multinivel/multifase. Por supuesto, utilizar un esquema de modulación multinivel/multifase resulta en una constelación más "poblada" y, con un nivel de potencia constante, una disminución de la distancia mínima Euclidiana en la constelación. Esto tiene ciertamente un efecto negativo sobre el desempeño de error del esquema de modulación/codificación como un todo. Pero, como se verá más adelante, esta reducción de la distancia Euclidiana mínima en la constelación puede ser compensada mediante el incremento en la distancia de Hamming debido a codificación tal que el desempeño total resulta notablemente mejorado.

Como ejemplo es posible suponer que en la etapa de codificación se desea utilizar una taza de 2/3. Si la velocidad de la fuente es R bits/seg, el número de símbolos/seg de salidas binarias codificadas será $\frac{3}{2}R$. Si se desea utilizar una constelación tal que el requerimiento de ancho de banda sea igual al requerimiento de ancho de banda de la señal no codificada (que no exista expansión de ancho de banda), se debe asignar m dimensiones para cada símbolo de salida binaria tal que el número de dimensiones/seg sea igual al número de dimensiones/seg de los datos no codificados, o sea R. En consecuencia, se debe tener

$$R = \frac{3}{2}Rm$$

y en consecuencia

$$m = \frac{2}{3} \text{ dimensión / símbolo binario}$$

Esto significa que la constelación se deberá diseñar de forma que tenga dos dimensiones por cada tres símbolos binarios. Pero tres símbolos binarios son equivalentes a ocho puntos en la constelación, y en consecuencia la conclusión es que es posible alcanzar el objetivo con una constelación de ocho puntos en un espacio bidimensional. Una constelación de ese tipo es por supuesto la correspondiente al esquema de modulación PSK octal. En consecuencia, la conclusión general es que si se utiliza una taza de 2/3 en conjunto con un esquema de modulación PSK octal no existirá expansión de ancho de banda.

Es posible examinar ahora cuanta ganancia de codificación es posible obtener con ese esquema. Suponiendo que la potencia disponible es P, sin codificación se tiene que

$$\mathcal{E}_b = \frac{P}{R}$$

y en consecuencia la distancia Euclidiana mínima entre dos secuencias es

$$d^2 = \frac{4P}{R}$$

Si se mapean dos bits de información en un punto de un constelación PSK octal, la energía de este punto es

$$\mathcal{E}_s = \frac{2P}{R}$$

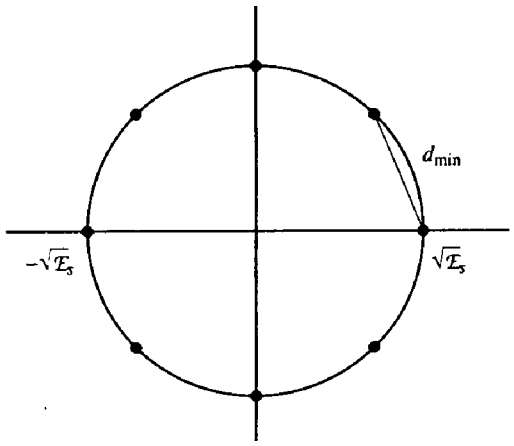

Figura 8.32: Constelación PSK $M = 8$ utilizada para codificación eficiente en ancho de banda.

De esto es posible obtener una expresión para la distancia Euclidiana mínima en la constelación (ver la figura 8.32) como

$$d_{min}^2 = 4\frac{2P}{R}\sin^2\frac{\pi}{8} = 2(2-\sqrt{2})\frac{P}{R}$$

Obviamente la distancia Euclidiana mínima se ha disminuido. Para ver este efecto es posible obtener la pérdida debido al uso de esta constelación.

$$\left(\frac{d_{min}^2}{d^2}\right)^{-1} = \frac{2}{2-\sqrt{2}} = 2+\sqrt{2} = 3.4142 \cong 5.33\,\text{dB}$$

Esta pérdida debe compensarse mediante el código. Obviamente, el código de taza 2/3 empleado aqui no solo debe compensar esta pérdida sino que también debe proveer una ganancia adicional para justificar su uso en el esquema completo de modulación-codificación. Es posible utilizar cualquier código convolucional o de bloques que provea la distancia mínima requerida para lograr cierta ganancia de codificación total. Por ejemplo, si se requiere una ganancia de codificación total de 3 dB, el código provee una ganancia una ganancia de 8.33 dB para compensar la pérdida de 5.33 dB debido a la modulación. Un código que provea esa alta ganancia de codificación es ciertamente un código muy complejo (restricción de longitud grande) que requiere codificadores y decodificadores sofisticados. Sin embargo, mediante la interpretación de la modulación y la codificación como una entidad única, como discutido en la próxima sección, será posible ver que puede lograrse un desempeño comparable utilizando un esquema de codificación mucho más simple.

8.7.2 Modulación codificada en trellis (TCM) (*)

La modulación codificada en trellis (TCM), es un método simple para diseñar esquemas de modulación codificados que puedan lograr un buen desempeño general. Este esquema de modulación - codificación se basa en el concepto de *mapeo por particionamiento de conjuntos* desarrollado por Ungerboeck (1982). El mapeo por particionamiento de conjuntos puede utilizarse conjuntamente con ambos tipos de códigos, convolucionales y de bloques, pero debido a la existencia de un algoritmo de decodificación con desición suave simple y óptimo (el algoritmo de Viterbi), se ha empleado mayoritariamente con códigos convolucionales. Cuando se utiliza con códigos convolucionales, el esquema de modulación resultante se conoce como *modulación codificada en trellis.*

Conceptos del particionamiento de conjuntos. El punto clave en el particionamiento de una constelación es hallar subconjuntos de la constelación que son similares y los puntos dentro de cada partición que estén máximamente separados. Comenzando con la constelación original, se la particiona en dos subconjuntos que son congruentes (complementarios) y los puntos dentro de cada partición está maximamente

separados. Luego se aplica el mismo principio a cada partición y se continúa. El punto en el cual se para el particionamiento depende del código que se utilize, lo cual se discutirá más adelante.

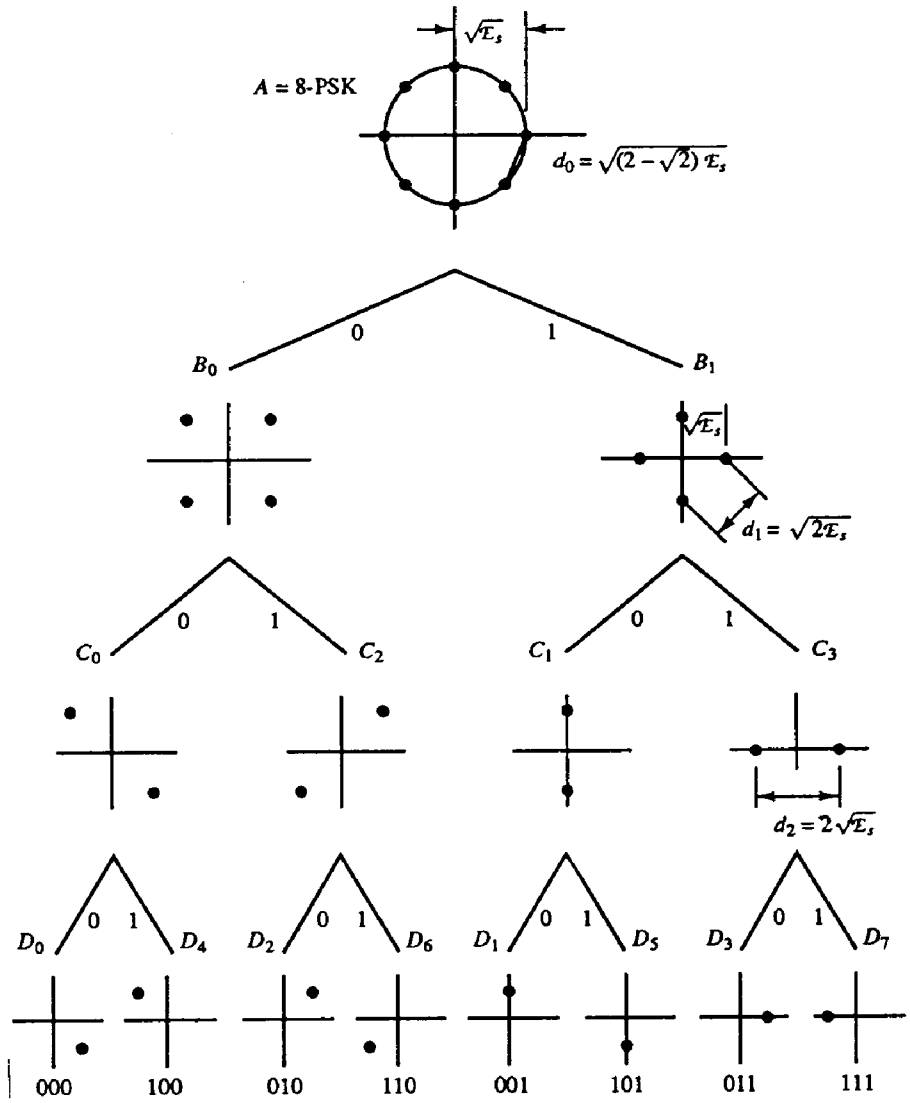

Figura 8.33: Particionamiento de conjuntos para una constelación PSK octal.

Un ejemplo de particionamiento de conjuntos se muestra en la figura 8.33. Se comienza con una constelación PSK octal con puntos de señal sobre un círculo de radio $\mathcal{E}_s$. La distancia mínima en la constelación será

$$d_0 = \sqrt{(2-\sqrt{2})\mathcal{E}_s}$$

Esta constelación se particiona en dos, B_0 y B_1. Es posible notar que B_0 y B_1 son complementarias. Existen varias formas en las que la constelación original PSK octal puede particionarse en dos subconjuntos complementarios, pero B_0 y B_1 proveen la partición de distancia máxima. Esta distancia es facilmente deducible como

$$d_1 = \sqrt{2\mathcal{E}_s}$$

Es posible continuar el proceso y particionar B_0 y B_1 para obtener C_0. C_1, C_2 y C_3. La partición tiene una distancia interna que ha disminuido a

$$d_2 = 2\sqrt{\mathcal{E}_s}$$

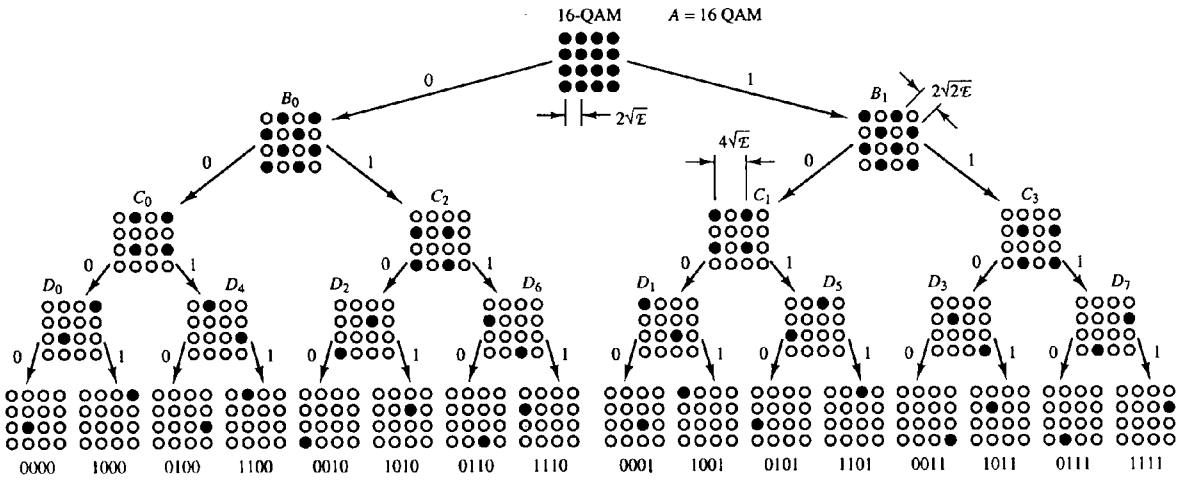

Figura 8.34: Particionamiento de conjuntos para una constelación PSK $M = 16$.

Es posible ir un paso más adelante para obtener ocho particiones, cada una conteniendo un único punto. Los subconjuntos correspondientes son D_0 a D_7. Otro ejemplo del particionamiento de conjuntos aplicado a una constelación QAM se muestra en la figura 8.34. El lector puede verificar que este particionamiento sigue las reglas generales del particionamiento de conjuntos descripto anteriormente.

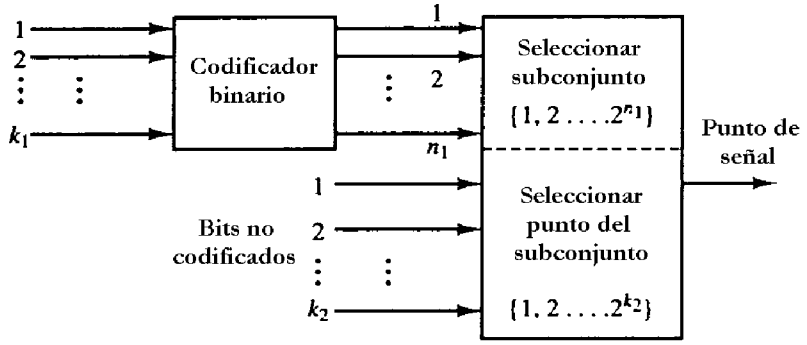

Figura 8.35: Diagrama en bloques de un sistema de modulación codificada.

Modulación codificada. El diagrama de bloques de un esquema de modulación codificada se muestra en la figura 8.35. Un bloque de k bits de entrada se divide en dos subbloques de longitudes k_1 y k_2 respectivamente. Los primeros k_1 bits se aplican a un codificador binario (N_1, k_1). La salida del codificador consiste en n_1 bits. Esos bits se utilizan para elegir una de las 2^{n_1} particiones de la constelación. Esto significa que la constelación ha sido particionada en 2^{n_1} subconjuntos. Después que la constelación ha sido elegida, los k_2 bits remanentes se utilizan para elegir uno de los puntos de la constelación elegida. Esto significa que existen 2^{k_2} puntos en cada partición. En consecuencia, la partición que se utiliza contiene 2^{n_1} subconjuntos y cada subconjunto contiene 2^{k_2} puntos. Esto determina una regla para obtener el tamaño de la constelación requerido y cuantos pasos de particionamiento deben seguirse sobre la constelación.

Ungerboeck (1982) mostró que eligiendo $n_1 = k_1 + 1$ y $k_2 = 1$ y utilizando códigos convolucionales simples es posible diseñar esquemas de modulación codificados que logran una ganancia total de codificación entre 3

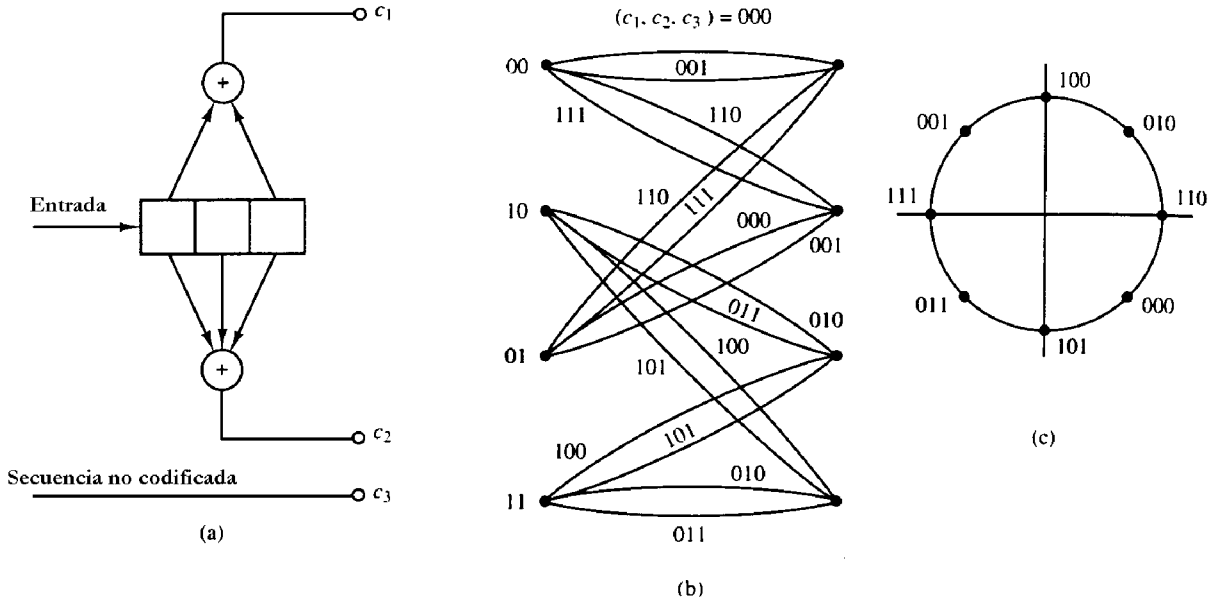

Figura 8.36: Esquema TCM simple: (a) codificador, (b) trellis de cuatro estados y (c) mapeo de los bits codificados (c_1, c_2, c_3) en los puntos de señal.

y 6 dB. Uno de tales esquemas se muestra en la figura 8.36. En este esquema de codificación $k_1 = 1$, $n_1 = 2$ y $k_2 = 1$. La constelación contiene $2^{n_1+k_2} = 8$ puntos, que están particionados en $2^{n_1} = 4$ subconjuntos cada uno de los cuales contiene $2^{k_2} = 2$ puntos. La constelación elegida aquí es PSK octal y se particiona como mostrado anteriormente (ver figura 8.33). El código convolucional empleado tiene una taza $\frac{k_1}{n_1} = \frac{1}{2}$. La restricción de longitud de este código es un parámetro de diseño y puede elegirse para proveer la ganancia de codificación deseada. Para restricciones de longitud mayores, obviamente, se obtiene mayores ganancias de codificación al precio de incrementar la complejidad del codificador - decodificador. En este ejemplo simple la restricción de longitud se eligió igual a 3. El diagrama trellis (de una etapa) para este código se muestra también en la figura 8.36.

El diagrama trellis mostrado en la figura 8.36 es el correspondiente a un código convolucional ordinario. La diferencia principal es que aquí se tienen dos caminos que conectan dos estados. La razón de esto es la existencia de un bit extra, $k_2 = 1$, que selecciona un punto en cada partición. En realidad, los dos caminos paralelos que conectan dos estados corresponden a una partición, y cualquier camino simple corresponde a un punto en la partición. Una pregunta final debe responderse: ¿cual es el mapeo óptimo entre la transición entre los estados de un código convolucional y las particiones?. A través de simulaciones computacionales extensivas y razonamiento heurístico es posible obtener las siguientes reglas.

1. Las transiciones en paralelo (cuando ocurren) corresponden a puntos de señal en una única partición en la etapa siguiente del particionamiento. En el ejemplo anterior, $C_0 = \{D_0, D_4\}$. $C_2 = \{D_2, D_6\}$, $C_1 = \{D_1, D_5\}$, y $C_3 = \{D_3, D_7\}$ corresponden a transiciones en paralelo. Esos puntos están separados por una distancia Euclidiana máxima de $d_2 = 2\sqrt{\mathcal{E}_s}$.

2. Las transiciones que se originan y vuelven a cualquier estado se asignan en la siguiente etapa del particionamiento que tiene una partición pariente simple en la etapa precedente. En el ejemplo anterior esas particiones serían $B_0 = \{C_0, C_2\}$ y $B_1 = \{C_1, C_3\}$. La distancia máxima en este caso es $d_1 = \sqrt{2\mathcal{E}_s}$.

3. Los puntos de señal deberían ocurrir con igual frecuencia.

Para ver como se desempeña el esquema de modulación codificada en trellis de la figura 8.36, es posible hallar la distancia Euclidiana mínima entre dos caminos que se originan en un nodo y se juntan en otro nodo. Esta distancia, conocida como *distancia Euclidiana libre*, D_{fed}, es una característica importante de un esquema TCM. Un candidato obvio para D_{fed} es la distancia Euclidiana entre dos transiciones en paralelo. La distancia Euclidiana entre dos transiciones en paralelo es $d_2 = 2\sqrt{\mathcal{E}_s}$. Otro camino candidato se muestra en la figura 8.37. Sin embargo, la distancia Euclidiana entre esos dos caminos es $d^2 = d_0^2 + 2d_1^2 = 4.58\mathcal{E}_s$.

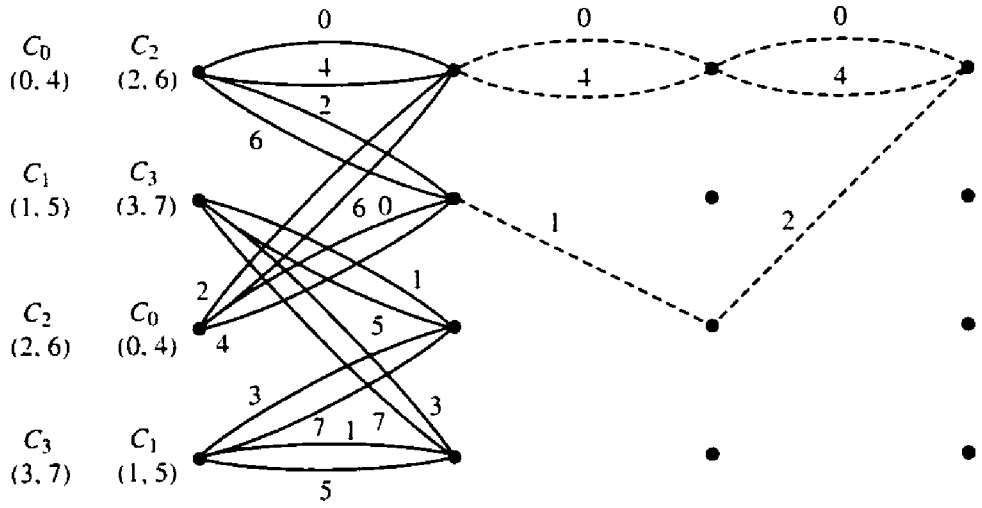

Figura 8.37: Dos caminos candidatos de distancia mínima.

obviamente, es mayor que la distancia entre las dos transiciones en paralelo. Es fácil verificar que para este código la distancia Euclidian libre es $D_{fed} = d_2 = 2\sqrt{\mathcal{E}_s}$. Para comparar este resultado con el esquema no codificado, es posible notar que en un esquema no codificado

$$d^2_{nocodif} = 4\mathcal{E}_b = 4\frac{P}{R}$$

y en un esquema codificado

$$d^2_{codif} = 4\mathcal{E}_s = 8\mathcal{E}_b$$

En consecuencia, la ganancia de codificación está dada por

$$G_{cod} = \frac{d^2_{codif}}{d^2_{nocodif}} = 2 \cong 3 \text{ dB}$$

De esta forma, este esquema simple de codificación es capaz de lograr una ganancia de codificación de 3 dB sin incrementar el ancho de banda. Obviamente, el precio pagado por este mejor desempeño es el incremento en la complejidad del par codificador - decodificador.

En lugar de una trellis de cuatro etapas, una trellis con un mayor número de etapas produce mayores ganancias de codificación. A través de simulaciones computacionales extensivas Ungerboeck indicó que con 8, 16, 32, 128 y 256 etapas, pueden lograrse ganancias de codificación en el rango de 3.6 dB a 5.75 dB. El codificador de una trellis de ocho estados se muestra en la figura 8.38.

Decodificación de códigos TCM. La decodificación de la modulación codificada en trellis se realiza en dos etapas. Dado que cada transición en la trellis corresponde a una partición del conjunto de señales y cada partición generalmente corresponde a un número de puntos de señal, el primer paso es hallar los puntos más problables en cada partición. Esto se logra hallando en cada partición el punto más próximo en distancia Euclidiana al punto recibido. Este primer paso en la decodificación de un esquema de modulacióm codificada en trellis se denomina *decodificación de subconjuntos*. Después de este paso, a cada transición en la trellis corresponde un único punto (el más probable) y una única distancia Euclidiana (la distancia entre el punto recibido y el punto más probable). El segundo paso del procedimiento de decodificación es utilizar

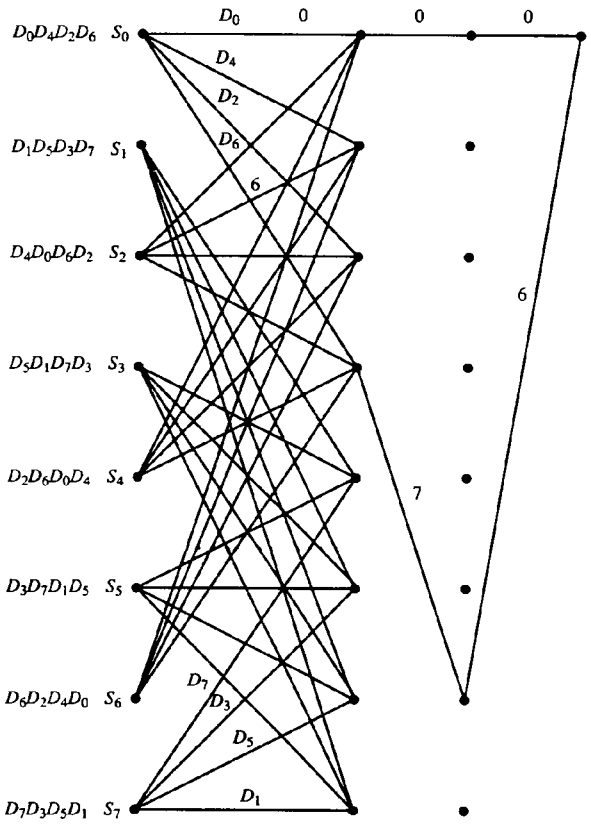

Figura 8.38: Codificador trellis de ocho estados para modulación codificada PSK octal.

esta distancia Euclidiana para hallar un camino a través de la trellis cuya distancia Euclidiana total de la secuencia recibida sea mínima. Esto se logra aplicando el algoritmo de Viterbi.

La modulación codificada en trellis se utiliza ampliamente en modems de alta velocidad. Sin codificación, los modems de alta velocidad logran velocidades hasta unos 9600 bits/seg con una constelación QAM con $M = 16$. La ganancia de codificación introducida por TCM ha hecho posible incrementar la velocidad de transmisión por lo menos en un factor de 2.

8.8 Aplicaciones prácticas de codificación (*)

En las secciones previas se vió que la codificación puede emplearse para mejorar la SNR efectiva y mejorar así el desempeño de un sistema de comunicaciones digitales. Los códigos de bloques y convolucionales o combinaciones en la forma de códigos concatenados como los discutidos anteriormente se han aplicado en el diseño de sistemas de comunicaciones donde el ancho de banda no es un problema esencial, o sea donde es permisible alguna expansión de ancho de banda. Por otro lado, en los casos donde el ancho de banda es un problema crucial, como en comunicaciones digitales sobre canales telefónicos, es posible emplear la modulación codificada. Mediante la utilización de codificación, el desempeño de sistemas prácticos de comunicaciones digitales se ha mejorado en hasta 9 dB, dependiendo de la aplicación y el tipo de código utilizado. En esta sección se discutirán aplicaciones de codificación en comunicaciones espaciales, modems para líneas telefónicas y reproductores de discos compactos (CD).

8.8.1 Codificación para aplicaciones espaciales (*)

Los canales de comunicaciones espaciales se caracterizan por SNR muy bajas y sin restricciones de ancho de banda. La potencia transmitida se obtiene usualmente de celdas solares a bordo y en consecuencia está limitada a 20-30 Watts. Las dimensiones físicas de la antena transmisora están también totalmente limitadas y en consecuencia lo está la ganancia asociada. La enorme distancia entre el transmisor y el receptor y la carencia de repetidores resulta en una SNR muy baja en el receptor. El ruido de canal puede caracterizarse

como gaussiano, de forma que esos canales se modelan muy bien como AWGN. Dado que el ancho de banda no es una preocupación principal puede aplicarse códigos convolucionales o de bloques.

En 1975 se envió a Marte a los Vikings, naves orbitatórias y de aterrizaje. En esa misión se utilizó un código de bloques (32,6) para proveer una ganancia de codificación de aproximadamente 4 dB con respecto a un sistema PSK no codificado con una probabilidad de error de 10^{-6}. Posteriormente, en la misión Voyager a los planetas exteriores (Marte, Júpiter y Saturno) se utilizó códigos convolucionales con decodificación de Viterbi. Dos códigos que fueron diseñados por el Jet Propulsion Laboratory para esa misión fueron un código convolucional (2,1) con restricción de longitud $L = 7$ y

$$\begin{aligned} \boldsymbol{g}_1 &= [1\ 1\ 0\ 1\ 1\ 0\ 1] \\ \boldsymbol{g}_2 &= [1\ 0\ 0\ 1\ 1\ 1\ 1] \end{aligned}$$

y un código convolucional (3,1) con restricción de longitud $L = 7$ y

$$\begin{aligned} \boldsymbol{g}_1 &= [1\ 1\ 0\ 1\ 1\ 0\ 1] \\ \boldsymbol{g}_2 &= [1\ 0\ 0\ 1\ 1\ 1\ 1] \\ \boldsymbol{g}_3 &= [1\ 0\ 1\ 0\ 1\ 1\ 1] \end{aligned}$$

El primer código tiene una distancia libre $d_{free} = 10$ y el segundo $d_{free} = 15$. Ambos códigos fueron decodificados utilizando el algoritmo de Viterbi y un esquema de decodificación suave en el cual se cuantizaba la salida con $Q = 8$ niveles. El primer código provee una ganancia de codificación de 5.1 dB con respecto a un sistema PSK no codificado operando con una probabilidad de error de 10^{-5}. El segundo código provee una ganancia de 5.7 dB. Ambos códigos operan a aproximadamente 4.5 dB del limite teórico predicho por la fórmula de Shannon.

En las misiones Voyager siguientes a Urano en 1986 se utilizó un código convolucional (2,1) con $L = 7$ como código interno concatenado con un código Reed-Solomon (255,223) como código externo. Una decodificación de Viterbi seguida de un decodificador Reed-Solomon en la estación terrestre permitió una ganancia total de codificación de 8 dB con una probabilidad de error de 10^{-6}. Este sistema operaba a una velocidad de aproximadamente 30 kbits/seg.

Para ciertos proyectos de comunicaciones espaciales se ha utilizado otros algoritmos de decodificación. Para el proyecto Pioner 9 de la NASA se diseñó un código convolucional con $L = 21$ con secuencias generadoras (en representación octal) dadas por

$$\begin{aligned} \boldsymbol{g}_1 &= [4\ 0\ 0\ 0\ 0\ 0\ 0] \\ \boldsymbol{g}_2 &= [7\ 1\ 5\ 4\ 7\ 3\ 7] \end{aligned}$$

que utilizaba un algoritmo de Fano con un esquema de decodificación de decisión suave y ocho niveles de cuantización de salida. Otro esquema de codificación frecuentemente utilizado, particularmente para las misiones Pionner 10, 11 y 12 y también las misiones alemanas de orbitaje solar Helios A y B, fué un código convolucional con $L = 32$. Las secuencias generadoras para este código (en representación octal) fueron las siguientes

$$\begin{aligned} \boldsymbol{g}_1 &= [7\ 3\ 3\ 5\ 3\ 3\ 6\ 7\ 6\ 7\ 2] \\ \boldsymbol{g}_2 &= [5\ 3\ 3\ 5\ 3\ 3\ 6\ 7\ 6\ 7\ 2] \end{aligned}$$

Este código tiene una distancia libre $d_{free} = 23$. Para la decodificación se utilizó nuevamente el algoritmo de decodificación de Fano con ocho niveles de cuantización de salida. Se ha utilizado también lógica de decodificación por mayoría en varios esquemas de codificación diseñados para los satélites de comunicaciones INTELSAT. Como ejemplo, un código (8,7) con $L = 48$ diseñado para operar a 64 kbits/seg sobre un satélite INTELSAT fué capaz de mejorar la probabilidad de error de 10^{-4} para 5×10^{-8}.

8.8.2 Codificación para modems telefónicos (*)

Los canales de línea telefónica se caracterizan por un ancho de banda limitado, típicamente de 300 a 3300 Hz y una SNR alta, usualmente 28 dB o más. En consecuencia, al diseñar esquemas de codificación para canales telefónicos se está frente a una restricción de ancho de banda. Esto contrasta con los canales de comunicaciones espaciales, donde la restricción básica es la potencia limitada. Esto corresponde al caso de $r \gg 1$ en la figura 8.38. Dado que el ancho de banda está limitado, es necesario utilizar esquemas de señalización de baja dimensionalidad, y dado que la potencia es abundanete, es posible emplear esquemas de modulación multinivel. Como se discutió en la sección anterior, un esquema apropiado en este caso es la modulación codificada en trellis.

Historicamente, los primeros modems para canales telefónicos (hasto los años 60) utilizaban modulación FSK con detección asincrónica y lograban velocidades de transmisión en el rango de 300 a 1200 bits/seg. Posteriormente, la primera generación de modems sincrónicos empleaba modulación PSK cuaternaria logrando velocidades de 2400 bits/seg. Los avances en las técnicas de ecualización permitieron constelaciones más sofisticadas, lo cual resultó en velocidades de transmisión más altas. Esos avances incluyeron modems PSK octales que lograban una velocidad de 4800 bits/seg, y modems QAM 16 con velocidades de 9600 bits/seg. Al principio de la década del 80 se introdujeron modems que utilizaban una constelación QAM 64 con velocidades de transmisión de 14400 bits/seg. Todas esas mejoras resultaron de avances en las técnicas de ecualización y procesamiento de señales y también en las características de las líneas telefónicas.

La aparición de la modulación codificada en trellis hizo posible el diseño de sistemas de modulación codificados que mejoraron el desempeño completo del sistema sin requerir ancho de banda en exceso. Los esquemas TCM basados en variaciones de los códigos de Ungerboeck, introducidos por Wei (1984), fueron adoptaron como un estándar por los comités de la CCITT. Esos códigos se basan en códigos convolucionales lineales y no lineales que garantizan la invarianza a rotaciones de fase de 180^o y 90^o. Esto es crucial en aplicaciones donde se emplea la codificación diferencial para evitar ambigüedades de fase por la utilización de un PLL para la estimación de la fase de la portadora en el receptor. Estos códigos logran una ganancia comparable a los códigos de Ungerboeck con el mismo número de estados pero al mismo tiempo proveen la invarianza de fase requerida. En la figura 8.39 se muestra la combinación de un codificador diferencial, el codificador convolucional no lineal y el mapeo de señal para un sistema de modulación codificada en trellis de 8 estados que fué adoptado en el estándar CCITT V.32.

8.8.3 Codificación para discos compactos (*)

En los capítulos 4, 5 y 6 se discutió los problemas de codificación de fuente y códigos de modulación para discos compactos. En esta sección se considerará las técnicas de corrección de errores empleadas en la grabación de audio digital en discos compactos.

El medio de almacenamiento para la grabación digital de audio sobre un disco compacto es un disco plástico de 120 mm de diámetro, 1.2 mm de espesor y un paso de surco (track) de 1.6 μm. En el momento de la reproducción este disco se lee mediante un haz laser a una velocidad de 1.25 m/seg. Dentro de la espiral del surco sobre el disco existen depresiones (pits) y areas planas entre pits denominadas lands. El audio digital se almacena mediante la longitud de esos pits y lands. Un 1 se representa mediante una transición de un pit a un land o viceversa, mientras un 0 corresponde a la no existencia de transición (modulación NRZI). Como se discutió previamente en el Capítulo 5, las restricciones sobre el tamaño físico de los pits y lands hacen necesario el empleo de códigos de corrida limitada (RLL). En la grabación de disco compacto se utiliza el código de modulación EFM 8-14 con $d = 2$ y $\kappa = 10$.

La fuente principal de errores en un disco compacto son las imperfecciones en la fabricación del disco, tal como burbujas de aire en el material plástico, y los daños sobre el disco, tal como impresiones digitales o rayas, polvo o desgastes de la superfície. Dado que cada pit mide casi 0.5 μm de ancho y alrededor de 0.9 a 3.3 μm de largo, esas fuentes de error resultan en ráfagas de error que afectan varios bits de información adyacentes, por lo que pueden usarse técnicas conocidas para la corrección de errores en ráfaga. Como se discutió anteriormente, los códigos Reed-Solomon son particularmente atractivos para tales aplicaciones.

Para detección y corrección de errores se emplea dos códigos Reed-Solomon[8], como se muestra en la figura 8.40. El primer código, $\mathcal{C}_1$, es un código RS (28,24), y el segundo código, $\mathcal{C}_2$ es un código RS (32,28). El alfabeto sobre el cual están definidos esos códigos consiste en secuencia binarias de longitud 8, que coincide con la definición de un símbolo.

[8] Son, en realidad, códigos Reed-Solomon acortados, obtenidos de códigos Reed-Solomon haciendo algunos bits de información igual a cero, y en consecuencia reduciendo k y n en una constante manteniendo la distancia mínima intacta.

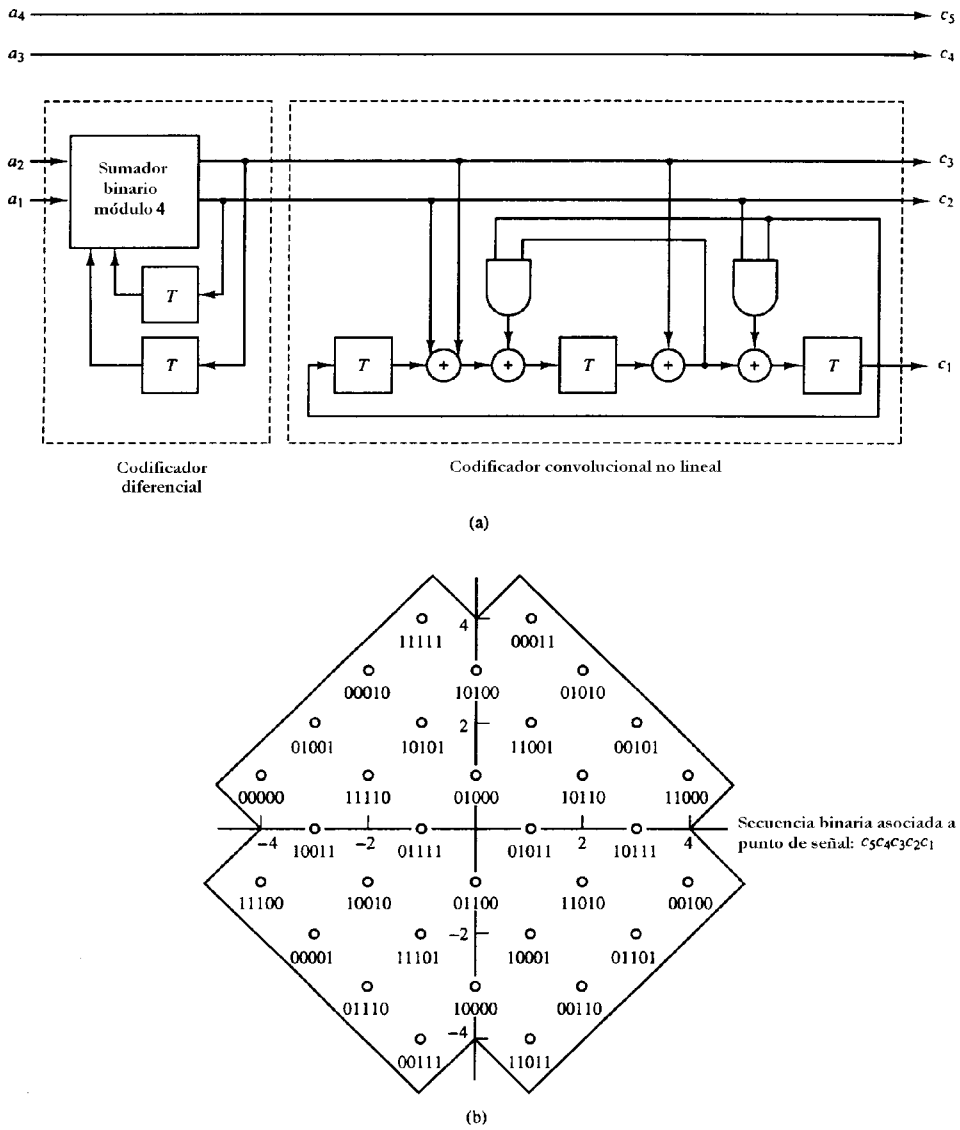

Figura 8.39: (a) Codificador diferencial, codificador convolucional no lineal, y (b) constelación adoptada por el estándar V.32.

La secuencia de entrada al codificador $\mathcal{C}_1$ consiste en 24 símbolos (usualmente conocido como un frame), el cual se codifica en 28 símbolos. Los 28 símbolos a la salida del codificador $\mathcal{C}_1$ se *entrelazan* (interleaving) para reducir el efecto de errores en ráfaga. Luego se codifican mediante $\mathcal{C}_2$ a 32 símbolos. En la salida de $\mathcal{C}_2$, los símbolos impares de cada frame se agrupan con los símbolos pares del frame próximo para formar un frame nuevo. A la salida del codificador $\mathcal{C}_2$, correspondiendo a cada conjunto de 6 muestras de audio, se tiene 32 símbolos de 8 bits. Se agrega un símbolo adicional de 8 bits que contiene la información de control y display (C y D), lo que determina un total de 33 símbolos por frame.

La salida se aplica luego a un codificador EFM 8-14 que mapea cada símbolo en una secuencia binaria de longitud 14. Como se vió en el Capítulo 5, se adiciona tres bits más, denominados bits de mezclado, por cada símbolo para asegurar que el mezclador de las palabras de código satisface la restricción de longitud de corrida. De esta forma la longitud de cada secuencia es 17. Luego, se completa un frame adicionando un patrón de sincronización de 24 bits y 3 bits de mezclado adicionales para garantizar la restricción de longitud de corrida después del mezclador. Esto lleva el número total de bits/frame (6 muestras, o $6 \times 2 \times 16 = 192$ bits de audio) a

$$32 \times 17 + 24 + 3 = 588$$

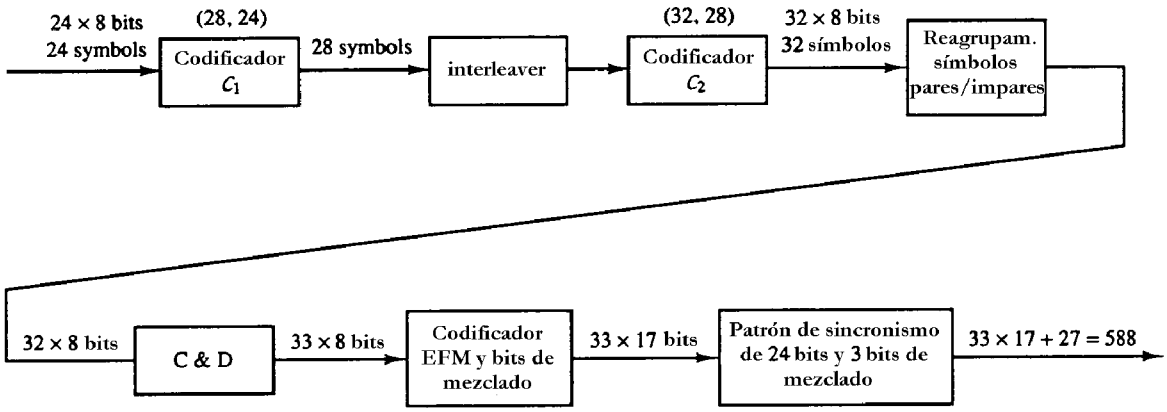

Figura 8.40: Codificación para la grabación de disco compacto.

bits de canal (ver la figura 5.30). El número de bits de canal por segundo está dado por

$$\frac{44100 \times 588}{6} = 4.321.800$$

Para un disco compacto que puede almacenar 67 minutos de música esto resulta en

$$67 \times 60 \times 4321800 = 17.373.636.000$$

bits de canal. El número de bits de información (bits de audio) está dado por

$$67 \times 60 \times 44100 \times 192 = 5.673.024.000$$

Esto significa que ha tenido lugar una expansión de 3 veces. Estos bits extra se utilizan para proteger de errores los bits de audio digital (los códigos RS) y también permiten garantizar que se satisface la restricción de longitud de corrida (el código EFM). La utilización eficiente de los códigos RLL hace posible almacenar más de 17 Mbits de canal con menos de 2000 Mpits.

En el momento de la reproducción, se separan primero los bits de sincronización y de mezclado, y luego se invierte la operación de entrelazado de 32 símbolos. El resultado entra luego al decodificador para el código $\mathcal{C}_2$. Este código tiene una distancia mínima de 5 y en consecuencia es capaz de correguir hasta dos errores. El decodificador, sin embargo, se diseña para corregir solo un error. Luego de acuerdo a la relación 8.7,

$$e_c + e_d \leq d_{min} - 1$$

es posible detectar hasta tres errores con certeza y cuatro o más errores con alta probabilidad. Si se encuentra un error único, se puede corregir; si se detectan errores múltiples, entonces los 28 símbolos se señalizan como no confiables. Después de la operación inversa del entrelazado los símbolos pasan por el decodificador del código $\mathcal{C}_1$. El decodificador de $\mathcal{C}_1$ maneja errores únicos. Si falla, se señalizan todos los símbolos de salida. Si en su entrada existen tres o más señalizaciones, las copia a la salida.

En la salida del segundo decodificador el símbolo correspondiente a posiciones no confiables se completa mediante interpolación a partir de las otras posiciones. Utilizando esta técnica de codificación - decodificación algo compleja junto con métodos de procesamiento de señales, puede cancelarse errores en ráfaga de hasta 12.000 bits de datos, lo cual corresponde a una longitud de surco de 7.5 mm sobre el disco.