

Estándares y protocolos de comunicación
Máster Universitario en Internet de las Cosas:
Tecnologías Aplicadas

Sesión 9

Protección frente a errores. Capacidad de canal

Marcelino Lázaro

Departamento de Teoría de la Señal y Comunicaciones
Universidad Carlos III de Madrid



1 / 76

Índice de contenidos

- Codificación de canal y capacidad de canal
 - ▶ Introducción al concepto de codificación de canal
 - ▶ Capacidad de canal
 - ★ Canales discretos sin memoria (DMCs)
 - ★ Canales gaussianos
- Códigos de canal
 - ▶ Códigos bloque
 - ▶ Códigos convolucionales
 - ▶ Códigos de rejilla (TCM)
 - ▶ Concatenación de códigos
 - ▶ Turbo códigos
 - ▶ Códigos LDPC

Materiales disponibles en

www.tsc.uc3m.es/~mlazaro/Docencia/MasterIoT.html

Introducción

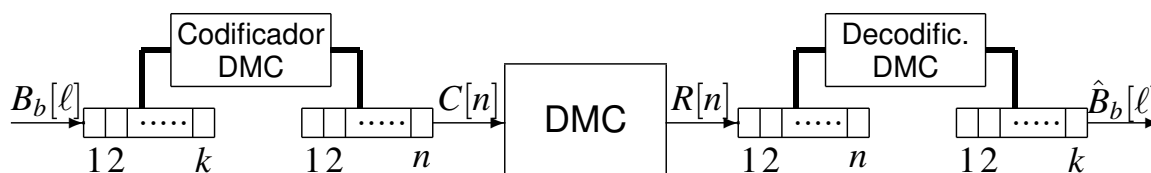
- Los sistemas de comunicaciones comenten errores
- Objetivo de un sistema de comunicaciones
 - ▶ BER < Calidad
- Alternativas para reducción de los errores
 - ▶ Aumentar la energía/potencia de la señal
 - ★ Limitaciones: Económicas, físicas, legales, interferencias, ...
 - ▶ Teorema de codificación de canales con ruido (Shannon)
 - ★ Introducción de bits de redundancia
 - ★ Tasa de codificación: R

$$R = \frac{\text{número de bits de información}}{\text{número de bits transmitidos (info+redundancia)}}$$

- ★ Capacidad del canal: C (bits/uso)
- ★ Posibilidad de reducción de la BER de forma arbitraria

$$R < C$$

Teorema de codificación de canal



DMC: *Discrete Memoryless Channel*

$$\text{Tasa del código: } R = \frac{k}{n}$$

Teorema de codificación de canal (Shannon 1948):

Capacidad de canal (DMC): $C = \max_{p_X(x_i)} I(X, Y)$

$I(X, Y)$: información mutua entre la entrada X y la salida Y del canal

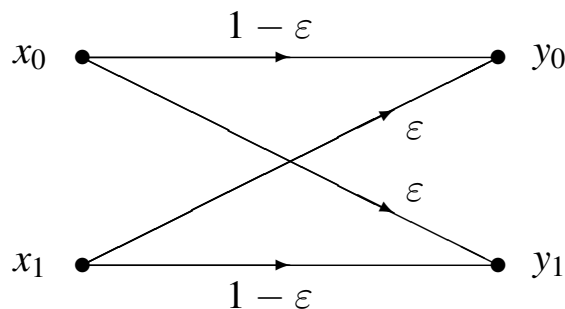
- 1 Si la tasa de transmisión R es menor que C , entonces para cualquier $\delta > 0$ existe un código con una longitud de bloque n suficientemente larga cuya probabilidad de error es menor que δ
- 2 Si $R > C$, la probabilidad de error de cualquier código con cualquier longitud de bloque está limitada por un valor no nulo
- 3 Existen códigos que permiten alcanzar la capacidad del canal $R = C$

Capacidad de canal - Canales digitales

- Modelo de canal discreto sin memoria (DMC)
 - ▶ Entrada y salida: variables aleatorias X e Y
 - ▶ Probabilidades de transición $p_{Y|X}(y_j|x_i)$
- Capacidad de canal

$$C = \max_{p_X(x_i)} I(X, Y) \text{ bits/uso}$$

- ▶ Ejemplo: Canal binario simétrico (BER= ε)



$$C = 1 - H_b(\varepsilon) \text{ bits/uso}$$

Ejemplo - Canal binario simétrico

- Modelo para canal digital binario con $BER = \varepsilon$
- Cálculo de la información mutua entrada / salida

$$\begin{aligned} I(X, Y) &= H(Y) - H(Y|X) = H(Y) - \sum_{i=0}^1 p_X(x_i) H(Y|X = x_i) \\ &= H(Y) - \sum_{i=0}^1 p_X(x_i) \left[- \sum_{j=0}^1 p_{Y|X}(y_j|x_i) \log p_{Y|X}(y_j|x_i) \right] \\ &= H(Y) - \sum_{i=0}^1 p_X(x_i) [-\varepsilon \log(\varepsilon) - (1 - \varepsilon) \log(1 - \varepsilon)] \\ &= H(Y) - \sum_{i=0}^1 p_X(x_i) H_b(\varepsilon) = H(Y) - H_b(\varepsilon) \end{aligned}$$

- Cálculo de la capacidad de canal
 - ▶ Se busca el máximo de la información mutua
 - ★ Para este canal, se obtiene cuando $H(Y)$ es máxima
 - ★ $H(Y)$ es máxima cuando los símbolos de salida son equiprobables
 - ★ Para este canal, ocurre cuando los símbolos de entrada son equiprobables

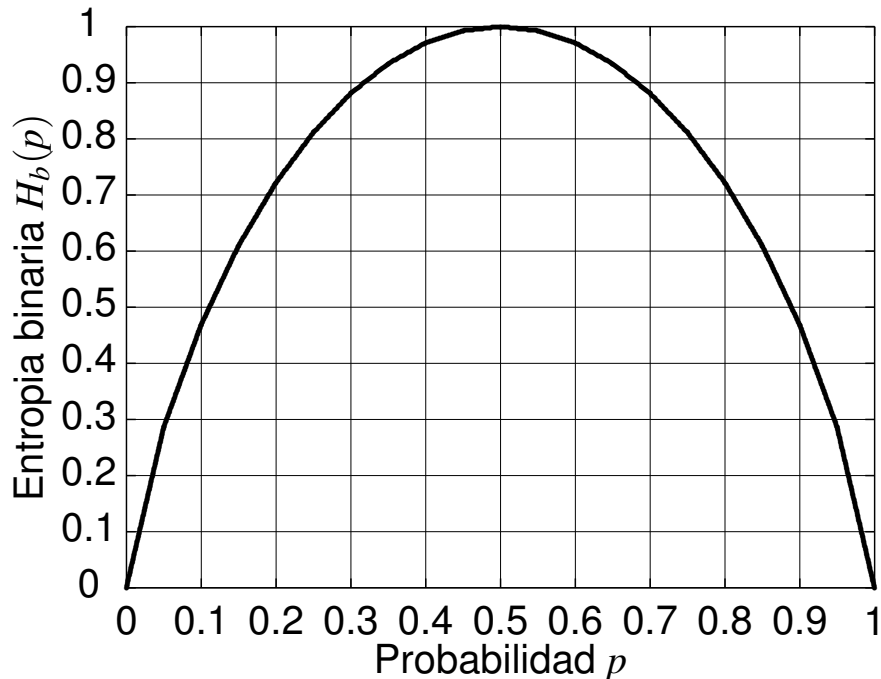
$$C = 1 - H_b(\varepsilon)$$

- ▶ Probabilidades de X que maximizan $I(X, Y)$: $p_X(x_0) = p_X(x_1) = \frac{1}{2}$

Entropía binaria $H_b(p)$

Entropía de una v.a. binaria con $p_X(x_0) = p$ y $p_X(x_1) = 1 - p$

$$H_b(p) = -p \log_2(p) - (1 - p) \log_2(1 - p)$$



Capacidad de canal para el canal gaussiano

- Capacidad de canal en las siguientes condiciones:
 - ▶ Potencia transmitida: P_X watt.
 - ▶ Ancho de banda: B Hz
 - ★ Potencia de ruido: $P_Z = N_0 B$ watt.
- Modelo de relación entrada salida en un canal gaussiano

$$Y = X + Z$$

Z es una variable aleatoria gaussiana, de media nula y varianza P_Z

- Cálculo de la capacidad mediante la información mutua

$$C = \max_{f_X(x) \mid E[X^2] \leq P_X} I(X, Y)$$

Restricción $E[X^2] \leq P_X$ dada por la limitación en potencia

- Resultado

$$C = B \log \left(1 + \frac{P_X}{N_0 B} \right) \text{ bits/s}$$

Se obtiene para $f_X(x)$ gaussiana

Evolución de la capacidad de canal

- Capacidad del canal gaussiano

$$C = B \log \left(1 + \frac{P_X}{N_0 B} \right) \text{ bits/s}$$

- Depende de dos parámetros

- ▶ Potencia de la señal transmitida, P_X
- ▶ Ancho de banda disponible en Hz, B

- Capacidad de canal en función de la potencia transmitida P_X

$$\lim_{P_X \rightarrow \infty} C = \infty$$

- ▶ Se puede aumentar C de forma arbitraria aumentando P_X
- ▶ Aumento lineal de C requiere aumento exponencial de P_X

- Capacidad de canal en función del ancho de banda (B Hz)

$$\lim_{B \rightarrow \infty} C = \frac{P}{N_0} \log_2(e) = 1,44 \frac{P}{N_0}$$

uc3m Universidad Carlos III de Madrid  El incremento de B no permite un incremento arbitrario de C Marcelino Lázaro, 2019 Teoría de la Señal Protección frente a errores. Capacidad de canal 9/76

Límites para la transmisión en un canal gaussiano

- Algunas definiciones

- ▶ Tasa de transmisión binaria: R_b bits/s
- ▶ Relación señal a ruido: $SNR = \frac{P_X}{P_Z} = \frac{P_X}{N_0 B}$
- ▶ Tasa binaria (eficiencia) espectral: $\eta = \frac{R_b}{B}$ bits/s/Hz
- ▶ Energía media por bit: $E_b = \frac{P_X}{R_b}$
- ▶ Relación E_b/N_0 : $\frac{E_b}{N_0} = \frac{P_X}{R_b N_0} = \frac{P_X}{N_0 B} \times \frac{B}{R_b} = \frac{SNR}{\eta}$

- Sistema de comunicaciones práctico

$$R_b < C \rightarrow R_b < B \log (1 + SNR) \text{ bits/s}$$

- ▶ Dividiendo por B en ambos lados y reordenando

$$\eta < \log (1 + SNR), \quad \eta < \log \left(1 + \eta \frac{E_b}{N_0} \right)$$

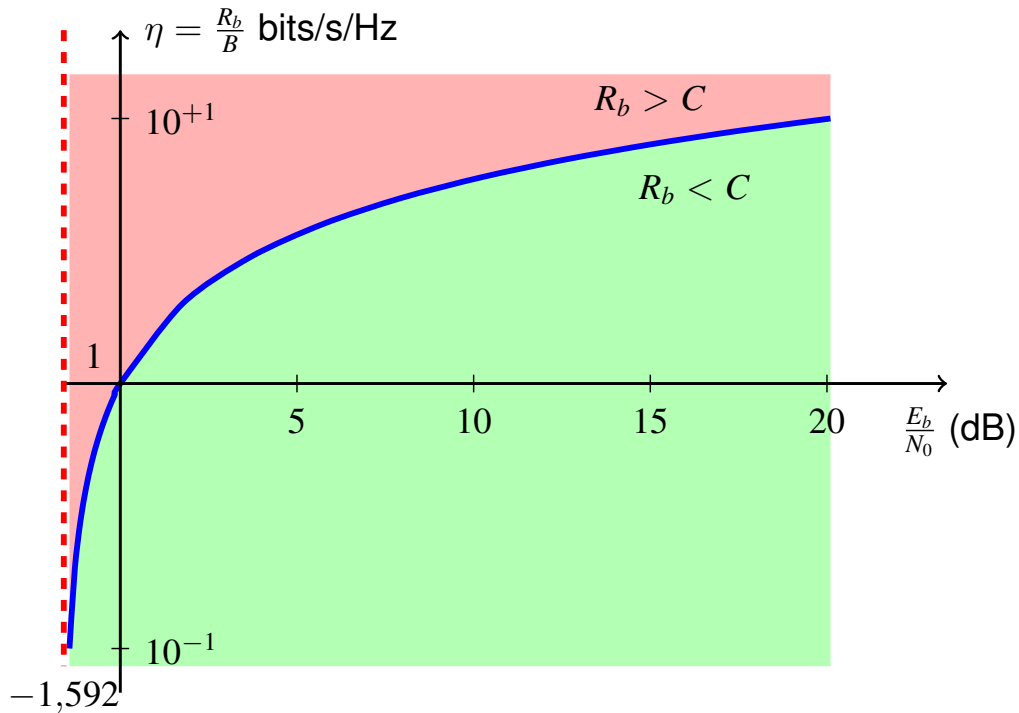
$$SNR > 2^\eta - 1, \quad \frac{E_b}{N_0} > \frac{2^\eta - 1}{\eta}$$

Cuando $\eta \rightarrow 0$

$$\frac{E_b}{N_0} = \ln 2 = 0,693 \approx -1,6 \text{ dB}$$

Tasa binaria espectral frente a E_b/N_0

- Se representa sobre el plano η vs $\frac{E_b}{N_0}$ la curva $\frac{E_b}{N_0} = \frac{2^\eta - 1}{\eta}$
 - Divide el plano en dos regiones: sistemas con $R_b > C$ (prácticos) y con $R_b < C$



Relación señal a ruido normalizada

- Cota inferior para SNR

$$SNR > 2^\eta - 1$$

- Definición de SNR normalizada

$$SNR_{norm} = \frac{SNR}{2^\eta - 1}$$

- Cota inferior sobre SNR_{norm}

$$SNR_{norm} > 1 \text{ (0 dB)}$$

Tipos de códigos

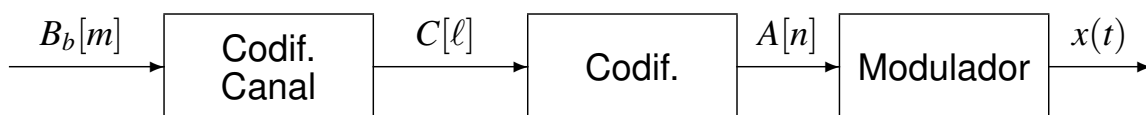
- Mecanismo de introducción de la redundancia
 - ▶ Códigos bloque
 - ★ Bloques de k bits se codifican de forma independiente
 - ★ Diccionario del código: k bits sin codificar / n bits codificados
 - ★ Concepto clave: distancia entre palabras código
 - ★ Ejemplo: código de repetición de orden $n - 1$
- | Bits sin codificar ($k = 1$) | Bits codificados (n) |
|--------------------------------|--------------------------|
| 1 | 11...1 |
| 0 | 00...0 |
- ▶ Códigos convolucionales
 - ★ Codificación continua mediante filtrado digital
- Capacidad del código
 - ▶ Códigos de detección de errores
 - ▶ Códigos de corrección de errores
 - Estadístico para la decisión
 - ▶ Salida dura: decodificación a partir de los bits decididos $\hat{C}[\ell]$
 - ▶ Salida blanda: decodificación a partir de la salida del demodulador $q[n]$
 - ★ Mejores prestaciones pero mayor complejidad

Borrado de bits: se "marcan" los bits/símbolos dudosos

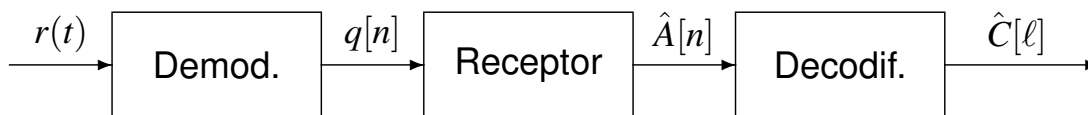
 Marcelino Lázaro, 2019 Teoría de la Señal Protección frente a errores. Capacidad de canal 13 / 76

Salida blanda / salida dura

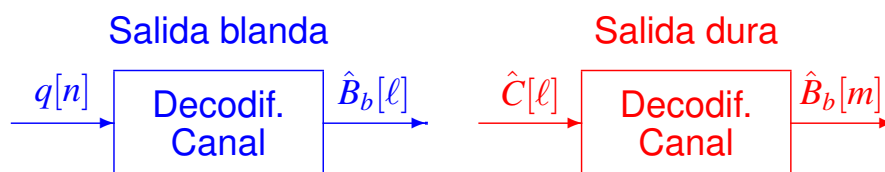
- Diagrama de bloques del transmisor



- Diagrama de bloques del receptor



- Decodificador de canal: salida blanda / salida dura



$B_b[m]$: bits de información (sin codificar)

$C[\ell]$: bits codificados

Salida dura / salida blanda

- Ejemplo: código repetición orden 2, modulación 2-PAM
 - ▶ Asignación binaria sobre la constelación: $0 \equiv -1 / 1 \equiv +1$
 - ▶ Observación blanda: $\mathbf{q} = \{-0,01, +1,2, -0,05\}$
 - ▶ Observación dura: $\hat{c} = \{0, 1, 0\}$
- Decodificación
 - ▶ Decodificación dura: por mayoría $\hat{B} = 0$
 - ▶ Decodificación blanda: comparar la observación \mathbf{q} con $\mathbf{q}_0\{-1, -1, -1\}$ y con $\mathbf{q}_1\{+1, +1, +1\}$

$$d(\mathbf{q}, \mathbf{q}_0) = \sqrt{(-0,01 - (-1))^2 + (+1,2 - (-1))^2 + (-0,05 - (-1))^2} = 2,59$$

$$d(\mathbf{q}, \mathbf{q}_1) = \sqrt{(-0,01 - (+1))^2 + (+1,2 - (+1))^2 + (-0,05 - (+1))^2} = 1,47$$

- Probabilidad de error para una BER= ε sobre la 2-PAM

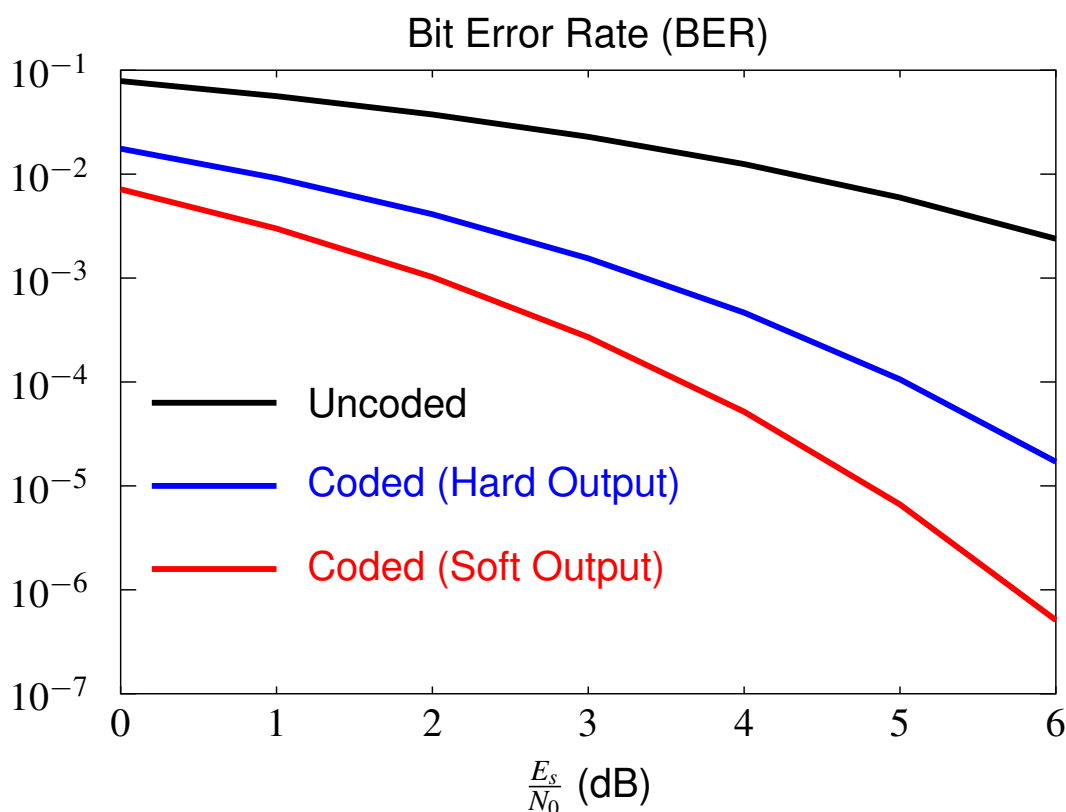
- ▶ Salida dura

$$P_e^{Dura} = 3 \varepsilon^2 (1 - \varepsilon) + \varepsilon^3, \quad \varepsilon = Q\left(\frac{1}{\sqrt{N_0/2}}\right) = Q\left(\sqrt{2\frac{E_s}{N_0}}\right)$$

- ▶ Salida blanda

$$P_e^{Blanda} = Q\left(\frac{d(\mathbf{q}_0, \mathbf{q}_1)}{2\sqrt{N_0/2}}\right) = Q\left(\frac{\sqrt{3}}{\sqrt{N_0/2}}\right) = Q\left(\sqrt{\frac{6E_s}{N_0}}\right)$$

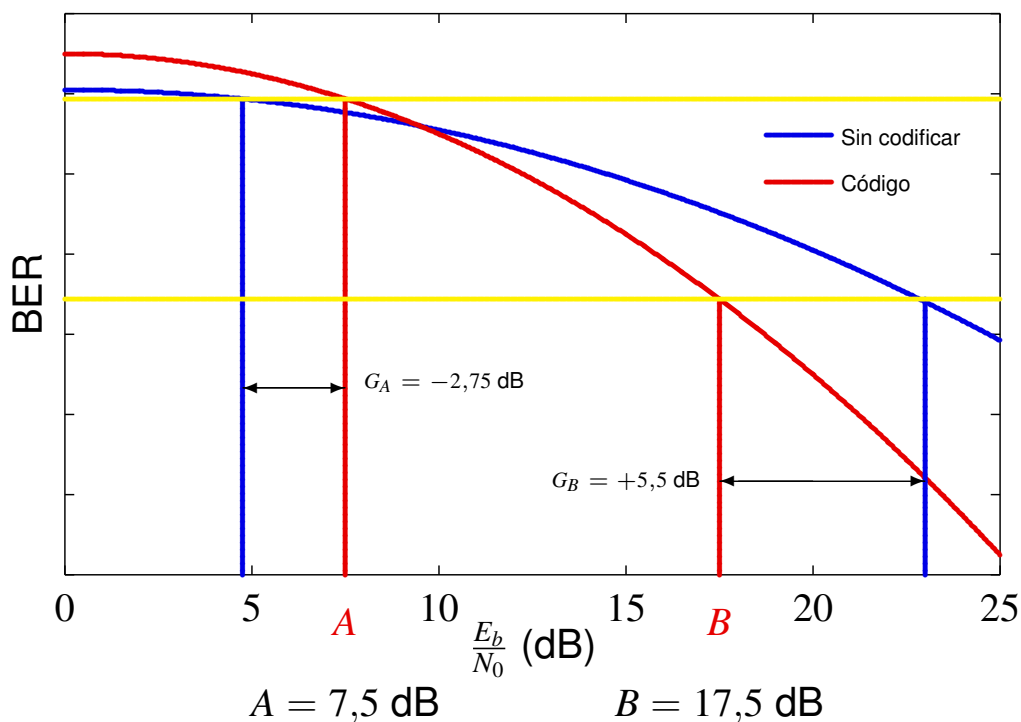
Salida dura / salida blanda



Ganancia de codificación

- Definición: diferencia en decibelios entre las relaciones E_b/N_0 necesarias para alcanzar una determinada BER sin codificar y utilizando la codificación
 - ▶ E_b : Energía media por bit de información
- Permite comparar las prestaciones de distintos códigos
- Depende de la BER (o de E_b/N_0)
 - ▶ Habitualmente se referencia a un valor concreto de E_b/N_0
- Puede ser positiva a partir de un cierto valor de E_b/N_0

Ganancia de codificación



Códigos bloque - Definiciones

- Codificación independiente de bloques de k bits

- ▶ Conversión en bloques de n bits \rightarrow Tasa $R = k/n$

- Definiciones para los bloques de bits

- ▶ Información:

$$\mathbf{b}_i = [b_i[0], b_i[1], \dots, b_i[k-1]], \quad i = 0, 1, \dots, 2^k - 1$$

- ▶ Codificados:

$$\mathbf{c}_i = [c_i[0], c_i[1], \dots, c_i[n-1]], \quad i = 0, 1, \dots, 2^k - 1$$

- ▶ Diccionario del código: mensaje (palabra sin codificar) \rightarrow palabra código

$$\mathbf{b}_i \rightarrow \mathbf{c}_i$$

- Peso de una palabra código $w(\mathbf{c}_i)$

- ▶ Número de unos de la palabra

- Distancia de Hamming entre dos palabras código $d^H(\mathbf{c}_i, \mathbf{c}_j)$

- ▶ Número de bits diferentes entre ambas palabras

- Distancia mínima del código: $d_{min}^H \equiv d_{min}$

- ▶ Mínima distancia de Hamming entre dos palabras código distintas

Codigos bloque - Ejemplo

- Diccionario del código para un código bloque $k = 2 \quad n = 6$

i	\mathbf{b}_i	\mathbf{c}_i
0	00	001110
1	01	010011
2	10	100100
3	11	111001

- Peso de las palabras

$$w(\mathbf{c}_0) = 3, \quad w(\mathbf{c}_1) = 3, \quad w(\mathbf{c}_2) = 2, \quad w(\mathbf{c}_3) = 4,$$

- Distancias de Hamming entre dos palabras

$$d(\mathbf{c}_0, \mathbf{c}_1) = 4, \quad d(\mathbf{c}_1, \mathbf{c}_2) = 5$$

$$d(\mathbf{c}_0, \mathbf{c}_2) = 3, \quad d(\mathbf{c}_1, \mathbf{c}_3) = 3$$

$$d(\mathbf{c}_0, \mathbf{c}_3) = 5, \quad d(\mathbf{c}_2, \mathbf{c}_3) = 4$$

- Distancia mínima del código

$$d_{min} = 3$$

Estimador óptimo - Salida dura

- Observación condicionada a la transmisión de c_i

$$\mathbf{r} = \mathbf{c}_i + \mathbf{e}, \quad \mathbf{e} = [e[0], e[1], \dots, e[n-1]]$$

- Modelo probabilístico del patrón de error (BER = ε)

$$p_{E[j]}(e[j]) = \varepsilon^{e[j]} (1 - \varepsilon)^{1-e[j]} = \begin{cases} \varepsilon, & e[j] = 1 \\ 1 - \varepsilon, & e[j] = 0 \end{cases}$$

- Verosimilitud (probabilidad condicional de la observación)

- ▶ Error: $e[j] = r[j] - c[j]$
- ▶ Verosimilitud para cada bit dado el bit de la observación $r[j]$

$$p_{R[j]|C[j]}(r[j]|c[j]) = \varepsilon^{e[j]} (1 - \varepsilon)^{1-e[j]} = \varepsilon^{r[j]-c_i[j]} (1 - \varepsilon)^{1-(r[j]-c_i[j])}$$

- ▶ Verosimilitud de una palabra código para la observación \mathbf{r}

$$p_{R|C}(\mathbf{r}|\mathbf{c}_i) = \prod_{j=0}^{n-1} \varepsilon^{r[j]-c_i[j]} (1 - \varepsilon)^{1-(r[j]-c_i[j])}$$

- Estimador de máxima verosimilitud (ML)

$$\hat{\mathbf{c}} = \mathbf{c}_i = \arg \min_{\mathbf{c}_i} d^H(\mathbf{r}, \mathbf{c}_i)$$

Capacidades de detección y corrección con salida dura

- Prestaciones dependen de distancias de Hamming

- ▶ Un error no será detectado si los errores de transmisión lo transforman en otra palabra del código
- ▶ Un error ocurre cuando el número de errores en la transmisión de la palabra codificada hace que la palabra recibida esté a una menor distancia de Hamming de otra palabra del código

- Prestaciones determinadas por la distancia mínima del código: d_{min}

- ▶ Capacidad de detección:

$$d = d_{min} - 1 \text{ errores}$$

- ▶ Capacidad de corrección:

$$t = \left\lfloor \frac{d_{min} - 1}{2} \right\rfloor \text{ errores}$$

Estimador óptimo - Salida blanda

- Constelación: M símbolos ($m = \log_2(M)$ bits/símbolo)
- Secuencia de símbolos para una palabra código

$$\mathbf{c}_i \rightarrow \mathbf{A}_i = [A_i[0], A_i[1], \dots, A_i[n' - 1]], \quad n' = \frac{n}{m}$$

- Modelo probabilístico de la observación condicionada a transmitir \mathbf{c}_i

$$\mathbf{q} = \mathbf{A}_i + \mathbf{e}, \quad \mathbf{e} = [e[0], e[1], \dots, e[n' - 1]]$$

- Modelo probabilístico del error:

$$f_E(e[j]) = N(0, \sigma_z^2)$$

- Verosimilitud (probabilidad condicional de la observación):

$$f_{\mathbf{q}|\mathbf{A}}(\mathbf{q}|\mathbf{A}_i) = N(\mathbf{A}_i, \sigma_z^2)$$

- Estimador de máxima verosimilitud: minimizar distancia euclídea

$$\hat{\mathbf{c}} = \mathbf{c}_i, \quad i = \arg \min_i d^E(\mathbf{q}, \mathbf{A}_i)$$

Ejemplo de código bloque y asignación de símbolos

- Diccionario del código para un código bloque $k = 2$ $n = 6$

i	\mathbf{b}_i	\mathbf{c}_i
0	00	001110
1	01	010011
2	10	100100
3	11	111001

- Asignación de símbolos depende de la constelación y asignación binaria

Ejemplo A: 2-PAM ($n' = n$)	
	$A_0 = [a_0, a_0, a_1, a_1, a_1, a_0] = [-1, -1, +1, +1, +1, -1]$ $A_1 = [a_0, a_1, a_0, a_0, a_1, a_1] = [-1, +1, -1, -1, +1, +1]$ $A_2 = [a_1, a_0, a_0, a_1, a_0, a_0] = [+1, -1, -1, +1, -1, -1]$ $A_3 = [a_1, a_1, a_1, a_0, a_0, a_1] = [+1, +1, +1, -1, -1, +1]$

Ejemplo B: 4-PAM ($n' = \frac{n}{2}$)	
	$A_0 = [a_0, a_2, a_3] = [-3, +1, +3]$ $A_1 = [a_1, a_0, a_2] = [-1, -3, +1]$ $A_2 = [a_3, a_1, a_0] = [+3, -1, -3]$ $A_3 = [a_2, a_3, a_1] = [+1, +3, -1]$

Ejemplo C: 4-QAM ($n' = \frac{n}{2}$)	
	$A_0 = [a_0, a_2, a_3] = \left[\begin{bmatrix} -1 \\ -1 \end{bmatrix}, \begin{bmatrix} +1 \\ -1 \end{bmatrix}, \begin{bmatrix} +1 \\ +1 \end{bmatrix} \right]$ $A_1 = [a_1, a_0, a_2] = \left[\begin{bmatrix} -1 \\ +1 \end{bmatrix}, \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \begin{bmatrix} +1 \\ -1 \end{bmatrix} \right]$ $A_2 = [a_3, a_1, a_0] = \left[\begin{bmatrix} +1 \\ +1 \end{bmatrix}, \begin{bmatrix} -1 \\ +1 \end{bmatrix}, \begin{bmatrix} -1 \\ -1 \end{bmatrix} \right]$ $A_3 = [a_2, a_3, a_1] = \left[\begin{bmatrix} +1 \\ -1 \end{bmatrix}, \begin{bmatrix} +1 \\ +1 \end{bmatrix}, \begin{bmatrix} -1 \\ +1 \end{bmatrix} \right]$

Ejemplo de decodificación sobre salida blanda

- Diccionario del código para un código bloque $k = 2n = 6$

i	b_i	c_i
0	00	001110
1	01	010011
2	10	100100
3	11	111001

- Asignación de símbolos

Ejemplo A: 2-PAM ($n' = n$)	$A_0 = [a_0, a_0, a_1, a_1, a_1, a_0] = [-1, -1, +1, +1, +1, -1]$
	$A_1 = [a_0, a_1, a_0, a_0, a_1, a_1] = [-1, +1, -1, -1, +1, +1]$
	$A_2 = [a_1, a_0, a_0, a_1, a_0, a_0] = [+1, -1, -1, +1, -1, -1]$
	$A_3 = [a_1, a_1, a_1, a_0, a_0, a_1] = [+1, +1, +1, -1, -1, +1]$

n	0	1	2	3	4	5
$q[n]$	+0,8	-1,2	-0,7	+0,2	-0,5	-1,1

$$d(q, A_0) = \sqrt{(+0,8 - (-1))^2 + (-1,2 - (-1))^2 + (-0,7 - (+1))^2 + (+0,2 - (+1))^2 + (-0,5 - (+1))^2 + (-1,1 - (-1))^2} = \sqrt{9,07}$$

$$d(q, A_1) = \sqrt{(+0,8 - (-1))^2 + (-1,2 - (+1))^2 + (-0,7 - (-1))^2 + (+0,2 - (-1))^2 + (-0,5 - (+1))^2 + (-1,1 - (+1))^2} = \sqrt{16,27}$$

$$d(q, A_2) = \sqrt{(+0,8 - (+1))^2 + (-1,2 - (-1))^2 + (-0,7 - (-1))^2 + (+0,2 - (+1))^2 + (-0,5 - (-1))^2 + (-1,1 - (-1))^2} = \sqrt{1,07}$$

$$d(q, A_3) = \sqrt{(+0,8 - (+1))^2 + (-1,2 - (+1))^2 + (-0,7 - (+1))^2 + (+0,2 - (-1))^2 + (-0,5 - (-1))^2 + (-1,1 - (+1))^2} = \sqrt{13,87}$$

Decisión: $\hat{c} = c_2 = 100100$

Ejemplo de decodificación sobre salida blanda (II)

- Diccionario del código para un código bloque $k = 2n = 6$

i	b_i	c_i
0	00	001110
1	01	010011
2	10	100100
3	11	111001

- Asignación de símbolos

Ejemplo B: 4-PAM ($n' = \frac{n}{2}$)	$A_0 = [a_0, a_2, a_3] = [-3, +1, +3]$
	$A_1 = [a_1, a_0, a_2] = [-1, -3, +1]$
	$A_2 = [a_3, a_1, a_0] = [+3, -1, -3]$
	$A_3 = [a_2, a_3, a_1] = [+1, +3, -1]$

n	0	1	2
$q[n]$	+0,8	+2,2	-0,7

$$d(q, A_0) = \sqrt{(+0,8 - (-3))^2 + (+2,2 - (+1))^2 + (-0,7 - (+3))^2} = \sqrt{29,57}$$

$$d(q, A_1) = \sqrt{(+0,8 - (-1))^2 + (+2,2 - (-3))^2 + (-0,7 - (+1))^2} = \sqrt{33,17}$$

$$d(q, A_2) = \sqrt{(+0,8 - (+3))^2 + (+2,2 - (-1))^2 + (-0,7 - (-3))^2} = \sqrt{20,37}$$

$$d(q, A_3) = \sqrt{(+0,8 - (+1))^2 + (+2,2 - (+3))^2 + (-0,7 - (-1))^2} = \sqrt{0,77}$$

Decisión: $\hat{c} = c_3 = 111001$

Códigos bloque lineales

- Código $C(k, n)$
- Base del código: k palabras código linealmente independientes

$$\{\mathbf{g}_0, \mathbf{g}_1, \dots, \mathbf{g}_{k-1}\}$$
$$\mathbf{g}_i = [g_i[0], g_i[1], \dots, g_i[n-1]]$$

- Palabra código: combinación lineal de los k elementos de la base

$$\mathbf{c}_i = b_i[0] \mathbf{g}_0 + b_i[1] \mathbf{g}_1 + \dots + b_i[k-1] \mathbf{g}_{k-1}$$

Coeficientes de la expansión: k bits de información (sin codificar) $b_i[\ell]$

- Propiedades

- ▶ Todos los elementos de la base pertenecen al código

$$\mathbf{c}_i = \mathbf{g}_\ell \rightarrow w(\mathbf{b}_i) = 1, \quad b_i[\ell] = 1$$

- ▶ $\mathbf{c}_0 = \mathbf{0} = [0, 0, \dots, 0]$ pertenece al código

★ Asociada a $\mathbf{b}_0 = \mathbf{0} = [0, 0, \dots, 0]$

- ▶ Toda combinación lineal de palabras código $\in C(k, n)$
- ▶ Todas las palabras del código tienen a otra palabra a distancia d_{min}
- ▶ Por tanto, \mathbf{c}_0 tiene otra palabra a d_{min}

$$d_{min} = \min_{\mathbf{c}_i \neq \mathbf{c}_0} w(\mathbf{c}_i)$$

Matriz generadora del código

- Agrupación de la base en una matriz $k \times n$

$$\mathbf{G} = \begin{bmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \\ \vdots \\ \mathbf{g}_{k-1} \end{bmatrix} = \begin{bmatrix} g_0[0] & g_0[1] & \dots & g_0[n-1] \\ g_1[0] & g_1[1] & \dots & g_1[n-1] \\ \vdots & \vdots & \ddots & \vdots \\ g_{k-1}[0] & g_{k-1}[1] & \dots & g_{k-1}[n-1] \end{bmatrix}$$

- Obtención de las palabras código

$$\mathbf{c}_i = \mathbf{b}_i \mathbf{G}$$

- Códigos sistemáticos: el mensaje \mathbf{b}_i forma parte de \mathbf{c}_i

$$\mathbf{c}_i = [\mathbf{b}_i | \mathbf{p}_i] \rightarrow \mathbf{G} = [\mathbf{I}_k | \mathbf{P}]$$

$$\mathbf{c}_i = [\mathbf{p}_i | \mathbf{b}_i] \rightarrow \mathbf{G} = [\mathbf{P} | \mathbf{I}_k]$$

Matriz generadora del código - Ejemplo

- Código $C(2, 5)$

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

- Palabras código

b_i	c_i
0 0	0 0 0 0 0
0 1	1 0 1 0 1
1 0	0 1 1 1 0
1 1	1 1 0 1 1

- Código sistemático
- Distancia mínima del código: $d_{min} = 3$
 - ▶ Detecta 2 errores
 - ▶ Corrige 1 error

Matriz de chequeo de paridad

- Matriz $(n - k) \times n$: complemento ortogonal de \mathbf{G}

$$\mathbf{G} \mathbf{H}^T = \mathbf{0} \text{ matriz de } k \times (n - k) \text{ ceros}$$

- Códigos sistemáticos

$$\mathbf{G} = [\mathbf{I}_k | \mathbf{P}] \rightarrow \mathbf{H} = [\mathbf{P}^T | \mathbf{I}_{n-k}]$$

$$\mathbf{G} = [\mathbf{P} | \mathbf{I}_k] \rightarrow \mathbf{H} = [\mathbf{I}_{n-k} | \mathbf{P}^T]$$

- Identificación de palabras código

$$c_i \mathbf{H}^T = b_i \mathbf{G} \mathbf{H}^T = \mathbf{0} \text{ vector de } n - k \text{ ceros}$$

- Decodificación mediante síndrome

- ▶ Modelo de transmisión: $r = c_i + e$ (e : patrón de error)
- ▶ Síndrome

$$s = r \mathbf{H}^T = (c_i + e) \mathbf{H}^T = e \mathbf{H}^T$$

- ▶ Decodificación: Tabla de síndromes

Matriz de chequeo de paridad y tabla de síndromes - Ejemplo

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

- Tabla de síndromes

e	s
0 0 0 0 0	0 0 0
1 0 0 0 0	1 0 0
0 1 0 0 0	0 1 0
0 0 1 0 0	0 0 1
0 0 0 1 0	0 1 1
0 0 0 0 1	1 0 1
¿?	1 1 0
¿?	1 1 1

110 $\rightarrow e_1 = 11000, e_2 = 00011, e_3 = 10110, e_4 = 01101$

111 $\rightarrow e_1 = 10010, e_2 = 01001, e_3 = 11100, e_4 = 00111$

Solución: elegir uno de los dos patrones de dos errores (e_1 o e_2)

- ▶ Que haya dos bits erróneos es más probable que haya tres o más errores

Decodificación por síndrome

- Proceso de decodificación alternativo y equivalente a máxima verosimilitud sobre salida dura
- Pasos a seguir

- 1 Cálculo del síndrome

$$s = r H^T$$

- 2 Identificación del patrón de error (tabla de síndromes)

$$s \rightarrow e$$

- 3 Corrección de errores (decisión de palabra código)

$$\hat{c} = r + e = c_i$$

- 4 Decodificación

$$\hat{c} = c_i \rightarrow \hat{b} = b_i$$

Más sencillo en códigos sistemáticos

Ventaja de trabajar con G y H

- Implementación del codificador y del decodificador
 - ▶ Número de palabras del código: 2^k
 - ★ $k = 2, n = 5$: 4 palabras
 - ★ $k = 247, n = 255$: $2^{247} \approx 2,26 \cdot 10^{74}$ palabras
 - ▶ Número de síndromes posibles
 - ★ $k = 2, n = 5$ ($t = 1$): 8 síndromes
 - ★ $k = 247, n = 255$ ($t = 1$): 256 síndromes
- Alternativa: códigos estructurados
 - ▶ Implementación mediante registros de desplazamiento
 - ★ Definición mediante polinomios generadores

Método de eliminación - Ejemplo

- Método para obtener una matriz de chequeo para un código no sistemático
- Sustitución de filas por combinaciones lineales de otras
 - ▶ 1ª fila: 1ª+2ª filas
 - ▶ 2ª fila: 1ª fila
- Código $C(2, 5)$

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

- Palabras código

b_i	c_i
00	00000
01	01110
10	11011
11	10101

- Mismas palabras código / distintas asignaciones
 - ▶ La misma matriz H es válida para generar la tabla de síndromes

Límite de Hamming

- Número de síndromes con redundancia $r = n - k$:

$$2^{n-k} = 2^r$$

- Límite de Hamming: para corregir t errores la mínima redundancia necesaria es

$$r \geq \log_2 V(n, t), \quad V(n, t) = \sum_{j=0}^t \binom{n}{j}$$

- ▶ $V(n, t)$: Esfera de Hamming de radio t

- Interpretación con número de síndromes disponibles

$$\binom{n}{0} + \binom{n}{1} + \dots + \binom{n}{t} \leq 2^{n-k}$$

- ▶ Igualdad: Códigos perfectos

Códigos perfectos

- Códigos de repetición (decisión por mayoría)

- ▶ n impar, $k = 1$, $t = \frac{n-1}{2}$ ($d_{min} = n$)

- Códigos de Hamming

- ▶ Para $m \geq 3$, $n = 2^m - 1$, $k = 2^m - m - 1$, $t = 1$ ($d_{min} = 3$)
- ▶ Matriz de chequeo: en las columnas aparecen todas las posibles combinaciones binarias de $(n - k)$ bits, excepto la todo ceros

- ★ Ejemplo: Código Hamming (4,7)

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

- Código de Golay

- ▶ $n = 23$, $k = 12$, $t = 3$ ($d_{min} = 7$)

Código de Golay

● Matriz generadora sistemática

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

● Código de Golay extendido

- ▶ Se añade un bit de paridad: $d_{min} = 8$, (detecta $d = 7$ errores)

★ No es un código perfecto

- ▶ Código $k = 12, n = 24$, tasa $R = \frac{1}{2}$

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

Prestaciones - Decodificación dura

- Probabilidad de error de bit (BER) en la transmisión: ε
- Se comenten errores cuando se excede la capacidad de corrección del código
- Código de corrección de t errores

$$P_e = \sum_{e=t+1}^n \binom{n}{e} \varepsilon^e (1 - \varepsilon)^{n-e}$$

- Códigos que corrigen todos los patrones de t errores y a patrones de $t + 1$ errores

$$P_e = \left[\binom{n}{t+1} - a \right] \varepsilon^{t+1} (1 - \varepsilon)^{n-t-1} + \sum_{e=t+2}^n \binom{n}{e} \varepsilon^e (1 - \varepsilon)^{n-e}$$

- Codificación tipo Gray o pseudo-Gray para SNR alta

$$BER \approx \frac{1}{k} P_e$$

Prestaciones - Decodificación blanda

- Probabilidad de error

$$P_e \approx c Q \left(\frac{d_{min}^E}{2\sqrt{N_0/2}} \right)$$

- d_{min}^E : mínima distancia euclídea entre las secuencias de símbolos correspondientes a dos palabras código diferentes

$$d_{min}^E = \min_{j \neq i} d^E(\mathbf{A}_i, \mathbf{A}_j)$$

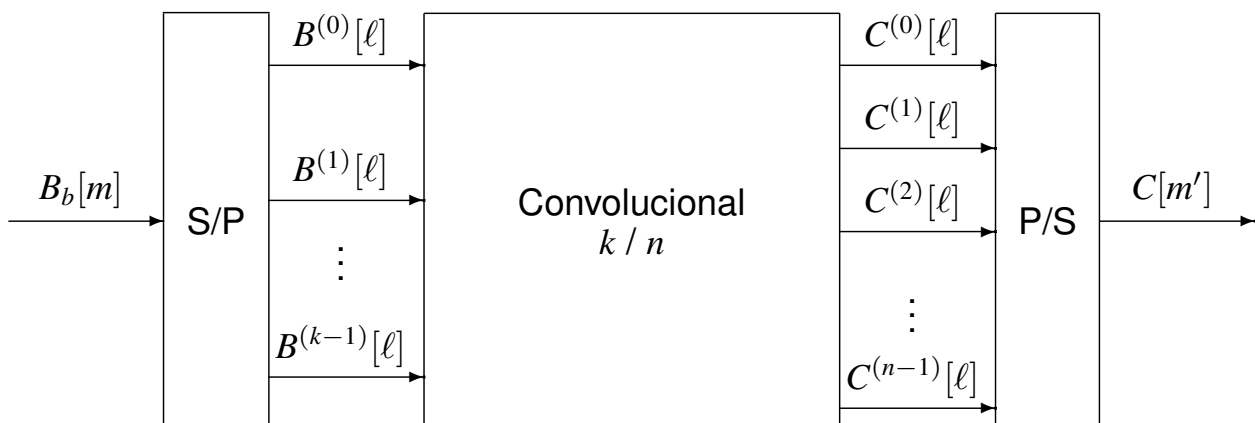
- c : máximo número de palabras cuyas secuencias de símbolos están a distancia mínima (euclídea) de la de una dada
- En general, d_{min}^E depende de la constelación y de la asignación binaria
 - ▶ Modulaciones binarias (constelación de 2 símbolos \mathbf{a}_0 y \mathbf{a}_1)

$$d^E(\mathbf{c}_i, \mathbf{c}_j) = d^E(\mathbf{a}_0, \mathbf{a}_1) \sqrt{d^H(\mathbf{c}_i, \mathbf{c}_j)}$$

$$P_e \approx c Q \left(\frac{d^E(\mathbf{a}_0, \mathbf{a}_1)}{2\sqrt{N_0/2}} \sqrt{d_{min}^H} \right)$$

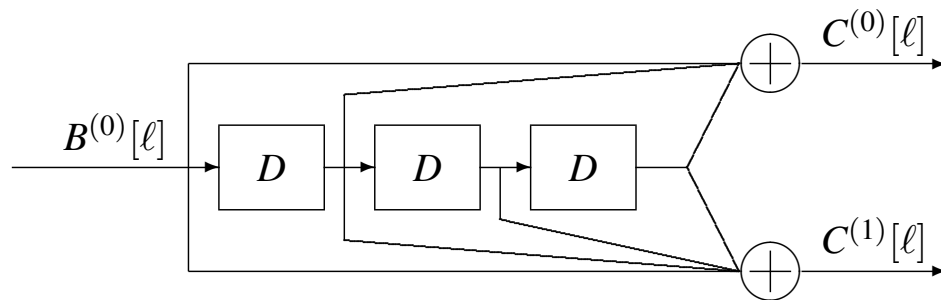
Códigos convolucionales

- Conversor serie / paralelo (S/P)
- Codificador convolucional
- Conversor paralelo / serie (P/S)



Códigos convolucionales

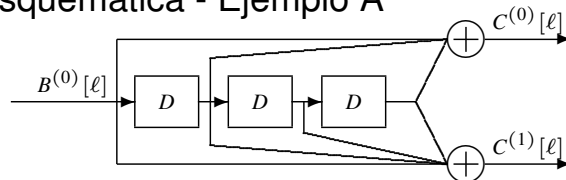
- Introducción de la redundancia mediante filtrado
 - ▶ Introducción de memoria
- Tasa $R = k/n$: banco de filtros con
 - ▶ k entradas
 - ▶ n salidas



- Notación:
 - ▶ Entradas: $B^{(i)}[\ell]$, con $i = 0, 1, \dots, k - 1$
 - ▶ Salidas: $C^{(j)}[\ell]$, con $j = 0, 1, \dots, n - 1$

Representaciones de los códigos convolucionales

- Representación esquemática - Ejemplo A



- Relación entrada salidas

$$C^{(0)}[\ell] = B^{(0)}[\ell] + B^{(0)}[\ell - 1] + B^{(0)}[\ell - 3]$$

$$C^{(1)}[\ell] = B^{(0)}[\ell] + B^{(0)}[\ell - 1] + B^{(0)}[\ell - 2] + B^{(0)}[\ell - 3]$$

- Representación de secuencias con polinomios en D - Transformada D

$$B^{(i)}(D) = \sum_{\ell} B^{(i)}[\ell] D^{\ell}$$

- ▶ Propiedad de la representación en D respecto a retardos

$$B^{(i)}[\ell - d] \leftrightarrow B^{(i)}(D) D^d$$

Representaciones de los códigos convolucionales (II)

- Notación mediante polinomios en D

$$C^{(0)}(D) = B^{(0)}(D) \{1 + D + D^3\}$$

$$C^{(1)}(D) = B^{(0)}(D) \{1 + D + D^2 + D^3\}$$

- Notación matricial (polinomios):

$$C(D)_{1 \times n} = B(D)_{1 \times k} G(D)_{k \times n}$$

- Matriz generadora de tamaño $k \times n$

- ▶ Cada elemento es un polinomio en D
- ▶ Elemento fila i columna j : contribución a la salida j -ésima de la entrada i -ésima
- ▶ Ejemplos

- ★ Ejemplo anterior (A): $k = 1, n = 2$

$$G(D) = [1 + D + D^3, 1 + D + D^2 + D^3]_{1 \times 2}$$

- ★ Otro ejemplo (B): $k = 2, n = 3$

$$G(D) = \begin{bmatrix} 1 + D & D & D \\ D & 1 & D \end{bmatrix}_{2 \times 3}$$

Paso a representación esquemática - Ejemplo B

$$G(D) = \begin{bmatrix} 1 + D & D & D \\ D & 1 & D \end{bmatrix}_{2 \times 3}$$

- Número de entradas del banco de filtros:

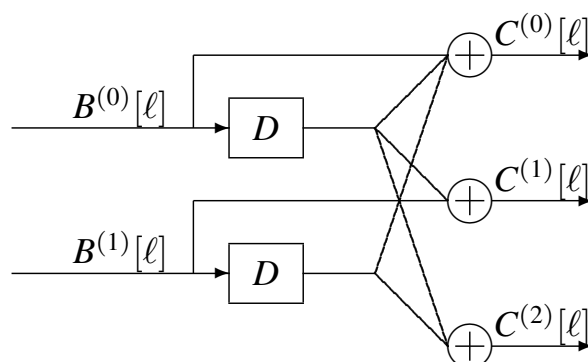
- ▶ Número de filas de la matriz $G(D)$: $k = 2$

- Número de salidas del banco de filtros:

- ▶ Número de columnas de la matriz $G(D)$: $n = 3$

- Número de memorias (elementos de retardo) de cada entrada:

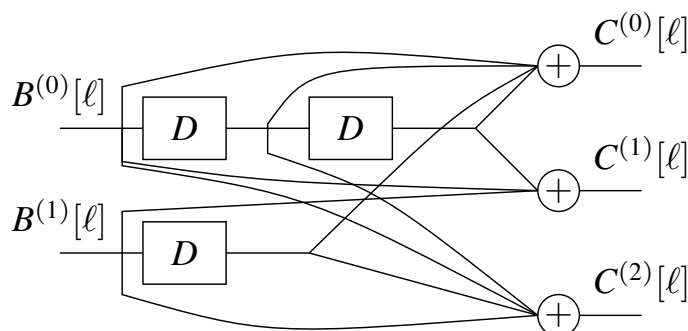
- ▶ Máximo grado de los polinomios en su correspondiente fila: $M^{(0)} = 1, M^{(1)} = 1$



Paso a representación esquemática - Ejemplo C

$$G(D) = \begin{bmatrix} 1 + D + D^2 & 1 + D^2 & 1 + D \\ D & 1 & 1 + D \end{bmatrix}_{2 \times 3}$$

- Número de entradas del banco de filtros:
 - ▶ Número de filas de la matriz $G(D)$: $k = 2$
- Número de salidas del banco de filtros:
 - ▶ Número de columnas de la matriz $G(D)$: $n = 3$
- Número de memorias (elementos de retardo) de cada entrada:
 - ▶ Máximo grado de los polinomios en su correspondiente fila: $M^{(0)} = 2, M^{(1)} = 1$



Parámetros de interés

- Memoria total del código: M_t
 - ▶ Número total de unidades de retardo (memorias)

$$M_t = \sum_{i=0}^{k-1} M^{(i)}$$

- ▶ Memoria de la entrada i -ésima:

$$M^{(i)} = \max_j \text{grado}(g_{i,j}(D))$$

- Longitud de restricción: K
 - ▶ Máxima longitud de la respuesta al impulso del codificador (máximo número de instantes de tiempo en los que un bit afecta a la salida del codificador)

$$K = 1 + \max_{i,j} \text{grado}(g_{i,j}(D)) = 1 + K_E$$

- ▶ K_E : memoria de la respuesta al impulso del codificador
- ▶ En general las prestaciones aumentan con K

Códigos sistemáticos

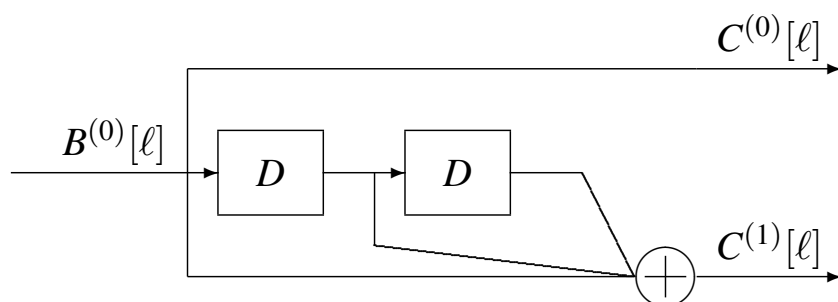
- Matriz de generación

$$G(D) = [I_k \ P(D)]$$

$$G(D) = [P(D) \ | \ I_k]$$

- ▶ Las entradas se “copian” en algunas de las salidas
- ▶ Ejemplo (D)

$$G(D) = [1, \ 1 + D + D^2]$$



Códigos sistemáticos - otro ejemplo (E)

$$G(D) = \left[\begin{array}{ccc|cc} 1 + D + D^2 & 1 + D^2 & 1 + D & 1 & 0 \\ D & 1 & 1 + D & 0 & 1 \end{array} \right]$$

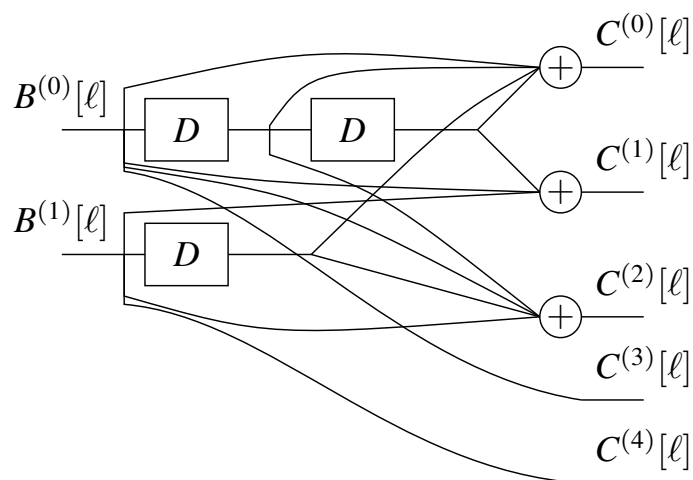


Diagrama de rejilla

- Definición del estado

- ▶ Contenido de las memorias del codificador

$$\psi[\ell] = [B^{(0)}[\ell-1], \dots, B^{(0)}[\ell-M^{(0)}], \dots, B^{(k-1)}[\ell-1], \dots, B^{(k-1)}[\ell-M^{(k-1)}]]$$

- Etiquetado de la rejilla

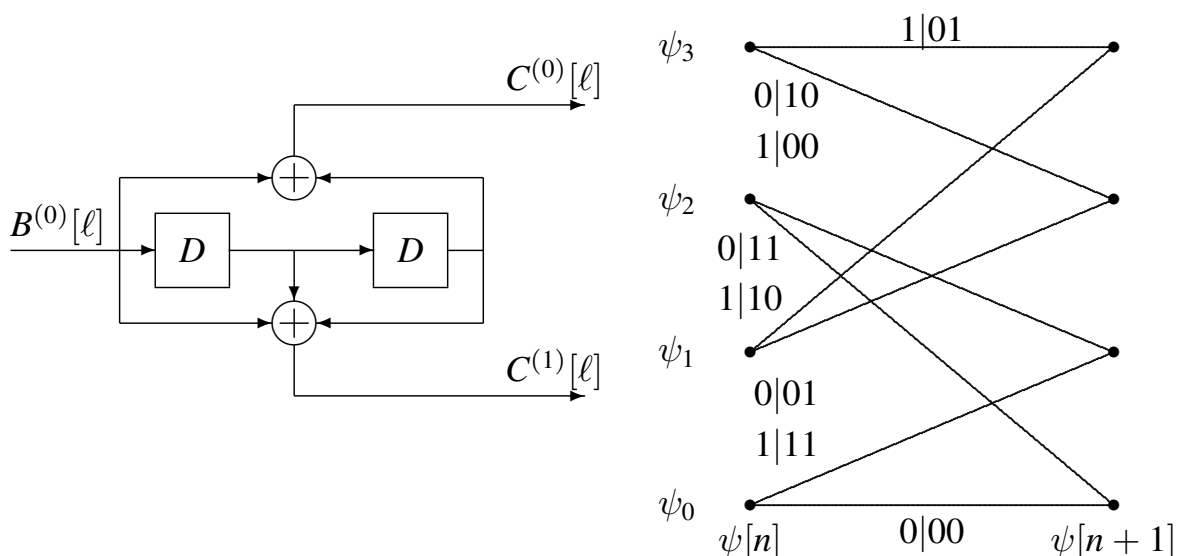
- ▶ Bits a la entrada del codificador (sin codificar)
- ▶ Bits a la salida del codificador (codificados)

$$B^{(0)}[\ell], B^{(1)}[\ell], \dots, B^{(k-1)}[\ell] \left| C^{(0)}[\ell], C^{(1)}[\ell], \dots, C^{(n-1)}[\ell] \right.$$

Ejemplo - Convolutacional F

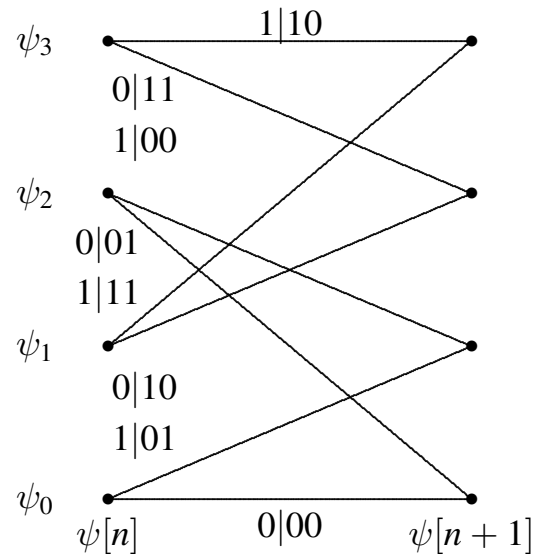
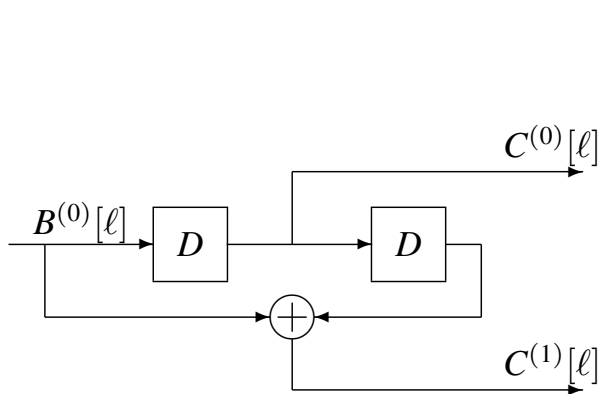
- Estado: $\psi[\ell] = [B^{(0)}[\ell-1], B^{(0)}[\ell-2]]$

- Estados: $\psi_0 = [0, 0], \psi_1 = [1, 0], \psi_2 = [0, 1], \psi_3 = [1, 1]$



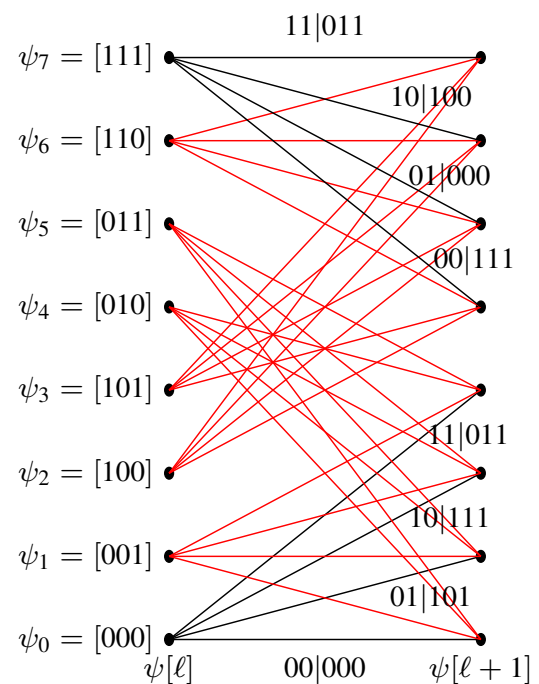
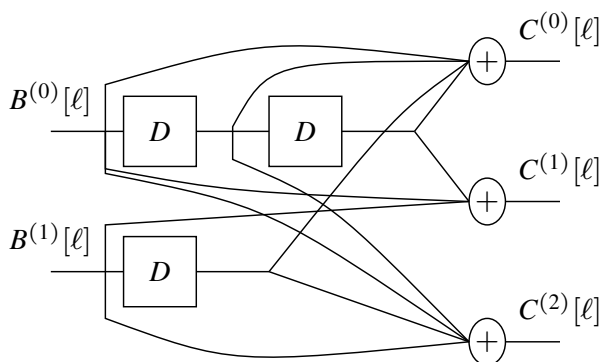
Ejemplo - Convolutacional G

- Estado: $\psi[\ell] = [B^{(0)}[\ell - 1], B^{(0)}[\ell - 2]]$
- Estados: $\psi_0 = [0, 0], \psi_1 = [1, 0], \psi_2 = [0, 1], \psi_3 = [1, 1]$



Ejemplo - Convolutacional C

- Estado: $\psi[\ell] = [B^{(0)}[\ell - 1], B^{(0)}[\ell - 2], B^{(1)}[\ell - 1]]$



Cabecera y número de transiciones en la rejilla

- Codificación de L bloques de k bits ($k \times L$ bits)
 - ▶ Conversión serie / paralelo
 - ★ L instantes discretos para $B^{(i)}[\ell], \ell \in \{0, 1, \dots, L-1\}$
- Longitud de los bits codificados con información: $n \times L$
- Cabecera para inicializar el codificador
 - ▶ $k \times K_E$ ceros
 - ★ Ceros en todas las k entradas para K_E instantes discretos
 - ★ Estado inicial: $\psi[0] = \psi_0 = [0, 0, \dots, 0]$
 - ★ Estado final: $\psi[L + K_E] = \psi_0 = [0, 0, \dots, 0]$
- Número de transiciones sobre la rejilla

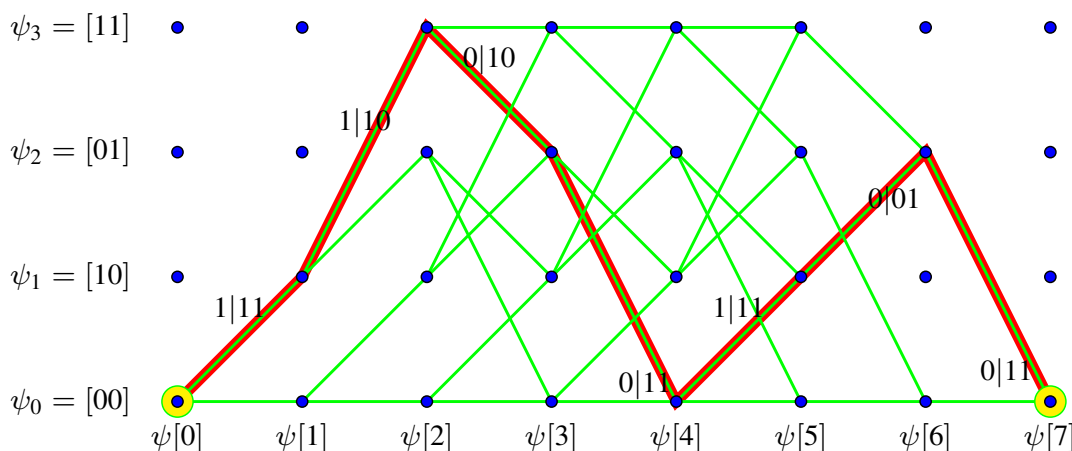
$$L + K_E \text{ transiciones}$$

- Longitud de bits codificados para $k \times L$ bits de información

$$\underbrace{n \times L}_{\text{bits info}} + \underbrace{n \times K_E}_{\text{bits cabecera}} = (L + K_E)n$$

Secuencia de bits : camino a través de la rejilla

- Convolutacional F : $k = 1, n = 2, K_E = 2$
 - ▶ Cabecera de $k \times K_E$ ceros: 00
- Bloque de $k \times L$ bits, con $L = 5$: $2^{k \times L} = 2^5 = 32$ caminos
 - ▶ Ejemplo: $B[m] = 11001$



- Secuencia codificada: $n \times (L + K_E)$ bits

$$C[m'] = 11\ 10\ 10\ 11\ 11\ 01\ 11$$

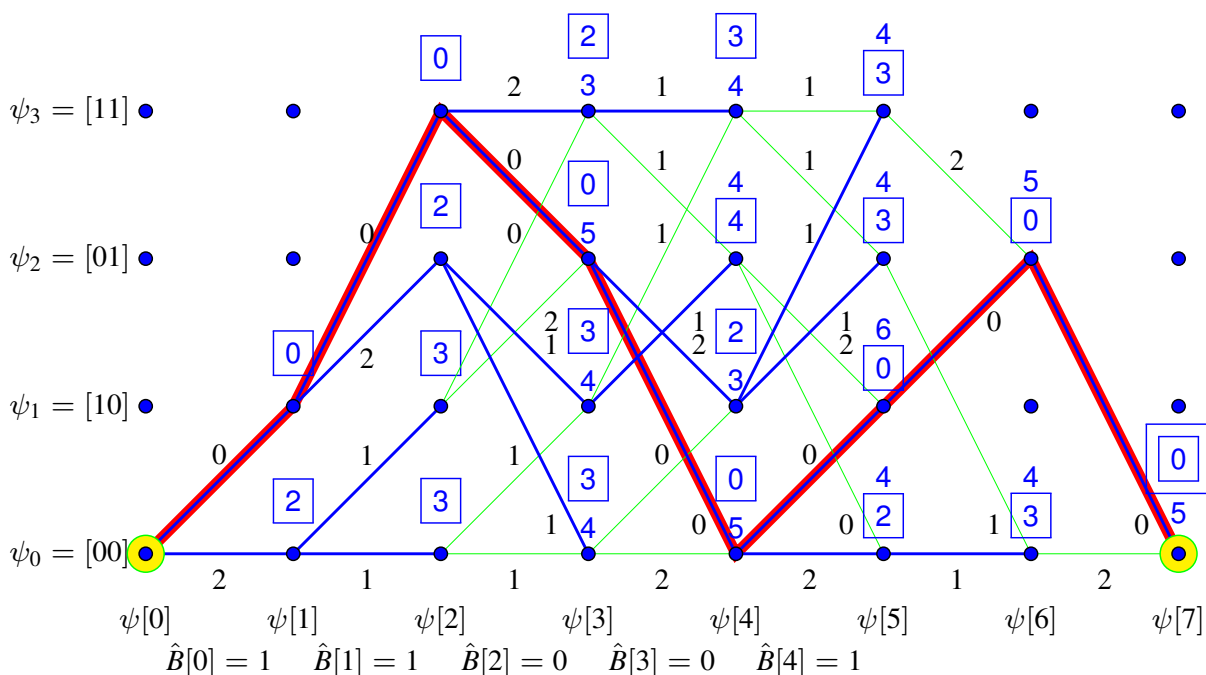
Decodificación - Algoritmo de Viterbi

- Recuperación de la secuencia más verosímil
- Estados inicial/final
 - ▶ Cabecera de referencia (habitualmente ceros “*bit flushing*”)
- Salida dura: observación de bits decididos
 - ▶ Secuencia con el menor número de bits codificados distintos a la observación
 - ▶ Métrica de rama: distancia de Hamming con la observación
- Salida blanda: observación de $q[n]$
 - ▶ Secuencia cuyos símbolos asociados están a la menor distancia euclídea de la observación
 - ▶ Métrica de rama: $|q[n] - A_i[n]|^2$
 - ★ Hay que tener en cuenta la constelación que se utiliza para hacer la conversión de las etiquetas de la rejilla básica a símbolos de la constelación ($A_i[n]$)
 - ▶ Mejores prestaciones con salida blanda

Decodificación con salida dura (Conv. F)

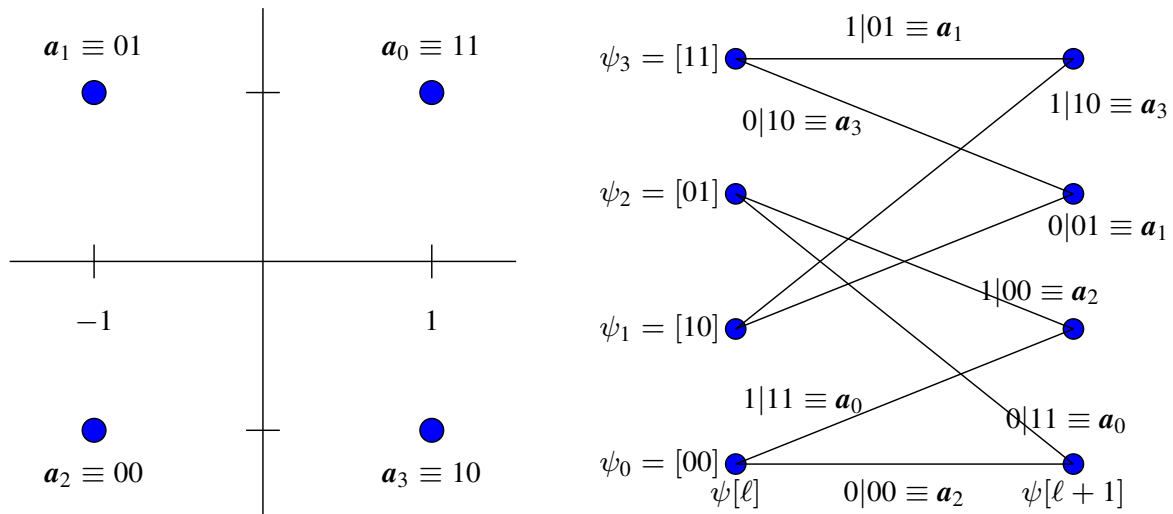
- Secuencia recibida

$$R[m'] = 11\ 10\ 10\ 11\ 11\ 01\ 11$$



Decodificación con salida blanda - Métrica (Conv. F)

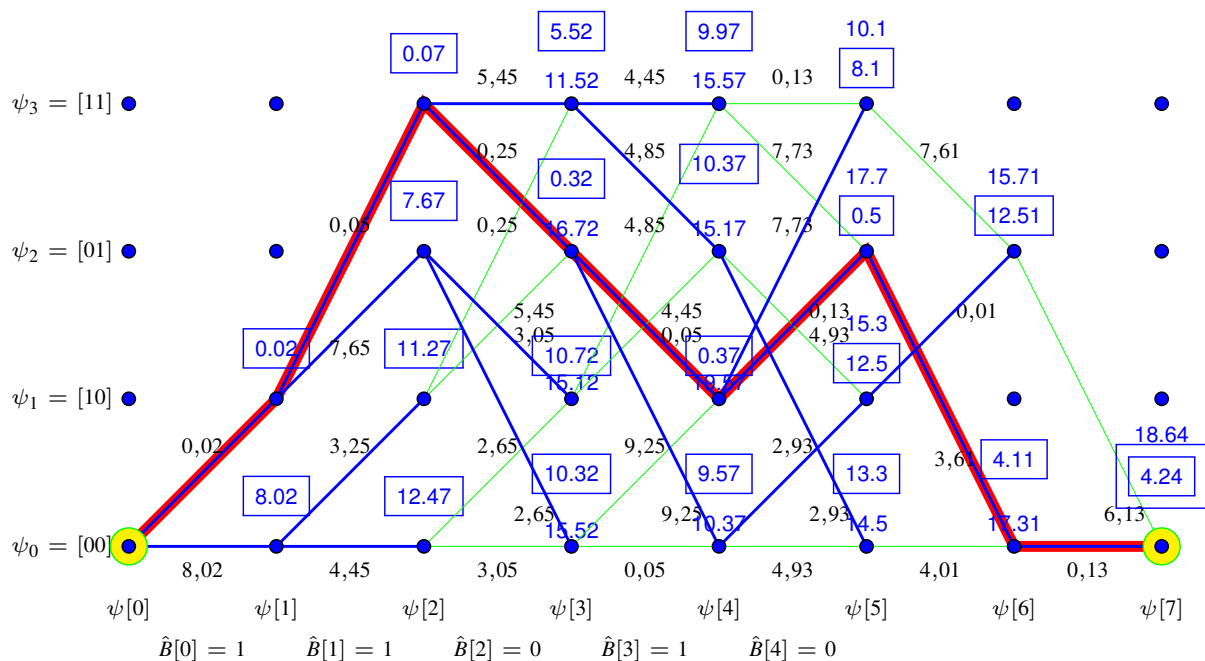
- Dependencia de la constelación y asignación binaria
- Ejemplo 4-QAM



Decodificación con salida blanda - Ejemplo

- Secuencia recibida

$$q[m] = \begin{bmatrix} +1,1 \\ +0,9 \end{bmatrix} \begin{bmatrix} +1,1 \\ -0,8 \end{bmatrix} \begin{bmatrix} +0,7 \\ -0,6 \end{bmatrix} \begin{bmatrix} -1,2 \\ -1,1 \end{bmatrix} \begin{bmatrix} -0,7 \\ +1,2 \end{bmatrix} \begin{bmatrix} -0,9 \\ +1,0 \end{bmatrix} \begin{bmatrix} +0,8 \\ +0,7 \end{bmatrix}$$



Prestaciones

- Salida dura

$$P_e \approx c \sum_{e=t+1}^{nz} \binom{nz}{e} \varepsilon^e (1 - \varepsilon)^{nz-e}$$

- ▶ D_{min}^H : mínima distancia de Hamming entre salidas para secuencias distintas
- ▶ z : longitud del evento erróneo de distancia mínima
- ▶ $t = \left\lfloor \frac{D_{min}^H - 1}{2} \right\rfloor$ (capacidad de corrección sobre nz bits)
- ▶ ε : probabilidad de error de bit del sistema (BER)

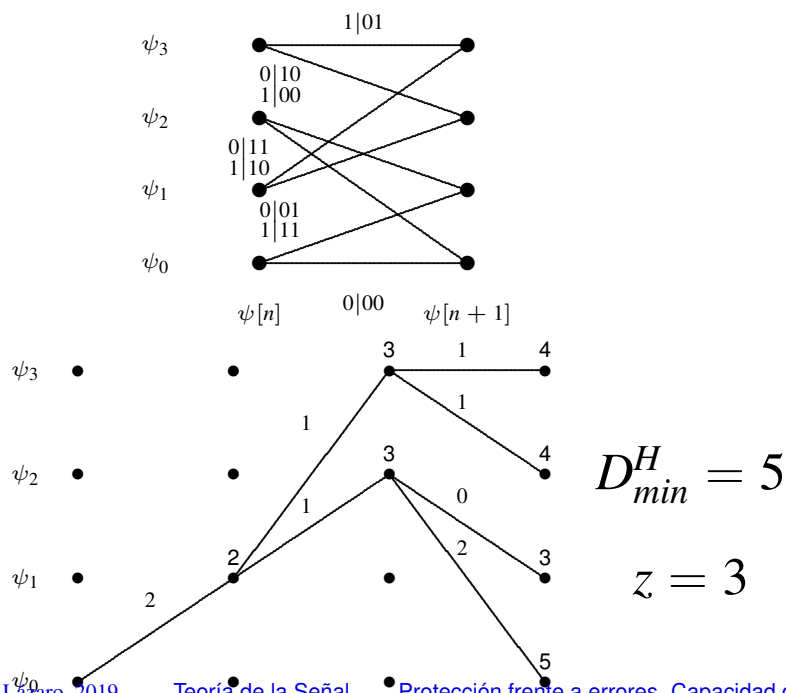
- Salida blanda

$$P_e \approx c Q \left(\frac{D_{min}^E}{2\sqrt{N_0/2}} \right)$$

- ▶ D_{min}^E : mínima distancia euclídea entre salidas para secuencias distintas

Calculo de D_{min}^H

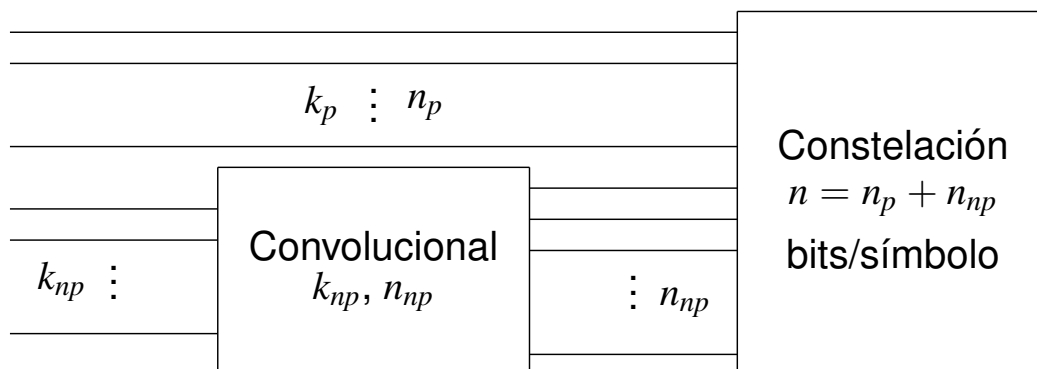
- Comparación con la secuencia de todo zeros
 - ▶ Métrica de rama: número de unos en los n bits codificados
- Puede usarse Viterbi para calcular distancias sobre sucesos erróneos



Códigos TCM (*Trellis Coded Modulation*)

- Diseño conjunto de un codificador convolucional y del codificador del transmisor (asignación binaria en la constelación del sistema)
- Uso eficiente del ancho de banda disponible en el canal
 - ▶ Introducción de redundancia aumentando el orden de la constelación
 - ★ El orden de ajusta al tamaño del codificador (número de salidas)
- Reducción de la redundancia mediante “transiciones en paralelo”
 - ▶ Protección aumentando la distancia entre símbolos

Códigos TCM - Definiciones



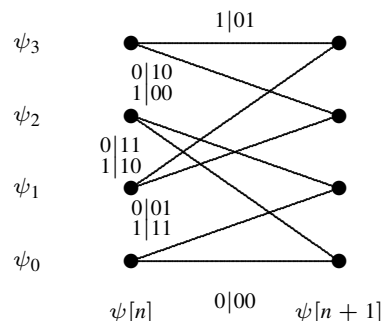
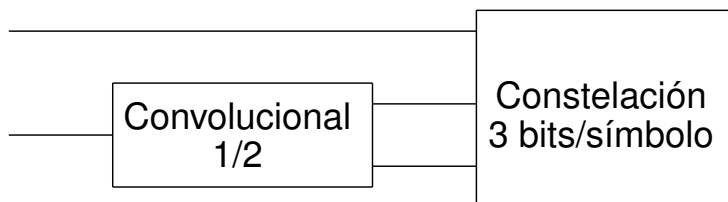
- k entradas
 - ▶ k_p “transiciones paralelo” (sin protección)
 - ▶ k_{np} “transiciones no paralelo” (protegidas)
- n salidas
 - ▶ $n_p = k_p$ “transiciones paralelo” (sin codificar)
 - ▶ n_{np} “transiciones no paralelo” (codificadas)
- Constelación: 2^n símbolos ($n = n_p + n_{np}$ bits/símbolo)

Códigos TCM - Asignación binaria

- Reglas de diseño: reglas de Ungerboeck
 - ▶ Asignación binaria para minimizar la probabilidad de error
 - ▶ División de la constelación + asignación binaria
- La constelación se divide en sub-constelaciones
 - ▶ División secuencial en dos partes aumentando la distancia mínima en cada paso
 - ▶ Número final de sub-constelaciones: $2^{n_{np}}$
 - ▶ Número final de símbolos por sub-constelación: 2^{n_p}
- Asignación binaria
 - ▶ Transiciones paralelo: selección de un símbolo en una sub-constelación
 - ★ Protección “física” sobre la constelación (asignación de Gray)
 - ▶ Transiciones no paralelo: selección de una sub-constelación
 - ★ Protección mediante el codificador convolucional
 - ★ Máxima distancia: ramas que salen o llegan al mismo estado
 - ★ Equiprobabilidad en la asignación de símbolos (sub-constelaciones)

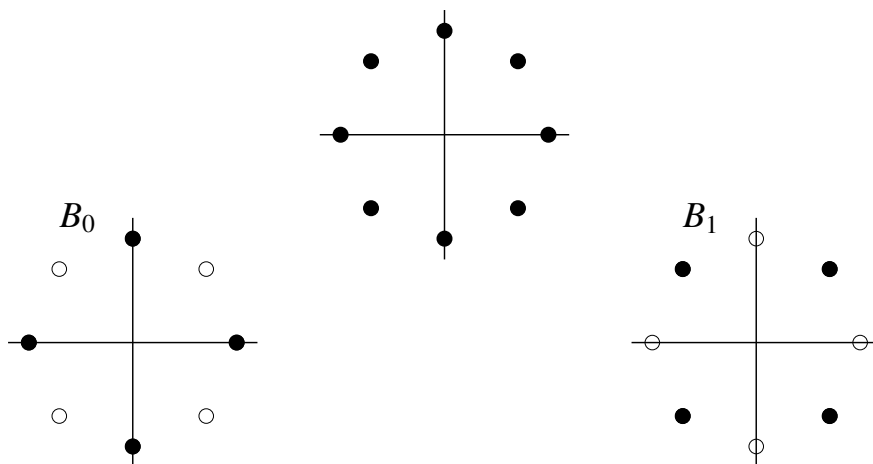
Ejemplo: $k_p = 1, k_{np} = 1, n_{np} = 2, 8\text{-PSK}$

- Tasa de codificación $R = 2/3$
 - ▶ Una transición paralelo
 - ▶ Codificador convolucional con tasa $1/2$
 - ★ Convolucional del Ejemplo D

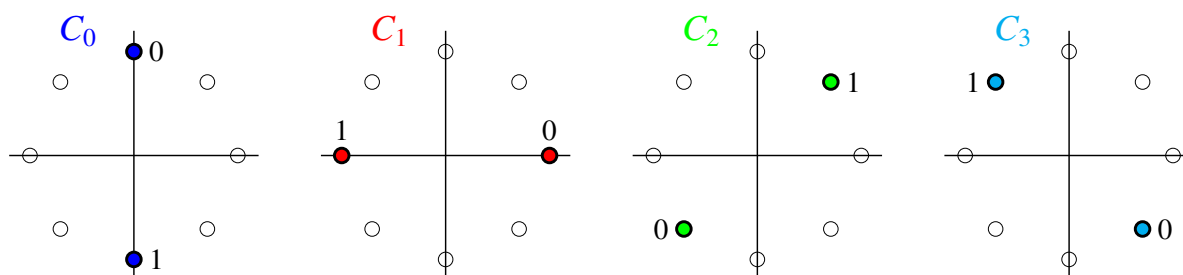


- Constelación con 3 bits por símbolo: 8 símbolos
 - ▶ 8-PSK

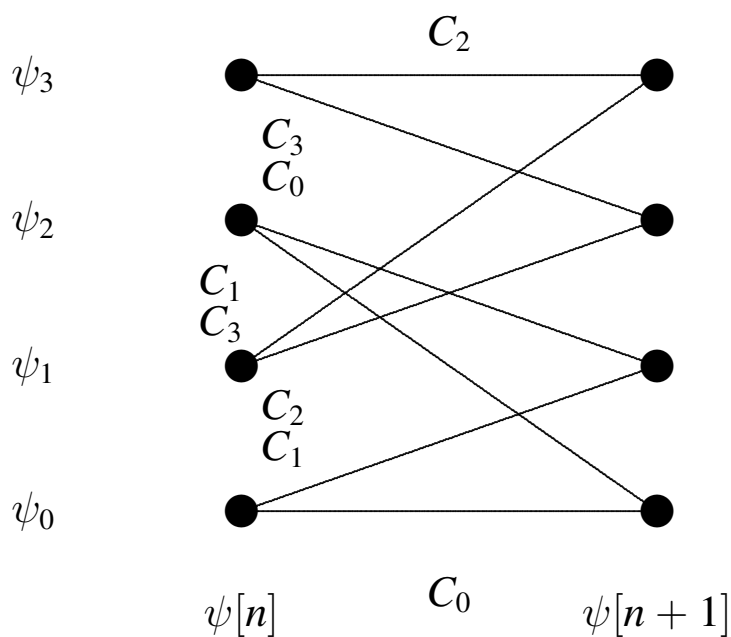
División de la constelación - Ejemplo



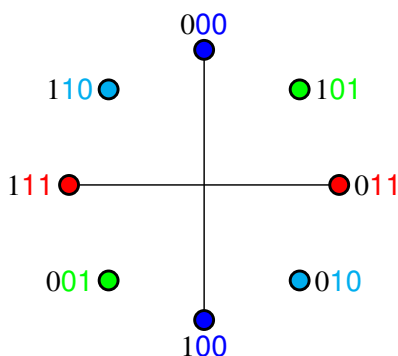
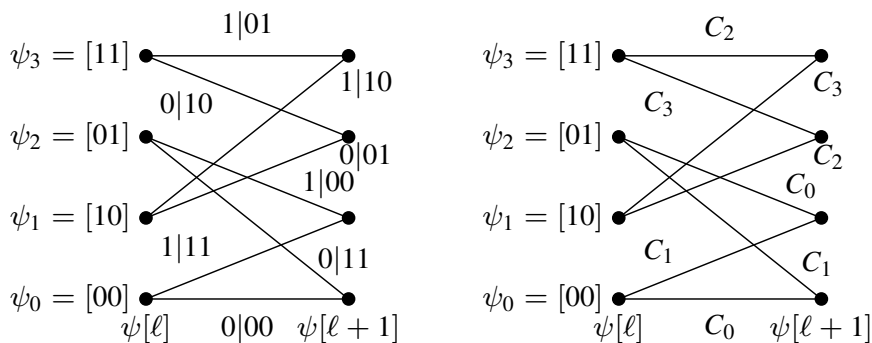
División final + Asignación del bit de la transición no paralelo



Asignación en las transiciones no paralelo - Ejemplo



Asignación final - Ejemplo (convolucional D)



Prestaciones

- Mínima distancia del código - El mínimo de:
 - ▶ Mínima distancia en paralelo
 - ★ Ejemplo (for normalized constellation): $d_p = 2$
 - Mínima distancia entre puntos en las sub-constelaciones
 - ▶ Mínima distancia no paralela
 - ★ Mínima distancia para el convolucional medida sobre la asignación de sub-constelaciones (en la rejilla)
 - ★ Ejemplo

$$d_{np} = \sqrt{d^2(C_0, C_1) + d^2(C_0, C_2) + d^2(C_0, C_1)} = 2,14$$

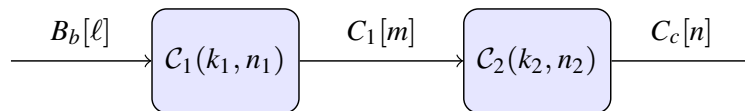
- Ganancia de codificación
 - ▶ Comparación con la constelación equivalente sin codificación de canal
 - ▶ Ejemplo: constelación de referencia 4-PSK

$$G = \frac{E_b^{SC}}{E_b^{TCM}} \frac{(d_{min}^{TCM})^2}{(d_{min}^{SC})^2} = 2 \text{ (3 dB)}$$

Códigos concatenados

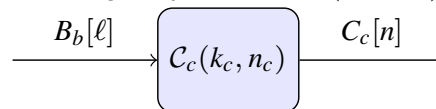
- Un código complejo puede en ocasiones obtenerse mediante la concatenación de dos códigos más simples
 - ▶ Concatenación en serie de dos códigos
 - ▶ Código entrada (interno) $\mathcal{C}_1(k_1, n_1)$, tasa $R_1 = k_1/n_1$
 - ▶ Código salida (externo) $\mathcal{C}_2(k_2, n_2)$, tasa $R_2 = k_2/n_2$
 - ▶ Tasa del código concatenado

$$R_c = \frac{k_1 k_2}{n_1 n_2} = R_1 \times R_2$$



- Relación de tamaños más habituales

- ▶ Caso $n_1/k_2 = c \in \mathbb{Z}$
 - ★ En este caso hay un código equivalente $\mathcal{C}_c(k_1, c n_2)$
- ▶ Caso $k_2/n_1 = c \in \mathbb{Z}$
 - ★ En este caso hay un código equivalente $\mathcal{C}_c(c k_1, n_2)$



Ejemplo con códigos bloque

- Códigos $\mathcal{C}_1(2, 3)$, $\mathcal{C}_2(3, 6)$

$$\mathbf{G}_1 = \begin{bmatrix} 101 \\ 011 \end{bmatrix} \equiv \begin{array}{|c|c|c|} \hline i & \mathbf{b}_i & \mathbf{c}_i \\ \hline 0 & 00 & 000 \\ \hline 1 & 01 & 011 \\ \hline 2 & 10 & 101 \\ \hline 3 & 11 & 110 \\ \hline \end{array}$$

$$\mathbf{G}_2 = \begin{bmatrix} 100011 \\ 010101 \\ 001111 \end{bmatrix} \equiv \begin{array}{|c|c|c|} \hline i & \mathbf{b}_i & \mathbf{c}_i \\ \hline 0 & 000 & 000000 \\ \hline 1 & 001 & 001111 \\ \hline 2 & 010 & 010101 \\ \hline 3 & 011 & 011010 \\ \hline 4 & 100 & 100011 \\ \hline 5 & 101 & 101100 \\ \hline 6 & 110 & 110110 \\ \hline 7 & 111 & 111001 \\ \hline \end{array}$$

i	\mathbf{b}_i	\mathbf{c}_i
0	000	000000
1	001	001111
2	010	010101
3	011	011010
4	100	100011
5	101	101100
6	110	110110
7	111	111001

- Código concatenado $\mathcal{C}_c(2, 6) = \mathcal{C}_1(2, 3) - \mathcal{C}_2(3, 6)$

$$\begin{array}{|c|c|c|} \hline i & \mathbf{b}_i & \mathbf{c}_i \\ \hline 0 & 00 & 000000 \\ \hline 1 & 01 & 011010 \\ \hline 2 & 10 & 101100 \\ \hline 3 & 11 & 110110 \\ \hline \end{array} \equiv \mathbf{G}_c = \begin{bmatrix} 101100 \\ 011010 \end{bmatrix}$$

$$\mathbf{G}_c = \mathbf{G}_1 \times \mathbf{G}_2, \text{ ya que } k_2 = n_1$$

Ejemplo con códigos bloque (II)

- Código concatenado $\mathcal{C}_c(3, 12) = \mathcal{C}_2(3, 6) - \mathcal{C}_2(3, 6)$

i	b_i	c_i	
0	000	000000	000000
1	001	001111	111001
2	010	010101	101100
3	011	011010	010101
4	100	100011	011010
5	101	101100	100011
6	110	110110	110110
7	111	111001	001111

$$\equiv \mathbf{G}_c = \begin{bmatrix} 100011011010 \\ 010101101100 \\ 001111111001 \end{bmatrix}$$

Ejemplo con códigos convolucionales

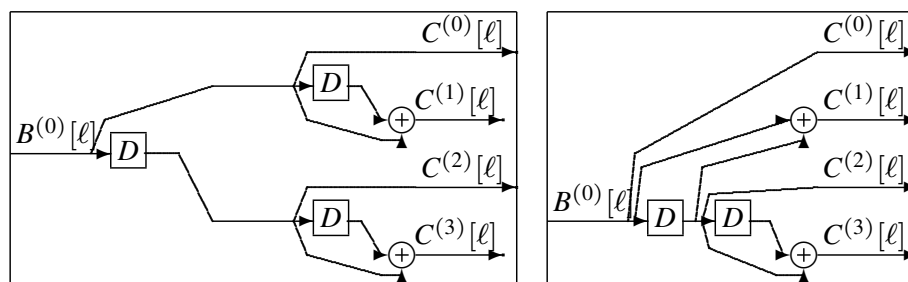
- Códigos $\mathcal{C}_1(1, 2), \mathcal{C}_2(2, 4)$

$$\mathbf{G}_1(D) = [1, D], \quad \mathbf{G}_2(D) = \begin{bmatrix} 1, & 1 + D, & 0, & 0 \\ 0, & 0, & 1, & 1 + D \end{bmatrix}$$

- ▶ El código \mathcal{C}_2 se implementa con dos códigos $\mathbf{G}'_2[1, 1 + D]$ en paralelo

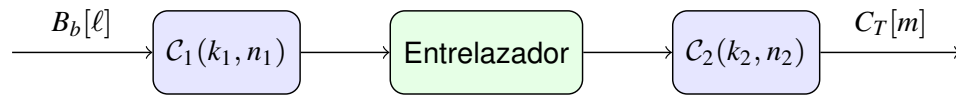
- Código concatenado $\mathcal{C}_c(1, 4) = \mathcal{C}_1(1, 2) - \mathcal{C}_2(2, 4)$

$$\mathbf{G}_c = [1, \quad 1 + D, \quad D, \quad D + D^2]$$

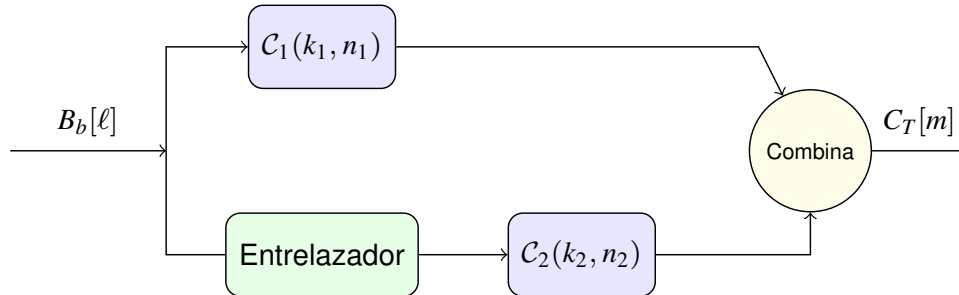


Turbo códigos

- Concatenación de códigos utilizando entrelazadores



Concatenación Serie

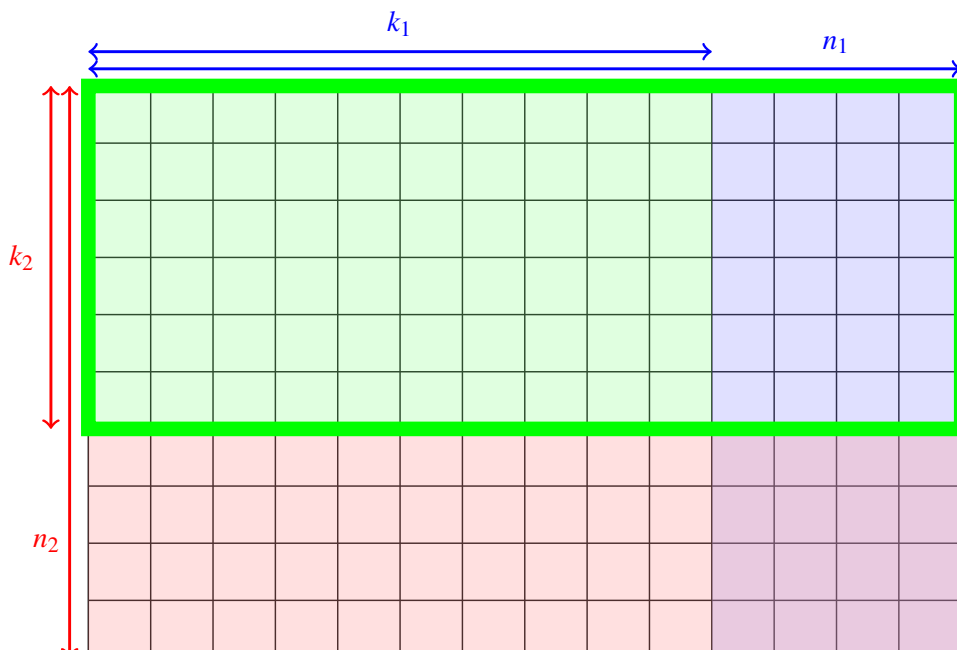


Concatenación Paralelo

- ▶ Codificadores
 - ★ Códigos convolucionales sistemáticos y recursivos (RSC)
- ▶ Entrelazador: reordena los bits
 - ★ Errores de un codificador en distintas palabras código del otro codificador
 - ★ Determina las prestaciones (se usan entrelazadores pseudo-aleatorios)

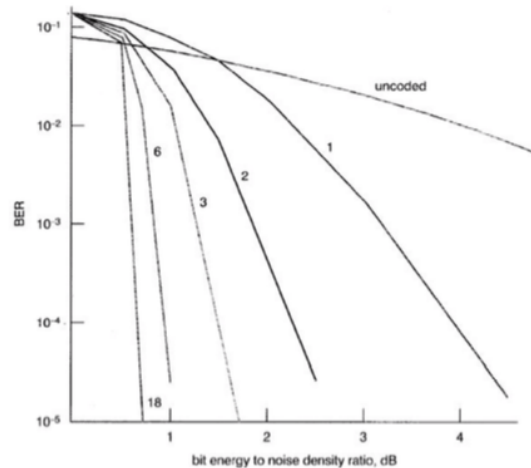
Entrelazador - Ejemplo concatenación serie ($k_2 \times n_1$)

- Entrelazador bloque
 - ▶ Entrada de bits escribiendo por filas
 - ▶ Salida de bits leyendo por columnas



Decodificación iterativa

- Decodificación iterativa (algoritmo BCJR)
 - ▶ Cada codificador proporciona su salida blanda
 - ★ LLR: Log-Likelihood Ratio
 - ▶ Intercambio de esta información entre codificadores
- Prestaciones: cercanas al límite de Shannon!!!



Códigos LDPC

LDPC: *Low Density Parity Check*

- Códigos bloque lineales
 - ▶ Códigos de gran tamaño
 - ★ Ejemplo: $\mathcal{C}(5000, 10000)$
 - ▶ Matriz de chequeo de paridad dispersa (pocos 1s)
- Representación mediante un grafo bipartido (grafo de Tanner)
 - ▶ Dos tipos de nodos
 - ★ Nodos de bits
 - ★ Nodos de chequeo
 - ▶ Aplicación del principio Turbo: decodificación iterativa
 - ★ Salidas blandas en cada iteración
 - ★ Algoritmos iterativos de tipo “*belief propagation*” (BCJR, MAP, SOVA)
- Excelentes prestaciones
 - ▶ Estado del arte actual, junto con los turbo códigos