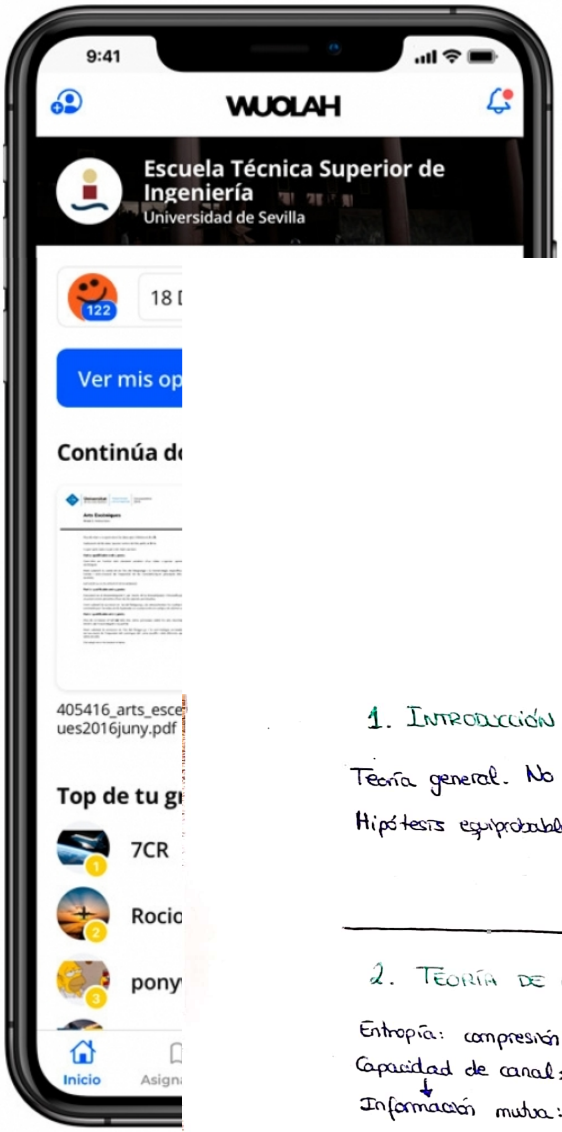
ResumenTeoria.pdf

MRA_Engineer

Comunicaciones Digitales

3º Grado en Ingeniería de las Tecnologías de Telecomunicación

Escuela Politécnica Superior
Universidad de Alcalá



1. INTRODUCCIÓN

Teoría general. No entra en examen.

Hipótesis equiprobable (no real) : $P(b=0) = P(b=1) = 0'5$ ⟶ misma probabilidad de bit '0' y '1'

---

2. TEORÍA DE LA INFORMACIÓN

Entropía: compresión de canal ⟶ $H(X)$ = Entropía

Capacidad de canal: velocidad de transmisión max. teniendo en cuenta el canal } límites: calidad, error

Información mutua: $I(X;Y)$: dos variables aleatorias porque el canal tiene entrada y salida.

$$P_e(s) \propto Q\left(\sqrt{(\cdot)\frac{E_s}{N_0}}\right) \longrightarrow \text{Si } R\uparrow \Rightarrow P_e\uparrow$$

(incremento velocidad de transmisión)

↓

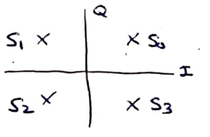
(incremento prob. de error)

↓

$\boxed{\text{Shannon}}$ No es totalmente cierto

ENTROPÍA

Símbolos :



$k=4$ (radix)

$$C = \{S_0, S_1, S_2 \dots S_{k-1}\}$$

$$P_{\text{símbolo}} = \frac{1}{k} = \frac{1}{4} = P(S=s_k) = p_k$$

Cantidad de información : $I(s_k) = \log_2\left(\frac{1}{p_k}\right)$ ⟶ Se incrementa la información cuantos más símbolos se transmitan

Si se envían siempre s_0 y de repente se envía S_1, ese S_1 aporta mucha información ⟵

Propiedad aditiva: $I(s_\ell, s_k) = I(s_\ell) + I(s_k)$

Entropía: promedio de contenido de información

$$H(\zeta) = \sum_{k=10}^{k-1} p_k \log_2 \left(\frac{1}{p_k}\right) \longrightarrow$$ $\log_2$ ya que la cantidad de información se mide en "**bits**"

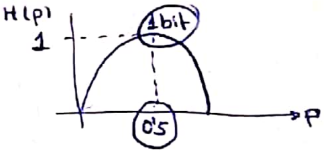
$$0 \leq \text{entropía} = H(\zeta) \leq \log_2(K)$$

Todo depende de la probabilidad de cada símbolo: EQUIPROBABILIDAD = MAX. ENTROPÍA

Ejemplo: $\{0, 1\}$

$\downarrow \quad \downarrow$

$p \quad 1-p$ = probabilidad



SOURCE CODING

La longitud depende de los codes que se asocien a los símbolos:

$S_0 = 001$
$S_1 = 1$
$S_2 = 10$

la longitud varía según nº de bits

Con la compresión se obtienen longitudes menores que si se codificara todo con un mismo número de bits.

Ejemplo: $K = 8 \longrightarrow$ 3 bits/símbolo

	p_k	$\bar{c}_k$
d_7	0	111
d_6	0	110
d_5	0	101
d_4	p_4	100
d_3	p_3	011
d_2	0	010
d_1	0	001
d_0	0	000

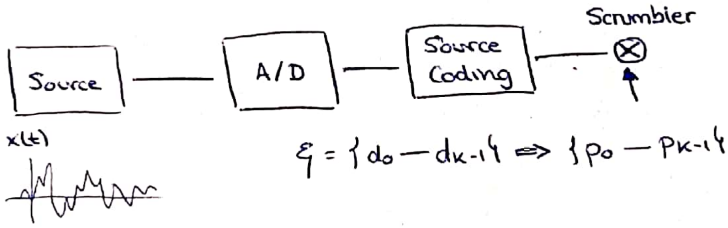
$\longrightarrow$ con 1 bit/símbolo es suficiente:

	$\bar{c}_k$
d_4	1
d_3	0

Entropía = 1 bit $\longrightarrow$ compresión
Teorema de Shannon

$\downarrow$

Esto es estúpido ya que solo se transmiten dos símbolos



$\xi = \{d_0 - d_{k-1}\} \Rightarrow \{p_0 - p_{k-1}\}$

longitud media: $\bar{L} = \sum_{k=0}^{k-1} p_k \cdot l_k \;\; \approx\!\longrightarrow H(\xi)$ nº medio bits/símbolo $\Rightarrow$ Entropía

l_k = longitud codeword correlación símbolo d_k

$d_k = b_0^{k} - b_{l_k-1}^{k}$

Eficiencia: cómo de buena es la codificación $\longrightarrow \eta = \dfrac{H(\xi)}{\bar{L}}$

El teorema de Shannon te dice que puedes comprimir hasta la entropía, pero no te dice como conseguirlo.

HUFFNAN CODING

Deden de tener diferente prefijo:

$d_j \rightarrow 0$
$d_e \rightarrow$ ~~01~~ debe empezar por 1 } Garantiza la invertibilidad.

- Codeword único
- Decodificación instantánea
- Longitud media : $H(\xi) \leq \bar{L} \leq H(\xi)+1$ → significa que perdemos un bit
- Desigualdad Kraft-McMillan

$$\left[\underbrace{H(\xi) = \bar{L}}_{\text{Codificación óptima}} \longrightarrow p_K = 2^{-l_k} \Rightarrow \eta = 100\% \right]$$

↳ probabilidades = 1/2, 1/4, 1/8, 1/16 ...

NOTA: se ordenan con orden de probabilidad decreciente
se agrupa de abajo a arriba.
Varianza alta = tenemos que esperar "mucho" para algunas palabras → velocidad de salida no uniforme
pequeña = todas las palabras de tamaño similar → velocidad uniforme

EXTENSIÓN DE FUENTE

Se puede agrupar como ξ^2 : segunda expansión de la fuente

$\xi = \{d_0, d_1, d_2 \ldots, d_7\}$ → todas las combinaciones de símbolos:
$d_0d_1, d_0d_2, d_0d_3 \ldots$
$d_1d_0, d_1d_2, \ldots$

$$\boxed{H(\xi^2) = 2 \cdot H(\xi)} \longrightarrow H(\xi^n) \leq \bar{L} \leq H(\xi^n)+1$$

$$\bar{L}_s = \frac{\bar{L}}{n}$$

$$\left[\begin{array}{l} H(\xi) \leq \bar{L}_s \leq H(\xi) + \frac{1}{n} \\ n \to \infty \Rightarrow \eta = 100\% \end{array} \right]$$

N símbolos : nº mensajes = $2^{N \cdot H(\xi)}$

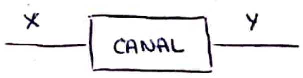
MUTUAL INFORMATION

1) **ENTROPÍA CONJUNTA**: a partir de dos variables aleatorias

$H(X,Y) \geq \max[H(X), H(Y)]$

$H(X,Y) \leq H(X) + H(Y)$

2) **ENTROPÍA CONDICIONAL**: más interesante, relaciona lo que envío y lo que llega.



$H(Y|Y) = \emptyset$

$H(Y|X) = H(Y)$ si X e Y son independientes

[Teorema de Bayes $\Rightarrow p(x,y) = p(Y|x) \cdot p(x)$] ⟶ repasar : esperanza $= E_x = \sum p(X=x) \cdot f(x)$

↓

si son x e y independientes : $p(x,y) = p(y) \cdot p(x)$

REGLA DE LA CADENA

$H(X,Y) = H(X) + H(Y|X)$

$H(Y|X) \leq H(Y)$

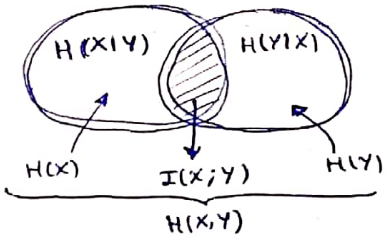
Con 3 variables aleatorias (Z = ruido) : $H(X,Y|Z) = H(X|Z) + H(Y|X,Z)$

3) **ENTROPÍA RELATIVA**: $H(x)$ es el nº de bits medio para representar la información en X.

4) **INFORMACIÓN MUTUA**:

$I(X;Y) = H(X) - H(X|Y) = I(Y;X)$ ⟶ simétrica!!

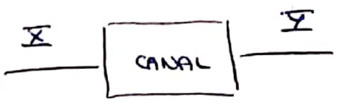
Puede dar $\emptyset$, si es así el canal no sirve para nada, perdemos la información.



$$\left\{ \begin{array}{l} \max: \quad H(X) = H(Y) \longrightarrow H(X|Y) = \emptyset \\ I(X;Y) \geq \emptyset \end{array} \right.$$

CANAL DISCRETO SIN MEMORIA

Tiene un alfabeto de entrada y otro de salida, donde cada "letra" tendrá probabilidad diferente



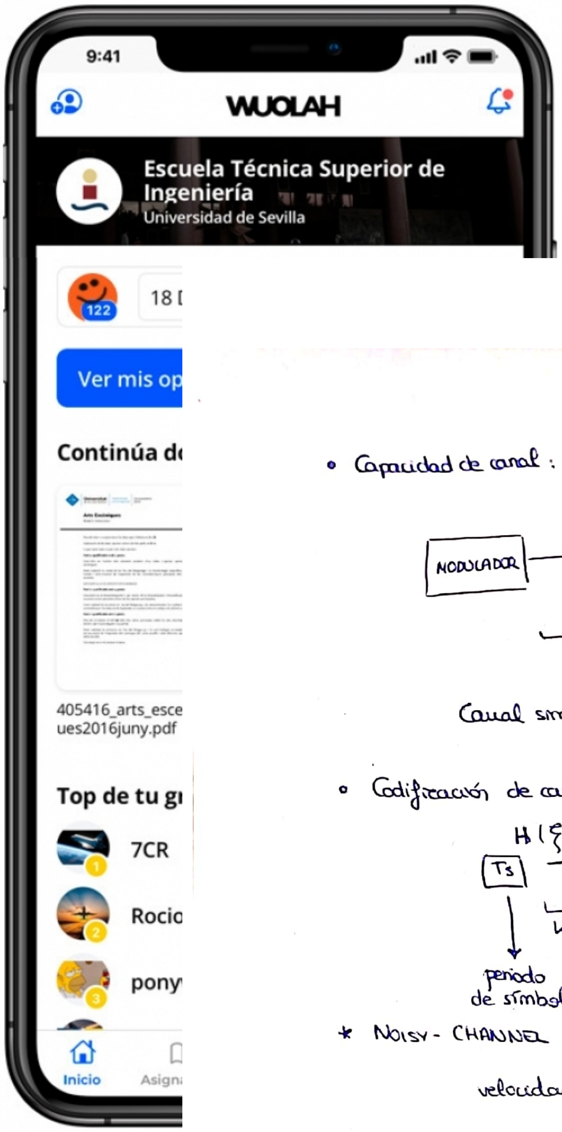
$\boxed{p(Y|X)}$

Se define el comportamiento del canal mediante una <u>matriz</u> <u>del</u> <u>canal</u> $(\overline{\overline{P}})$

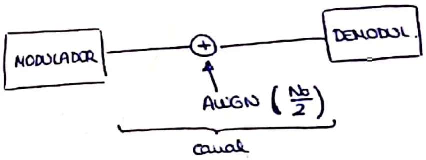
Canal simétrico:

$$\overline{\overline{P}} = \begin{array}{ccc} \left(\begin{matrix} 0'6 & 0'1 & 0'3 \\ 0'3 & 0'6 & 0'1 \\ 0'1 & 0'3 & 0'6 \end{matrix}\right) & \begin{matrix} =1 \\ =1 \\ =1 \end{matrix} \\ \begin{matrix} \shortparallel & \shortparallel & \shortparallel \\ 1 & 1 & 1 \end{matrix} & \end{array} \qquad \overline{\overline{P}} = \underbrace{\begin{array}{cc} \left(\begin{matrix} 0'6 & 0'3 & 0'1 \\ 0'3 & 0'6 & 0'1 \\ 0'1 & 0'3 & 0'6 \end{matrix}\right) & \begin{matrix} =1 \\ =1 \\ =1 \end{matrix} \\ \begin{matrix} \shortparallel & \neq & \neq \end{matrix} & \end{array}}_{\text{no simétrico}}$$

$$\overline{\overline{P}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \longrightarrow \text{caso óptimo}$$



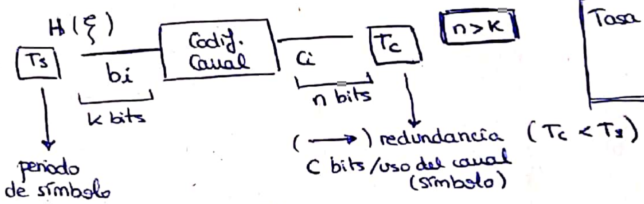
- Capacidad de canal : $C = \max_{p_x} [I(X;Y)]$ máximo de la información mutua



$\boxed{\text{depende únicamente de la matriz de canal : } \bar{\bar{P}}}$

Canal simétrico => símbolos equiprobables => $\bar{\bar{P}}$ simétrica

- Codificación de canal: permite "proteger" la información , añadiendo redundancia



$\boxed{\text{Tasa de codificación: } R = \frac{k}{n} < 1}$

* NOISY - CHANNEL CODING THEOREM :

velocidad de transmisión $= \frac{H(\xi)}{T_s}$ (bits/s)

tasa hipotética máxima $= \frac{C}{T_c}$ (bits/s)

$$\boxed{\frac{H(\xi)}{T_s} \leq \frac{C}{T_c}}$$

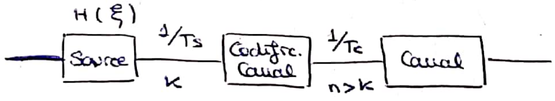
↓

Si esto se cumple se puede encontrar una codificación que garantice error de transmisión nulo.

$$\boxed{R = \frac{k}{n} = \frac{T_c}{T_s}}$$

capacidad ⟶ $I(X;Y) \longrightarrow \text{máx} = \boxed{C}$ bits/uso del canal

$C > H(\xi)\frac{k}{n} \Rightarrow P_e \rightarrow 0$



$R = \frac{k}{n} = \frac{T_c}{T_s}$.

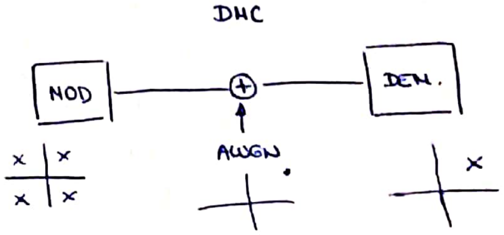
en el caso de matriz simétrica : $C = 1 - H(\rho)$

$\left[C = \log_2(\text{nº elementos}) - H(\rho)\right]$ ↳ = primera fila matriz

↓

más sencillo que calcular el máximo .

Entropia e Información mutua para RRW aleatorias continuas



- Entropía Diferencial / Continua

  variable aleatoria gaussiana: $h(X) = \frac{1}{2} \log_2 (2\pi e \sigma_X^2)$

NOTA: no se va a utilizar

AWGN CHANNEL CAPACITY THEOREM

$$SNR = \frac{S}{\sigma_N^2} \longrightarrow C = \frac{1}{2} \log_2 (1 + SNR) = \frac{1}{2} \log_2 \left(1 + \frac{S}{N \cdot B_0} \right)$$

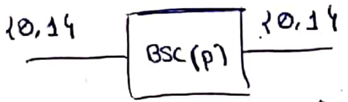
$$\boxed{\text{Límite absoluto} = C = B \cdot \log_2 \left(1 + \frac{S}{N_0 B} \right)}$$ (bits/s)

Shannon-Hartley

→ tasa de transmisión debe ser menor

PUENTE AL CODIFICADOR DE CANAL

Códifiacador de canal :

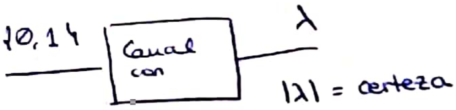


$$\bar{P} = \begin{pmatrix} 1-p & p \\ p & 1-p \end{pmatrix}$$

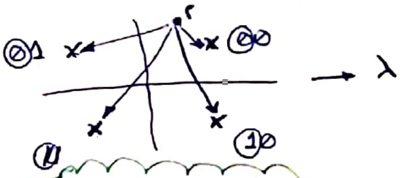
entrada
salida = discreta

HARD – DEMODULATION

LLR (log-likehood Ratios)



$|\lambda|$ = certeza



entrada = discreta
salida = continua

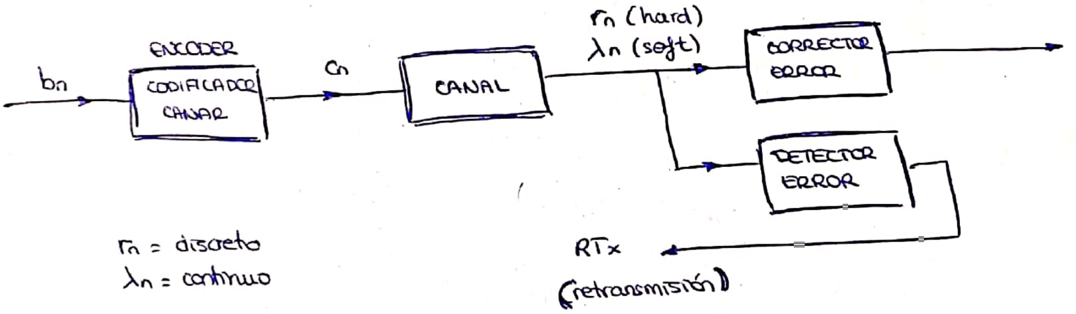
Demodulación a posterioria (ÓPTIMO)
Observando la salida (r)

$$\log\left(\frac{P(b_k=1|r)}{P(b_k=0|r)}\right)$$

SOFT – DEMODULATION

CODIFICACIÓN DE CANAL

BLOCK 3 : CHANNEL CODING



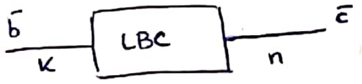
r_n = discreto
λ_n = continuo

\+ Distancia de Hamming : diferencia de bits

$\bar{c} = (1,1\ 1\ 0)$
$\hookrightarrow \bar{\omega}(\bar{c}) = 3$
$\bar{d} = (1010)$

$d_H(\bar{c},\bar{d}) = 1$ (solo 1 bit diferente)
$\bar{c}+\bar{d} = (0100)$ → coincide

BINARY LINEAR BLOCK CODES



$n > k$

$\underbrace{\bar{b} \in GF(2)^k}_{2^k}$ $\qquad$ $\underbrace{\bar{c} \in GF(2)^n}_{2^n}$

$2^n > 2^k$

subconjunto: $\underbrace{C(n,k)}_{\text{dim } k} \in GF(2)^n$
$\downarrow$
2^k

codificación
↓
correspondencia

n = 4 → 16 vectores pero necesito 8.

K=3

entrada:
(000) → (0000)
(001) → (0011)
(010) → (0101)
(011) → (0110)
(100) → (1001)
(101) → (1010)
(110) → (1100)
(111) → (1111)

→ no se utilizan todos los element.

código = conjunto
⇓
subespacio vectorial

Si llegan elementos que no están, sabemos que hay un error!!!.

REGLAS : × = AND
\+ = XOR

$\bar{c} \cdot \bar{h}_i^T = 0$ → Ecuación de chequeo / comprobación
se confirma la corrección del código generado.

$$\text{LBC} \begin{cases} \vec{\bar{G}} = \begin{pmatrix} \bar{g}_1 \\ \vdots \\ \bar{g}_K \end{pmatrix} \rightarrow \text{matriz generadora} \quad \underline{K \times N \ (k \times n)} \\ \bar{\bar{H}} = \begin{pmatrix} \bar{h}_1 \\ \vdots \\ \bar{h}_{n-k} \end{pmatrix} \rightarrow \text{matriz comprobadora} \quad \underline{(n-k) \times n} \end{cases}$$

$$\boxed{\begin{array}{l} \bar{c} \in C(n,K) \Longleftrightarrow \vec{c} = \bar{b} \cdot \bar{\bar{G}} \\ \qquad\qquad\qquad\quad \bar{b} \in GF(2)^K \\ \bar{c} \in C(n,K) \Longleftrightarrow \bar{c} \cdot \bar{H}^T = \vec{0} \\ \quad \hookrightarrow (\bar{c} \in GF(2)^n) \end{array}}$$

$\bar{\bar{G}} \cdot \bar{\bar{H}}^T = \bar{\bar{0}}$ → define mismo código binario

$\bar{\bar{G}}$ => da información sobre el <u>codificador</u>

$\bar{\bar{H}}$ => da información sobre el <u>decodificador</u>

En el caso anterior: $\bar{\bar{G}} = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$ → código de ~~generación~~ paridad par

$\bar{\bar{H}} = (1111)$

↓

$\vec{C} \cdot \bar{\bar{H}}^T = c_1 + c_2 + c_3 + c_4$ (+ = XOR!!) $= 0$ (OK)

$\neq 0$ → $\bar{\bar{G}}$ y $\bar{\bar{H}}$ definen diferentes códigos $\bar{\bar{E}}$

SISTEMATICIDAD:

$$\bar{\bar{G}}_S \left(\begin{array}{|ccc|c} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{array} \right)$$

↳ bit de chequeo

$c_1 = ((001) 1)$ $b_1 = (001)$

$c_2 = ((010) 1)$ $b_2 (010)$

NO SISTEMÁTICA:

$$\bar{G}_{NS} = \left(\begin{array}{|ccc|c} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{array} \right) \longrightarrow \text{define el mismo código}$$

¿Cómo obtener $\bar{\bar{G}}$ o $\bar{\bar{H}}$?

$\bar{\bar{G}}_s \longleftrightarrow \bar{\bar{H}}_s$ ⟶ $\bar{\bar{G}}_s = \left(\begin{array}{|ccc|c} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{array} \right)$

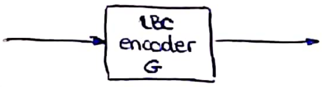
matriz identidad

$\bar{\bar{H}}_s = (\boxed{1 \ 1 \ 1} \ 1)$

Si la matriz no es sistemática se debe buscar esta realizando operaciones.

¿Cómo saber cómo de buenos son los códigos? PROPIEDADES DETECTORAS Y CORRECTORAS.

BINARY LINEAR BLOCK CODES



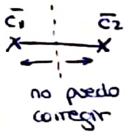
$$\bar{\bar{G}}_s = \begin{pmatrix} 1 & 0 & 0 & 0 & \vdots & 1 \\ 0 & 1 & 0 & 0 & \vdots & 1 \\ 0 & 0 & 1 & 0 & \vdots & 1 \\ 0 & 0 & 0 & 1 & \vdots & 1 \end{pmatrix} \longrightarrow \bar{\bar{H}}_s = (1111 \vdots 1)$$

$(n-k)xn$

$k=4$
$n=5$

$d_{min} = 2$

$\bar{r}$ $d=1$
$t=0$



no puedo corregir

$$\bar{\bar{G}}s = (1 \quad 1 \quad 1 \quad 1 \quad 1) \longrightarrow k=1, \; n=5$$

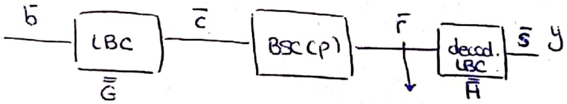
$$\bar{\bar{H}}s = \begin{pmatrix} 1 & 0 & 0 & 0 & \vdots & 1 \\ 0 & 1 & 0 & 0 & \vdots & 1 \\ 0 & 0 & 1 & 0 & \vdots & 1 \\ 0 & 0 & 0 & 1 & \vdots & 1 \end{pmatrix} \longrightarrow (n-k)xn$$

¿cuantas palabras tiene?

$2^k = 2^1 = \underline{\underline{2}}$

$d_{min} = 5 \longrightarrow d=4$
$t=2$

* Distancia mínima del código:



y se comparan las palabras dos a dos y se escoge el <u>valor mínimo</u>.

$$\boxed{d_{min} \leq (n-k)+1}$$

Cota de Singleton

$\bar{c} + \bar{e}$
↓
vector error
$(e_1, \ldots, e_n) \longrightarrow$ nº '1' = nº errores $\longrightarrow$ <u>en qué bit!!!</u>

$\bar{s} = \bar{r} \cdot \bar{\bar{H}}^T$

* Posibilidades:

- Detección de error (ARQ) : si $\omega(e) = d_{min} - 1 = d$
- Corrección de error (FEC) : si $\omega(e) = (d_{min} - 1)/2 = t$

[NOTA => $\omega(e)$ = peso de error]

d_{min} se puede calcular con H como el número de columnas que debo sumar en binario para obtener vector $\emptyset$.

nº codewords = 2^k
nº posibilidades recepción = 2^n
} un LBC detecta hasta patrones de $(2^n - 2^k)$

síndrome = $S_i = r \cdot H^T \longrightarrow$ permite distinguir 2^{n-k} posibilidades de error

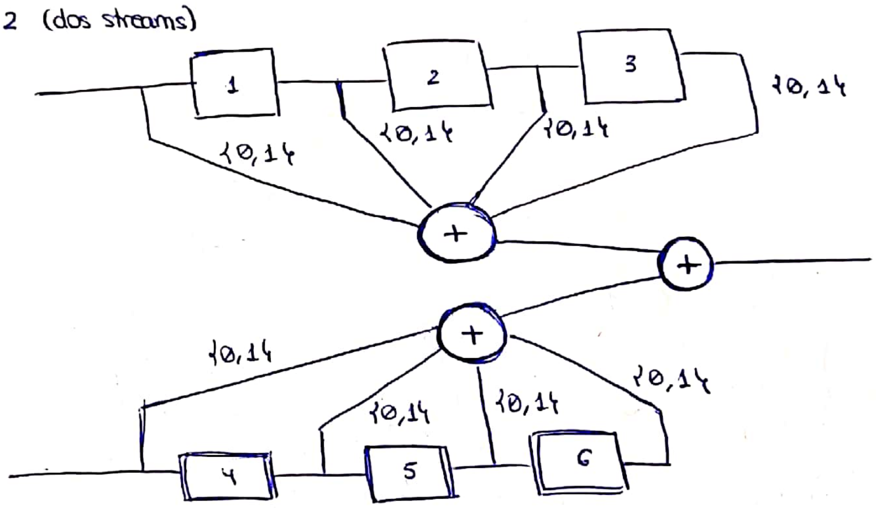
$\hat{c} = \bar{r} + \bar{e}_i \longrightarrow$ si el error escogido/estimado es correcto $\longrightarrow \hat{c} = \hat{b} \cdot G$

MULTIPLE - ERROR CORRECTION $\longrightarrow$ NO ENTRA

CONVOLUTIONAL CODES

nº registros desplazamiento = nº streams de entrada

K=2 (dos streams)



Las ramas que su peso sea '0', no se tienen en cuenta.

n = 1

posiciones de memoria = 6 = ν

↓

$2^6 = 2^\nu$

CC (n, K, ν)

$R = \frac{K}{n}$

Una salida 0101 = $D + D^3 = g(D)$

$b(D) = 101 = 1 + D^2$

↓

g(D)

||

respuesta del sistema al impulso:

$$C(D) = b(D) \cdot g(D) = (D + D^3) + (D^3 + D^5)$$

$$= D + \underbrace{(1+1)}_{GF(2)} D^3 + D^5 = D + D^5$$

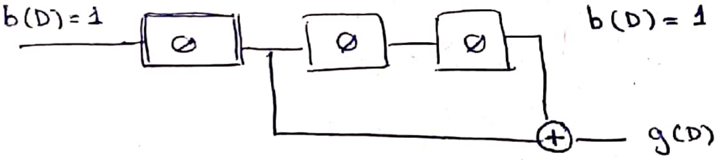
$= 010001$

GF(2)

||

0

(operación lógica)



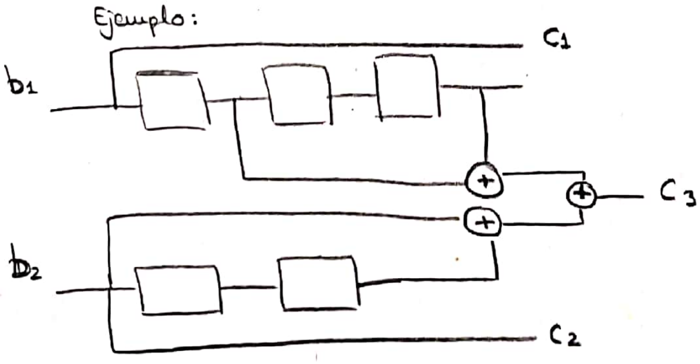
} respuesta al impulso, inicialización a 0 de los registros.

Expresión matricial:

$$G(D) = \begin{pmatrix} g_1^{(1)}(D) & g_1^{(2)}(D) & \cdots & g_1^{(n)}(D) \\ g_2^{(1)}(D) & \vdots & & \vdots \\ \vdots & & & \end{pmatrix}$$

→ FILA = registro desplazamiento

Ejemplo:



→ reflejo b1 en C1

→ reflejo b1 en C2

→ reflejo b1 en C3

$$\overline{G}(D) = \begin{pmatrix} 1 & 0 & D + D^3 \\ 0 & 1 & 1 + D^2 \end{pmatrix}$$

reflejo b2 en C1

b2 en C2

b2 en C3

⇓

CC (3, 2, 5)

n ↓ → C1, C2, C3

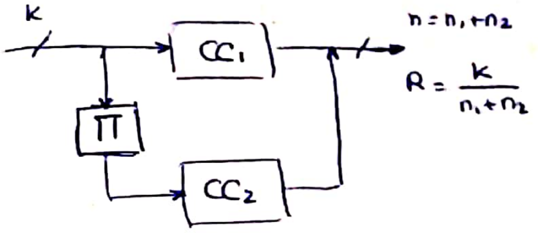
K ↓ → b1, b2

V ↓ → 5 registros

Diferencia demodulación dura y blanda : 2 dB

Se pueden localizar errores de bucles en el trellis.

Puncturing : se eliminan bits , consiguiendo R variables



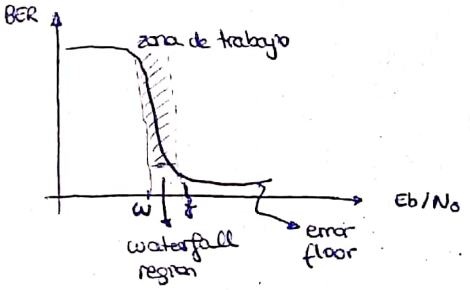
TURBO CODES :

- pocos estados
- resultados muy buenos
- se aproximan al límite teórico de Shannon.
- SISO da una estimación de probabilidad a posteriori

⇒ Modelo de decodificación **iterativa**

$\boxed{\Pi}$ = permutador : reordena los bits

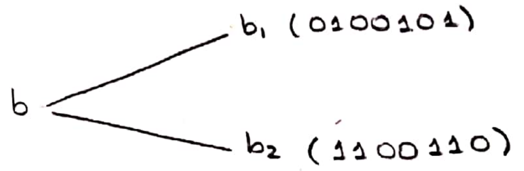
$\boxed{CC}$ → deben ser recursivos



- Variando N (nº bits empleados) :

N↑ → nº bits ↑ → BER ↓ → R ↓

N↓ → nº bits ↓ → BER ↓ → R ↑



$$\bar{b}(D) = (D + D^4 + D^6,\ 1 + D + D^4 + D^5)$$

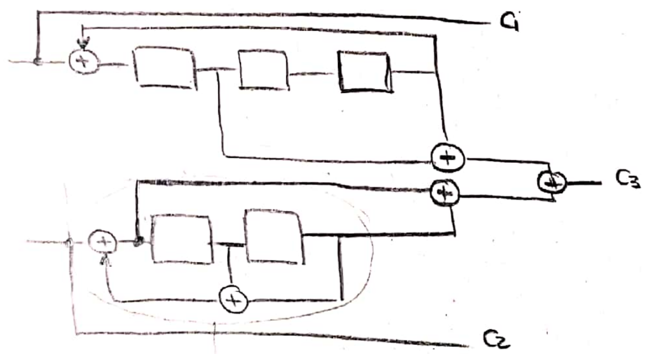
$$\bar{C}(D) = (C_1,\ C_2,\ C_3) \longrightarrow \begin{cases} C_1 = D + D^4 + D^6 = b_1 \\ C_2 = 1 + D + D^4 + D^5 = b_2 \\ C_3 = \bar{\bar{G}}(D) \cdot \bar{b}(D) \end{cases}$$

CC sistemático sin realimentación:

$$\bar{\bar{G}}(D) = \left(\boxed{\begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix}} \ \begin{matrix} D+D^3 \\ 1+D^2 \end{matrix} \right)$$

↳ matriz identidad = sistemático

Si modificamos:

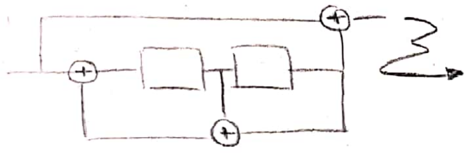


$$\bar{\bar{G}}(D) = \left(\begin{matrix} 1 & 0 & \frac{D+D^3}{1+D^3} \\ 0 & 1 & \frac{1+D^2}{1+D+D^2} \end{matrix} \right)$$

↓ sigue siendo sistemático ya que C_1, C_2 no tienen en cuenta la realimentación

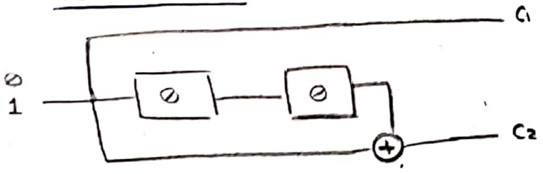
la realimentación se expresa mediante la división.

si fuera



$$1 + \frac{D^2}{1+D+D^2}$$

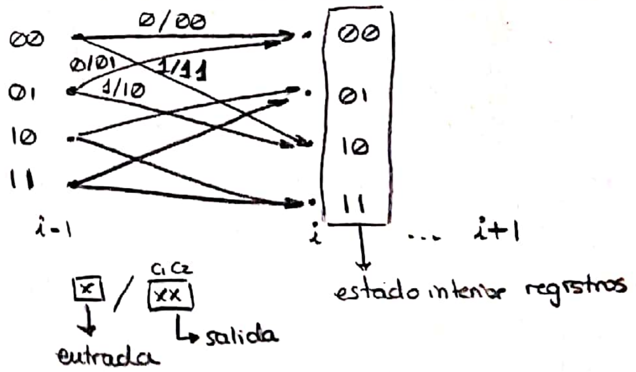
FSM: trellis



CC(2, 1, 2)

transición = $2^k / 2^n$

estado = 2^v



Se suelen añadir unos bits de terminación para que se empiece/acabe en el estado '0', así se sabe de donde se parte y donde se acaba.

[LOW DENSITY PARITY CHECK CODES]

Van a sustituir a los turbo códigos en ciertas áreas (LDPC), por ejemplo en 5G.

Diferencia con LBC:

* matriz chequeo (H) → cada fila = una ecuación → LDPC : casi todos elementos 0
ecuaciones sencillas

* densidad : $r = \frac{\rho}{n} = \gamma$

$H_{J\times n}$, ρ = '1's / fila, γ = '1's / columna

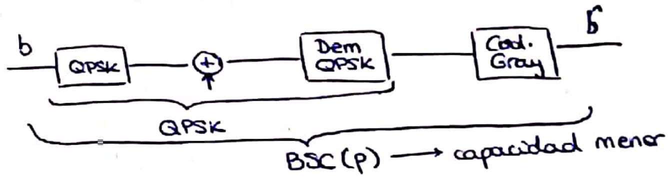
* matriz H sobredimensionada, hay dependencia, si se elimina la dependencia se consigue LBC.

LDPC : es un LBC expandido, que garantiza ecuaciones de paridad dispersas.

[FOUNTAIN CODE] [CÓDIGOS POLARES]

TEMA 4 : TÉCNICAS DE ACCESO AL MEDIO

- medio
- canal ⟶ parcial 1 :



- FDMA ⟶ móviles 1G
- TDMA ⟶ móviles 2G
- SDMA ⟶ móviles 3G
- OFDMA ⟶ móviles 4G

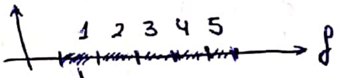
[Multiple Access vs Multiplexing] -

Acceso múltiple : uplink ⟶ no hay coordinación

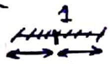
Multiplexación : downlink ⟶ coordinación (no hay colisiones, hay entidad centralizada)

Duplexación : FDD
TDD

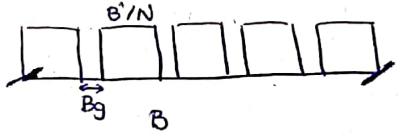
[FDMA] se divide un ancho de banda
un canal para un usuario
uso de filtros sintonizados



↓ una comunicación unidireccional
o bidireccional (duplex) :



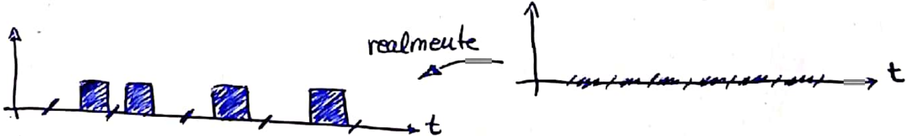
hay que dejar intervalos de guarda entre canales para filtrar correctamente :



$$B' = \underbrace{B}_{\text{total}} - (N-1)\underbrace{Bg}_{\text{ancho de banda de guarda}} \longrightarrow \text{FDD, TDD} : \frac{B'}{2}$$

$$R_b = kR_s = K \cdot \underbrace{\frac{B'}{2N}}_{\text{conformado rectangular}}$$

[TDMA] división en slots de tiempo ⟶ comunicaciones digitales solo.



fenómeno de propagación por posiciones diferentes : no se sabe exactamente cuándo llega

hay tiempo de guarda para evitar colisiones

MUY IMPORTANTE

$$T_f = \frac{(D_c + D_s)N_s}{B/2} + N_s \cdot T_g \longrightarrow \underset{\text{(efectiva)}}{R_b} = K\frac{D_s}{T_f} \longrightarrow \underset{\text{(nominal)}}{R_b} < \frac{D_s}{N_s}$$

$\frac{D_s}{N_s}$ → hay Tg, hay Dc

(nominal) : $\underline{\underline{R_b = K \cdot \frac{B}{2}}}$ instantánea

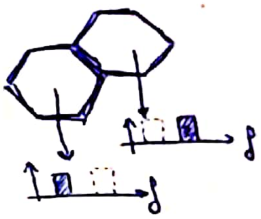
$$T_s = \frac{T_f}{N_s} \longrightarrow \text{tiempo de slot}$$

Dc = símbolos de control
Ds = información (símbolos)
Ns = nº slots
B/2 = conformado rectangular
Tg = tiempo de guarda

Space division multiple access : [SDMA]

organiza las estaciones bases de la telefonía móvil en celdas.

las zonas no deben interferir con el resto, cada una a una frecuencia



[CDMA]

utiliza la interferencia

no se divide el ancho de banda, es asíncrona (cada uno transmite "cuando quiere")

técnica de espectro ensanchado : mala, desaprovecho BW.

si se elige bien c(t), la señal con la que se codifica, su uso ↑↑

factor de ensanchamiento : $N = T_b / T_c$ (crítico!!)

$[BW' = BW \cdot N]$

asignando diferentes secuencias código a cada usuario, se puede extraer la información enviándose todos juntos, interfiriéndose incluso.

límites: nivel de interferencia razonable, códigos libres.

N = nº chips / símbolo o transmisión

Ganancia de procesamiento : $PG = 10 \log(N)$ → marca como de inmune soy como se resiste a las interferencias

Tipos : $\underline{\underline{\text{DS}}}$, FH, TH

OFDM

Multicarrier modulation : se emplean diferentes portadoras

se segmenta el ancho de banda y se puede hacer igualación de forma eficiente.

Se basa en la DFT (discrete Fourier transform)