



Codificación de Canal

Introducción & códigos bloque

Manuel A. Vázquez
Jose Miguel Leiva
Joaquín Míguez

21 de febrero de 2024

Índice

- 1 **Introducción**
 - Modelos de canal
 - Fundamentos
- 2 **Codificación**
- 3 **Decodificación**
 - Decodificación dura
 - Decodificación blanda
 - Ganancia de codificación
- 4 **Códigos bloque lineales**
 - Fundamentos
 - Decodificación
- 5 **Códigos cíclicos**
 - Polinomios
 - Decodificación

Índice

- 1 **Introducción**
 - Modelos de canal
 - Fundamentos
- 2 **Codificación**
- 3 **Decodificación**
 - Decodificación dura
 - Decodificación blanda
 - Ganancia de codificación
- 4 **Códigos bloque lineales**
 - Fundamentos
 - Decodificación
- 5 **Códigos cíclicos**
 - Polinomios
 - Decodificación

Codificación (de canal)

Objetivo

Añadir redundancia a la información transmitida para ser capaces de recuperarla aún cuando se produzcan errores.



Ejemplo: código de repetición

- 0 → 000
- 1 → 111

de manera que, e.g.,

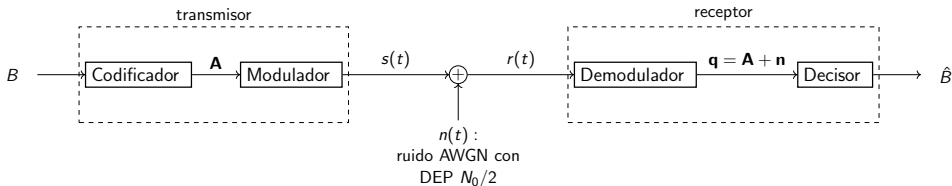
010 → 000 111 000

¿Qué deberíamos *decidir* que se transmitió si recibimos

010 100 000 ?

¡000 (en lugar de 010)!

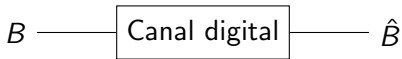
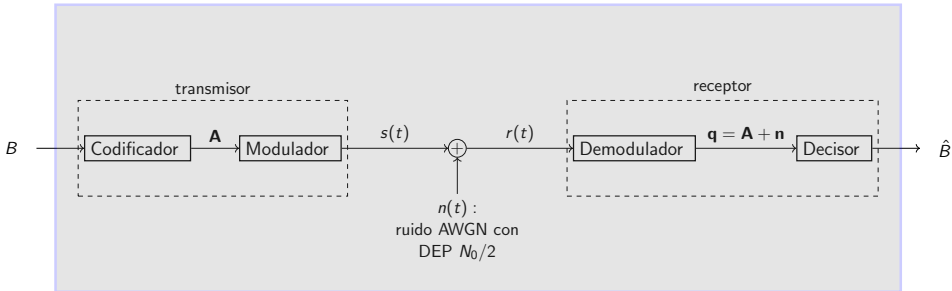
Sistema de Comunicaciones Digital



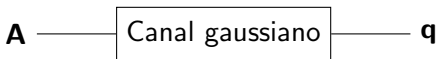
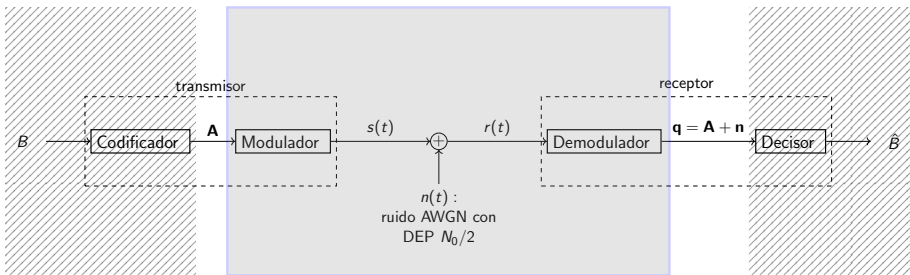
Se puede estudiar este modelo a diferentes niveles de abstracción...

- Canal digital
- Canal gaussiano

Canal digital



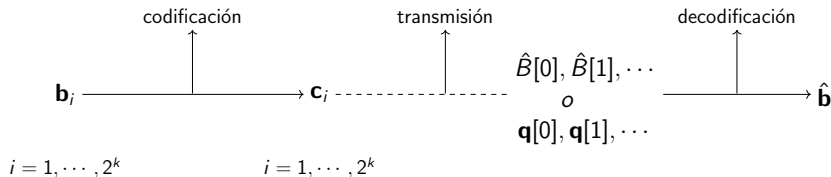
Canal Gaussiano (con entrada digital)



Algunos conceptos básicos

- **Código**

Transformación de una secuencia de k bits, $\mathbf{b} \in \{\mathbf{b}_1, \mathbf{b}_2, \dots\}$, en otra de $n > k$ bits, $\mathbf{c} \in \{\mathbf{c}_1, \mathbf{c}_2, \dots\}$.



- **Probabilidad de error de \mathbf{b}_i**

$$P_e^i = Pr\{\hat{\mathbf{b}} \neq \mathbf{b}_i | \mathbf{b} = \mathbf{b}_i\}, \quad i = 1, \dots, 2^k$$

- **Probabilidad de error máxima:** $P_e^{\text{máx}} = \text{máx}_i P_e^i$
- **Tasa:** La tasa de un código define el número de bits de información k que transporta una palabra código de longitud n .

$$R = k/n$$

Probabilidad de error *de palabra código vs. de bit*

- P_e : probabilidad de error de palabra código

$$P_e = \frac{\# \text{ palabras código incorrectas recibidas}}{\# \text{ total de palabras código transmitidas}} = \frac{v}{w}$$

- **BER (Bit Error Rate)**: probabilidad de error de bit

$$BER = \frac{\# \text{ bits incorrectos}}{\# \text{ bits transmitidos}}$$

(si cada palabra código transporta un único bit de información, coinciden)

$$\left. \begin{array}{l} \text{peor caso} \rightarrow BER = \frac{v \times k}{w \times k} = P_e \\ \text{mejor caso} \rightarrow BER = \frac{v \times 1}{w \times k} = \frac{P_e}{k} \end{array} \right\} \Rightarrow \frac{P_e}{k} \leq BER \leq P_e$$

Teorema de Codificación

Teorema: Codificación de canal (Shannon, 1948)

Si C es la capacidad de un canal, es posible transmitir a tasa $R < C$ de forma fiable.

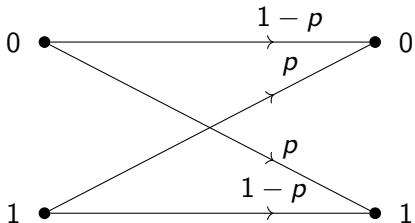
Capacidad

Es el máximo de la información mutua entre la entrada y la salida del canal.

Transmisión fiable

Existe una secuencia de códigos $(n, k) = (n, nR)$ tal que, cuando $n \rightarrow \infty$, $P_e^{\text{máx}} \rightarrow 0$.

Teorema de codificación de canal: ejemplo



Sean 4 canales binarios con

$$p = 0,15 \Rightarrow C_1 = 0,39$$

$$p = 0,17 \Rightarrow C_3 = 0,34$$

$$p = 0,13 \Rightarrow C_2 = 0,44$$

$$p = 0,19 \Rightarrow C_4 = 0,29$$

$$C = 1 - H_b(p),$$

siendo p la BER del canal y H_b la entropía binaria.

y un código de tasa $R = 1/3 = 0,33$.



Teorema de codificación de canal

Un código con $R=1/3$ solo respeta el límite de Shannon en los tres primeros casos.

Teorema de codificación de canal: ejemplo

En la figura se muestra como evoluciona la probabilidad de error de palabra código con n (solo tiende a 0 cuando $R < C$).

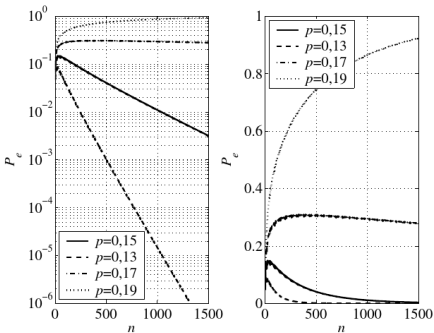


Figura: Izquierda: escala logarítmica; derecha: escala lineal

Definiciones

Definición: Redundancia

El número de bits, $r = n - k$, añadidos por el codificador.

Permite expresar la tasa del código como $R = \frac{k}{n} = \frac{n-r}{n} = 1 - \frac{r}{n}$

Definición: Distancia de Hamming...

...entre dos secuencias binarias es el número de bits en los que difieren.

Proporciona una medida de cómo de diferentes son dos secuencias de bits. Por ejemplo, $d_H(1010, 1001) = 2$.

Definición: Distancia mínima de un código

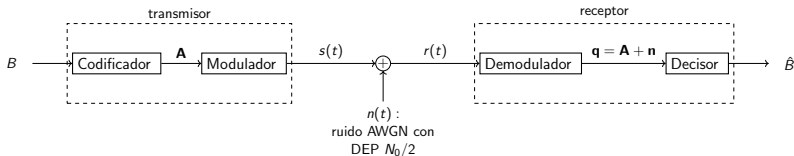
$$d_{min} = \min_{i \neq j} d_H(\mathbf{c}_i, \mathbf{c}_j)$$

Índice

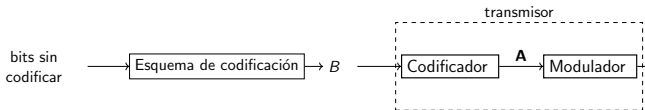
- 1 Introducción
 - Modelos de canal
 - Fundamentos
- 2 Codificación
- 3 Decodificación
 - Decodificación dura
 - Decodificación blanda
 - Ganancia de codificación
- 4 Códigos bloque lineales
 - Fundamentos
 - Decodificación
- 5 Códigos cíclicos
 - Polinomios
 - Decodificación

Codificación

En el modelo habitual de un sistema de comunicaciones digital,



el esquema de codificación se coloca *antes* del sistema



y tenemos

$$\left. \begin{array}{l} B[0] = C[0] \\ B[1] = C[1] \\ \vdots \quad \quad \vdots \end{array} \right\} \text{palabra código}$$

Índice

- 1 Introducción
 - Modelos de canal
 - Fundamentos
- 2 Codificación
- 3 Decodificación**
 - Decodificación dura
 - Decodificación blanda
 - Ganancia de codificación
- 4 Códigos bloque lineales
 - Fundamentos
 - Decodificación
- 5 Códigos cíclicos
 - Polinomios
 - Decodificación

Decodificación dura

- Decodificación a *nivel de bit*
- Se basa en el **canal digital**



- La entrada al decodificador son los bits que salen del **Detector**, los \hat{B} 's.
- La métrica utilizada es la **distancia de Hamming**.

Notación

$\mathbf{c}_i = [C^i[0], C^i[1], \dots, C^i[n-1]] \equiv i$ -ésima palabra código

$\mathbf{r} = [\hat{B}[0], \hat{B}[1], \dots, \hat{B}[n-1]] \equiv$ palabra recibida

Decodificación dura: criterio de decisión

- Criterio *Máximo a Posteriori* (MAP): decidir \mathbf{c}_i si

$$p(\mathbf{c}_i|\mathbf{r}) > p(\mathbf{c}_j|\mathbf{r}) \quad \forall j \neq i$$

- Si las palabras código son equiprobables, entonces es equivalente a Máxima Verosimilitud (ML),

$$p(\mathbf{r}|\mathbf{c}_i) > p(\mathbf{r}|\mathbf{c}_j) \quad \forall j \neq i$$

- Las verosimilitudes se pueden expresar en función de d_H

$$p(\mathbf{r}|\mathbf{c}_i) = \epsilon^{d_H(\mathbf{r}, \mathbf{c}_i)} (1 - \epsilon)^{n - d_H(\mathbf{r}, \mathbf{c}_i)}$$

$\epsilon \equiv$ probabilidad de error de bit *del canal*

- Si $\epsilon < 0,5$ ML equivale a decidir \mathbf{c}_i si

$$d_H(\mathbf{r}, \mathbf{c}_i) < d_H(\mathbf{r}, \mathbf{c}_j) \quad \forall j \neq i.$$

Decodificación dura: detección vs. corrección de errores

Suponiendo que ocurrieron errores durante la transmisión, hay dos escenarios posibles:

- **No** lo **detectamos**
(solo detectamos errors si $\mathbf{r} \neq \mathbf{c}_i \quad i = 1, \dots, 2^k$)
- Lo **detectamos**, y tenemos que tomar una decisión:
 - No nos arriesgamos a **corregirlos** y solicitamos la *retransmisión*
(no podemos corregir *con confianza*)
 - **intentamos corregirlos**
(¡¡implica un riesgo!!)

En este último escenario necesitamos una *política*: en este curso **siempre** intentamos corregir los errores.

Decodificación dura: detección

- Detectamos un error de palabra código cuando se producen **menos de** d_{min} errores de bit.
- Probabilidad de no detectar que ha ocurrido algún error (se tienen que dar d_{min} errores)

$$P_{nd} \leq \sum_{m=d_{min}}^n \binom{n}{m} \epsilon^m (1 - \epsilon)^{n-m}$$

donde ϵ es la probabilidad de error de bit en el sistema, y d_{min} es la mínima distancia entre palabras código.



Una cota de la probabilidad de error...

...porque hay casos en los que d_{min} errores no hacen que una palabra código se convierta en otra $\Rightarrow \leq$ en lugar de $=$

Decodificación dura: corrección (*siempre* corregimos)

- Se decodifica correctamente si hay menos de $d_{min}/2$ bits erróneos \Rightarrow el código puede corregir **hasta**

$$t = \lfloor (d_{min} - 1)/2 \rfloor \text{ errores.}$$

- Probabilidad de error de corrección:

$$P_e \leq \sum_{m=t+1}^n \binom{n}{m} \epsilon^m (1 - \epsilon)^{n-m}$$



Una cota de la probabilidad de error...

...puesto que podríamos corregir bien más de t errores (simplemente no está garantizado) $\Rightarrow \leq$ en lugar de $=$

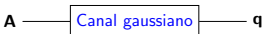


Cota aproximada

El primer elemento del sumatorio aproxima bien el error si ϵ es pequeño y d_{min} grande.

Decodificación blanda

- Decodificación a *nivel de elemento de la constelación*
- Se basa en el **canal gaussiano**



con

$$\mathbf{q} = \mathbf{A} + \mathbf{n}$$

donde \mathbf{n} es un vector de ruido gaussiano.

- La entrada al decodificador son las observaciones que salen del **Demodulador**, los \mathbf{q} 's.
- La métrica utilizada es la **distancia euclídea**

Notación

$m \equiv \#$ bits transportados por cada \mathbf{A}

$\tilde{\mathbf{c}}_i = [\mathbf{A}^{(i)}[0], \mathbf{A}^{(i)}[1], \dots, \mathbf{A}^{(i)}[n/m - 1]] \equiv i$ -ésima pal. cód.

$\tilde{\mathbf{r}} = [\mathbf{q}[0], \mathbf{q}[1], \dots, \mathbf{q}[n/m - 1]] \equiv$ palabra recibida

Decodificación blanda: corrección

- La probabilidad de error *de palabra código* se puede aproximar por

$$P_e \approx \kappa Q \left(\frac{d_{min}/2}{\sqrt{N_0/2}} \right) \quad (1)$$

donde κ es el *kiss number*.

Definición: kiss number

Es el máximo número de palabras código que están a una distancia d_{min} de una dada.

Ganancia de codificación

- Si igualamos la *BER* con y sin codificación, la **ganancia de codificación** se obtiene como

$$G = \frac{(E_b/N_0)_{sc}}{(E_b/N_0)_c}$$

- Hay que distinguir entre la obtenida con decodificación dura y blanda.

Para calcular las E_b/N_0 's individuales, a menudo es útil...



Regla de Stirling

$$Q(x) \approx \frac{1}{2} e^{-\frac{x^2}{2}}$$

Ganancia de codificación: ejemplo



Sea una constelación binaria antipodal 2-PAM ($\pm\sqrt{E_s}$), y el código

\mathbf{b}_i	\mathbf{c}_i
00	000
01	011
10	110
11	101

Ganancia de codificación: ejemplo - decodificación dura

- Este código no corrige errores puesto que $t = \lfloor (d_{min} - 1)/2 \rfloor = 0$, y la probabilidad de error de palabra código es

$$P_e \leq \sum_{m=1}^3 \binom{3}{m} \epsilon^m (1 - \epsilon)^{n-m} \approx 3\epsilon$$

donde $\epsilon = Q(\sqrt{2E_s/N_0})$.

- Probabilidad de error de bit

$$BER \approx \frac{2}{3} 3Q\left(\sqrt{\frac{2E_s}{N_0}}\right)$$

- Y para expresarlo en términos de E_b , tenemos en cuenta $2E_b = 3E_s$, quedando

$$BER \approx 2Q\left(\sqrt{\frac{4E_b}{3N_0}}\right)$$

Ganancia de codificación: ejemplo - decodificación blanda

- Decidimos \mathbf{b} a partir de la salida del canal gaussiano,

$$\mathbf{q} = (\mathbf{q}[0], \mathbf{q}[1], \mathbf{q}[2]) = (\mathbf{A}[0] + \mathbf{n}[0], \mathbf{A}[1] + \mathbf{n}[1], \mathbf{A}[2] + \mathbf{n}[2])$$

- Equivalente al decisor para la constelación

$$\begin{pmatrix} -\sqrt{E_s} \\ -\sqrt{E_s} \\ -\sqrt{E_s} \end{pmatrix}, \begin{pmatrix} -\sqrt{E_s} \\ \sqrt{E_s} \\ \sqrt{E_s} \end{pmatrix}, \begin{pmatrix} \sqrt{E_s} \\ \sqrt{E_s} \\ -\sqrt{E_s} \end{pmatrix}, \begin{pmatrix} \sqrt{E_s} \\ -\sqrt{E_s} \\ \sqrt{E_s} \end{pmatrix}$$

que tiene distancia (euclídea) mínima $d_{min} = 2\sqrt{2E_s}$

- Utilizando (1) la probabilidad de error de palabra código es

$$P_e \approx 3Q\left(\sqrt{\frac{4E_s}{N_0}}\right)$$

- BER en función de E_b :

$$BER \approx 2Q\left(\sqrt{\frac{8E_b}{3N_0}}\right)$$

Ganancia de codificación: ejemplo - decodificación dura vs. blanda

- Sin codificar, tenemos que $E_b = E_s$ y

$$BER_{sc} = \epsilon = Q(\sqrt{2E_b/N_0})$$

- Ganancia con decodificación dura
 - Igualamos BER_c y BER_{sc}
 - Aproximación: igualamos solo los argumentos de $Q(\cdot)$

$$G = \frac{(E_b/N_0)_{sc}}{(E_b/N_0)_c} = 2/3 \approx -1,76dB$$

- ¡Hemos perdido codificando! (lógico, dado que el código no corrige errores)
- Decodificación blanda

$$G = 4/3 \approx 1,25dB$$

- Ahora sí sacamos partido a la codificación

Índice

- 1 **Introducción**
 - Modelos de canal
 - Fundamentos
- 2 **Codificación**
- 3 **Decodificación**
 - Decodificación dura
 - Decodificación blanda
 - Ganancia de codificación
- 4 **Códigos bloque lineales**
 - Fundamentos
 - Decodificación
- 5 **Códigos cíclicos**
 - Polinomios
 - Decodificación

Códigos bloque lineales

Campo de Galois módulo 2 ($GF(2)$)

$$a + b = (a + b)_2$$

$$a \cdot b = (a \cdot b)_2$$

Definición: Código bloque lineal




Un código bloque lineal es un código en el que cualquier combinación lineal de palabras código es también una palabra código.

Propiedades


- Es un subespacio de $GF(2)^n$ con 2^k elementos.
- La palabra todo-ceros pertenece al código.
- Toda palabra tiene al menos otra a d_{min} .
- d_{min} coincide con el menor peso (número de 1s) de las palabras no nulas.

Códigos bloque lineales: estructura

Elementos de un código lineal (n, k)

- **b** es el mensaje,  $1 \times k$
- **c** es la palabra código,  $1 \times n$
- **r** es la palabra recibida,  $1 \times n$ con

$$\mathbf{r} = \mathbf{c} + \mathbf{e}$$

- **e** es el ruido  $1 \times n$
- **G** es la matriz **generadora**,

(para codificar)

 $k \times n$

- **H** es la matriz de **control (o chequeo) de paridad**,
- (para decodificar)

 $n - k \times n$

Codificación

El *mapeo* $\mathbf{b} \rightarrow \mathbf{c}$ se hace multiplicando por una matriz i.e.,

$$\mathbf{c} = \mathbf{bG}.$$

Recordar:

- \mathbf{b} es $1 \times k$
- \mathbf{G} es $k \times n$
- \mathbf{c} es $1 \times n$



Propiedad

Cada fila de \mathbf{G} es una palabra código.

Matriz de control de paridad

La matriz de control de paridad, \mathbf{H} , es el *complemento ortogonal* de \mathbf{G} , de manera que

$$\mathbf{cH}^T = \mathbf{0} \Leftrightarrow \mathbf{c} \text{ is a codeword}$$

Por conveniencia,

Definición: Síndrome

El síndrome de la secuencia recibida \mathbf{r} es

$$\mathbf{s} = \mathbf{rH}^T \quad (\text{de dimensiones } 1 \times (n - k))$$

Entonces,

$$\mathbf{s} = \mathbf{0} \Leftrightarrow \mathbf{r} \text{ es una palabra código.}$$



Relación entre el síndrome y el error

$$\mathbf{s} = \mathbf{rH}^T = (\mathbf{c} + \mathbf{e})\mathbf{H}^T = \mathbf{cH}^T + \mathbf{eH}^T = \mathbf{eH}^T$$

Decodificación dura: decodificación por síndrome

El **criterio de mínima distancia** exige calcular la d_H entre la palabra recibida, \mathbf{r} , y todas las palabras código...pero podemos hacer decodificación por *síndrome*.

De antemano:

Construir una tabla con el síndrome asociado a cada posible error,

error (\mathbf{e})	síndrome(\mathbf{s})	(Si varios errores dan lugar al mismo síndrome, se elige el más verosímil, i.e., el de menor peso)
⋮	⋮	

En operación: dada la palabra recibida, \mathbf{r} :

- 1 Obtener síndrome $\mathbf{s} = \mathbf{rH}^T$.
- 2 Buscar en la tabla el patrón de error, \mathbf{e} , con ese síndrome
- 3 *Deshacer* el error

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

Códigos sistemáticos

Definición: Código sistemático

Un código en el que el mensaje siempre aparece en la secuencia codificada (en el mismo sitio).

Se puede forzar esto a través de la matriz generadora,

$$\mathbf{G} = [\mathbf{I}_k \quad \mathbf{P}] \quad \text{o} \quad \mathbf{G} = [\mathbf{P} \quad \mathbf{I}_k]$$

- Los k primeros/últimos bits en \mathbf{c} son iguales a \mathbf{b} y los $n - k$ restantes son de redundancia.
- Si $\mathbf{G} = [\mathbf{I}_k \quad \mathbf{P}]$ se puede demostrar

$$\mathbf{H} = [\mathbf{P}^T \quad \mathbf{I}_{n-k}]$$



Ejercicio

Demuéstralo!!

Ejemplo de código sistemático: Hamming (7, 4)

Matriz Generadora:

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

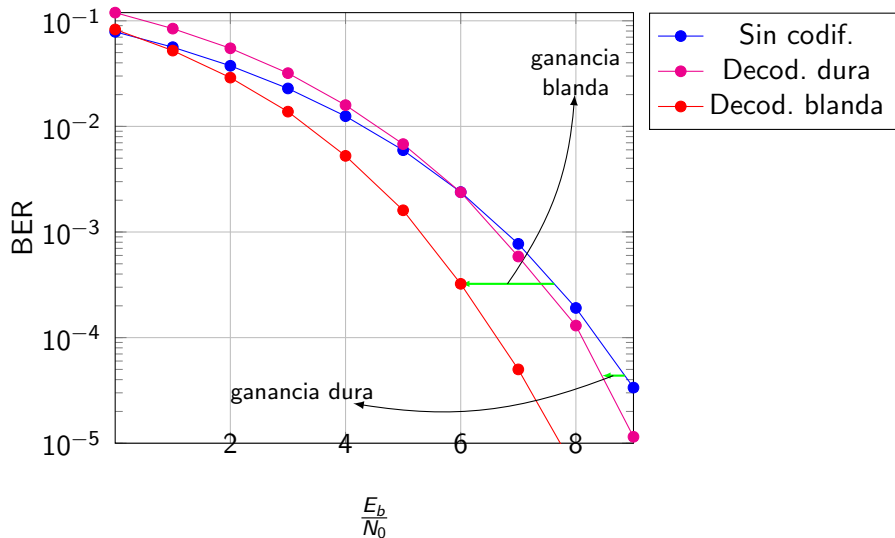
Matriz de control de paridad:

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

Cualquier código de Hamming:

- Es *perfecto*
- $d_{min} = 3$
- $k = 2^j - j - 1$ y $n = 2^j - 1 \forall j \in \mathbb{N} \geq 2$
 - $j = 2 \rightarrow (3, 1)$
 - $j = 3 \rightarrow (7, 4)$
 - $j = 4 \rightarrow (15, 11)$

Hamming (7, 4): ganancia de codificación



Hamming (7, 4): decodificación

De antemano aplicamos

$$\mathbf{s} = \mathbf{eH}^T$$

sobre cada \mathbf{e} que suponga un único bit erróneo (el código solo puede corregir 1 bit):

error	síndrome
0000000	000
1000000	101
0100000	110
0010000	111
0001000	011
0000100	100
0000010	010
0000001	001



Ejemplo: $\mathbf{r} = [1100101]$

$$\mathbf{s} = \mathbf{rH}^T = [1100101] \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = [110]$$

por lo que $\mathbf{e} = [0100000]$ y

$$\hat{\mathbf{c}} = \mathbf{r} + \mathbf{e} = \mathbf{r} = [1\mathbf{0}00101].$$

Códigos equivalentes



Calcular H a partir de G

Si el código es sistemático, tenemos una manera sencilla de calcular la matriz de control de paridad...

...pero y si no lo es? Si el código **no** es sistemático, se pueden hacer operaciones sobre su matriz generadora, G , para transformarla en una de un código sistemático *equivalente*, $G' = [I_k \ P]$.

Las operaciones *permitidas* son:

Sobre filas reemplazar una fila con una combinación lineal de si misma y otras filas o intercambiar filas.

Sobre columnas intercambiar columnas.

Definición: Códigos equivalentes

Dos códigos son equivalentes si tienen las mismas palabras código (quizás tras reordenar los bits).

Índice

- 1 Introducción
 - Modelos de canal
 - Fundamentos
- 2 Codificación
- 3 Decodificación
 - Decodificación dura
 - Decodificación blanda
 - Ganancia de codificación
- 4 Códigos bloque lineales
 - Fundamentos
 - Decodificación
- 5 Códigos cíclicos
 - Polinomios
 - Decodificación

Códigos cíclicos



Valores de k y n altos

Trabajar con matrices no es eficiente!!

Definición: Código cíclico

Es un código bloque lineal en el que cualquier desplazamiento *circular* de una palabra código da como resultado otra palabra código.

En un código cíclico,

- Si $[c_0, c_1, \dots, c_{n-1}]$ es palabra código, entonces $[c_{n-1}, c_0, c_1, \dots, c_{n-2}]$ también lo es
 - i.e., cualquier palabra código es igual a otra desplazada circularmente.

Representación polinómica de palabras código

La palabra código $[c_0, c_1, \dots, c_{n-1}]$ se representa mediante el polinomio

$$c(x) = c_0 + c_1x + c_2x^2 + \dots + c_{n-1}x^{n-1}$$

¿Cómo se consigue

$$[c_0, c_1, \dots, c_{n-1}] \rightarrow [c_{n-1}, c_0, \dots, c_{n-2}]$$

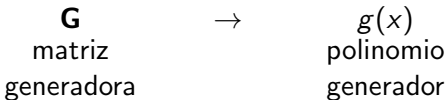
matemáticamente? Multiplicando $c(x)$ por x módulo $(x^n - 1)$, i.e.,

$$\begin{aligned} xc(x) &= c_0x + c_1x^2 + \dots + c_{n-1}x^n = c_0x + \dots + c_{n-1}x^n + c_{n-1} - c_{n-1} \\ &= c_{n-1}(x^n - 1) + c_0x + c_1x^2 + \dots + c_{n-2}x^{n-1} \end{aligned}$$

Entonces

$$(xc(x))_{x^n-1} = \underbrace{c_{n-1} + c_0x + c_1x^2 + \dots + c_{n-2}x^{n-1}}_{[c_{n-1}, c_0, \dots, c_{n-2}]}$$

Codificación



La codificación se lleva a cabo multiplicando, módulo $x^n - 1$, el polinomio que representa a \mathbf{b}_i por un **polinomio generador** $g(x)$,

$$c(x) = (b(x)g(x))_{x^n-1}$$

El polinomio generador, $g(x)$,

- es de grado $r = n - k$,
- debe ser un polinomio irreducible

Decodificación

$$\begin{array}{ccc} \mathbf{H} & \rightarrow & h(x) \\ \text{matriz de} & & \text{polinomio de} \\ \text{control de paridad} & & \text{control de paridad} \end{array}$$

El polinomio de control de paridad, $h(x)$,

- es de grado $r' = n - k - 1$,
- debe cumplir

$$(g(x)h(x))_{x^{n-1}} = 0.$$

Como en cualquier código bloque lineal, podemos hacer **decodificación por síndrome**:

$$s(x) = (r(x)h(x))_{x^{n-1}}$$