

# Codificación de Canal

Enrique Alexandre

Dpto. de Teoría de la Señal y Comunicaciones

Curso 2021/2022

# Índice

## 1. Introducción

- Motivación
- Parámetros importantes
- Detección y corrección de errores
- ¿Y la probabilidad de error?

## 2. Códigos de repetición

- Definición
- Cálculo de la probabilidad de error

## 3. Códigos de paridad

## 4. Códigos bloque

- Decodificación de máxima verosimilitud
- Códigos Hamming

## 5. Decodificación dura y blanda

- Definición
- Ejemplo

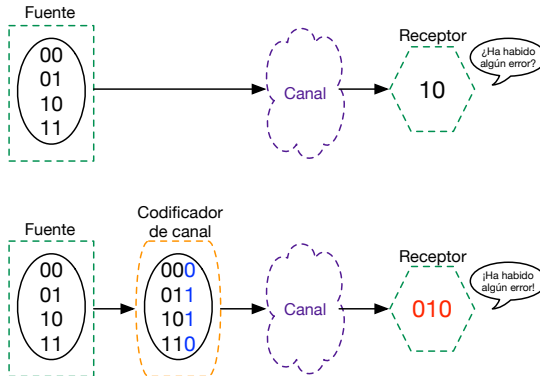
## 6. Códigos convolucionales

- Introducción
- Árbol del código
- Diagrama de rejilla
- Diagrama de estados
- Algoritmo de Viterbi

# Introducción

## Motivación

- Existencia de canales ruidosos
  - Necesidad de transmitir información libre de errores



# Introducción

## Parámetros importantes

- **Tasa de codificación:** Relación entre el número de bits de información y el número de bits totales transmitidos.
- **Distancia Hamming:** Número de elementos distintos entre dos vectores.
- **Distancia mínima:** La menor distancia Hamming entre dos palabras código válidas.

# Introducción

## Detección y corrección de errores

- Dado un código con una distancia mínima  $d_{\min}$ :
  - Se pueden **detectar**  $d_{\min} - 1$  errores.
  - Se pueden **corregir**  $(d_{\min} - 1)/2$  errores.

# Introducción

## ¿Y la probabilidad de error?

- Supongamos que la probabilidad de error en un bit es  $p_e$ .
- La probabilidad de que ocurran  $i$  errores en una palabra código con  $n$  bits se puede calcular como:

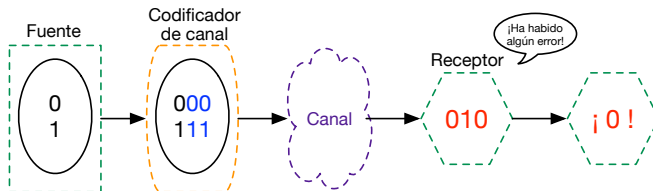
$$p(i, n) = \binom{n}{i} \cdot p_e^i \cdot (1 - p_e)^{n-i} \approx \binom{n}{i} \cdot p_e^i$$

$$\binom{n}{i} = \frac{n!}{i!(n-i)!}$$

# Códigos de repetición

## Definición

- Posiblemente la solución más sencilla:
  - Repito cada bit transmitido  $n$  veces
- Por ejemplo, si  $n = 3$ :



# Códigos de repetición

## Cálculo de la probabilidad de error

- Supongamos lo siguiente:
  - $p_e = 10^{-6}$
  - Código de repetición con  $k = 1$  y  $n = 3$ .
- La probabilidad de que haya algún error pero no lo detectemos será:

$$p(3, 3) = p_e^3 = 10^{-18}$$

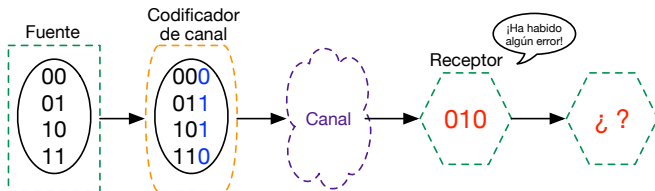
- La probabilidad de que haya algún error y no lo podamos corregir:

$$p = p(2, 3) + p(3, 3) = 3 \cdot p_e^2 - 2p_e^3 \approx 3 \cdot 10^{-12}$$



# Códigos de paridad

- Se utiliza  $n = k + 1$ .
- El bit de paridad es tal que el número de unos en la palabra código sea par (paridad par) o impar (paridad impar).



# Códigos bloque

- En realidad agrupan también a los anteriores.
- Un par de definiciones:
  - **Código lineal:**
    - La suma de dos vectores código es otro vector código.
    - El vector 0 forma parte del código.
  - **Peso de un vector:**
    - Es el número de unos que tiene el vector.
  - **Distancia mínima:**
    - Se puede calcular como el peso mínimo del código

# Códigos bloque

- Supongamos un mensaje a transmitir,  $\mathbf{m} = (m_1, m_2, \dots, m_k)$ .
- Y el correspondiente vector transmitido:  $\mathbf{x} = (x_1, x_2, \dots, x_n)$ .
- En un código bloque, ambos vectores se relacionan matricialmente como:

$$\mathbf{x} = \mathbf{m} \cdot \mathbf{G}$$

- Donde  $\mathbf{G}$  es la **matriz generadora** del código, de dimensiones  $(k \times n)$ .
- Decimos que un **código es sistemático** si la matriz generadora es tal que:

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

# Códigos bloque

- ¿Y cómo decodificamos el mensaje?
- Se utiliza la matriz de paridad  $\mathbf{H}$ , de dimensiones  $(n - k \times n)$ :

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

- Si recibimos un vector  $\mathbf{y}$  (que en realidad será el vector transmitido  $\mathbf{x}$  más ruido ( $\mathbf{e}$ ), haremos lo siguiente:

$$\mathbf{s} = \mathbf{y} \cdot \mathbf{H}^T$$

- A  $\mathbf{s}$  se le llama **síndrome**, y si es nulo, la palabra código  $\mathbf{y}$  es válida.

# Códigos bloque

- Y los errores, ¿cómo se corrigen?
- Es posible demostrar que el síndrome depende exclusivamente del error cometido ( $\mathbf{e}$ ), y no del vector transmitido ( $\mathbf{x}$ ).
- Pero hay un problema:
  - Si  $\mathbf{x}$  tiene  $n$  bits, hay  $2^n$  posibles vectores de error.
  - El síndrome sólo tiene  $n - k$  bits:
    - Sólo hay  $2^{n-k}$  síndromes posibles

# Códigos bloque

## Decodificación de máxima verosimilitud

- Creamos una tabla con los síndromes correspondientes a los  $2^{n-k-1}$  vectores de error más probables.
- ¿Qué hará el decodificador?
  - Calcula el síndrome:  $\mathbf{s} = \mathbf{y} \cdot \mathbf{H}^T$
  - Busca en la tabla ese síndrome
  - Obtiene el vector de error correspondiente ( $\hat{\mathbf{e}}$ )
  - Obtiene cuál sería la palabra decodificada:  $\mathbf{y} + \hat{\mathbf{e}}$

# Códigos bloque

## Códigos Hamming

- No hemos visto cómo se define la matriz generadora de un código bloque.
- Hay muchos métodos, pero quizás el más conocido es el de **códigos Hamming**.
- Se caracterizan por:
  - Bits de control:  $q = n - k \geq 3$ .
  - $n = 2^q - 1$
  - Independientemente del valor de  $q$ ,  $d_{\min} = 3$ :
    - Podemos detectar hasta 2 errores
    - Podemos corregir hasta 1 error.

# Códigos bloque

## Códigos Hamming

- Para crear la matriz de comprobación de paridad, colocamos todos los vectores binarios posibles en las columnas, ordenados de forma que quede la matriz identidad al final:

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



# Códigos bloque

## Códigos Hamming

- A partir de aquí, es inmediato calcular  $\mathbf{G}$ , sabiendo que  $\mathbf{H} = [\mathbf{P}^T | \mathbf{I}_q]$ :

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

# Decodificación dura y blanda

## Definición

- **Decodificación dura:**

- Es lo que hemos venido haciendo hasta ahora.
- La palabra recibida se decodifica bit a bit.
- Se elige aquella palabra código a una distancia Hamming menor.

- **Decodificación blanda:**

- En este caso se elige el símbolo a una menor distancia euclídea.
- La probabilidad de error va a ser menor

# Decodificación dura y blanda

## Ejemplo

- Supongamos lo siguiente:
  - Código de paridad con  $n = 3$ .
  - Utilizamos una modulación PAM unipolar con amplitudes 0V y 1V.
  - Suponemos además que:
    - Se transmite el símbolo **011**.
    - Es decir, transmitimos por el canal la señal 0V – 1V – 1V.
    - En el receptor obtenemos: 0,2V – 0,45V – 0,7V

# Decodificación dura y blanda

## Ejemplo

- Si utilizamos decodificación dura:
  - El receptor obtendrá:
    - $0,2V \rightarrow 0$
    - $0,45V \rightarrow 0$
    - $0,7V \rightarrow 1$
  - Es decir, recibimos el vector **001**, que sabemos que es incorrecto.
  - Hay cuatro posibles palabras código: **000**, **011**, **101** y **110**.
  - Las distancias Hamming de nuestra palabra a cada una de éstas es: 1, 1, 1 y 3 respectivamente.
  - Por tanto, elegiremos una de las tres primeras al azar.
  - Y acertaremos **1 de cada 3 veces**.

# Decodificación dura y blanda

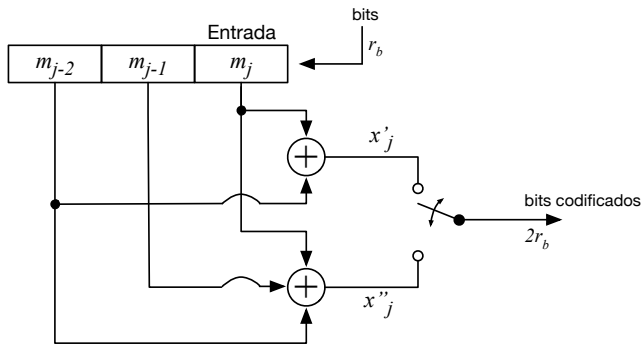
## Ejemplo

- Si utilizamos decodificación blanda:
  - Recibimos el mismo vector que antes:  $0,2V - 0,45V - 0,7V$ .
  - En este caso calculamos la distancia euclídea desde este vector a cada uno de los posibles símbolos de mi constelación:
    - $000 \rightarrow (0 - 0,2)^2 + (0 - 0,45)^2 + (0 - 0,7)^2 = 0,73$
    - $011 \rightarrow (0 - 0,2)^2 + (1 - 0,45)^2 + (1 - 0,7)^2 = 0,43$
    - $101 \rightarrow (1 - 0,2)^2 + (0 - 0,45)^2 + (1 - 0,7)^2 = 0,93$
    - $110 \rightarrow (1 - 0,2)^2 + (1 - 0,45)^2 + (0 - 0,7)^2 = 1,43$
  - Finalmente elegiremos el símbolo **011**.

# Códigos convolucionales

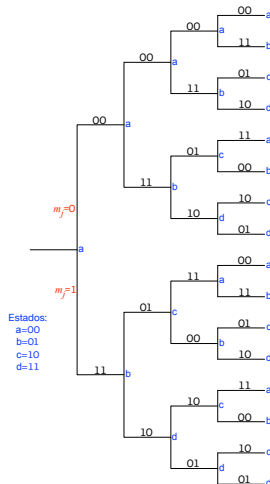
## Introducción

- Principal diferencia con lo anterior: tienen memoria
- Ejemplo: Código  $(n, k, L) = (2, 1, 2)$ .



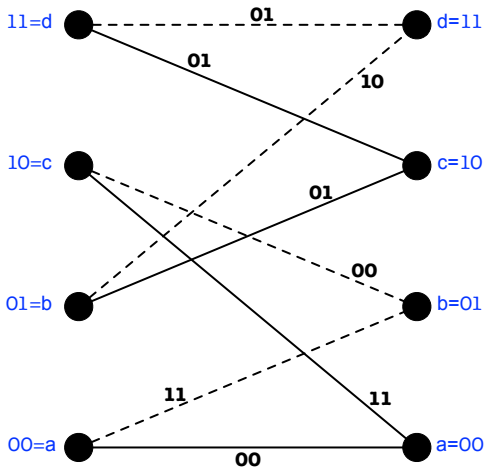
# Códigos convolucionales

## Árbol del código



# Códigos convolucionales

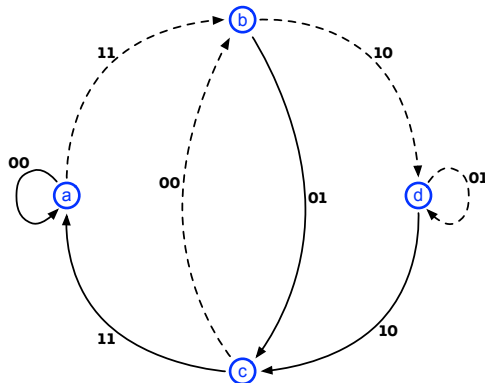
## Diagrama de rejilla





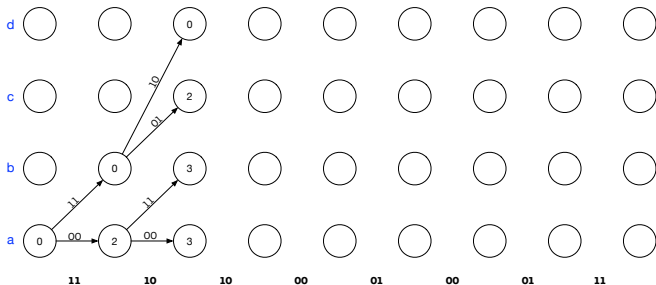
# Códigos convolucionales

## Diagrama de estados



# Códigos convolucionales

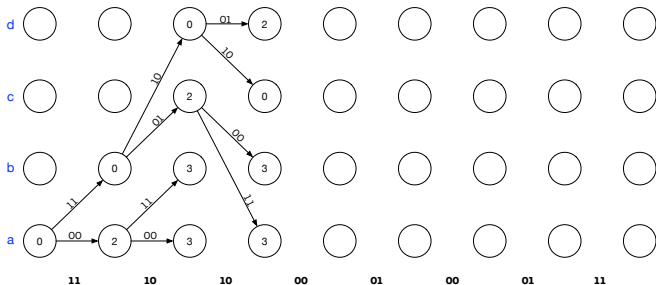
## Algoritmo de Viterbi





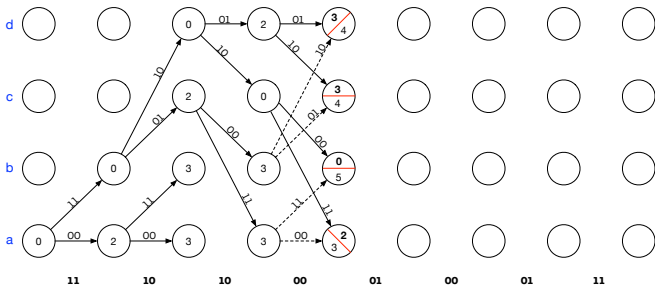
# Códigos convolucionales

## Algoritmo de Viterbi



# Códigos convolucionales

## Algoritmo de Viterbi



# Códigos convolucionales

## Algoritmo de Viterbi

