

# TEORÍA DE LAS COMUNICACIONES

## TRABAJO PRÁCTICO DE SIMULACIÓN

### PARTE 2: Codificación de fuente y canal

Ivan Svetlich

06/08/2021

## 1 Codificación de canal

### Códigos de bloque lineales

Un código de bloque lineal  $(n, k)$  puede representarse con una matriz generadora  $G$ , que mapea las  $2^k$  palabras de fuente posibles a sus correspondientes palabras de código de longitud  $n$ . Si la matriz  $G$  tiene la forma

$$G = [I_k | P] \quad (1)$$

donde  $I_k$  es una matriz identidad de  $k \times k$  y  $P$  es una matriz de  $k \times (n - k)$ , entonces el código resultante se denomina sistemático. En este tipo de códigos los primeros  $k$  elementos de la palabra de código son la secuencia del mensaje, y los siguientes  $n - k$  son los bits de paridad. Esta característica otorga una facilidad al momento de la decodificación, pues basta con seleccionar los primeros  $k$  elementos para recuperar los bits de fuente.

En este caso se requiere diseñar un código de bloque lineal  $(9, 5)$  con codificación sistemática, así que su matriz generadora será

$$G = [I_5 | P] \quad (2)$$

La matriz de paridad propuesta es

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

por lo que la matriz generadora resulta

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

Dado un código  $C$  asociado a una matriz  $G$ , su complemento ortogonal define un código  $C^\perp$  asociado a una matriz generadora  $H$ , cuyas filas son ortogonales a las de  $G$ . Como toda la palabra  $c$  perteneciente a  $C$  es ortogonal a las filas de  $H$

$$cH^t = 0 \quad (3)$$

Esta propiedad permite controlar si una palabra pertenece al conjunto de palabras de código de  $C$ , y por lo tanto  $H$  se denomina matriz de control de paridad. En el caso de los códigos sistemáticos  $H$  esta dada por

$$H = [P^t | I_{n-k}] \quad (4)$$

donde  $P^t$  denota la matriz de paridad  $P$  transpuesta. Para el código propuesto, la matriz de control de paridad resulta

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

Las capacidades de detección y corrección de errores de un código de bloque lineal están determinadas por la distancia mínima entre dos palabras de código  $d_{min}$ , que en este tipo de códigos coincide con el peso mínimo  $w_{min}$ . Para que se produzca un fallo en la detección de errores, estos se deben dar de forma tal que la secuencia recibida sea una palabra de código. Esto puede ocurrir solo si la cantidad de errores es mayor o igual a  $d_{min}$ . Entonces, la capacidad de detección es

$$t_d = d_{min} - 1 \quad (5)$$

De forma similar, un fallo en la corrección de errores puede ocurrir cuando la cantidad de errores en una secuencia sea tal que la palabra de código más cercana ya no sea la transmitida, sino otra de las palabras de código posibles. Entonces, la capacidad de corrección es

$$t_c = \left\lfloor \frac{d_{min} - 1}{2} \right\rfloor \quad (6)$$

Mediante simulación se halló que la distancia mínima del código propuesto es  $d_{min} = 3$ . Utilizando las ecuaciones (5) y (6)

- $t_d = 2$
- $t_c = 1$

La probabilidad de error de palabra de un sistema con codificación mediante un código de bloque lineal  $(n, k)$  corrector de  $t_c$  errores, utilizando decisión dura, es

$$P_{ep} = \sum_{i=t_c+1}^n p^i (1-p)^{n-i} \quad (7)$$

donde  $p$  es la probabilidad de error en el canal. Si  $p \ll 1$  esta expresión se puede aproximar como

$$P_{ep} \simeq \binom{n}{t_c+1} p^{t_c+1} \quad (8)$$

A partir de (8) se puede obtener el valor aproximado de la probabilidad de error de bit

$$P_{eb} \simeq \frac{t_c+1}{n} \binom{n}{t_c+1} p^{t_c+1} \quad (9)$$

El objetivo fundamental de la codificación es reducir la probabilidad de error de bit de fuente respecto de una comunicación sin codificación para una dada energía invertida, o bien reducir la energía necesaria para lograr una determinada probabilidad de error. La ganancia de un código se mide como

$$G_c = \frac{\frac{E_b}{N_0} \text{ con cod.}}{\frac{E_b}{N_0} \text{ sin cod.}} @ P_{eb} \quad (10)$$

La ganancia asintótica cuando  $P_{eb} \rightarrow 0$  o  $E_b/N_0 \rightarrow \infty$  determina una cota máxima para la ganancia de un código. En el caso de decisión dura se puede hallar como

$$G_a = \frac{k}{n} \left\lfloor \frac{d_{min} - 1}{2} \right\rfloor \quad (11)$$

Para el código propuesto su valor es

$$G_a = \frac{5}{9} \left\lfloor \frac{3+1}{2} \right\rfloor = \frac{10}{9} = 0.458 \text{ dB} \quad (12)$$

En la figura 1 se muestran la curva de probabilidad de error de bit de fuente cuando se utiliza el código como corrector de errores (en azul). A modo comparativo se incluyen las curvas de probabilidad de error obtenidas a partir de las expresiones teóricas para los casos con codificación (en rojo) y sin codificación (en verde). Se observa que la codificación es menos eficiente para valores bajos de  $E_b/N_0$  y supone una ganancia respecto de una comunicación sin codificación a partir de 6 dB aproximadamente. A medida que  $P_{eb} \rightarrow 0$  las curvas se tornan paralelas y la distancia entre ellas corresponde a la ganancia asintótica del código.

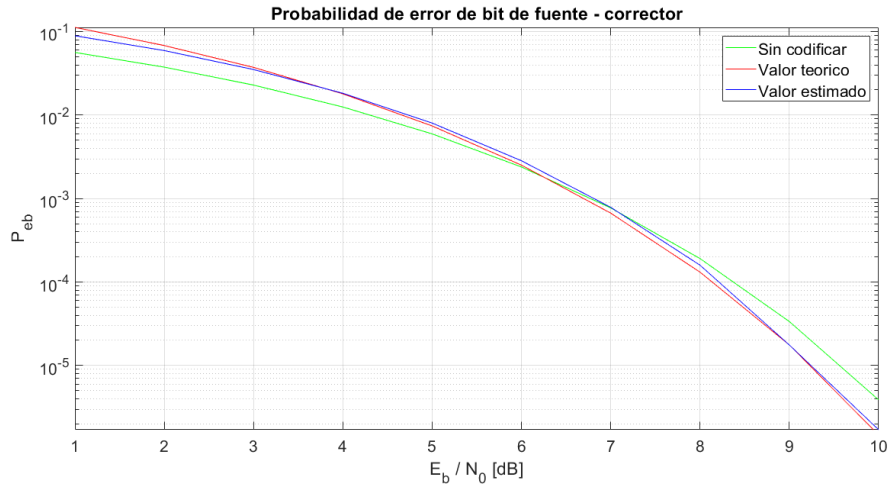


Figura 1: Probabilidad de error de bit de fuente (corrector).

De forma similar se releva la curva de probabilidad de error de fuente cuando se utiliza el código como detector de errores (figura 2). Evidentemente, la aproximación teórica utilizada es pesimista y resulta en una cota holgada para el valor obtenido mediante simulación. A diferencia del caso anterior, el código detector ofrece un mejor desempeño en todo el rango estudiado. Este resultado coincide con las capacidades de detección y corrección calculadas previamente.

## 2 Codificación de fuente

### Algoritmo de codificación de Huffman

El objetivo de los algoritmos de compresión de datos es representar a una fuente con el menor número de bits que permita su recuperación por parte

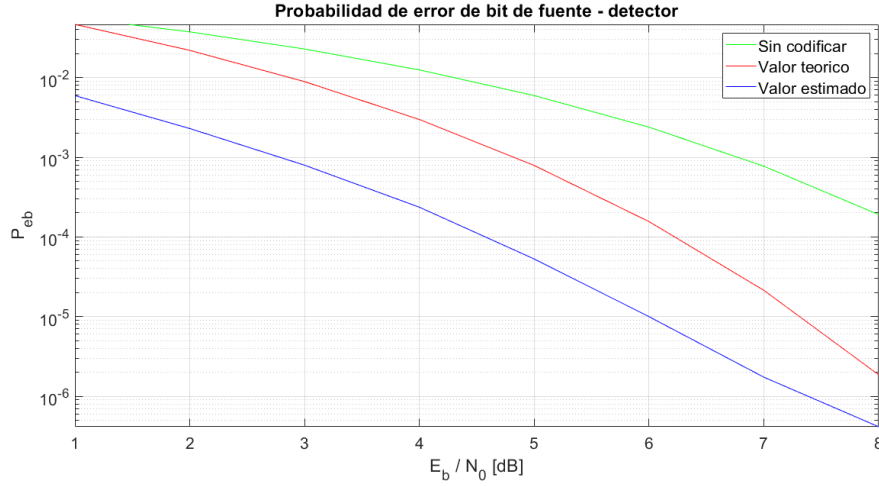


Figura 2: Probabilidad de error de bit de fuente (detector).

del receptor. Dependiendo de la tolerancia frente a la distorsión de la información, la compresión puede ser con o sin pérdidas. En una compresión sin pérdidas, la fuente se comprime de manera tal que la reconstrucción de la información es perfecta. En cambio, una compresión con pérdidas implica una distorsión de la fuente dentro de un límite tolerado.

El algoritmo de codificación de Huffman es uno de los principales métodos de compresión sin pérdidas de fuentes discretas. Se trata de un algoritmo óptimo, en el sentido de que el largo promedio de las palabras de código es el mínimo posible. Además, como las palabras de código están sujetas a la condición de prefijo (ninguna palabra es un prefijo de otra de mayor longitud), la secuencia recibida es unívoca e instantáneamente decodificable.

Según el teorema de codificación de fuente de Shannon, el mínimo largo de palabra promedio posible está determinado por la entropía de la fuente, si el código es unívoco. Con este criterio se define la eficiencia del código como

$$\eta = \frac{H(S)}{\bar{L}} \quad (13)$$

donde  $H(S)$  es la entropía y  $\bar{L}$  el largo promedio. Si para un código de prefijo se tiene que  $\bar{L} > H(S)$ , puede mejorarse utilizando una fuente extendida  $S^n$ , que se obtiene tomando como mensajes la combinación de  $n$  mensajes de la fuente original.

En este caso se requiere diseñar el algoritmo de Huffman para comprimir un archivo utilizando fuentes extendidas de orden 2 y 3. Para estimar las probabilidades de cada uno de los mensajes posibles, se calculan mediante simulación las frecuencias relativas dentro del archivo (tabla 1).

A partir de los valores hallados, se obtienen las palabras de código correspondientes a cada mensaje, siguiendo el criterio de que a los mensajes menos probables se les asigna el bit 1. En la tabla 2 se muestran las probabilidades resultantes del procedimiento, para fuente de orden 2.

Tabla 1: Fuente extendida de orden 2

| <b>1</b>          | <b>2</b>             | <b>3</b>               | <b>4</b>          |
|-------------------|----------------------|------------------------|-------------------|
| $p_3 = 0.50569$   | $p_3 = 0.47646$      | $p_3 = 0.50569$        | $p_{0,1,2,3} = 1$ |
| $p_0 = 0.47646$   | $p_0 = 0.47646$      | $p_{0,1,2} = 0.494317$ |                   |
| $p_2 = 0.0090886$ | $p_{2,1} = 0.017857$ |                        |                   |
| $p_1 = 0.0087686$ |                      |                        |                   |

En la tabla 3 se muestran las palabras de código correspondientes a cada mensaje.

Tabla 2: Palabras de código (orden 2)

| <b>Mensaje</b> | <b>Palabra de código</b> | <b>Longitud</b> |
|----------------|--------------------------|-----------------|
| $m_0 = 0\ 0$   | 1 0                      | 2               |
| $m_1 = 0\ 1$   | 1 1 1                    | 3               |
| $m_2 = 1\ 0$   | 1 1 0                    | 3               |
| $m_3 = 1\ 1$   | 0                        | 1               |

Aunque en el archivo, los dos mensajes posibles (pixel blanco y pixel negro) son prácticamente equiprobables, la utilización de fuentes extendidas logra compresión gracias a los patrones presentes en la imagen. Al haber zonas amplias de un mismo color, la frecuencia de los mensajes correspondientes a una sucesión de símbolos iguales es mucho mayor a la de las transiciones. Como la codificación de Huffman asigna palabras de código de menor longitud a los mensajes más probables, esto se traduce en una reducción del tamaño total del archivo.

## Referencias

- [1] S. Haykin. *Communications Systems, 4th Edition*. John Wiley & Sons, 2001.
- [2] J. Proakis and M. Saleh. *Digital Communications, 5th Edition*. McGraw Hill, 2007.