Teoría de la Información y Codificación

Trabajo Práctico de Simulación

Nihany Santiago (03012/3)
santi.nihany@gmail.com

Índice

1 Introducción

1.1 Resumen

El propósito de este trabajo ha sido investigar cómo un esquema de codificación de canal mejora la confiabilidad de un enlace digital en presencia de ruido aditivo gaussiano blanco (AWGN). Para ello se ha seleccionado un código de bloque lineal sistemático (n,k) = (10,14) combinándolo con una modulación BPSK de decisión dura. Se analizó el comportamiento de la comunicación utilizando un detector como corrector y detector. Además, se realizó una cuantificación de la ganancia real que aporta el código y una comparación con las fórmulas teóricas.

1.2 Marco teórico

1.2.1 Ruido AWGN y modulación BPSK

El **canal AWGN** (Additive White Gaussian Noise) se caracteriza por añadir a cada muestra de la señal transmitida un valor de ruido gaussiano con media cero y densidad espectral de potencia constante $\frac{N_0}{2}$. Matemáticamente, si $s(t)$ es la señal enviada, la recibida es:

$$r(t) = s(t) + n(t), n(t) \sim \mathbb{N}\left(0, \frac{N_0}{2}\right)$$

La modulación **BPSK** (Binary Phase Shift Keying) representa los bits mediante dos símbolos antipodales:

$$0 \to +A, 1 \to -A$$

donde la probabilidad de error de bit teórica del canal es:

$$P_{\text{eb}} = Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$$

1.2.2 Códigos de bloque lineales

Un **código de bloque lineal** (n, k) traduce **palabras de fuente** de longitud k a **palabras código** de longitud n.

Para realizar la codificación, se utiliza una **matriz generadora G** que convierte una palabra de fuente u en palabra código $v = u \cdot G$. Las filas de G son palabras de código y las palabras de código son combinación lineal de las filas de G. Esta matriz se define:

$$G = [I_k \mid P]$$

de modo que los primeros k bits de v coincidan con u, y la submatriz P aporta la redundancia.

Para la decodificación es utilizada la **matriz de verificación de paridad H^T**, que satisface $G \cdot H^T = 0$ y permite calcular el **síndrome** $s = v \cdot H^T$. En forma sistemática:

$$H = [P^T \mid I_{\text{n-k}}]$$

Cuando el cálculo del síndrome es cero, se puede afirmar que hubo error en la transmisión.

El **peso** (w_H) de una palabra de código es el número de sus elementos que son distintos de cero y la **distancia** entre dos palabras de código es la distancia de Hamming (d_H) entre ellos, es decir, el número de elementos en los que difieren.

Un código con $d_{\min}$ ($\min_{\mathrm{i,j}}\left\{d_{\mathrm{H}}\left(v_i, v_j\right)\right\}$) corrige hasta:

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

errores arbitrarios por palabra, y detecta hasta:

$$t_d = d_{\min} - 1$$

La **cota de Hamming** establece la cantidad de bits de redundancia $(n - k)$ necesarios en para formar un código de bloque lineal que puede corregir hasta t_c errores. Se expresa como:

$$\sum_{i=0}^{t_c} \binom{n}{i} \leq 2^{n-k}$$

El miembro izquierdo cuenta el número de esquemas de error que se deben poder corregir y el miembro derecho representa la cantidad de síndromes posibles. Esto implica que, para que un código pueda corregir t_c errores, debe haber suficientes síndromes distintos para representar todos los errores posibles.

1.2.3 Ganancia asintótica de codificación

La **tasa** del código es $R = \frac{k}{n}$. Bajo decisión dura, la ganancia de codificación asintótica en dB se define como:

$$G_a = 10 \log(R \cdot t_c) = 10 \log\left(\frac{k}{n} \cdot \left\lfloor \frac{d_{\min} - 1}{2} \right\rfloor\right)$$

Este valor indica el ahorro teórico de energía por bit de información al operar en la región asintótica de bajas tasas de error.

1.2.4 Demodulación

- **Decisión dura (hard-decision)**: el demodulador obtiene bits discretos $\{0, 1\}$ y el decodificador trabaja directamente sobre ellos. Es simple y rápido, pero desprecia la información de confiabilidad de cada bit.
- **Decisión blanda (soft-decision)**: se emplean los valores continuos recibidos (por ejemplo, log-likelihood ratios) para ponderar las probabilidades de cada bit, mejorando el rendimiento en aproximadamente 2–3 dB respecto a la decisión dura, a costa de mayor complejidad computacional y de memoria.

En este trabajo, se modelará el sistema utilizando decisiones duras.

2 Discusión

2.1 Diseño del sistema

2.1.1 Parámetros del código $(n = 14, k = 10)$

Para nuestro código de bloque trabajamos con palabras de longitud total $n = 14$, de las cuales $k = 10$ son bits de información y $n - k = 4$ bits de paridad. Aplicando la cota de Hamming,

$$\sum_{i=0}^{t_c} \binom{n}{i} \leq 2^{n-k} = 16$$

se comprueba que para $t_c = 1$ se cumple $\binom{14}{0} + \binom{14}{1} = 15 \leq 16$, mientras que para $t_c = 2$ ya supera el límite. De ello se deduce:

- Corrección de errores: $t_c = 1$
- Distancia mínima: $d_{\min} = 2t_c + 1 = 3$
- Detección de errores: $t_d = d_{\min} - 1 = 2$

Este código puede corregir automáticamente un error por palabra y detectar hasta dos sin riesgo de mal corrección.

2.1.2 Construcción de las matrices G y H

Se adopta la forma sistemática

$$G = [I_{10} \mid P]$$

con

$$P = \begin{pmatrix} 1100 \\ 0110 \\ 0011 \\ 1001 \\ 1010 \\ 0101 \\ 1101 \\ 1110 \\ 0111 \\ 1011 \end{pmatrix}$$

de modo que las filas de P sean distintas y con peso $w_H \geq d_{\min} - 1$.

La matriz de paridad se construye como:

$$H = [P^T \mid I_4]$$

2.2 Simulación

El rango de relación señal-ruido por bit empleado es $\frac{E_b}{N_0} = [1..7\text{dB}]$ (paso 0.25 dB).

2.2.1 Generación de secuencias de prueba

Para cada valor de $\frac{E_b}{N_0}$ se determina un número de palabras M suficientemente grande — al menos 10^6 o adaptado a la probabilidad teórica de error de bit— y se construye la matriz U con palabras de fuente aleatorias de largo k.

2.2.2 Codificación y modulación BPSK

Cada fila de U se convierte en palabra código multiplicando la matriz por la matriz generadora y obteniendo $V = U \cdot G$. Luego, el mapeo BPSK transforma ceros en +1 y unos en −1. El canal AWGN añade ruido gaussiano de varianza $\frac{N_0}{2}$ a cada símbolo.

2.2.3 Decodificación

A partir de la salida R, se calcula la matriz de síndromes:

$$S = R \cdot H^T$$

Utilizando las matrices R y S podemos obtener identificar los errores en la trasmisión y construir la matriz V_e con las palabras de código originales. Como definimos anteriormente, las filas de S no nulas indican que hubo un error de transmisión en la palabra de código asociada a esa fila. Podemos emplear dos modos para construir V_e:

1. **Modo corrector**: Cada síndrome s_i no nulo se compara con las filas de H^T. Si el síndrome concuerde con la fila número j, se invierte el bit j en la palabra recibida.

2. **Modo detector**: Se eliminan de R las palabras correspondientes a los síndromes no nulos.

2.2.4 Cálculo de métricas

Para calcular los errores de bit: obtenemos una matriz de error E comparando la matriz de bits estimados U_e (primeras k columnas de V_e) con los transmitidos U:

$$E = U \oplus U_e$$

La cantidad de **filas no nulas** de E será el número de **palabras erradas**. Se estima la P_{ep} dividiendo por la cantidad total de filas M.

La cantidad de **elementos no nulos** de E será el número de **bits errados**. Se estima la P_{eb} dividiendo por la cantidad total de elementos $M \times k$.

2.2.5 Cálculo de ganancia

La ganancia de código se mide para una P_{eb} dada:

$$G_c = \frac{\frac{E_b}{N_0} \text{ sin cod.}}{\frac{E_b}{N_0} \text{ con cod.}} |_{@P_{\text{eb}}}$$

Calculamos $\frac{E_b}{N_0}$ con cod. para cada iteración utilizando la función inversa de $Q(.)$ y la probabilidad de error de bit asociada. Obtenemos G_c restando estos valores por los $\frac{E_b}{N_0}$ sin cod. asociados a cada iteración.

3 Conclusiones