

Práctica 6

Códigos Hamming

Resumen

Vamos a ver ahora un tercer ejemplo de códigos de canal: los códigos bloque, y en particular los códigos Hamming.

1 Introducción

Los códigos bloque son una familia de códigos detectores/correctores de errores que se caracterizan por agrupar la información en bloques que se procesan de forma matricial. Un caso particular de códigos bloque son los códigos Hamming, aunque existen otros tipos como los códigos Reed-Solomon, los códigos Hadamard o los códigos Reed-Muller. Los códigos Hamming fueron desarrollados inicialmente en 1950 como método para corregir los errores en las tarjetas perforadas que se utilizaban en los ordenadores de la época. Este esquema en particular utilizaba cuatro bits de información a los que se añadían tres bits de redundancia.

Los códigos Hamming se caracterizan por tener la capacidad de detectar hasta dos errores y corregir hasta un error. Esto hace que se utilicen en aplicaciones donde la tasa de errores es baja, como por ejemplo en la memoria de ordenadores (las memorias ECC RAM utilizan este tipo de codificación).

2 Desarrollo de la práctica

1. Generaremos una función en Matlab que, dado un valor de q devuelva las matrices de generación y de comprobación de paridad de un Código Hamming:

```
function [G, H] = MatricesHamming(q)
```

2. Crearemos ahora una función en Matlab que, dado un array de dimensiones $1 \times N$ con bits para transmitir, genere un array $1 \times N \frac{n}{k}$ con las palabras código obtenidas al utilizar un código Hamming matriz generadora G .

La función debe tener el siguiente encabezado:

```
function codigo = CodificadorHamming(bits, G)
```

3. A continuación generaremos la función correspondiente al decodificador, que dado un vector $1 \times n \cdot N$, compruebe cada una de las palabras código y corrija aquellos errores que detecte.

```
function bits = DecodificadorHamming(codigo, H)
```

La función deberá, en primer lugar, generar la tabla de síndromes. Posteriormente se utiliza la matriz H para obtener el vector con los síndromes de decodificación y se compara cada uno de ellos con la entrada correspondiente de la tabla para obtener el vector corregido.

Para generar la tabla con los síndromes asociados a los errores más comunes vamos a considerar los $2^q - 1$ vectores de error $1 \times n$ con menor número de unos.

4. Para probar el sistema construido, vamos a volver a utilizar una vez más un vector con 10.000 bits aleatorios, que haremos pasar por el codificador, canal y decodificador configurados con $q = 3$ y $p = 0.02$:

```
bitsTx = randi([0,1],[1,1e4]);  
codigoTx = CodificadorHamming(bitsTx, 3);  
codigoRx = Canal(codigoTx, 0.02);  
bitsRx = DecodificadorHamming(codigoRx, 3);
```

¿Cuál es la probabilidad de error obtenida tras la decodificación?

5. Podemos comprobar también este codificador utilizando como entrada un mensaje de texto. Representad tanto el mensaje original, como el recibido (sin corregir los errores) y el corregido.

3 ¿Qué entregar?

- Todas las funciones creadas.
- Script del ejemplo de Matlab (códigos bloque), tanto con una señal aleatoria como con el texto.

4 Apartado opcional

Igual que habéis hecho en la práctica 4, podéis comprobar el sistema con un archivo de audio, y compararlo con el caso de un código de repetición.