

Práctica 4

Código de repetición

Resumen

Esta práctica introduce los códigos de canal más sencillos que existen: los códigos de repetición.

1 Introducción

Los códigos de repetición son quizás la alternativa más simple a la hora de generar un código de detección/corrección de errores. Se basan simplemente en repetir cada bit transmitido n veces (normalmente n es impar), de forma que en recepción lo único que hay que hacer es decidir qué bit se ha transmitido utilizando un criterio de mayoría.

A pesar de su simplicidad, los códigos de repetición se utilizan todavía en algunas aplicaciones, como por ejemplo formando parte de otros sistemas de codificación de canal conocidos como turbo códigos. Además, tienen la ventaja de que es posible modificar la tasa de codificación de forma dinámica según varíen las condiciones del canal de transmisión, algo que no muchos códigos son capaces de soportar.

2 Desarrollo de la práctica

1. Generaremos en primer lugar una función en Matlab que, dado un array de dimensiones $1 \times N$ con bits para transmitir, genere un array $1 \times (nN)$ con las palabras código obtenidas al utilizar un código de repetición con n repeticiones por bit.

La función debe tener el siguiente encabezado:

```
function codigo = CodificadorRepeticion(bits, n)
```

2. A continuación generaremos la función correspondiente al decodificador, que dado un vector $1 \times (nN)$, compruebe cada una de las palabras código y corrija aquellas en las que se detecte algún error tal y como hemos visto.

```
function bits = DecodificadorRepeticion(codigo, n)
```

3. Por último, generar una función que simule un canal que introduce errores en los bits con probabilidad p .

```
function codigoRx = Canal(codigoTx, p)
```

4. Para probar el sistema construido, simularemos un vector con 10.000 bits aleatorios, que haremos pasar por el codificador, canal y decodificador configurados con $n = 3$ y $p = 0.02$:

```
bitsTx = randi([0,1], [1,1e4]);  
codigoTx = CodificadorRepeticion(bitsTx, 3);
```

```
codigoRx = Canal(codigoTx, 0.02);  
bitsRx = DecodificadorRepeticion(codigoRx, 3);
```

¿Cuál es la probabilidad de error final obtenida? Comprobar que coincide con lo visto en teoría.

3 ¿Qué entregar?

- Todas las funciones creadas.
- Script del ejemplo de Matlab (códigos de repetición).

4 Apartado opcional

Con todas las funciones que habéis generado, podéis probar cómo funcionarían para el caso de un fichero de sonido.

Cargad el fichero de audio Brahms_mono.wav. Utilizad para ello la función `audioread`, con el parámetro extra '`native`' para que se carguen las muestras en formato entero de 16 bits, que es el original.

Ahora vamos a transmitir esta señal por un canal con una probabilidad de error de bit de 0.0001. Podéis escuchar el resultado utilizando las funciones `audioplayer` y `play`.