

Práctica 5

Códigos de paridad

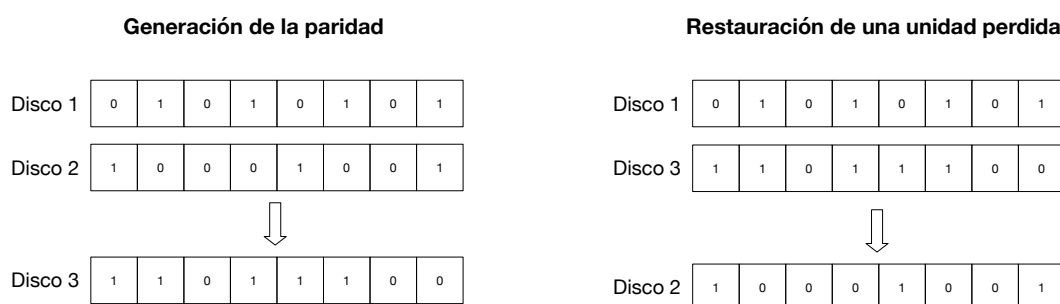
Resumen

Continuamos estudiando los códigos de canal con un segundo ejemplo: los códigos de paridad

1 Introducción

Los códigos de paridad representan otro ejemplo de códigos muy simples para la detección de errores. En este caso, dada una palabra de $n - 1$ bits, se añade un bit extra (bit de paridad) de forma que el número total de “unos” en la palabra código final sea par (paridad par) o impar (paridad impar). Su principal ventaja es la simplicidad, aunque tienen la desventaja de que sólo permiten detectar un número impar de errores y que no son capaces de corregirlos. Se utilizan habitualmente en aplicaciones hardware donde se puede repetir una operación en caso de encontrar algún error. Los buses PCI o SCSI utilizan este tipo de códigos.

Otra aplicación interesante es en los sistemas RAID de discos duros. En este caso, si uno de los discos falla, se puede reconstruir la información combinando la información de los otros discos. Si tenemos, por ejemplo, un sistema RAID-5 con tres discos duros en el que los dos primeros tienen la siguiente información: “01010101” y “10001001” respectivamente, la información del tercer disco se calcula siguiendo un esquema de paridad, de forma que quedaría: “11011100”. Si ahora falla cualquiera de los discos, su información se puede rescatar realizando la misma operación con los datos de los otros dos, tal y como se muestra en la siguiente figura:



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 $1 \times (N \frac{n}{n-1})$ con las palabras código obtenidas al utilizar un código de paridad par con palabras de n bits, siendo el último el de paridad.

La función debe tener el siguiente encabezado:

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

2. A continuación generaremos la función correspondiente al decodificador, que dado un vector $1 \times (N \frac{n}{n-1})$, compruebe cada una de las palabras código y marque aquellas en las que se haya detectado algún error, devolviendo un vector $1 \times N$.

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

3. Para probar el sistema construido, vamos a volver a utilizar 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 = CodificadorParidad(bitsTx, 3);  
codigoRx = Canal(codigoTx, 0.02);  
bitsRx = DecodificadorParidad(codigoRx, 3);
```

¿Cuál es la probabilidad de que no se haya detectado algún error?

3 ¿Qué entregar?

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