Práctica 2
Codificación de canal

1 Introducción

Además del fichero CodFuente.py que hemos utilizado en la Práctica 1, en esta práctica utilizaremos algunos archivos adicionales:

- `Practica2a.ipynb`, `Practica2b.ipynb`, `Practica2c.ipynb` y `Practica2d.ipynb`: Se trata de los archivos principales de cada una de las partes de la práctica, que serán los que deberemos ejecutar.
- `CodCanal.py`. Archivo con una serie de funciones relacionadas con la codificación de canal
  - **repeticion_cod**
  - **repeticion_dec**
  - **paridad_cod**
  - **paridad_dec**
  - **matrices_hamming**
  - **hamming_cod**
  - **hamming_dec**
- `Canal.py`. Archivo con las funciones necesarias para simular un canal de transmisión.
  - **canalAWGN**

Dentro del archivo *CodCanal.py* veréis que faltan por desarrollar algunas funciones que se han dejado en blanco:

- **numErrores**: Esta función debe tomar a su entrada los mensajes a la entrada y a la salida del sistema en formato *string* y devolver a su salida el número de errores detectados entre ambos:
- **repeticion_cod**: Esta función debe tomar a su entrada un mensaje en formato binario (*String*), y el parámetro k del codificador de repetición, y devolver a su salida el mensaje codificado.
- **repeticion_dec**: Esta es la función inversa a la anterior. Dado un mensaje recibido del que sabemos que ha sido codificado con un código de repetición con parámetro k, debe devolver el mensaje binario decodificado, con los posibles errores corregidos.

2 Códigos de repetición

En el fichero `Practica2a.ipynb` se encuentra un ejemplo de códigos de repetición. Los códigos de repetición son quizás la opción más simple de código corrector de errores. Se basan sencillamente en repetir cada uno de los bits n veces (normalmente n es impar) de forma que en recepción se decide qué bit se ha transmitido votando por mayoría en esos n valores recibidos.

Para poder ejecutar este ejemplo debéis completar las funciones mencionadas anteriormente: `numErrores`, `repeticion_cod` y `repeticion_dec`.

3 Códigos de paridad

Los códigos de paridad son más eficientes que los de repetición en cuanto a la cantidad de información extra a transmitir pero no permiten la corrección de errores, sino sólo su detección. En el archivo `Practica2b.ipynb` hay un ejemplo de este tipo de códigos. Tan sólo debéis completar el código necesario para calcular el tamaño del archivo en cada una de las fases de la transmisión.

4 Códigos Hamming

Una alternativa más elaborada de código de canal son los códigos bloque, y en particular los códigos Hamming. Este tipo de códigos permiten tanto la detección como la corrección de errores (en concreto la detección de 2 y la corrección de 1 error). Su funcionamiento es bastante sencillo, y se basa en operaciones algebraicas con dos matrices: una para codificar y otra para decodificar la información. Podéis encontrar este ejemplo en el archivo `Practica2c.ipynb`.

5 Sistema completo

Este último ejemplo muestra un sistema de comunicaciones completo, incluyendo codificador de fuente, codificador de canal, modulador y canal. Se permite además la entrada de archivos de texto, imágenes o ficheros de audio. Podéis probar con distintas combinaciones de codificadores, moduladores y canales (cantidad de ruido) y ver el efecto en el sistema global.

6 ¿Qué hay que entregar?

- Todos los archivos con el código completado tal y como se indica antes.
- Un documento de texto en el que se responda a lo siguiente:
  - ¿Qué diferencias aprecias entre los distintos codificadores de canal en función del ruido en el canal?
  - ¿Qué conclusiones puedes extraer a partir de las pruebas con el simulador del sistema global?