

TRANSMISIÓN DE DATOS 2007/08		
Práctica 7: Sistema Codificación-Decodificación (Fuente y Canal)		Grupo
		Puesto
Apellidos, nombre		Fecha
Apellidos, nombre		Enero 2008

El objetivo de esta práctica es el de familiarizar al alumno con el proceso completo de codificación y decodificación de fuente, simulación de transmisión a través de un canal con ruido y utilización de codificación de canal para protección frente a errores.

En el fichero FuenteCanal.zip se encuentra el código y ficheros de prueba necesarios para la realización de la práctica. Para ello descomprima y copie los ficheros al directorio de realización de la práctica.

Justo antes de finalizar la práctica, comprima el proyecto en un fichero TxDatosP7GXzz.zip (siendo X el grupo -A, B ó C-, y zz el número de pareja) conéctese al sistema de entrega de prácticas de la Intranet y entréguelo en el grupo que corresponda (A, B o C). Guárdese adicionalmente una **copia personal**, para posibles futuras reutilización del código en prácticas posteriores. Recuerde borrar su trabajo del ordenador del puesto de prácticas.

1 Introducción

En esta práctica se llevará a cabo la implementación del proceso completo de codificación de fuente y simulación de transmisión de la información a través de canales con ruido haciendo uso de la codificación de canal para protección frente al mismo. Posteriormente se llevará a cabo la decodificación de la información transmitida (canal y fuente).

Para ello se implementarán diferentes sistemas siguiendo el siguiente esquema genérico:



A partir de una fuente de información original se llevará a cabo el proceso de codificación de fuente con alguno de los métodos estudiados: codificación *Huffman*, cuantificación PCM/G711 o codificación transformacional. Sobre los datos obtenidos se llevará a cabo una codificación de canal (códigos lineales o códigos convolucionales). Finalmente se simulará el envío de la información a través de un canal con ruido y se observarán los efectos del ruido y de la aplicación de códigos de canal en la transmisión.

El conjunto de sistemas a desarrollar serán los siguientes:

- Codificación/decodificación de una cadena de texto mediante codificación *Huffman*. Protección frente a errores mediante códigos lineales.
- Codificación/decodificación de muestras de audio mediante codificación PCM. Protección frente a errores mediante códigos convolucionales.

2 Codificación Huffman + Códigos Lineales

En este apartado se implementará un sistema de codificación/decodificación mediante el método *Huffman* simulando el envío a través de un canal con ruido y añadiendo una codificación mediante códigos lineales para protección frente a errores en la transmisión. Se proporcionan las funciones `decodificaHuffman` y `generarRuido` utilizadas en prácticas anteriores (ver detalles en práctica 3).

Para la decodificación tras la aplicación de códigos lineales se proporciona la función `decodificacionCodLineal` con la siguiente especificación:

```
function [secuenciaDecodificada] = decodificacionCodLineal(secuenciaCodificada, n,k,
                                                         matrizStandard, H)
```

- **Recibe:**
 - `secuenciaCodificada` == Una secuencia de bits (como vector fila) resultado de la codificación mediante la aplicación de un código lineal $C(n,k)$. Nótese que las funciones implementadas para codificación por códigos lineales generan matrices (una palabra código por fila) mientras que esta función recibe los códigos como una secuencia.
 - `n` == longitud (en bits) de cada palabra codificada.
 - `k` == longitud (en bits) de cada palabra original.
 - `matrizStandard` == Matriz standard asociada al código lineal $C(n,k)$.
 - `H` = matriz de chequeo de paridad asociada al código lineal $C(n,k)$.
- **Devuelve:**
 - `secuenciaDecodificada` == La secuencia de bits resultado de la decodificación con corrección de errores.

Importante: La función proporcionada hace uso de la función **corregirPalabra** implementada en la práctica 5. Por tanto es necesario que dicha función sea completada y esté disponible en el directorio de trabajo para el correcto funcionamiento del proceso de decodificación.

2.1 Ejercicio 1

Complete el script que se proporciona en el fichero `ejercicio1.m` para la implementación y prueba de un sistema de codificación/decodificación *Huffman* mediante la codificación y posterior decodificación de la cadena de texto “*Esta es una prueba de codificación y decodificación Huffman con codificación de canal para la práctica 7.*”. Para simular la protección frente a errores en la transmisión se llevará a cabo una codificación de canal mediante códigos lineales.

En el primer paso se deberá llevar a cabo la codificación mediante *Huffman* de la cadena de texto especificada. Esta codificación se utilizará como base para los pasos siguientes.

En el segundo paso se simulará el envío y posterior decodificación de la información a través de un canal con ruido. Para ello genere una máscara de ruido con los siguientes parámetros: *probabilidadRafaga=0.03*, *longitudMediaRafaga=1*, *variacionLongitudRafaga=0*, *semillaAleatoria=12*. Aplique la máscara de ruido a la información codificada y lleve a cabo su decodificación. Esta decodificación servirá de referencia para comparar con la codificación mediante códigos lineales.

En el tercer paso se simulará el proceso completo de codificación *Huffman* seguida de codificación de canal mediante códigos lineales. Para ello partiendo de la codificación *Huffman* del primer paso aplique una codificación mediante código lineal $C(5,2)$. Lleve a cabo el proceso de decodificación utilizando la función `decodificacionCodLineal` proporcionada (seguida de `decodificaHuffman`). Compruebe que el proceso completo de codificación y decodificación mostrado en la figura se realiza correctamente.



Nota: Recuerde que las funciones implementadas para codificación lineal reciben matrices en las que cada fila es una palabra a codificar utilice las funciones `filasEnSecuencia` y `secuenciaEnFilas` para la implementación del ejercicio.

Rellene la siguiente tabla indicando el número de bits necesarios para enviar la cadena de texto anterior en los siguientes casos: **a)** sin codificación, **b)** codificación mediante codificación *Huffman* y **c)** codificación mediante *Huffman* y código lineal C(5,2).

Sin Codificación	<i>Huffman</i>	<i>Huffman</i> + Cod. Lineal C(5,2)

Complete el cuarto paso en el que además de llevar a cabo la codificación de la cadena de texto mediante *Huffman* y posterior aplicación de codificación lineal C(5,2) (tal y como se realiza en el paso 3) se simule la transmisión a través de un canal con ruido. Genere una máscara de ruido de longitud adecuada con los siguientes parámetros: *probabilidadRafaga*=0.03, *longitudMediaRafaga*=1, *variacionLongitudRafaga*=0, *semillaAleatoria*=12. Posteriormente lleve a cabo el proceso de decodificación de la cadena de texto. Compare los resultados con los del paso 2 (canal con ruido sin codificación. lineal) y comente los resultados:

Complete el paso 5 llevando repitiendo el mismo experimento que en el paso anterior pero en este caso aplicando una codificación mediante código lineal C(6,2). ¿Se obtiene alguna mejora en el resultado? Comente los resultados obtenidos en los pasos 4 y 5 mediante códigos lineales con diferentes parámetros.

Comente la relación entre los bits necesarios para la codificación en los distintos experimentos implementados y la resistencia frente a errores en el canal de comunicación ofrecida:

3 Codificación PCM + Códigos Convolucionales

En este apartado se implementará la secuencia completa de codificación y decodificación mediante PCM realizando además la simulación de un canal con ruido y añadiendo una codificación convolucional para protección frente a errores en la transmisión. Para ello se utilizarán como funciones auxiliares las funciones `getSecuenciaBits` utilizada para convertir una secuencia de valores a una secuencia de bits y `getSecuenciaNumerica` para el proceso inverso (véase práctica 3). Así mismo se proporcionan las funciones `secuenciaEnFilas` y `filasEnSecuencia` necesarias para transformar las secuencias de bits a codificar en matrices aceptadas por las funciones para la codificación convolucional.

3.1 Ejercicio 2

Complete el script suministrado *Ejercicio2.m* siguiendo las indicaciones del mismo. En primer lugar se realizará una codificación PCM de la muestra de audio suministrada con 8 bits y valor de sobrecarga $V=4$ (Nota: En este caso se limitará la longitud de la muestra a los 1000 primeros elementos para permitir una rápida ejecución de las pruebas). Se utilizará la función `getSecuenciaBits` para obtener la secuencia de bits correspondiente. A continuación (paso 3) se llevará a cabo la decodificación de la secuencia de bits mediante el proceso inverso (utilizando `getSecuenciaNumerica` y `decodificaPCM`).

En el cuarto paso se llevará a cabo una primera simulación de canal ruidoso. Genere una máscara de error para la secuencia de bits codificada. Utilice los siguientes parámetros: *probabilidadRafaga=0.1*, *longitudMediaRafaga=1*, *variacionLongitudRafaga=0*, *semillaAleatoria=2*. ¿Cuál es el error de reconstrucción? :

En el paso 5 se añadirá una codificación convolucional tras la codificación PCM con parámetros $n=4$ y $k=2$. Utilícese para ello la matriz proporcionada en el script. Recuerde que para la codificación será necesario pasar la secuencia de bits a filas con la función `secuenciaEnFilas` y realizar el proceso inverso tras la decodificación (función `filasEnSecuencia`). ¿Cuántos bits son necesarios para la codificación de las 1000 muestras con y sin codificación convolucional? ¿A qué se debe la diferencia?

Escriba la potencia de error obtenida tras la decodificación:

En el paso 6 se deberá repetir el experimento realizando la codificación PCM seguida de una codificación convolucional. En este caso se simulará la transmisión a través de un canal con ruido. Para ello genere una secuencia de ruido con los siguientes parámetros: *probabilidadRafaga=0.1*, *longitudMediaRafaga=1*, *variacionLongitudRafaga=0*, *semillaAleatoria=2*.

Lleve a cabo la decodificación de las muestras ¿Cuál es la potencia de error obtenida en este caso?

Compare los resultados obtenidos en este experimento con aquellos en los que se simulaba una transmisión con canal ruidoso sin la aplicación de codificación convolucional. ¿En cual de los dos casos se obtiene una mejor transmisión? ¿Por qué? ¿Cuáles son las ventajas y desventajas de cada uno de los métodos?

A continuación lleve a cabo una modificación en el script de forma que todas las secuencias de error se generen con $probabilidadRafaga=0.01$. ¿Qué resultados se obtienen comparando la transmisión sin ruido, con ruido y aplicando o no la codificación convolucional? Escriba y comente los resultados:

Por último se llevará a cabo una prueba reduciendo el número de bits para la codificación PCM. La probabilidad de error para las secuencias de ruido generadas será $probabilidadRafaga=0.1$. Rellene la siguiente tabla con las potencias de error obtenidas:

Codificación PCM	Sin canal ruidoso Sin cod. convolucional	Con canal ruidoso Sin cod. convolucional	Con canal ruidoso Con cod. Convolucional
8 bits			
6 bits			
4 bits			

Repita el experimento con probabilidad de error $probabilidadRafaga=0.01$. Rellene la siguiente tabla:

Codificación PCM	Sin canal ruidoso Sin cod. convolucional	Con canal ruidoso Sin cod. convolucional	Con canal ruidoso Con cod. Convolucional
8 bits			
6 bits			
4 bits			

Compare los resultados y comente los resultados y conclusiones obtenidos.