

TRANSMISIÓN DE DATOS 2006/07		
Práctica 5: Códigos Lineales		Grupo
		Puesto
Apellidos, nombre		Fecha
Apellidos, nombre		30 Noviembre – 14 Diciembre 4 Diciembre – 11 Diciembre

El objetivo de esta práctica es familiarizar al alumno con la codificación de canal mediante códigos lineales.

Para llevar a cabo la práctica, desarrolle cada ejercicio en un fichero MATLAB dentro del directorio P5.

Se proporcionan, accesibles desde la página de la asignatura una serie de ficheros como guía para las funciones que se deben implementar. Todos estos ficheros están accesibles en el fichero TxDatosP5GXX.zip.

Además del código de las funciones que se piden en cada apartado se debe adjuntar los scripts que se desarrollen para poner a prueba dichas funciones y completar los ejercicios propuestos.

Justo antes de finalizar la práctica, comprima todo el directorio P5 en un fichero TxDatosP5GXX.zip (donde G indicará el grupo, A o B, y XX el puesto del laboratorio asignado), conéctese al sistema de entrega de prácticas de la Intranet y entréguelo en el grupo que corresponda (A o B). Guárdese adicionalmente una **copia personal**, para posibles futuras reutilización del código en prácticas posteriores.

2.1 Códigos Lineales

Un código (n,k) está formado por 2^k secuencias de longitud n , correspondiendo cada una de las palabras código con uno de los posibles 2^k mensajes. Según se seleccionen las palabras códigos y se asignen mensajes con palabras códigos tendremos distintos códigos con mejores o peores prestaciones.

Los códigos lineales son códigos que cumplen la siguiente propiedad:

Cualquier combinación lineal (suma módulo 2 - \oplus -) de palabras código es palabra código. (1)

Esta condición implica que la palabra código nula (todos sus elementos nulos) será siempre palabra código.

Adicionalmente se suele trabajar con el subconjunto de códigos lineales que cumplan que la propiedad siguiente:

La palabra código correspondiente a un mensaje suma de dos mensajes es igual a la suma de las palabras códigos correspondientes

$$\text{Sean } C(x1)=c1 \text{ y } C(x2)=c2, \text{ entonces } C(x3=x1 \oplus x2)=c1 \oplus c2 \quad (2)$$

Para generar las palabras códigos se pueden usar las condiciones anteriores. Sin embargo, normalmente se suele hacer uso de una matriz generatriz (o generadora). Esta matriz está formada por las palabras código correspondientes a los mensajes $(1\ 0\ 0 \dots 0)$, $(0\ 1\ 0 \dots 0)$, $(0\ 0\ 1 \dots 0)$, ..., $(0\ 0\ 0 \dots 1)$.

Generalmente se hace uso de la matriz generatriz en forma sistemática, que se caracteriza por asignar a los mensajes palabras código que empiecen (o terminen) por el mismo.

$$G = [I_k | P]$$

Donde I_k es la matriz unidad $[k \times k]$ y P la matriz de paridad $[k \times (n-k)]$.

Para calcular la palabra código c que corresponde a un mensaje x (gracias a las propiedades de linealidad) basta con multiplicar el mensaje por la matriz G

$$c = x G$$

Para detectar la presencia de un error se hace uso de la matriz de chequeo de paridad, que se obtiene a partir de la matriz G ($G H^T = 0$)

$$H = [P^T | I_{(n-k)}]$$

Para chequear la presencia de un error (puede haber errores que no se detecten) se calcula el síndrome

$$s = c H^T$$

Si el síndrome es distinto del vector nulo se detecta un error, que puede ser corregible o no.

Para la corrección de errores se hace uso de la matriz estándar, que se crea sistemáticamente según se vio en teoría:

- La primera fila está formada por las palabras código, empezando por la palabra nula que forma parte de cualquier código lineal
- A partir de la primera fila, cada fila se forma sumando a la primera fila un COSET LEADER que es una secuencia de n bits de peso mínimo que no aparecido ya en las filas anteriores.
- Esto se repite hasta completar $(n-k)$ filas, que por k columnas, genera una matriz con todas las posibles secuencias de n bits.
- Los COSET LEADERS son los patrones de error corregibles.

Para corregir errores, se calcula el síndrome de la palabra recibida y la de los COSETs, y se suma a la palabra recibida el COSET LEADER de igual síndrome. Si el patrón de error detectado coincide con el que dio lugar al error, el código corrige el error. En caso contrario (el patrón de error no es uno de los líderes) el error no se corrige (y en algunos casos podría no detectarse).

2.1.1 Ejercicio 1: Desarrollo de generador de códigos lineales

Desarrollar una función MATLAB que devuelva una matriz generatriz sistemática G de un código (n,k) .

```
function [G]=matrizGeneratriz(n,k)
```

Escribir la matriz generatriz generada del código lineal $(5,3)$

Escribir la matriz generatriz generada del código lineal (7,3)

(Nota:Recuerdese adjuntar el script con el que se llama a la funcion implementada con los distintos ejemplos)

Desarrollar una función MATLAB que reciba un mensaje x (cada palabra del mensaje como una fila de la matrix x) y una matriz generatriz G y devuelva las palabras de código correspondientes a cada palabra del mensaje (cada palabra código en una fila de la matrix 'codigo').

```
function [codigo] = codifica(x,G)
```

Escribir los mensajes del código (5,3) y sus correspondientes palabras código.

<i>Mensaje</i>	<i>Palabra código</i>

Escribir los mensajes del código (7,3) y sus correspondientes palabras código.

<i>Mensaje</i>	<i>Palabra código</i>

(Nota:Recuerdese adjuntar el script con el que se llama a la funcion implementada con los distintos ejemplos)

2.1.2 Ejercicio 2: Codificación lineal

Escribir un script MATLAB que haciendo uso de las funciones implementadas anteriormente lleve a cabo las siguientes tareas:

Generar aleatoriamente una secuencia de 21 bits y almacenarla.

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Codificarla con los códigos anteriores.

(5,3) Teóricamente

--

(5,3) Programa Matlab

--

(7,3) Teóricamente

--

(7,3) Programa Matlab

--

2.1.3 Ejercicio 3: Desarrollo de detector de errores

Desarrollar una función MATLAB que devuelva la matriz de chequeo de paridad H correspondiente a una matriz generatriz G, teniendo en cuenta los valores recibidos de n y k.

```
function [H] = matrizChequeoParidad(G, n, k)
```

Escribir la matriz H del código (5,3)

Escribir la matriz H del código (7,3)

(Nota: Recuérdese adjuntar el script con el que se llama a la función implementada con los distintos ejemplos)

2.1.4 Ejercicio 4: Cálculo de síndrome

Desarrollar una función MATLAB que reciba una secuencia de palabras de código c (cada palabra de código como una fila de la matriz c) y una matriz de chequeo de paridad H y devuelva los síndromes correspondientes a cada palabra de código recibida (cada síndrome en una fila de la matriz 'síndromes'). La función devolvera también un vector columna 'errores' en el que se indicara, para cada palabra de código, un '0' si es una palabra de código válida o un '1' si contiene error.

```
function [síndromes, errores] = calculoSíndrome(c, H)
```

Calcular el síndrome de las secuencias código generadas anteriormente.

(5,3) Teóricamente

(5,3) Programa Matlab

(7,3) Teóricamente

(7,3) Programa Matlab

(Nota:Recuerdese adjuntar el script con el que se llama a la funcion implementada con los distintos ejemplos)

2.1.5 Ejercicio 5: Detección de errores

Escribir un script MATLAB que haciendo uso de las funciones implementadas anteriormente (en los casos en que proceda) lleve a cabo las siguientes tareas:

Calcular una secuencia de errores para la secuencia de código (5,3) con probabilidad de error 0,2

--

Escribir la secuencia recibida correspondiente a la secuencia código y al error generados.

--

Calcular una secuencia de errores para la secuencia de código (7,3) con probabilidad de error 0,2

--

Escribir la secuencia recibida correspondiente a la secuencia código y al error generados.

--

Calcular el síndrome de las secuencias código con error generadas anteriormente.

(5,3) Teóricamente

(5,3) Programa Matlab

(7,3) Teóricamente

(7,3) Programa Matlab

2.1.6 Ejercicio 6: Cálculo de la matriz estándar

Desarrollar una función MATLAB que a partir de la lista de palabras código correspondientes a un código (especificadas en la matriz 'palabrasCodigo' fila por fila) devuelva la matriz standard correspondiente a dicho código (se recomienda que dicha matriz se implemente como un tipo 'cell'):

```
function [matrizStandard] = generaMatrizStandard(palabrasCodigo)
```

Calcule la matriz estándar correspondientes al código anterior (5,3)

Desarrolle un script MATLAB que intente corregir los errores del código (5,3)

Escriba el mensaje decodificado

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Mensaje original

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Y comente el resultado en función del código utilizado.

--