

PRACTICA 1

INTRODUCCION A LA SIMULACION DE

SISTEMAS DE COMUNICACIONES EN

LENGUAJE MATLAB

LA PRACTICA 1 SE DIVIDE EN TRES PARTES:

-PRACTICA 1A: CALCULO MATRICIAL

-PRACTICA 1B: SIMULACION DE UN AMPLIFICADOR CLASE C

-PRACTICA 1C: CODIFICACION FUENTE

PRACTICA 1A: CALCULO MATRICIAL

El objetivo de esta práctica es la familiarización del alumno con algunas de las herramientas para cálculo matricial que tiene MATLAB, así como la implementación de alguna función que complemente las propias de MATLAB.

En primer lugar se generarán dos matrices cuadradas de orden n , de forma aleatoria. Para ello el programa deberá preguntar el orden de la matriz. La distribución utilizada para la generación de los números aleatorios será una gaussiana.

GENERACION DE LAS MATRICES

```
%*****PRACTICA_1A.m*****
% TITULO: CALCULO MATRICIAL
% ASIGNATURA: LABORATORIO DE SIMULACION DE COMUNICACIONES
% GRADO EN INGENIERIA EN SISTEMAS DE TELECOMUNICACION
close all % Cierra todas las pantallas que hubiera anteriormente
clear % Borra todas las variables
clc % Borra pantalla
n=input('MATRIZ CUADRADA DE ORDEN: ');
randn('state',0) % Para que todos obtengamos las mismas matrices
                % randn -->distribución gaussiana
A=fix(10*randn(n)) % Para que las matrices sean de números enteros
                % fix(x) nos da el número entero de x.
a=det(A) % Para determinar que la matriz obtenida no es singular
B=fix(10*randn(n))
b=det(B), pause
clc
```

MANIPULACION DE LAS MATRICES

Con las matrices generadas, se deberán realizar las siguientes comprobaciones:

- 1) La transpuesta de la suma de dos matrices es la suma de las transpuestas, esto es:

$$(A+B)' = A' + B'$$

- 2) La transpuesta del producto de dos matrices es el producto de las transpuestas en orden inverso, esto es:

$$(A*B)' = B'*A'$$

- 3) El determinante del producto de dos matrices cuadradas es el producto de los determinantes de cada una de ellas, y además cumple la propiedad conmutativa, esto es:

$$|A*B| = |A|*|B| = |B*A|$$

- 4) Si A y B son dos matrices no singulares (matriz singular es una matriz cuadrada cuyo determinante es cero), el producto de A por B es una matriz no singular y se cumple:

$$\text{inv}(A*B) = \text{inv}(B)*\text{inv}(A)$$

MANIPULACION DE LAS MATRICES

5) Si A es una matriz no singular, se cumple:

$$\text{inv}(A') = (\text{inv}(A))'$$

6) Si k es escalar (tomar $k = 5$) y A es una matriz cuadrada no singular, entonces se cumple:

$$\text{inv}(k \cdot A) = (1/k) \cdot \text{inv}(A)$$

7) El determinante de la inversa de una matriz es igual a la inversa el determinante de la matriz, esto es:

$$|\text{inv}(A)| = 1/|A|$$

8) Si A es una matriz cuadrada con determinante distinto de cero su inversa se puede calcular como:

$$\text{inv}(A) = (1/|A|) \cdot [\text{adjuntos}]$$

NOTA: Para la realización de la última comprobación será necesario implementar una función que calcule los adjuntos de una matriz, a la cual se la llamará `matr_adj(A)`. Como es necesario calcular los adjuntos uno a uno, se aconseja que esta función llame a otra, denominada `adjuntos(A,i,j)` que nos devuelva el adjunto del elemento de la matriz a_{ij}

PRACTICA 1B: SIMULACION DE UN AMPLIFICADOR CLASE C

El objetivo de esta práctica es que el alumno maneje algunas de las funciones básicas que ofrece MATLAB, desarrolle nuevas funciones y utilice el entorno gráfico para la presentación de datos. Para ello deberá emplear, siempre que sea posible, las funciones de MATLAB y en caso de que no sea posible, deberá desarrollar sus propias funciones. En todo caso, los datos se deben presentar gráficamente.

AMPLIFICADOR CLASE C

Se va a suponer que un amplificador ideal en clase C (ver la siguiente figura) se compone de los siguientes bloques:

- 1.- Recortador de amplitud
- 2.- Amplificador lineal
- 3.- Filtro paso bajo



AMPLIFICADOR CLASE C

Si $x(t)$ es la señal a la entrada del recortador de amplitud a su salida se obtiene la señal $x_L(t)$ relacionada con la entrada por la expresión:

$$x_L(t) = \begin{cases} x(t) - L; & \text{si } x(t) \geq L \\ 0; & \text{Resto} \end{cases}$$

Así el comportamiento de este bloque es semejante al que produce un rectificador de media onda.

La señal a la salida del amplificador lineal es la señal de entrada amplificada g veces.

$$x_A(t) = g \cdot x_L(t)$$

El filtro paso bajo es un filtro de Butterworth de orden 7, el cual se puede simular digitalmente por un filtro IIR que cumple con la siguiente función de transferencia:

$$H(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_7 \cdot z^{-7}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_7 \cdot z^{-7}}$$

AMPLIFICADOR CLASE C

La simulación consiste en introducir un tono de amplitud de 5 voltios y frecuencia 1MHz, al que se le hace pasar por el amplificador en clase C mencionado anteriormente. Como salida se debe obtener:

- a) La señal a la entrada del amplificador clase C.
- b) El espectro de la señal a la entrada del amplificador clase C. Para calcular el espectro de las señales, utilizar la función *spectrum*, dada a continuación.
- c) La señal a la salida del recortador.
- d) El espectro de la señal a la salida del recortador.
- e) La señal a la salida del amplificador lineal.
- f) El espectro de la señal a la salida del amplificador lineal.
- g) Función de transferencia del filtro $H(e^{j\Omega})$. Utilizar para ello la función de MATLAB *freqz* disponible en la *toolbox signal*.
- h) La señal a la salida del amplificador clase C. Utilizar para filtrar la función *filter*, disponible en la *toolbox signal*.

AMPLIFICADOR CLASE C

- i) El espectro de la señal a la salida del amplificador clase C.
- j) Diagrama polar con la situación de los ceros y los polos de la función de transferencia del filtro. Utilizar para ello la función *polozer*, dada a continuación.

DATOS

$f_c = 1\text{MHz}$	Frecuencia del tono de entrada.
$A_c = 10\text{Vpp}$	Amplitud del tono de entrada.
$f_m = 128\text{MHz}$	Frecuencia de muestreo.
$N = 1024$	Número de muestras.
$L = 1\text{V}$	Amplitud del limitador.
$g = 10$ (veces)	Ganancia del amplificador.
$B_w = 1,1\text{MHz}$	Ancho de banda del filtro paso bajo.
$Z = 50\text{ Ohm}$	Impedancia de trabajo.
B,A	El cálculo de los coeficientes del filtro paso bajo se realizará: $[B,A] = \text{butter}(7, B_w/(f_m/2))$; esta función está disponible en la <i>toolbox signal</i> .

AMPLIFICADOR CLASE C

```
%*****spectrum.m*****
% TITULO: FUNCION PARA REPRESENTAR EL ESPECTRO DE UNA SEÑAL
% ASIGNATURA: LABORATORIO DE SIMULACION DE COMUNICACIONES
% GRADO EN INGENIERIA EN SISTEMAS DE TELECOMUNICACION
%*****
% y=spectrum(x,T,Z,s,lin_log)
% x: señal a analizar (real)
% T: período de muestreo
% Z: impedancia (ohmios)
% s: título que se quiere aparezca en el espectro
% lin_log: representación lineal (lin) o logarítmica (log)
%*****
function y=spectrum(x,T,Z,s,lin_log)
a=length(x);
aa=log(a)/log(2);
l=ceil(aa);
```

AMPLIFICADOR CLASE C

```
x1=fft(x,2^l);
y=x1.*conj(x1);
l=2^l; % l=length(y);
y=y/(l^2);
y=[y(1) 2*y(2:l/2)]/real(Z);
eje=(1/T)*(0:l/2-1)/l; % eje de abcisas
aux=(lin_log(1:3)=='lin');
if sum(aux)==3
    plot(eje,y*1e3,'g')
    ylabel('mW')
else
    plot(eje, 10*log10(y*1e3+eps),'g') % para no calcular el logaritmo de cero
    ylabel('dBm')
end
grid, title(s),xlabel('hertzios'), pause
end
```

AMPLIFICADOR CLASE C

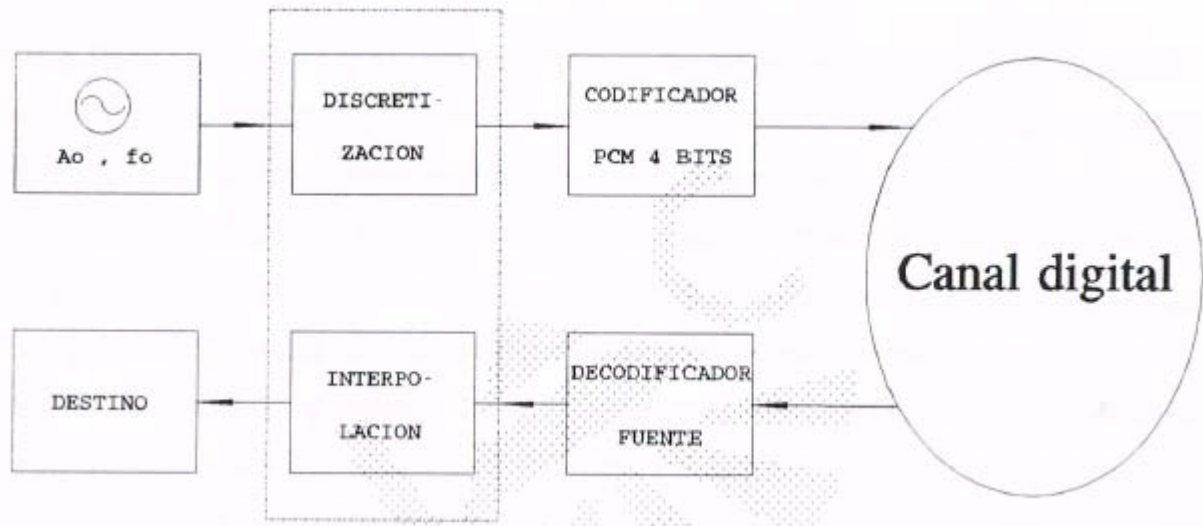
```
%*****polozero.m*****
% TITULO: FUNCION PARA REPRESENTAR EL DIAGRAMA DE POLOS Y CEROS
% AUTOR: OSCAR JIMENEZ
% ASIGNATURA: LABORATORIO DE SIMULACION DE COMUNICACIONES
% GRADO EN INGENIERIA EN SISTEMAS DE TELECOMUNICACION
%*****
% Esta función nos representa el diagrama de polos y ceros
% de un filtro digital IIR, definidos los vectores de entrada
% de acuerdo con la nomenclatura MATLAB, es decir:
% B: vector no recursivo, numerador de  $H(z)$ 
% A: vector de realimentación, denominador de  $H(z)$ 
%*****
function polozero(B,A)
theta=0:0.01:2*pi;
rho=ones(size(theta));
[Z,P]=tf2zp(B,A);
```

AMPLIFICADOR CLASE C

```
if max(abs(Z))>max(abs(P))
    polar(angle(Z),abs(Z),'yo')
    title('DIAGRAMA DE POLOS (*) Y CEROS (o) DEL FILTRO IIR')
    hold on
    polar(angle(P),abs(P),'g*')
    polar(theta,rho,'c')
else
    polar(angle(P),abs(P),'g*')
    title('DIAGRAMA DE POLOS (*) Y CEROS (o) DEL FILTRO IIR')
    hold on
    polar(angle(Z),abs(Z),'yo')
    polar(theta,rho,'c')
end
hold off
end
```

CODIFICACION FUENTE

En esta práctica, el alumno debe realizar la simulación en MATLAB de codificación fuente. Para ello, se ha escogido un esquema clásico PCM de 4 bits. Por su parte, en el extremo receptor, deberá simplemente realizar el proceso opuesto, el decodificador fuente.



CODIFICACION FUENTE

Generador de señal

Generador sinusoidal tipo coseno con los parámetros:

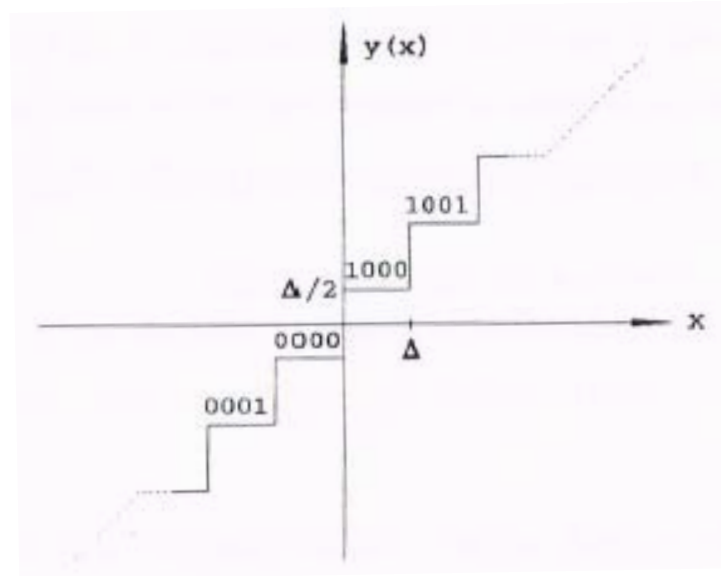
Amplitud: 1V
Frecuencia: 16Hz

Codificador PCM

Codificador de fuente analógica compuesto por un cuantificador uniforme de tipo Mid-Riser, seguido de un conversor de decimal a binario. Sus parámetros son:

Escalones del cuantificador: 16
Fondo de escala del cuantificador: 1V
Número de bits del conversor: 4
Bit de signo negativo: 0

CODIFICACION FUENTE



Canal digital

El canal digital recibe una secuencia binaria y una tasa de error a su entrada, dando a su salida una secuencia binaria modificada, de forma aleatoria, y de acuerdo con la tasa de error.

Nota: La tasa de error a utilizar es la que se conoce como B.E.R. (Bit Error Rate), la cual viene definida como:

$$BER = \frac{\text{Nº de bits erróneos}}{\text{Nº de bits transmitidos}}$$

CODIFICACION FUENTE

El número de bits que modificará el canal será entonces:

$$N_{be} = \text{round}(BER * \text{length}(B_{inx}))$$

Decodificador Fuente:

El decodificador fuente recibe una secuencia binaria en su entrada que convierte en una secuencia de números decimales (muestras) a su salida. Los parámetros de funcionamiento de este bloque son los mismos que los del codificador PCM

Interpolación:

A partir de las muestras cuantificadas obtiene la señal analógica definitiva que entrega al destino. El modo en que opera es empleando un filtro paso bajo como método de interpolación. Las características del filtro son las siguientes:

Tipo de filtro:	Butterworth, orden 8
Frecuencia de corte:	32Hz

CODIFICACION FUENTE

Sistema Global:

Como parámetros genéricos de la simulación se ha tomado:

Frecuencia de muestreo: 1024Hz

Número de muestras: 128

Proceso de simulación:

Seguidamente se describen las funciones que el alumno debe realizar empleando MATLAB. Junto a ellas aparecen también los parámetros de entrada y de salida requeridos.

1) CUANTIF: Debe realizar la cuantificación uniforme Mid-Riser de cualquier secuencia de datos analógicos introducidos. Para ello, recibe como parámetros el fondo de escala y el número de niveles del cuantificador y, como salida, devuelve una secuencia de números resultantes de la cuantificación de los números de entrada.

CODIFICACION FUENTE

Una vez programada la función, se comprobará en MATLAB generando una rampa de tensión que vaya desde un valor igual a menos el fondo de escala del cuantificador, hasta otro igual a su fondo de escala. Con esta señal como entrada, se obtendrá la característica de entrada-salida y la característica de entrada-error de salida del cuantificador programado.

2) CANALDIG: En función de la tasa de error, modificará la secuencia binaria, de forma aleatoria. Utilizar para ello distribución uniforme.

Habiendo terminado la programación de estas funciones y funcionando todas ellas correctamente, estudiar el fichero script que simula el sistema completo, cuyo código se adjunta. Realizar las funciones **BINADEC** y **DECABIN** que faltan, añadir las representaciones gráficas y ejecutarlo.

Realizar la simulación para distintas tasas de error, así como diferentes señales de entrada, x .

CODIFICACION FUENTE

```
%*****script_1_c.m*****  
% TITULO: SCRIPT PRACTICA 1_C  
% AUTOR: NOMBRE ALUMNO  
% ASIGNATURA: LABORATORIO DE SIMULACION DE COMUNICACIONES  
% GRADO EN INGENIERIA EN SISTEMAS DE TELECOMUNICACION  
%*****  
% Script que debe usarse como guión para la realización  
% de la práctica 1C  
% CODIFICACION FUENTE  
%*****  
% Inicio:  
% -----  
%  
close all  
clc,clear  
hold off
```

CODIFICACION FUENTE

% Constantes

% -----

%

fs=1024; % Frecuencia de muestreo

N=128; % Número de muestras

f0=16; % Frecuencia del tono

Ap=1; % Amplitud del tono

b=4; % Número de bits

FE=1; % Fondo de escala de cuantificación

% Cálculos intermedios:

% -----

%

t=0:1/fs:(N-1)/fs; % Eje de tiempos

NE=2^b; % Número de escalones

CODIFICACION FUENTE

% Señal de entrada:

% -----

%

% f1=20; % Frecuencia del tono 2.

% x=Ap*cos(2*pi*f0*t)+Ap*cos(2*pi*f1*t);

x=Ap*cos(2*pi*f0*t);

% Señal cuantificada:

% -----

%

Qx=cuantif(x,FE,NE);

% Señal codificada en binario 4 bits:

% -----

%

Binx=decabin(Qx,FE,b);

CODIFICACION FUENTE

% Señal a la salida del canal:

% -----

%

BER=input('Tasa de error, BER:'); % En tanto por uno
Biny=canaldig(Binx, BER);

% Señales con y sin errores pasadas a decimal:

% -----

%

Decixmal=binadec(Biny,FE,b);
Decixbien=binadec(Binx,FE,b);

% Señales con y sin errores filtradas paso bajo:

% -----

%

% Utilizar las funciones butter y filter

%

CODIFICACION FUENTE

% Representaciones gráficas. Al menos deben aparecer las siguientes:

% -----

%

% 1. Señal cuantificada

% 2. Errores introducidos por el canal. Se calcula haciendo la or-exclusiva

% entre la señal a su entrada y la señal a su salida

% 3. Señal decimal sin errores en el receptor

% 4. Señal decimal con errores en el receptor

% 5. Densidad espectral de potencia de la señal 3

% 6. Densidad espectral de potencia de la señal 4

CODIFICACION FUENTE

% 7. Señal sin errores después del filtro

% 8. Señal con errores después del filtro

% 9.....otras (a criterio del alumno)

end