

## Apéndice A Programas en MATLAB

### PROGRAMA 2.5.1

Capítulo 2  
Sección 2.5  
Ejemplo 2.5.1  
Figura 2.10

```
% *****  
%  
% Programa para graficar dos secuencias sinusoidales:  
%  
% a) Periódica: x1 = Acos(w*n) con w = pi/20  
% b) No-periódica: x2 = Acos(w*n) con w = 0.6  
% *****  
  
clear all;  
  
% Definiciones *****  
  
n = -40:40; % Muestras a graficar  
  
% Graficación de x1 = sin(pi*n/20) [periódica]  
  
x1 = sin(pi*n/20);  
  
subplot(211);  
stem(n,x1);  
title('(a) Señal periódica: x1 = sin(pi*n/20)');  
ylabel('x1(n)');  
xlabel(' n ');  
grid on;  
  
% Graficación de x2 = sin(0.6*n) [no-periódica]  
  
x2 = sin(0.6*n);  
  
subplot(212);  
stem(n,x2);  
title('(b) Señal no periódica: x2 = sin(0.6*n)');  
ylabel('x2(n)');  
xlabel(' n ');  
grid on;
```

## PROGRAMA 2.5.2

Capítulo 2  
Sección 2.5  
Ejemplo 2.5.2  
Figura 2.11

```
% *****  
%  
% Programa para graficar dos secuencias sinusoidales  
% con sus frecuencias separadas una cantidad de 2*pi:  
%  
%     a) x1 = sin(0.1*pi*n)  
%     b) x2 = sin((0.1*pi+2*pi)*n)  
%  
% *****  
  
clear all;  
  
% Definiciones *****  
  
n = -20:20;           % Muestras a graficar  
  
% Graficación de x1 = sin(0.1*pi*n)  
  
x1 = sin(0.1*pi*n);  
  
subplot(211);  
stem(n,x1);  
title('(a) x1 = sin(0.1*pi*n)');  
ylabel('x1(n)');  
xlabel(' n ');  
grid on;  
  
% Graficación de x2 = sin((0.1*pi+2*pi)*n)  
  
x2 = sin((0.1*pi+2*pi)*n);  
  
subplot(212);  
stem(n,x2);  
title('(b) x2 = sin((0.1*pi+2*pi)*n)');  
ylabel('x2(n)');  
xlabel(' n ');  
grid on;
```

## PROGRAMA 2.5.3

Capítulo 2  
Sección 2.5  
Figura 2.12

```
% *****
%
% Programa para observar el efecto de variar la
% frecuencia en una secuencia sinusoidal en el
% intervalo  $0 < f < 1/2$ :
%
%            $x = \cos(2\pi f n)$ 
%
% *****

clear all;

% Definiciones *****

n = 0:30;           % Muestras a graficar

% Graficación de  $x_0 = \cos(2\pi \cdot 0 \cdot n)$ 

f = 0;
x0 = cos(2*pi*f*n);

subplot(221);
stem(n,x0);
axis([0 30 -1.5 1.5]);
title('f = 0');
ylabel('x(n)');
xlabel(' n ');
grid on;

% Graficación de  $x_1 = \cos(2\pi \cdot (1/16) \cdot n)$ 

f = 1/16;
x1 = cos(2*pi*f*n);

subplot(222);
stem(n,x1);
axis([0 30 -1.5 1.5]);
title('f = 1/16');
ylabel('x(n)');
xlabel(' n ');
grid on;

% Graficación de  $x_2 = \cos(2\pi \cdot (1/8) \cdot n)$ 

f = 1/8;
x2 = cos(2*pi*f*n);
```

```

subplot(223);
stem(n,x2);
axis([0 30 -1.5 1.5]);
title('f = 1/8');
ylabel('x(n)');
xlabel(' n ');
grid on;

% Graficación de x3 = cos(2*pi*(1/2)*n)

f = 1/2;
x3 = cos(2*pi*f*n);

subplot(224);
stem(n,x3);
axis([0 30 -1.5 1.5]);
title('f = 1/2');
ylabel('x(n)');
xlabel(' n ');
grid on;

```

## PROGRAMA 6.2.1

### Capítulo 6 Sección 6.2 Figura 6.4

```
%*****  
%  
%           Programa en Matlab para graficar una  
%           SECUENCIA EXPONENCIAL COMPLEJA  
%  
%*****  
c=(pi/4)*i;  
n=-8:8;  
xexp=exp(c*n);  
  
subplot(221);  
stem(n,real(xexp));  
axis([-8 8 -1.5 1.5]);  
z=zeros(1,17);  
hold on;  
plot(n,z);  
title('Re[ x(n)=exp(j[pi/4]n) ]');  
ylabel(' Re [x(n)] ');  
xlabel(' n ');  
  
subplot(223);  
stem(n,imag(xexp));  
axis([-8 8 -1.5 1.5]);  
hold on;  
plot(n,z);  
title('Im[ x(n)=exp(j[pi/4]n) ]');  
ylabel(' Im [x(n)] ');  
xlabel(' n ');  
  
xexp=exp(-c*n);  
  
subplot(222);  
stem(n,real(xexp));  
axis([-8 8 -1.5 1.5]);  
hold on;  
plot(n,z);  
title('Re[ x(-k)=exp(-j[pi/4]k) ]');  
ylabel(' Re [x(-k)] ');  
xlabel(' k ');  
  
subplot(224);  
stem(n,imag(xexp));  
axis([-8 8 -1.5 1.5]);  
hold on;  
plot(n,z);  
title('Im[ x(-k)=exp(-j[pi/4]k) ]');  
ylabel(' Im [x(-k)] ');  
xlabel(' k ');
```

**PROGRAMA 6.2.2**Capítulo 6  
Sección 6.2  
Figura 6.14Capítulo 6  
Sección 6.3  
Figura 6.15

```
% *****  
%  
% Respuesta en frecuencia de un sistema LTI cuya respuesta  
%     al impulso está dado por:  $h(n) = d(n) + d(n-2)$   
%  
% *****  
  
clear all;  
  
% Definiciones *****  
  
w = 0:pi/236:pi;           % Barrido de frecuencia Figura 6.14  
%w = -2*pi:pi/236:2*pi; % Barrido de frecuencia Figura 6.15  
  
% Definición de los coeficientes del sistema *****  
  
h1 = [1 0 1];  
  
% Cálculo de la respuesta en frecuencia *****  
  
H1 = freqz(h1, 1, w);  
  
% Cálculo y graficación de la respuesta en magnitud *****  
  
m1 = abs(H1);  
  
subplot(211);  
plot(w/pi,m1);  
grid on;  
ylabel('Magnitud');  
xlabel('Frecuencia Normalizada: w/pi');  
  
% Cálculo y graficación de la respuesta en fase *****  
  
ph1 = angle(H1)*180/pi;  
  
subplot(212);  
plot(w/pi,ph1);  
grid on;  
ylabel('Fase [grados]');  
xlabel('Frecuencia Normalizada: w/pi');
```

## PROGRAMA 7.1.1

Capítulo 7  
Sección 7.  
Ejemplo 7.1.1  
Figuras 7.1 y 7.2

```
%*****  
%  
% Programa en Matlab para graficar la secuencia  
% x(n) = (a^n)u(n)  
%  
%*****  
  
n = 0:40;  
a = 0.9;  
x = a.^n;  
  
ze = -20:-1;  
z = zeros(1,20);  
  
% Gráfica de la secuencia *****  
  
figure(1);  
stem(n,x);  
axis([-20 40 -0.5 1.5]);  
hold on;  
stem(ze,z);  
title('x(n) = (a^n)u(n) con a=0.9');  
grid;  
ylabel('x(n)');  
xlabel('n');  
  
%*****  
% Programa en Matlab para graficar la magnitud y fase de  
% la FT X(w)=1/(1-a*exp(-jw)) para |w| < pi  
%*****  
  
w = -pi:pi/50:pi;  
  
% Gráfica de la magnitud *****  
  
figure(2);  
a = 0.9;  
X = (1./sqrt(1 - 2*a.*cos(w) + a^2));  
subplot(211);  
plot(w,abs(X));  
title('X(w)=1/(1-a*exp(-jw)) con a=0.9');  
grid;  
ylabel(' Magnitud ');  
xlabel('w [Rad/seg]');
```

```
% Gráfica de la fase *****  
  
fase = - atan(a.*sin(w)./(1 - a.*cos(w)));  
subplot(212);  
plot(w,fase);  
grid;  
ylabel(' Fase [Rad] ');  
xlabel('w [Rad/seg]');
```



## PROGRAMA 7.1.2

Capítulo 7  
Sección 7.1  
Ejemplo 7.1.2  
Figura 7.3

```
% *****
%
% Programa en Matlab para graficar la magnitud y la fase
% de la FT:
%  $H(w) = \frac{[\exp(-jMw/2)]}{(M+1)} \frac{[\sin((M+1)w/2)]}{\sin(w/2)}$ 
%
% *****

w = -pi:.05:pi;

% Gráfica de la magnitud *****

x = sin(5*w/2);
y = sin(w/2);
z = (1/5)*(x./y);
subplot(211);
plot(w,abs(z));
title('H(w)=[exp(-jMw/2)]/(M+1)[sin((M+1)w/2)/sin(w/2)]; M=4');
grid;
ylabel('Magnitud');
xlabel('w [Rad/seg]');

% Gráfica de la fase *****

H1 = exp(-j*2*w);
H2 = z;
fase = angle(H1) + angle(H2);
subplot(212);
plot(w,unwrap(fase));
grid;
ylabel('Fase [Rad]');
xlabel('w [Rad/seg]');
```

## PROGRAMA 7.3.2

Capítulo 7  
Sección 7.3  
Ejemplo 7.3.2  
Figura 7.6

```
% *****  
%  
% Programa en Matlab para graficar la secuencia original  
% "x(n)" y la secuencia muestreada "ys(n)".  
% Señal original: x(n) = [5 4 3 2 1 0 0 0 0 0]  
% Señal muestreada: ys(n) = [5 0 3 0 1 0 0 0 0 0]  
% *****  
  
clear all;  
  
% Definición de las secuencias *****  
  
x = [5 4 3 2 1 ,zeros(1,5)];  
ys = [5 0 3 0 1 ,zeros(1,5)];  
  
%Obtención de las FT de las secuencias *****  
  
[xft,w] = freqz(x,1,512,'whole');  
[ysft,w] = freqz(ys,1,512,'whole');  
  
% Gráficas de las secuencias y de las magnitudes de la FT  
  
n = 0:9; % Vector de tiempo discreto  
  
subplot(221);  
stem(n,x); % Gráfica de la secuencia original x(n)  
title('Secuencia original: x(n)');  
xlabel('n');  
grid on;  
  
subplot(222);  
plot((w/pi),abs(xft)); % Gráfica de la magnitud de la FT  
title('FT de la secuencia original x(n)');  
xlabel('Frecuencia Normalizada: w/pi');  
ylabel('Magnitud');  
grid on;  
  
subplot(223);  
stem(n,ys); % Gráfica de la secuencia muestreada ys(n)  
title('Secuencia muestreada ys(n)');  
xlabel('n');  
grid on;  
  
subplot(224);  
plot((w/pi),abs(ysft)); % Gráfica de la magnitud de la FT  
title('FT de la secuencia muestreada ys(n)');  
xlabel('Frecuencia Normalizada: w/pi');  
ylabel('Magnitud');  
grid on;  
axis([0 2 0 15]);
```

## PROGRAMA 7.3.3

Capítulo 7  
Sección 7.3  
Ejemplo 7.3.3  
Figura 7.8

```
% *****  
%  
% Programa en Matlab para graficar la secuencia original  
% "x(n)" y la secuencia comprimida "yd(n)".  
% Señal original: x(n) = [5 4 3 2 1 0 0 0 0 0]  
% Señal comprimida: yd(n) = [4 2 0 0 0 0 0 0 0 0]  
%  
% *****  
  
clear all;  
  
% Definición de las secuencias *****  
  
x = [5 4 3 2 1 0 0 0 0 0];  
yd = [4 2 0 0 0 0 0 0 0 0];  
  
% Obtención de las FT de las secuencias *****  
  
[xft,w] = freqz(x,1,512,'whole');  
[ydft,w] = freqz(yd,1,512,'whole');  
  
% Gráficas de las secuencias y de las magnitudes de la FT  
  
n = 0:9; % Vector de tiempo discreto  
subplot(221);  
stem(n,x); % Gráfica de la secuencia original x(n)  
title('Secuencia original: x(n)');  
xlabel('n');  
grid on;  
  
subplot(222);  
plot((w/pi),abs(xft)); % Gráfica de la magnitud de la FT  
title('FT de la secuencia original x(n)');  
xlabel('Frecuencia Normalizada: w/pi');  
ylabel('Magnitud');  
grid on;  
axis([0 2 0 15]);  
  
subplot(223);  
stem(n,yd); % Gráfica de la secuencia comprimida yd(n)  
title('Secuencia comprimida yd(n)');  
xlabel('n');  
grid on;  
  
subplot(224);  
plot((w/pi),abs(ydft)); % Gráfica de la magnitud de la FT  
title('FT de la secuencia comprimida yd(n)');  
xlabel('Frecuencia Normalizada: w/pi');  
ylabel('Magnitud');  
grid on;  
axis([0 2 0 15]);
```

## PROGRAMA 7.3.4

Capítulo 7  
Sección 7.3  
Ejemplo 7.3.4  
Figura 7.10

```
% *****  
%  
% Programa en Matlab para graficar la secuencia original  
% "x(n)" y la secuencia expandida "ye(n)".  
% Señal original: x(n) = [5 4 3 2 1 0 0 0 0 0]  
% Señal expandida: ye(n) = [0 5 0 4 0 3 0 2 0 1]  
%  
% *****  
  
clear all;  
  
% Definición de las secuencias *****  
  
x = [5 4 3 2 1 0 0 0 0 0];  
ye = [0 5 0 4 0 3 0 2 0 1];  
  
%Obtención de las FT de las secuencias *****  
  
[xft,w] = freqz(x,1,512,'whole');  
[yeft,w] = freqz(ye,1,512,'whole');  
  
% Gráficas de las secuencias y de las magnitudes de la FT  
  
n = 0:9; % Vector de tiempo discreto  
  
subplot(221);  
stem(n,x); % Gráfica de la secuencia original x(n)  
title('Secuencia original x(n)');  
xlabel('n');  
grid on;  
  
subplot(222);  
plot((w/pi),abs(xft)); % Gráfica de la magnitud de la FT  
title('FT de la secuencia original x(n)');  
xlabel('Frecuencia Normalizada: w/pi');  
ylabel('Magnitud');  
grid on;  
  
subplot(223);  
stem(n,ye); % Gráfica de la secuencia expandida ye(n)  
title('Secuencia expandida ye(n)');  
xlabel('n');  
grid on;  
  
subplot(224);  
plot((w/pi),abs(yeft)); % Gráfica de la magnitud de la FT  
title('FT de la secuencia expandida ye(n)');  
xlabel('Frecuencia Normalizada: w/pi');  
ylabel('Magnitud');  
grid on;  
axis([0 2 0 15]);
```

## PROGRAMA 8.4.2

Capítulo 8  
Sección 8.4.1.4  
Ejemplo 8.4.2  
Figura 8.30

```
% *****  
%  
% Programa en Matlab para observar el efecto de expandir  
% una secuencia senoidal con un factor de L = 3  
%  
% *****  
  
clear all;  
  
% Definiciones *****  
  
N = 50;      % Longitud de la secuencia  
L = 3;      % Factor de expansión  
wo = 0.12;  % Frecuencia de la señal de entrada  
  
% Generación de la secuencia de entrada *****  
  
n = 0:N-1;  
x = sin(2*pi*wo*n);  
  
% Generación de la secuencia de salida expandida *****  
  
y = zeros(1, L*length(x));  
y([1: L: length(y)]) = x;  
  
% Gráfica de las secuencias de entrada y salida *****  
  
subplot(2,1,1);  
stem(n,x);  
axis([0 N -1.2 1.2]);  
title('Secuencia de entrada: x(n)');  
xlabel('n');  
ylabel('Amplitud');  
grid on;  
  
subplot(2,1,2);  
stem(n,y(1:length(x)));  
axis([0 N -1.2 1.2]);  
title('Secuencia de salida expandida con L = 3: xe(n)');  
xlabel('n');  
ylabel('Amplitud');  
grid on;
```

## PROGRAMA 8.4.3

Capítulo 8  
Sección 8.4.1.4  
Ejemplo 8.4.3  
Figura 8.32

```
% *****  
%  
% Programa en Matlab para observar el efecto de comprimir  
% una secuencia senoidal por un factor de M = 3  
%  
% *****  
  
clear all;  
  
% Definiciones *****  
  
N = 50;      % Longitud de la secuencia  
M = 3;      % Factor de expansión  
wo = 0.042; % Frecuencia de la señal de entrada  
  
% Generación de la secuencia de entrada *****  
  
n = 0:N-1;  
m = 0:N*M - 1;  
x = sin(2*pi*wo*m);  
  
% Generación de la secuencia de salida comprimida *****  
  
y = x([1 : M : length(x)]);  
  
% Gráfica de las secuencias de entrada y salida *****  
  
subplot(2,1,1);  
stem(n,x(1:N));  
axis([0 N -1.2 1.2]);  
title('Secuencia de entrada: x(n)');  
xlabel('n');  
ylabel('Amplitud');  
grid on;  
  
subplot(2,1,2);  
stem(n,y);  
axis([0 N -1.2 1.2]);  
title('Secuencia de salida comprimida con M = 3: xd(n)');  
xlabel('n');  
ylabel('Amplitud');  
grid on;
```

## PROGRAMA 8.4.4

Capítulo 8  
Sección 8.4.1.4  
Ejemplo 8.4.4  
Figuras 8.34 y 8.35

```
% *****
%
% Programa en Matlab para observar el efecto de expandir
% una secuencia x(n) por un factor de L = 5, en el
% dominio de la frecuencia.
%
% *****

clear all;

% Generación de la secuencia de entrada *****
% Se calculan los coeficientes de un filtro fir2, los
% cuales corresponden a la respuesta al impulso unitario
% de dicho filtro, generando así una secuencia x(n) de
% de banda limitada

freq = [0 0.45 0.5 1];
mag = [0 1 0 0];
x = fir2(11, freq, mag);
n = 0:11;

% Generación de la secuencia de salida comprimida *****

L = 5;
y = zeros(1, L*length(x));
y([1 : L : length(y)]) = x;
m = 0:(length(y)-1);

% Gráfica de las secuencias de entrada y salida *****

figure(1);
subplot(2,1,1);
stem(n,x);
axis([0 12 -0.3 0.3]);
grid;
title('Secuencia de entrada: x(n)');
ylabel('Amplitud');
xlabel('n');
grid on;

subplot(2,1,2);
stem(m,y);
axis([0 60 -0.3 0.3]);
grid;
title('Secuencia expandida de salida con L=5: xe(n)');
ylabel('Amplitud');
xlabel('n');
grid on;
```

```

% Gráfica del espectro de las secuencias de entrada y
% salida *****

figure(2);

subplot(2,1,1);
[Xz, w] = freqz(x, 1, 512);
plot(w/pi, abs(Xz));
axis([0 1 0 1]);
grid;
title('Espectro (magnitud) de entrada: |x(jw)|');
xlabel('Frecuencia normalizada: w/pi');
ylabel('Magnitud');
grid on;

subplot(2,1,2);
[Yz, w] = freqz(y, 1, 512);
plot(w/pi, abs(Yz));
axis([0 1 0 1]);
grid;
title('Espectro (magnitud) de salida (L=5): |xe(jw)|');
xlabel('Frecuencia normalizada: w/pi');
ylabel('Magnitud');
grid on;

```



## PROGRAMA 8.4.5

Capítulo 8  
Sección 8.4.1.4  
Ejemplo 8.4.5  
Figuras 8.37 y 8.38

```
% *****  
%  
% Programa en Matlab para observar el efecto de comprimir  
% una secuencia x(n) por un factor de M = 2, en el  
% dominio de la frecuencia.  
%  
% *****  
  
clear all;  
  
% Generación de la secuencia de entrada *****  
% Se calculan los coeficientes de un filtro fir2, los  
% cuales corresponden a la respuesta al impulso unitario  
% de dicho filtro, generando así una secuencia x(n) de  
% de banda limitada  
  
freq = [0 0.42 0.48 1];  
mag = [0 1 0 0];  
x = fir2(39, freq, mag);  
n = 0:39;  
  
% Generación de la secuencia de salida comprimida *****  
  
M = 2;  
y = x([1 : M : length(x)]);  
m = 0:((length(x))/2)-1;  
  
% Gráfica de las secuencia de entrada y de salida *****  
  
figure(1);  
subplot(2,1,1);  
stem(n,x);  
title('Secuencia de entrada: x(n)');  
ylabel('Amplitud');  
xlabel('n');  
grid;  
  
subplot(2,1,2);  
stem(m,y);  
title('Secuencia comprimida de salida con M=2: xd(n)');  
ylabel('Amplitud');  
xlabel('n');  
grid;
```

```

% Gráfica del espectro de las secuencias de entrada y
% salida *****

figure(2);

subplot(2,1,1);
[Xz, w] = freqz(x, 1, 512);
plot(w/pi, abs(Xz));
grid;
title('Espectro (magnitud) de entrada: |x(jw)|');
xlabel('Frecuencia normalizada: w/pi');
ylabel('Magnitud');

subplot(2,1,2);
[Yz, w] = freqz(y, 1, 512);
plot(w/pi, abs(Yz));
grid;
title('Espectro (magnitud) de salida (M=2): |xd(jw)|');
xlabel('Frecuencia normalizada: w/pi');
ylabel('Magnitud');

```

## PROGRAMA 9.3.6

Capítulo 9  
Sección 9.3  
Ejemplo 9.3.6  
Figura 9.16

```
% *****
%
% Programa en Matlab para obtener la Transformada de
% Fourier a partir de la Transformada "z" dada a
% continuación:
%
%  $X(z)=[2z^2-0.167z]/[z^2-0.167z-0.167]$ 
%
% *****

clear all;

% Definiciones *****

w = -pi:.01:pi;           % Barrido de frecuencia
b = [2, -0.167, 0];      % Coeficientes del numerador
a = [1, -0.167, -0.167]; % Coeficientes del denominador

% Obtención de la FT *****

h = freqz(b,a,w);

% Gráfica de la magnitud y fase de la FT *****

subplot(211);
plot(w/pi,abs(h));
title('FT a partir de la TZ:  $X(z)=[2z^2-0.167z]/[z^2-0.167z-0.167]$  con  $z=e^{jw}$ ');
xlabel('Frecuencia normalizad: w/pi');
ylabel('Magnitud: |X(e^{jw})|');
grid;

subplot(212);
plot(w/pi,angle(h)*180/pi);
xlabel('Frecuencia normalizada: w/pi');
ylabel('Fase en grados');
grid;
```

## PROGRAMA 9.6.1

Capítulo 9  
Sección 9.6.1  
Ejemplo 9.6.1  
Figura 9.21

```
% *****  
% Programa en Matlab para obtener la Respuesta en  
% Frecuencia del Filtro de Promedio Móvil para M=5 y M=14  
% *****  
  
clear all;  
  
% Definiciones *****  
  
w = 0:pi/236:pi; % Barrido de frecuencia  
  
% Generación de los coeficientes del filtro *****  
  
h1 = ones(1,5)/5;  
h2 = ones(1,14)/14;  
  
% Cálculo de las respuestas en frecuencia *****  
  
H1 = freqz(h1, 1, w);  
H2 = freqz(h2, 1, w);  
  
% Cálculo y graficación de las respuestas en magnitud ***  
  
m1 = abs(H1);  
m2 = abs(H2);  
  
subplot(211);  
g1=plot(w/pi,m1,'k-',w/pi,m2,'k--');  
title('Respuesta en frecuencia del filtro de promedio móvil');  
ylabel('Magnitud');  
xlabel('Frecuencia normalizada: w/pi');  
legend('M = 5', 'M = 14');  
grid on;  
  
% Cálculo y graficación de las respuestas en fase *****  
  
ph1 = angle(H1)*180/pi;  
ph2 = angle(H2)*180/pi;
```

```

subplot(212);
plot(w/pi,ph1,'k-',w/pi,ph2,'k--');
ylabel('Fase en grados');
xlabel('Frecuencia normalizada: w/pi');
legend('M = 5','M = 14');
grid on;

```

### PROGRAMA 9.6.4a

Capítulo 9  
Sección 9.6.2  
Ejemplo 9.6.4  
Figura 9.27

```

% *****
%
% Respuesta en frecuencia (magnitud, fase y retardo de
% grupo) para el sistema: H(z)=(z-r)/z, con r=0.9
%
% *****

clear all;

% Definiciones *****

b = [1, -0.9];      % Coeficientes del numerador
a = [1];           % Coeficiente del denominador
w = -pi:.01:pi;    % Barrido de frecuencia

% Cálculo de la respuesta en frecuencia *****

H = freqz(b,a,w);

% Cálculo y graficación de las respuestas en magnitud,
% fase y retardo de grupo *****

mag = abs(H);
db = 20*log10(mag);
phase = angle(H)*180/pi;

% Calculo del retardo de grupo
num = 0.81 - 0.9*cos(w); % Numerador con r=0.9
den = 1.81 - 2*0.9*cos(w); % Denominador con r=0.9
gdelay = num./den;

subplot(311);
plot(w/pi,db);
title('Respuesta en frecuencia del sistema: H(z)=(z-0.9)/z');
ylabel('Magnitud en db');
grid;

subplot(312);
plot(w/pi,phase);
ylabel('Fase en grados');

```

```

grid;

subplot(313);
plot(w/pi,gdelay);
ylabel('Grpdelay en muestras');
xlabel('Frecuencia normalizad: w/pi');
grid;

```

### PROGRAMA 9.6.4b

Capítulo 9  
Sección 9.6.2  
Ejemplo 9.6.4  
Figura 9.28

```

% *****
%
% Programa en Matlab para obtener la respuesta en
% frecuencia (magnitud, fase y retardo de grupo) para
% sistema:  $H(z)=(z-r)/z$ , con  $r=0.9$  haciendo uso de la
% función "grpdelay"
%
% *****

clear all;

% Definiciones *****

r = 0.9;
w = -pi:.01:pi; % Barrido de frecuencia

% Función de Transferencia en (z) del sistema.
B = [1 -0.9]; % Coeficientes del numerador
A = [1]; % Coeficiente del denominador

% Función de Respuesta en frecuencia en (w) del sistema

H = 1 - r*cos(w) +j*sin(w);

% Cálculo y graficación de las respuestas en magnitud,
% fase y retardo de grupo *****

subplot(311);
plot(w/pi,20*log10(abs(H)));
title('Respuesta en frecuencia de la función:  $H(z)=(z-0.9)/z$ ');
ylabel('Magnitud en db');
grid;

subplot(312);
plot(w/pi,angle(H)*180/pi);
ylabel('Fase en grados');
grid;

subplot(313);

```

```
grpdelay(B,A,w);
```

## PROGRAMA 9.6.6

Capítulo 9  
Sección 9.6.6  
Figura 9.37

```
% *****  
%  
% Programa en Matlab para obtener la respuesta en  
% frecuencia (magnitud, fase y retardo de grupo) para un  
% cero complejo, definido por la función de transferencia:  
%  $H(z)=1-r*\exp\{j*teta\}*z^{-1}$ , con  $r=0.9$  y  $teta=\pi/2$ , para  
% el rango de frecuencia  $0<w<2*\pi$   
% *****  
  
clear all;  
  
% Definiciones *****  
  
w = 0:.05:2*pi; % Barrido de frecuencia  
  
% Función de Transferencia en (z) del sistema.  
  
b = [1,-0.9*j]; % Coeficientes del numerador  
a = [1]; % Coeficiente del denominador  
  
% Respuesta en frecuencia  
  
H = freqz(b,a,w);  
  
% Cálculo y graficación de las respuestas en magnitud,  
% fase y retardo de grupo *****  
  
% Cálculo y graficación de la magnitud  
  
HdB = 20*log10(abs(H));  
  
subplot(311);  
plot(w/pi,HdB);  
title('Respuesta en frecuencia de la función:  $H(z)=1-r*\exp\{j*teta\}*z^{-1}$ ');  
ylabel('Magnitud en dB');  
grid;
```

```
% Cálculo y graficación de la fase

fase = angle(H)*180/pi;

subplot(312);
plot(w/pi,fase);
ylabel('Fase en grados');
grid;

% Cálculo y graficación del retardo de grupo

gd = grpdelay(b,a,length(w),'whole');

subplot(313);
plot(w/pi,gd);
ylabel('Grpdelay en muestras');
xlabel('Frecuencia normalizada: w/pi');
grid;
```



**PROGRAMA 9.6.7**Capítulo 9  
Sección 9.6.7  
Figura 9.39

```
% *****
%
% Programa en Matlab para obtener la respuesta en
% frecuencia (magnitud y fase) para un
% sistema de segundo orden, definido por la función de
% transferencia:  $H(z)=z^2/(z^2-2r\cos(\theta)z+r^2)$ , para
% el rango de frecuencia  $0<w<\pi$ , con:
%     1)  $r=0.9$ ,  $\theta=\pi/6$ 
%     2)  $r=0.85$ ,  $\theta=2\pi/3$ 
% *****

clear all;

% Definiciones *****

w = 0:.1:pi;           % Barrido de frecuencia

r1 = 0.9;              % Sistema (1)
theta1 = pi/6;

r2 = 0.85;            % Sistema (2)
theta2 = 2*pi/3;

r3 = 0.8;              % Sistema (3)
theta3 = pi/6;

% Funciones de Transferencia en (z) del sistema.

b1 = [1];              % Coef. del num. (1)
a1 = [1, -2*r1*cos(theta1), r1*r1]; % Coef. del den. (1)

H1 = freqz(b1,a1,w);   % Respuesta en frecuencia (1)

b2 = [1];              % Coef. del num. (2)
a2 = [1, -2*r2*cos(theta2), r2*r2]; % Coef. del den. (2)

H2 = freqz(b2,a2,w);   % Respuesta en frecuencia (2)
```

```

b3 = [1]; % Coef. del num. (3)
a3 = [1, -2*r3*cos(theta3), r3*r3]; % Coef. del den. (3)

H3 = freqz(b3,a3,w); % Respuesta en frecuencia (3)

% Cálculo de las respuestas en magnitud y fase *****

H1dB = 20*log10(abs(H1));
H2dB = 20*log10(abs(H2));
H3dB = 20*log10(abs(H3));

ph1 = angle(H1)*180/pi;
ph2 = angle(H2)*180/pi;
ph3 = angle(H3)*180/pi;

% Graficación de la magnitud

subplot(211);
plot(w/pi,H1dB,'k-',w/pi,H2dB,'k--',w/pi,H3dB,'k+');
title('Respuesta en frecuencia del sistema:  $H(z)=z^2/(z^2-2r\cos(\theta)z+r^2)$ ');
ylabel('Magnitud en dB');
legend('r=0.9, theta=pi/6','r=0.85, theta=2*pi/3','r=0.8, theta=pi/6');
grid;

% Graficación de la fase

subplot(212);
plot(w/pi,ph1,'k-',w/pi,ph2,'k--',w/pi,ph3,'k+');
xlabel('Frecuencia normalizada: w/pi');
ylabel('Fase en grados');
grid;

```

## PROGRAMA 10.2.1

Capítulo 10  
Sección 10.2.1  
Figura 10.8

```
% *****  
%  
% Respuesta en frecuencia del Filtro FIR Pasa Bajas de  
% Promedio Móvil para M=2, M=3 y M=4  
%  
% *****  
  
clear all;  
  
% Definiciones *****  
  
w = 0:pi/236:pi; % Barrido de frecuencia  
  
% Generación de los coeficientes del filtro *****  
  
h1 = ones(1,2)/2; % M = 2 (dos muestras)  
h2 = ones(1,3)/3; % M = 3 (tres muestras)  
h3 = ones(1,4)/4; % M = 4 (cuatro muestras)  
  
% Cálculo de las respuestas en frecuencia *****  
  
H1 = freqz(h1, 1, w);  
H2 = freqz(h2, 1, w);  
H3 = freqz(h3, 1, w);  
  
% Cálculo y graficación de las respuestas en magnitud ***  
  
m1 = abs(H1);  
m2 = abs(H2);  
m3 = abs(H3);  
  
subplot(211);  
plot(w/pi,m1,'k-',w/pi,m2,'k--',w/pi,m3,'k:');  
title('Respuesta en frecuencia del filtro FIR de Promedio Movil pasa-  
bajas')  
ylabel('Magnitud');  
xlabel('Frecuencia normalizada: w/pi');  
legend('M = 2', 'M = 3', 'M = 4');  
grid on;
```

```
% Cálculo y graficación de las respuestas en fase *****
```

```
ph1 = angle(H1)*180/pi;  
ph2 = angle(H2)*180/pi;  
ph3 = angle(H3)*180/pi;
```

```
subplot(212);  
plot(w/pi,ph1,'k-',w/pi,ph2,'k--',w/pi,ph3,'k:');  
ylabel('Fase en grados');  
xlabel('Frecuencia normalizada: w/pi');  
grid on;
```

## PROGRAMA 10.2.2

Capítulo 10  
Sección 10.2.2  
Figura 10.10

```
% *****  
%  
% Respuesta en frecuencia del Filtro FIR Pasa Altas de  
% Promedio Móvil para M=2, M=3 y M=4  
%  
% *****
```

```
clear all;
```

```
% Definiciones *****
```

```
w = 0:pi/236:pi; % Barrido de frecuencia
```

```
% Generación de los coeficientes del filtro *****
```

```
h1 = [1 -1]./2; % M = 2 (dos muestras)  
h2 = [1 -1 1]./3; % M = 3 (tres muestras)  
h3 = [1 -1 1 -1]./4; % M = 4 (cuatro muestras)
```

```
% Cálculo de las respuestas en frecuencia *****
```

```
H1 = freqz(h1, 1, w);  
H2 = freqz(h2, 1, w);  
H3 = freqz(h3, 1, w);
```

```
% Cálculo y graficación de las respuestas en magnitud ***
```

```
m1 = abs(H1);  
m2 = abs(H2);  
m3 = abs(H3);
```

```
subplot(211);  
plot(w/pi,m1,'k-',w/pi,m2,'k--',w/pi,m3,'k:');  
title('Respuesta en frecuencia del filtro FIR de Promedio Movil pasa-  
altas');  
ylabel('Magnitud');  
xlabel('Frecuencia normalizada: w/pi');  
legend('M = 2', 'M = 3', 'M = 4');  
grid on;
```

```
% Cálculo y graficación de las respuestas en fase *****
```

```
ph1 = angle(H1)*180/pi;  
ph2 = angle(H2)*180/pi;  
ph3 = angle(H3)*180/pi;
```

```
subplot(212);  
plot(w/pi,ph1,'k-',w/pi,ph2,'k--',w/pi,ph3,'k:');  
ylabel('Fase en grados');  
xlabel('Frecuencia normalizada: w/pi');  
grid on;
```

### PROGRAMA 10.2.3

Capítulo 10  
Sección 10.2.3  
Figura 10.12

```
% *****  
%  
% Programa en Matlab para obtener la respuesta en  
% frecuencia (magnitud) para el  
% filtro pasa-bajas IIR:  
%  $H(z) = (1-\alpha)(1+z^{-1})/2*(1-\alpha*z^{-1})$   
%  
% *****
```

```
clear all;
```

```
% Definiciones *****
```

```
w = 0:.05:pi; % Barrido de frecuencia
```

```
% Función de Transferencia en (z) del sistema.
```

```
alfa = 0.8;
```

```
b1 = [1-alfa,1-alfa]; % Coeficientes del numerador  
a1 = [2,-2*alfa]; % Coeficiente del denominador  
H1 = freqz(b1,a1,w);
```

```
alfa = 0.7;
```

```
b2 = [1-alfa,1-alfa]; % Coeficientes del numerador  
a2 = [2,-2*alfa]; % Coeficiente del denominador  
H2 = freqz(b2,a2,w);
```

```
alfa = 0.5;
```

```
b3 = [1-alfa,1-alfa]; % Coeficientes del numerador  
a3 = [2,-2*alfa]; % Coeficiente del denominador  
H3 = freqz(b3,a3,w);
```

```
% Cálculo y graficación de las respuestas en magnitud ***
```

```
m1 = abs(H1);  
m2 = abs(H2);
```

```

m3 = abs(H3);

subplot(211);
plot(w/pi,m1,'k-',w/pi,m2,'k--',w/pi,m3,'k:');
title('Respuesta en frecuencia del filtro IIR pasa-bajas');
ylabel('Magnitud');
xlabel('Frecuencia normalizada: w/pi');
legend('alfa = 0.8','alfa = 0.7','alfa = 0.5');
grid on;

% Cálculo y graficación de las respuestas en fase ****

ph1 = angle(H1)*180/pi;
ph2 = angle(H2)*180/pi;
ph3 = angle(H3)*180/pi;

subplot(212);
plot(w/pi,ph1,'k-',w/pi,ph2,'k--',w/pi,ph3,'k:');
ylabel('Fase en grados');
xlabel('Frecuencia normalizada: w/pi');
grid on;

```

## PROGRAMA 10.2.4

Capítulo 10  
Sección 10.2.4  
Ejemplo 10.2.1  
Figuras 10.13 y 10.14

```
% *****
%
% Programa en Matlab para obtener la respuesta en
% frecuencia (magnitud) para el filtro peine:
%  $H_1(z) = (1/2)(1+z^{-M})$  y  $H_2(z) = (1/2)(1-z^{-M})$ 
% para  $M = 5$ 
%
% *****

clear all;

% Definiciones *****

w = 0:.05:2*pi;           % Barrido de frecuencia

% Función de Transferencia en (z) del sistema.

b1 = [1,0,0,0,0,1];      % Coeficientes del numerador
a1 = [2];                % Coeficiente del denominador
H1 = freqz(b1,a1,w);

b2 = [1,0,0,0,0,-1];    % Coeficientes del numerador
a2 = [2];                % Coeficiente del denominador
H2 = freqz(b2,a2,w);

% Cálculo y graficación de la respuesta en magnitud (H1)*

m1 = abs(H1);

subplot(211);
plot(w/pi,m1);
title('Respuesta en frecuencia del filtro peine:  $H_1(z) = (1/2)(1+z^{-5})$ ');
ylabel('Magnitud');
xlabel('Frecuencia normalizada: w/pi');
grid on;

% Cálculo y graficación de la respuesta en fase (H1) ***

ph1 = angle(H1)*180/pi;
```

```

subplot(212);
plot(w/pi,ph1);
ylabel('Fase en grados');
xlabel('Frecuencia normalizada: w/pi');
grid on;

% Cálculo y graficación de la respuesta en magnitud (H2)*
m2 = abs(H2);

figure(2);
subplot(211);
plot(w/pi,m2);
title('Respuesta en frecuencia del filtro peine: H1(z)= (1/2)(1-z^-5)');
ylabel('Magnitud');
xlabel('Frecuencia normalizada: w/pi');
grid on;

% Cálculo y graficación de la respuesta en fase (H2) ***
ph2 = angle(H2)*180/pi;

subplot(212);
plot(w/pi,ph2);
ylabel('Fase en grados');
xlabel('Frecuencia normalizada: w/pi');
grid on;

```



**PROGRAMA 10.2.5a**Capítulo 10  
Sección 10.2.5  
Ejemplo 10.2.2  
Figura 10.17

```
% *****  
%  
% Programa en Matlab para obtener la respuesta en  
% frecuencia (magnitud, fase y retardo de grupo) para el  
% filtro pasatodo:  $H(z)=(a+z^{-1})/(1+a*z^{-1})$ , con  
%  $a=-1/2$  y  $a=-4/5$ .  
% *****  
  
clear all;  
  
% Definiciones *****  
  
w = 0:.05:pi; % Barrido de frecuencia  
  
% Función de Transferencia en (z) del sistema *****  
  
% Filtro 1 (a = -1/2)  
b1 = [-1/2,1]; % Coeficientes del numerador  
a1 = [1,-1/2]; % Coeficiente del denominador  
H1 = freqz(b1,a1,w);  
  
% Filtro 2 (a = -4/5)  
b2 = [-4/5,1]; % Coeficientes del numerador  
a2 = [1,-4/5]; % Coeficiente del denominador  
H2 = freqz(b2,a2,w);  
  
% Cálculo y graficación de las respuestas en magnitud ***  
  
m1 = abs(H1);  
m2 = abs(H2);  
  
subplot(311);  
plot(w/pi,m1,'k-',w/pi,m2,'k--');  
title('Respuesta en frecuencia del filtro pasa-todo:  $H(z)=(a+z^{-1})/(1+az^{-1})$ ');  
ylabel('Magnitud');  
legend('a = -1/2','a = -4/5');  
axis([0 1 0 2]);  
grid on;
```

```

% Cálculo y graficación de las respuestas en fase *****

ph1 = angle(H1)*180/pi;
ph2 = angle(H2)*180/pi;

subplot(312);
plot(w/pi,ph1,'k-',w/pi,ph2,'k--');
ylabel('Fase en grados');
grid on;

% Cálculo y graficación de las respuestas *****
% de retardo de grupo *****

gd1 = grpdelay(b1,a1,length(w));
gd2 = grpdelay(b2,a2,length(w));

subplot(313);
plot(w/pi,gd1,'k-',w/pi,gd2,'k--');
ylabel('Grpdelay en muestras');
xlabel('Frecuencia normalizada: w/pi');
grid on;

```

**PROGRAMA 10.2.5b**Capítulo 10  
Sección 10.2.5  
Ejemplo 10.2.3  
Figura 10.20

```
% *****
%
% Programa en Matlab para obtener la respuesta en
% frecuencia (magnitud, fase y retardo de grupo) para el
% filtro pasa-todo de segundo orden:
%  $H(z) = (0.81 - 1.2726z^{-1} + z^{-2}) / (1 - 1.2726z^{-1} + 0.81z^{-2})$ 
%
% *****

clear all;

% Definiciones *****

w = 0:.05:pi; % Barrido de frecuencia

% Función de Transferencia en (z) del sistema.

b = [0.81, -1.2726, 1]; % Coeficientes del numerador
a = [1, -1.2726, 0.81]; % Coeficiente del denominador
Hap = freqz(b, a, w);

% Cálculo de la respuesta en frecuencia del sistema

m1 = abs(Hap);
ph1 = angle(Hap)*180/pi;
gd1 = grpdelay(b, a, length(w));

% Graficación de la respuesta en magnitud ***

subplot(311);
plot(w/pi, m1);
ylabel('Magnitud');
title('Filtro pasa-todo:  $H(z) = (0.81 - 1.2726z^{-1} + z^{-2}) / (1 - 1.2726z^{-1} + 0.81z^{-2})$ ');
axis([0 1 0 2]);
grid on;

% Graficación de la respuesta en fase *****

subplot(312);
```

```
plot(w/pi,ph1);
ylabel('Fase en grados');
xlabel('Frecuencia Normalizada: w/pi');
grid on;
```

```
% Graficación de la respuesta de *****
% retardo de grupo *****
```

```
subplot(313);
plot(w/pi,gd1);
ylabel('Grpdelay en muestras');
xlabel('Frecuencia normalizada: w/pi');
grid on;
```

## PROGRAMA 10.2.6

Capítulo 10  
Sección 10.2.6.1  
Ejemplo 10.2.5  
Figuras 10.25 a 10.28

```
% *****
%
% Programa en Matlab para estudiar las secuencias de
% entrada, intermedia y de salida, y sus espectros
% correspondientes, para el sistema:
%       $H(z) = z(z-2) / [(z-1/4)(z-3/4)]$ 
%      y su compensador correspondiente:
%       $H_c(z) = [(z-1/4)(z-3/4)] / [2z(z-2)]$ 
% *****

clear all;

% Definiciones *****

w = 0:.1:2*pi;           % Barrido de frecuencia
n = 0:36;

% Secuencia de entrada x(n)

x1 = zeros(1,5);
x2 = ones(1,5);
x3 = zeros(1,27);
x  = [x1 x2 x3];

% Función de Transferencia H(z)

b = [1 -2 0];           % Coeficientes del numerador
a = [1 -1 3/16];        % Coeficiente del denominador

% Cálculo de la secuencia intermedia s(n) [salida de H(z)]

s = filter(b,a,x);

% Función de Transferencia del compensador Hc(z)

bc = [1 -1 3/16];       % Coeficientes del numerador
ac = [2 -1 0];          % Coeficiente del denominador

% Cálculo de la secuencia de salida de Hc(z): y(n)
```

```

y = filter(bc,ac,s);

% Cálculo de los espectros (magnitud y fase) de las
%      secuencias: x(n), s(n), y(n)

espx = abs(freqz(x,1,w));
esps = abs(freqz(s,1,w));
espy = abs(freqz(y,1,w));

phx = angle(freqz(x,1,w))*180/pi;
phs = angle(freqz(s,1,w))*180/pi;
phy = angle(freqz(y,1,w))*180/pi;

% Graficación de los espectros de magnitud de
% las secuencias: x(n), s(n), y(n)

figure(1);

subplot(311);
plot(w/pi,espx);
ylabel('|X(exp(jw))|');
title('Espectros de magnitud de las secuencias: x(n), s(n) y y(n)');
grid;

subplot(312);
plot(w/pi,esps);
ylabel('|S(exp(jw))|');
grid;

subplot(313);
plot(w/pi,espy);
ylabel('|Y(exp(jw))|');
xlabel('Frecuencia normalizada: w/pi');
grid;

% Graficación de los espectros de fase de
% las secuencias: x(n), s(n), y(n)

figure(2);

subplot(311);
plot(w/pi,phx);
ylabel('Fase[X(exp(jw))]');
title('Espectros de fase de las secuencias: x(n), s(n) y y(n)');
grid;

subplot(312);
plot(w/pi,phs);
ylabel('Fase[S(exp(jw))]');
grid;

subplot(313);
plot(w/pi,phy);
ylabel('Fase[Y(exp(jw))]');
xlabel('Frecuencia normalizada: w/pi');

```

```

grid;

% Graficación de los espectros (magnitud y fase) de
% las secuencias: x(n) y y(n)

figure(3);

subplot(211);
plot(w/pi,espx,'k-',w/pi,espy,'ko');
ylabel('Magnitud');
xlabel('Frecuencia normalizada: w/pi');
title('Espectros de magnitud y fase de las secuencias: x(n) y y(n)');
legend('X(exp(jw))','Y(exp(jw))');
grid;

subplot(212);
plot(w/pi,phx,'k-',w/pi,phy,'ko');
ylabel('Fase en grados');
xlabel('Frecuencia normalizada: w/pi');
grid;

% Graficación de las secuencias x(n), s(n), y(n)

figure(4);

subplot(311);
stem(n,x);
ylabel('x(n)');
title('Graficación de las secuencias x(n), s(n), y(n)');
grid;

subplot(312);
stem(n,s);
ylabel('s(n)');
grid;

subplot(313);
stem(n,y);
ylabel('y(n)');
xlabel('n');
grid;

```

## PROGRAMA 10.2.8

Capítulo 10  
Sección 10.2.8  
Ejemplo 10.2.7  
Figuras 10.32 a 10.35

```
% *****  
%  
% Programa en Matlab para estudiar el filtro FIR con  
% respuesta simétrica al impulso definida por la expresión:  
%  
%  $h(n) = \{0.1 \ 0.3 \ 0.7 \ 1 \ 0.7 \ 0.3 \ 0.1\}$   
%  
% *****  
  
clear all;  
  
% Definiciones *****  
  
w = 0:.05:pi; % Barrido de frecuencia  
n = 0:36;  
  
% Secuencia de entrada x(n)  
  
x1 = zeros(1,5);  
x2 = ones(1,5);  
x3 = zeros(1,27);  
x = [x1 x2 x3];  
  
% Función de Transferencia H(z)  
  
b = [0.1 0.3 0.7 1 0.7 0.3 0.1]; % Coef. del numerador  
a = [1]; % Coef. del denominador  
  
% Cálculo de la secuencia de salida de H(z): y(n)  
  
y = filter(b,a,x);  
  
% Cálculo de la respuesta en frecuencia del filtro FIR  
  
H1 = freqz(b, a, w);  
  
% Cálculo y graficación de la respuesta en magnitud ***  
  
m1 = abs(H1);
```



```

figure(1);

subplot(311);
plot(w/pi,m1);
title('Respuesta en frecuencia del filtro FIR');
ylabel('Magnitud');
grid on;

% Cálculo y graficación de la respuesta en fase *****

ph1 = angle(H1)*180/pi;

subplot(312);
plot(w/pi,ph1);
ylabel('Fase en grados');
grid on;

% Cálculo y graficación del retardo de grupo *****

gd1 = grpdelay(b,a,length(w));

subplot(313);
plot(w/pi,gd1);
ylabel('Grpdelay en muestras');
xlabel('Frecuencia normalizada: w/pi');
grid on;

% Graficación de las secuencias x(n), y(n) cuando
% la entrada es un pulso rectangular discreto

figure(2);

subplot(211);
stem(n,x);
title('Secuencias de entrada x(n)=pulso-rectangular y salida y(n) del
filtro FIR');
ylabel('x(n)');
xlabel('n');
axis([0 40 0 1.5]);
grid;

subplot(212);
stem(n,y);
ylabel('y(n)');
xlabel('n');
grid;

% Respuesta al impulso delta(n) del filtro FIR con
% respuesta al impulso h(n).
% Graficación de las secuencias x(n), y(n)

x1 = ones(1,1);      % Impulso unitario
x2 = zeros(1,5);
x3 = zeros(1,31);

```

```

x = [x1 x2 x3];

y = filter(b,a,x);

figure(3);

subplot(211);
stem(n,x);
title('Secuencias de entrada  $x(n)=\delta(n)$  y salida  $y(n)=h(n)$  del filtro FIR');
ylabel('x(n)');
xlabel('n');
axis([0 40 0 1.5]);
grid;

subplot(212);
stem(n,y);
ylabel('y(n)');
xlabel('n');
grid;

% Graficación de las secuencias x(n), y(n) cuando
% la entrada es una señal sinusoidal

n = 0:30;
f = 1/16;
x = cos(2*pi*f*n);

y = filter(b,a,x);

figure(4);

subplot(211);
stem(n,x);
title('Secuencias de entrada  $x(n)=\cos(2\pi f n)$  y salida y(n) del filtro FIR');
ylabel('x(n)');
xlabel('n');
grid;

subplot(212);
stem(n,y);
ylabel('y(n)');
xlabel('n');
grid;

```

## PROGRAMA 12.3.1

Capítulo 12  
Sección 12.3.1  
Figura 12.7

```
% *****  
%  
% Mapeo de w vs. OMEGA vía la Transformación  
% Bilineal  
%  
% *****  
  
omega = -10:.02:10;  
  
w = 2*atan(omega);  
  
plot(omega,w);  
grid;  
title('w = 2*arctan(omega)');  
ylabel('w');  
xlabel('omega');
```

## PROGRAMA 12.3.2

Capítulo 12  
Sección 12.3.1  
Ejemplo 12.3.1  
Figuras 12.8 y 12.9

```
% *****
%
% Programa en Matlab para obtener la respuesta en
% frecuencia (ganancia en dB y fase) para:
%
% 1) filtro analógico pasabanda con:
%   omegar = frecuencia de resonancia = 4 Rad/seg
% 2) filtro digital IIR pasabanda con:
%   wr = frecuencia de resonancia = pi/2
%
% *****
%
% El filtro digital se obtuvo a partir del filtro analógico
% mediante la TRANSFORMACION BILINEAL
%
% *****

clear all;

% Función de Transferencia en (s) del FILTRO ANALOGICO

numa = [1 0.1];
dena = [1 0.2 16.01];
[Ha,omega] = freqs(numa,dena);

HadB = 20*log10(abs(Ha)); % Ganancia en dB
pha = angle(Ha)*180/pi; % Fase en grados

% Función de Transferencia en (z) del FILTRO DIGITAL

numd = [0.128 0.006 -0.122];
dend = [1 0 0.975];
[Hd,w] = freqz(numd,dend);

HddB = 20*log10(abs(Hd)); % Ganancia en dB
phd = angle(Hd)*180/pi; % Fase en grados

%*****
% Graficacion de la Respuesta en Frecuencia del
% FILTRO ANALOGICO
%*****
```

```

% Graficación de la Ganancia en dB *****

figure(1);

subplot(211);
plot(omega,HadB);
title('Respuesta en frecuencia del filtro ANALOGICO Ha(s) del Ejemplo
12.3.1');
ylabel('Magnitud en dB');
xlabel('Frecuencia analógica: omega');
axis([0 8 -40 20]);
grid on;

% Graficación de la Fase *****

subplot(212);
plot(omega,pha);
ylabel('Fase en grados');
xlabel('Frecuencia analógica: omega');
axis([0 8 -100 100]);
grid on;

%*****
%   Graficación de la Respuesta en Frecuencia del
%           FILTRO DIGITAL
%*****

% Graficación de la Ganancia en dB *****

figure(2);

subplot(211);
plot(w/pi,HddB);
axis([0 1 -40 20]);
title('Respuesta en frecuencia del filtro DIGITAL G(z) del Ejemplo
12.3.1');
ylabel('Ganancia en dB');
xlabel('Frecuencia normalizada: w/pi');
grid;

% Graficación de la Fase *****

subplot(212);
plot(w/pi,phd);
ylabel('Fase en grados');
xlabel('Frecuencia normalizada: w/pi');
grid;

```

### PROGRAMA 12.3.3

Capítulo 12  
Sección 12.3.1  
Ejemplo 12.3.2  
Figuras 12.10 y 12.11

```
% *****  
%  
% Programa en Matlab para obtener la respuesta en  
% frecuencia (ganancia en dB y fase) para:  
%  
% 1) filtro analógico pasa-bajas con:  
%   omegac = frecuencia de corte = 5 Rad/seg  
% 2) filtro digital IIR pasa-bajas con:  
%   wc = frecuencia de corte = 0.2*pi  
%  
% *****  
% El filtro digital se obtuvo a partir del filtro analógico  
% mediante la TRANSFORMACION BILINEAL  
%  
% *****  
  
clear all;  
  
% Función de Transferencia en (s) del FILTRO ANALOGICO  
  
numa = [5];  
dena = [1 5];  
[Ha,omega] = freqs (numa,dena);  
  
HadB = 20*log10 (abs (Ha)); % Ganancia en dB  
pha = angle (Ha)*180/pi; % Fase en grados  
  
% Función de Transferencia en (z) del FILTRO DIGITAL  
  
numd = [0.245 0.245];  
dend = [1 -0.509];  
[Hd,w] = freqz (numd,dend);  
  
HddB = 20*log10 (abs (Hd)); % Ganancia en dB  
phd = angle (Hd)*180/pi; % Fase en grados  
  
%*****  
% Graficacion de la Respuesta en Frecuencia del  
% FILTRO ANALOGICO  
%*****
```

```

% Graficación de la Ganancia en dB *****

figure(1);

subplot(211);
plot(omega,HadB);
title('Respuesta en frecuencia del filtro ANALOGICO Ha(s) del Ejemplo
12.3.2');
ylabel('Magnitud en dB');
xlabel('Frecuencia analógica: omega');
axis([0 70 -40 20]);
grid on;

% Graficación de la Fase *****

subplot(212);
plot(omega,pha);
ylabel('Fase en grados');
xlabel('Frecuencia analógica: omega');
axis([0 70 -100 0]);
grid on;

%*****
%   Graficacion de la Respuesta en Frecuencia del
%           FILTRO DIGITAL
%*****

% Graficación de la Ganancia en dB *****

figure(2);

subplot(211);
plot(w/pi,HddB);
axis([0 1 -40 20]);
title('Respuesta en frecuencia del filtro DIGITAL H(z) del Ejemplo
12.3.2');
ylabel('Ganancia en dB');
xlabel('Frecuencia normalizada: w/pi');
grid;

% Graficación de la Fase *****

subplot(212);
plot(w/pi,phd);
ylabel('Fase en grados');
xlabel('Frecuencia normalizada: w/pi');
grid;

```

## PROGRAMA 12.3.4

Capítulo 12  
Sección 12.3.1  
Ejemplo 12.3.3  
Figuras 12.13 a 12.16

```
% *****  
%  
% Programa en Matlab para obtener la respuesta en  
% frecuencia (ganancia en dB y fase) para:  
%  
% 1) filtro analógico pasa-bajas Ha(s) con  
%   omegap = frecuencia de paso = 0.4142135  
%   rizo pasa-banda < 0.5 dB  
%   omegas = frecuencia de rechazo = 1.1708496  
%   atenuación mínima en la banda de rechazo > 15 dB  
%   N = orden del filtro = 3  
%   omegac = frecuencia de corte = 0.588148  
%  
% 2) filtro digital IIR pasa-bajas G(z) con:  
%   wp = frecuencia de paso = 0.25pi  
%   rizo pasa-banda < 0.5 dB  
%   ws = frecuencia de rechazo = 0.55pi  
%   atenuación mínima en la banda de rechazo > 15 dB  
%  
% *****  
% El filtro digital se obtuvo a partir del filtro analógico  
% mediante la TRANSFORMACION BILINEAL  
%  
% *****  
  
clear all;  
  
% Función de Transferencia en (s) del FILTRO ANALOGICO  
  
numa = [0.203451];  
dena = [1 1.176296 0.691836 0.203451];  
[Ha,omega] = freqs(numa,dena);  
  
HadB = 20*log10(abs(Ha)); % Ganancia en dB  
pha = angle(Ha)*180/pi; % Fase en grados  
  
% Función de Transferencia en (z) del FILTRO DIGITAL  
  
numd = 0.0662272*conv([1 1],[1 2 1]);  
dend = conv([1 -0.2593284],[1 -0.6762858 0.3917468]);  
[Hd,w] = freqz(numd,dend);  
  
% *****  
% NOTA:  
% En este caso el comando "conv" se usa para realizar  
% la MULTIPLICACION de los dos vectores que aparecen  
% como argumentos de dicho comando.  
% *****
```



```

HddB = 20*log10(abs(Hd)); % Ganancia en dB
phd = angle(Hd)*180/pi; % Fase en grados

%*****
% Graficacion de la Respuesta en Frecuencia del
% FILTRO ANALOGICO
%*****

% Graficación de la Ganancia en dB *****

figure(1);

subplot(211);
plot(omega,HadB);
title('Respuesta en frecuencia del filtro ANALOGICO Ha(s) del Ejemplo
12.3.3');
ylabel('Magnitud en dB');
xlabel('Frecuencia analógica: omega');
axis([0 3 -40 20]);
grid on;

% Graficación de la Fase *****

subplot(212);
plot(omega,pha);
ylabel('Fase en grados');
xlabel('Frecuencia analógica: omega');
axis([0 3 -200 200]);
grid on;

%*****
% Detalle de la Respuesta en Magnitud del
% FILTRO ANALOGICO
%*****

% Detalle de la Ganancia en dB *****

figure(2);

subplot(311);
plot(omega,HadB);
title('Detalle de la respuesta en magnitud del filtro ANALOGICO Ha(s) del
Ejemplo 12.3.3');
ylabel('Magnitud en dB');
axis([0.3 0.5 -1 0]);
grid on;

subplot(312);
plot(omega,HadB);
ylabel('Magnitud en dB');
axis([0.5 0.7 -5 -1]);
grid on;

```

```

subplot(313);
plot(omega,HadB);
ylabel('Magnitud en dB');
xlabel('Frecuencia analógica: omega');
axis([1 1.2 -19 -14]);
grid on;

%*****
%   Graficación de la Respuesta en Frecuencia del
%           FILTRO DIGITAL
%*****

% Graficación de la Ganancia en dB *****

figure(3);

subplot(211);
plot(w/pi,HddB);
axis([0 1 -40 20]);
title('Respuesta en frecuencia del filtro DIGITAL G(z) del Ejemplo
12.3.3');
ylabel('Ganancia en dB');
xlabel('Frecuencia normalizada: w/pi');
grid;

% Graficación de la Fase *****

subplot(212);
plot(w/pi,phd);
ylabel('Fase en grados');
xlabel('Frecuencia normalizada: w/pi');
grid;

%*****
%   Detalle de la Respuesta en Magnitud del
%           FILTRO DIGITAL
%*****

% Detalle de la Ganancia en dB *****

figure(4);

subplot(211);
plot(w/pi,HddB);
title('Detalle de la respuesta en magnitud del filtro DIGITAL G(z) del
Ejemplo 12.3.3');
ylabel('Magnitud en dB');
xlabel('Frecuencia normalizada: w/pi');
axis([0.2 0.3 -1 1]);
grid on;

subplot(212);
plot(w/pi,HddB);
ylabel('Magnitud en dB');
xlabel('Frecuencia normalizada: w/pi');
axis([0.5 0.6 -20 -10]);
grid on;

```

## PROGRAMA 12.3.5

Capítulo 12  
Sección 12.3.3  
Ejemplo 12.3.4  
Figuras 12.17 y 12.18

```
% *****
%
% Programa en Matlab para obtener la función de
% transferencia y la respuesta en frecuencia de
% un filtro digital IIR pasa-bajas elíptico con:
%
%     wp = frecuencia de paso = 0.4
%     rizo pasa-banda < 0.5 dB
%     atenuación mínima en la banda de rechazo = 40 dB
%
% *****

clear all;

% Definiciones *****

w = 0:pi/255:pi;           % Barrido de frecuencia

% Función de Transferencia en (z) del sistema.

N = 5;
Rp = 0.5;
Rs = 40;
Wn = 0.4;

[b,a] = ellip(N,Rp,Rs,Wn);
disp('Polinomio del numerador:'); disp(b);
disp('Polinomio del denominador:'); disp(a);

% Obtención de la respuesta en frecuencia *****

h = freqz(b,a,w);
H = 20*log10(abs(h));
ph = angle(h)*180/pi;

% Graficación de la respuesta en frecuencia *****

figure(1);
subplot(211);
plot(w/pi,H);
axis([0 1 -60 5]);
ylabel('Ganancia en dB');
xlabel('Frecuencia normalizada: w/pi');
grid;

subplot(212);
plot(w/pi,ph);
ylabel('Fase en grados');
xlabel('Frecuencia normalizada: w/pi');
grid;
```

```
% Detalle de la respuesta en frecuencia *****  
  
figure(2);  
plot(w/pi,H);  
axis([0 0.5 -3.5 0.5]);  
ylabel('Ganancia en dB');  
xlabel('Frecuencia normalizada: w/pi');  
grid;
```

## PROGRAMA 12.3.6

Capítulo 12  
Sección 12.3.3  
Ejemplo 12.3.5  
Figuras 12.19 y 12.20

```
% *****
%
% Programa en Matlab para obtener la función de
% transferencia y la respuesta en frecuencia
% para un filtro digital IIR pasa-altas Chebyshev
% Tipo I con:
%
% orden del filtro = N = 4
% wp = frecuencia de paso = 0.7
% rizo pasa-banda < 1 dB
%
% *****

clear all;

% Definiciones *****

w = 0:pi/255:pi;          % Barrido de frecuencia

% Función de Transferencia en (z) del sistema.

N = 4;
Rp = 1;
Wn = 0.7;

[b,a] = cheby1(N,Rp,Wn,'high');
disp('Polinomio del numerador:'); disp(b);
disp('Polinomio del denominador:'); disp(a);

% Obtención de la respuesta en frecuencia *****

h = freqz(b,a,w);
H = 20*log10(abs(h));
ph = angle(h)*180/pi;

% Gráfica de la respuesta en frecuencia *****

figure(1);
subplot(211);
plot(w/pi,H);
axis([0 1 -60 5]);
ylabel('Ganancia en dB');
xlabel('Frecuencia normalizada: w/pi');
grid;

subplot(212);
plot(w/pi,ph);
ylabel('Fase en grados');
xlabel('Frecuencia normalizada: w/pi');
grid;
```

```
% Detalle de la respuesta en frecuencia *****  
  
figure(2);  
plot(w/pi,H);  
axis([0.6 1 -2 0.5]);  
ylabel('Ganancia en dB');  
xlabel('Frecuencia normalizada: w/pi');  
grid;
```

## PROGRAMA 12.4.2

Capítulo 12  
Sección 12.4.2  
Figuras 12.28 y 12.29

```
% *****
%
% Programa en Matlab para graficar las ventanas siguientes
% y obtener la respuesta en frecuencia (magnitud) de cada
% una de ellas:
%     Rectangular
%     Hamming
%     Hanning
%     Blackman
% *****

clear all;

% Definiciones para el cálculo de la ventana de Kaiser **

Q = 10;          % Posición relativa de la muestra central
P = (2*Q + 1);  % Longitud de la ventana (P = 11)

n = 0:2*Q;

% *****
% Coeficientes de la ventana rectangular
% *****

vrectangular = ones(1,P);

% Respuesta en frecuencia de la ventana rectangular

[mrectangular,omega] = freqz(vrectangular,sum(vrectangular),256);

% *****
% Coeficientes de la ventana Hamming
% *****

vhamming = hamming(P);

% Respuesta en frecuencia de la ventana rectangular

[mhamming,omega] = freqz(vhamming,sum(vhamming),256);

% *****
% Coeficientes de la ventana Hanning
% *****

vhanning = hanning(P);

% Respuesta en frecuencia de la ventana rectangular

[mhanning,omega] = freqz(vhanning,sum(vhanning),256);
```

```

% *****
% Coeficientes de la ventana Blackman
% *****

vblackman = blackman(P);

% Respuesta en frecuencia de la ventana rectangular

[mblackman,omega] = freqz(vblackman,sum(vblackman),256);

% *****
% Graficación de las ventanas y sus espectros de magnitud
% respectivos
% *****

figure(1);

subplot(221);
stem(vrectangular);
axis([1 21 0 1.2]);
title('Ventana Rectangular: w(n)');
xlabel('n');
grid on;

subplot(223);
plot(omega/pi,20*log10(abs(mrectangular)),'k-');
axis([0 1 -120 0]);
title('V. Rectangular: |W(exp[jw])|');
ylabel('Ganancia en dB');
xlabel('w/pi');
grid on;

subplot(222);
stem(vhamming);
axis([1 21 0 1.2]);
title('Ventana Hamming: w(n)');
xlabel('n');
grid on;

subplot(224);
plot(omega/pi,20*log10(abs(mhamming)),'k-');
axis([0 1 -120 0]);
title('V. Hamming: |W(exp[jw])|');
ylabel('Ganancia en dB');
xlabel('w/pi');
grid on;

figure(2);

subplot(221);
stem(vhanning);
axis([1 21 0 1.2]);
title('Ventana Hanning: w(n)');
xlabel('n');
grid on;

```



```

subplot(223);
plot(omega/pi,20*log10(abs(mhanning)), 'k-');
axis([0 1 -120 0]);
title('V. Hanning: |W(exp[jw])|');
ylabel('Ganancia en dB');
xlabel('w/pi');
grid on;

subplot(222);
stem(vblackman);
axis([1 21 0 1.2]);
title('Ventana Blackman: w(n)');
xlabel('n');
grid on;

subplot(224);
plot(omega/pi,20*log10(abs(mblackman)), 'k-');
axis([0 1 -120 0]);
title('V. Blackman: |W(exp[jw])|');
ylabel('Ganancia en dB');
xlabel('w/pi');
grid on;

```

### PROGRAMA 12.4.3

Capítulo 12  
Sección 12.4.2  
Ejemplo 12.4.1  
Figuras 12.32 a 12.34

```
% *****  
%  
% Programa en Matlab para obtener la respuesta al impulso  
% y la respuesta en frecuencia (ganancia en dB) para el  
% filtro digital FIR pasabajas:  
%  
% wc = frecuencia de corte: 1000 Hz  
% frecuencia de muestreo: 4000 muestras/segundo  
% longitud del filtro: 51 muestras  
% ventanas: Rectangular y Hamming  
%  
% *****  
  
clear all;  
  
% Definiciones *****  
  
m = 0:24; % (1/2) Muestras de h(n) del filtro  
n = 0:50; % Tiempo discreto "n"  
  
% Respuesta ideal deseada al impulso h(n) del filtro FIR  
% pasa-bajas  
  
hnum = sin((m-25)*pi/2);  
hden = (m-25)*pi;  
h1 = hnum./hden;  
h2 = 0.5;  
h3 = fliplr(h1);  
hdeseada = [h1, h2, h3];  
  
% Respuesta real al impulso h(n) del filtro FIR con la  
% ventana Rectangular  
  
vrectangular = ones(51,1);  
hrectangular = hdeseada.*vrectangular';  
  
% Respuesta real al impulso h(n) del filtro FIR con la  
% ventana Hamming  
  
vhamming = hamming(51);  
hhamming = hdeseada.*vhamming';  
  
% Respuesta en frecuencia de la magnitud para el  
% filtro FIR con las ventanas  
% Rectangular y Hamming  
  
[Hrectangular,w] = freqz(hrectangular,1,256);  
HrectangulardB = 20*log10(abs(Hrectangular));  
  
[Hhamming,w] = freqz(hhamming,1,256);  
HhammingdB = 20*log10(abs(Hhamming));
```

```

% Graficación de la respuestas *****

figure(1);

subplot(311)
stem(n,vrectangular);
title('Diseño del filtro FIR pasa-bajas con una ventana Rectangular');
ylabel('win(n) Rectangular');
xlabel('n');
axis([0 51 -0.2 1.2]);
grid;

subplot(312)
stem(n,hrectangular);
ylabel('h(n)');
xlabel('n');
axis([0 51 -0.2 0.6]);
grid;

subplot(313);
plot(w/pi,HrectangulardB);
ylabel('|H(exp[jw])| dB');
xlabel('Frecuencia normalizada: w/pi');
axis([0 1 -100 5]);
grid;

figure(2);

subplot(311)
stem(n,vhamming);
title('Diseño del filtro FIR pasa-bajas con una ventana Hamming');
ylabel('win(n) Hamming');
xlabel('n');
axis([0 51 -0.2 1.2]);
grid;

subplot(312)
stem(n,hhamming);
ylabel('h(n)');
xlabel('n');
axis([0 51 -0.2 0.6]);
grid;

subplot(313);
plot(w/pi,HhammingdB);
ylabel('|H(e[jw])| dB');
xlabel('Frecuencia normalizada: w/pi');
axis([0 1 -100 5]);
grid;

```

```
% Graficación de los espectros de magnitud en forma comparativa ****  
  
figure(3);  
  
plot(w/pi,HrectangulardB,'k-',w/pi,HhammingdB,'k-.');  
title('Filtros FIR pasa-bajas: espectros de magnitud |H(exp[jw])|');  
ylabel('Magnitud en dB');  
xlabel('Frecuencia normalizada: w/pi');  
legend('Filtro con V. Rectangular','Filtro con V. Hamming');  
grid;
```

## PROGRAMA 12.4.4

Capítulo 12  
Sección 12.4.3  
Figuras 12.35 a 12.38

```
% *****  
%  
% Programa en Matlab para graficar la ventana Kaiser:  
%  
% 1) Con una longitud N = 20 muestras y beta = 0, 3 y 6  
% 2) Con beta = 6 y longitudes de N = 10, 20 y 40 muestras  
%  
% *****  
  
clear all;  
  
%*****  
% 1) Variación de beta = 0, 3 y 6  
%*****  
  
% Definiciones *****  
  
N = 20; % Numero de muestras  
n = 0:19; % Tiempo discreto "n"  
  
beta0 = 0;  
beta3 = 3;  
beta6 = 6;  
  
% Coeficientes de la ventana de Kaiser  
  
wkaiser0 = kaiser(N,beta0);  
wkaiser3 = kaiser(N,beta3);  
wkaiser6 = kaiser(N,beta6);  
  
% Cálculo de la Respuesta en Frecuencia de la Magnitud  
% para la ventana de Kaiser normalizada  
  
[H0,w] = freqz(wkaiser0,sum(wkaiser0));  
Wkaiser0 = 20*log10(abs(H0));  
  
[H3,w] = freqz(wkaiser3,sum(wkaiser3));  
Wkaiser3 = 20*log10(abs(H3));  
  
[H6,w] = freqz(wkaiser6,sum(wkaiser6));  
Wkaiser6 = 20*log10(abs(H6));  
  
% Graficación de la ventana de Kaiser para diferentes valores  
% de beta
```

```

figure(1);

subplot(231);
stem(n,wkaiser0);
title('BETA = 0');
ylabel('Amplitud');
xlabel('Muestras: n');
axis([0 20 0 1.2]);
grid on;

subplot(232);
stem(n,wkaiser3);
title('BETA = 3');
xlabel('Muestras: n');
axis([0 20 0 1.2]);
grid on;

subplot(233);
stem(n,wkaiser6);
title('BETA = 6');
xlabel('Muestras: n');
axis([0 20 0 1.2]);
grid on;

% Graficación del espectro de magnitud de la ventana de
% Kaiser

subplot(234);
plot(w/pi,Wkaiser0);
ylabel('Ganancia en dB');
xlabel('w/pi');
axis([0 1 -100 0]);
grid on;

subplot(235);
plot(w/pi,Wkaiser3);
xlabel('w/pi');
axis([0 1 -100 0]);
grid on;

subplot(236);
plot(w/pi,Wkaiser6);
xlabel('w/pi');
axis([0 1 -100 0]);
grid on;

% Grafica comparativa del espectro de magnitud de la
% ventana de Kaiser para diferentes valores de BETA
figure(2);

plot(w/pi,Wkaiser0,'k-',w/pi,Wkaiser3,'k--',w/pi,Wkaiser6,'k-.');
title('Espectro de magnitud de la ventana Kaiser para diferentes valores
de BETA');
ylabel('Magnitud en dB');
xlabel('Frecuencia normalizada: w/pi');
legend('BETA = 0','BETA = 3','BETA = 6');
grid on;

```

```

%*****
% 2) Variación de la longitud N = 10, 20 y 40
%*****

% Definiciones *****

N10 = 10; % Numero de muestras
N20 = 20;
N40 = 40;
beta6 = 6;

% Coeficientes de la ventana de Kaiser

wkaiser10 = kaiser(N10,beta6);
wkaiser20 = kaiser(N20,beta6);
wkaiser40 = kaiser(N40,beta6);

% Cálculo de la respuesta en frecuencia de la ventana
% de Kaiser normalizada

[H10,w] = freqz(wkaiser10,sum(wkaiser10));
Wkaiser10 = 20*log10(abs(H10));

[H20,w] = freqz(wkaiser20,sum(wkaiser20));
Wkaiser20 = 20*log10(abs(H20));

[H40,w] = freqz(wkaiser40,sum(wkaiser40));
Wkaiser40 = 20*log10(abs(H40));

% Graficación de las ventanas de Kaiser para diferentes
% longitudes

figure(3);

n1 = 0:9;
subplot(231);
stem(n1,wkaiser10);
title('N = 10 (BETA = 6)');
ylabel('Amplitud');
xlabel('Muestras: n');
axis([0 40 0 1.2]);
grid on;

n2 = 0:19;
subplot(232);
stem(n2,wkaiser20);
title('N = 20 (BETA = 6)');
xlabel('Muestras: n');
axis([0 40 0 1.2]);
grid on;

```

```

n3 = 0:39;
subplot(233);
stem(n3,wkaiser40);
title('N = 40 (BETA = 6)');
xlabel('Muestras: n');
axis([0 40 0 1.2]);
grid on;

% Graficación del espectro de magnitud de la ventana de
% Kaiser

subplot(234);
plot(w/pi,Wkaiser10);
ylabel('Ganancia en dB');
xlabel('w/pi');
axis([0 1 -100 0]);
grid on;

subplot(235);
plot(w/pi,Wkaiser20);
xlabel('w/pi');
axis([0 1 -100 0]);
grid on;

subplot(236);
plot(w/pi,Wkaiser40);
xlabel('w/pi');
axis([0 1 -100 0]);
grid on;

% Grafica comparativa del espectro de magnitud de la
% ventana de Kaiser para diferentes valores de BETA

figure(4);

plot(w/pi,Wkaiser10,'k-',w/pi,Wkaiser20,'k--',w/pi,Wkaiser40,'k-.');
title('Espectro de magnitud de la ventana Kaiser para diferentes
longitudes (BETA=6)');
ylabel('Magnitud en dB');
xlabel('Frecuencia normalizada: w/pi');
legend('N = 10', 'N = 20', 'N = 40');
grid on;

```



## PROGRAMA 12.4.5

Capítulo 12  
Sección 12.4.3  
Ejemplo 12.4.2  
Figuras 12.41 a 12.43

```
% *****
%
% Programa en Matlab para obtener la respuesta en
% frecuencia (magnitud) de un filtro digital FIR
% pasa-bajas usando una ventana Kaiser con:
%
%     Wp = 0.3pi
%     Ws = 0.4pi
%     alfas = 50 dB
%     Wc = 0.35pi
%     beta = 4.533514
%     deltaf = 0.05
%     N = 61
%
% *****

clear all;

% Definiciones para el cálculo de la ventana de Kaiser **

N = 61;                % Orden del filtro
n = 0:60;              % Tiempo discreto "n"

beta = 4.533514;

% Coeficientes de la ventana de Kaiser

wkaiser = kaiser(N,beta);

% Respuesta en frecuencia de la ventana de Kaiser normalizada

[Wkaiser,w] = freqz(wkaiser,sum(wkaiser));

% Graficación de la Ventana Kaiser y su espectro de magnitud

figure(1);

subplot(211);
stem(n,wkaiser);
title('Ventana Kaiser para BETA = 4.533514 y N = 61');
xlabel('n');
grid on;

subplot(212);
plot(w/pi,20*log10(abs(Wkaiser)));
axis([0 1 -80 5]);
title('Espectro de magnitud de la ventana Kaiser');
ylabel('Ganancia en dB');
xlabel('Frecuencia normalizada: w/pi');
grid on;
```

```

% Definiciones para el cálculo de la respuesta al impulso
% del filtro pasa-bajas ideal

m = 0:29;           % Muestras de h(n) del filtro
Q = 30;           % Muestra central

% Respuesta al impulso h(n) del filtro pasa-bajas ideal

hnum = sin((m-Q)*0.35*pi);
hden = (m-Q)*pi;
h1 = hnum./hden;
h2 = 0.35;         % wc/pi
h3 = fliplr(h1);
hideal = [h1, h2, h3];

% Respuesta en frecuencia del filtro pasa-bajas ideal

[Hideal,w] = freqz(hideal,sum(hideal));

% Graficación de la respuesta al impulso y su espectro de
% magnitud

figure(2);

subplot(211);
stem(n,hideal);
title('Respuesta al impulso del filtro pasa-bajas ideal');
xlabel('n');
grid;

subplot(212);
plot(w/pi,20*log10(abs(Hideal)));
axis([0 1 -80 5]);
title('Espectro de magnitud del filtro pasa-bajas ideal');
ylabel('Ganancia en dB');
xlabel('Frecuencia normalizada: w/pi');
grid on;

% Multiplicación de la ventana Kaiser por la respuesta al impulso
% del filtro pasa bajas ideal

hdeseada = hideal.*wkaiser' % Coeficientes del filtro FIR
[Hdeseada,w] =freqz(hdeseada,1);

% Graficación de la respuesta al impulso del filtro FIR diseñado
% con la ventana Kaiser y gráficación del espectro en magnitud del
% filtro FIR

figure(3);

subplot(211);
stem(n,hdeseada);
title('Respuesta al impulso del filtro FIR diseñado con ventana Kaiser');
xlabel('n');
grid;

```

```
subplot(212);  
plot(w/pi,20*log10(abs(Hdeseada)));  
axis([0 1 -80 5]);  
title('Espectro de magnitud del filtro FIR diseñado con ventana Kaiser');  
ylabel('Ganancia en dB');  
xlabel('Frecuencia normalizada: w/pi');  
grid on;
```

## PROGRAMA 13.2.1

Capítulo 13  
Sección 13.2  
Figuras 13.6 a 13.12

```
% *****
%
% Programa en Matlab para comparar la respuesta en
% frecuencia de cuatro tipos de respuestas de filtros
% analógicos pasa-bajas con las siguientes especificaciones:
%
% orden del filtro:          N = 4
% frecuencia:              Wn = 1.0
% rizo en la banda de paso  < 1 dB
% atenuación mínima de rechazo: 40 dB
%
% Tipos de respuestas de filtros:
% Butterworth
% Chebyshev Tipo I
% Chebyshev Tipo II
% Bessel
% Elíptico
% *****

clear all;

% Definiciones *****

w = logspace(-1,1);          % Barrido de frecuencia

N = 4;
Wn = 1.0;
Rp = 1;
Rs = 40;

Wp = 2.0; % Solo para el filtro Chebyshev Tipo II

% Función de Transferencia en (s) del sistema.

[bbutter, abutter] = butter(N,Wn, 's');

[bcheby1, acheby1] = cheby1(N,Rp,Wn, 's');

[bcheby2, acheby2] = cheby2(N,Rs,Wp, 's');

[bbessel, abessel] = besself(N,Wn);

[bellip, aellip] = ellip(N,Rp,Rs,Wn, 's');

% Obtención de la respuesta en frecuencia *****

hbutter = freqs(bbutter, abutter, w);
Hbutter = 20*log10(abs(hbutter));
phbutter = angle(hbutter)*180/pi;
```

```

hcheby1 = freqs(bcheby1,acheby1,w);
Hcheby1 = 20*log10(abs(hcheby1));
pchcheby1 = angle(hcheby1)*180/pi;

hcheby2 = freqs(bcheby2,acheby2,w);
Hcheby2 = 20*log10(abs(hcheby2));
pchcheby2 = angle(hcheby2)*180/pi;

hbessel = freqs(bbessel,abessel,w);
Hbessel = 20*log10(abs(hbessel));
phbessel = angle(hbessel)*180/pi;

hellip = freqs(bellip,aellip,w);
Hellip = 20*log10(abs(hellip));
phellip = angle(hellip)*180/pi;

% Gráfica de la respuesta en frecuencia de la magnitud *

figure(1);
semilogx(w,Hbutter,'k-',w,Hcheby1,'k--',w,Hcheby2,'k:',w,Hbessel,'kd',w,Hellip,'kh');
legend('Butterworth','Chebyshev I','Chebyshev II','Bessel','Elíptico');
axis([0.1 10 -60 5]);
title('Filtro analógico pasa-bajas de orden 4');
ylabel('Ganancia en dB');
xlabel('Frecuencia normalizada: w/Wn (Wn = frecuencia de corte)');
grid;

% Detalle de la respuesta en frecuencia de la magnitud *

figure(2);
semilogx(w,Hbutter,'k-',w,Hcheby1,'k--',w,Hcheby2,'k:',w,Hbessel,'kd',w,Hellip,'kh');
axis([0.5 5 -60 5]);
legend('Butterworth','Chebyshev I','Chebyshev II','Bessel','Elíptico');
title('Filtro analógico pasa-bajas de orden 4');
ylabel('Ganancia en dB');
xlabel('Frecuencia normalizada: w/Wn (Wn = frecuencia de corte)');
grid;

% Calculo de los polos y ceros de la F.T. *****

[zbutter,pbutter,kbutter] = buttap(N);

[zcheby1,pcheby1,kcheby1] = cheblap(N,Rp);

[zcheby2,pcheby2,kcheby2] = cheb2ap(N,Rs);

[zbessel,pbessel,kbessel] = besslap(N);

[zellip,pellip,kellip] = ellipap(N,Rp,Rs);

figure(3);
zplane(zbutter,pbutter);
title('Filtro digital pasa-bajas Butterworth orden 4');

```

```
figure(4);
zplane(zcheby1,pcheby1);
title('Filtro digital pasa-bajas Chebyshev Tipo I orden 4');

figure(5);
zplane(zcheby2,pcheby2);
title('Filtro digital pasa-bajas Chebyshev Tipo II orden 4');

figure(6);
zplane(zbessel,pbessel);
title('Filtro digital pasa-bajas Bessel orden 4');

figure(7);
zplane(zellip,pellip);
title('Filtro digital pasa-bajas Eliptico orden 4');
```

## PROGRAMA 13.3.1

Capítulo 13  
Sección 13.3  
Figuras 13.13 y 13.14

```
% *****  
%  
% Programa en Matlab para obtener la respuesta en  
% frecuencia (magnitud) de filtros analógicos  
% pasa-bajas Butterworth de orden "N" = 3, 5 y 8  
% Wn = frecuencia de corte = 1.0  
%  
% *****  
  
clear all;  
  
% Definiciones *****  
  
omega = 0:0.05:10; % Barrido lineal de frecuencia  
  
Wn = 1;  
  
N3 = 3;  
N5 = 5;  
N8 = 8;  
  
% Función de Transferencia en (s) del sistema.  
  
[num3,den3] = butter(N3,Wn,'s');  
[num5,den5] = butter(N5,Wn,'s');  
[num8,den8] = butter(N8,Wn,'s');  
  
% Graficación de la magnitud (escala lineal) *****  
  
H3 = freqs(num3,den3,omega);  
H5 = freqs(num5,den5,omega);  
H8 = freqs(num8,den8,omega);  
  
plot(omega,abs(H3),'k-',omega,abs(H5),'k--',omega,abs(H8),'k:');  
title('Filtros Butterworth');  
ylabel('Magnitud');  
legend('N = 3','N = 5','N = 8');  
xlabel('Frecuencia omega (rad/seg)');  
axis([0 3 0 1.2]);  
grid on;  
  
% Graficación de la magnitud (escala logarítmica) *****  
  
omega = logspace(-1,1); % Barrido log de la frecuencia  
  
H3 = freqs(num3,den3,omega);  
H5 = freqs(num5,den5,omega);  
H8 = freqs(num8,den8,omega);
```

```
H3dB = 20*log10(abs(H3));
H5dB = 20*log10(abs(H5));
H8dB = 20*log10(abs(H8));

figure(2);

semilogx(omega, H3dB, 'k-', omega, H5dB, 'k--', omega, H8dB, 'k:');
axis([0.1 10 -160 10]);
hold on;
title('Filtros Butterworth');
ylabel('Magnitud en dB');
xlabel('Frecuencia omega (rad/seg)');
legend('N = 3', 'N = 5', 'N = 8');
axis([0.1 10 -160 10]);
grid on;
```



## PROGRAMA 13.4.1

Capítulo 13  
Sección 13.4  
Figuras 13.15 y 13.16

```
% *****  
%  
% Programa en Matlab para obtener la respuesta en  
% frecuencia (magnitud) de filtros analógicos  
% pasa-bajas Chebyshev Tipo I de orden N = 3, 5 y 8  
% Wn = frecuencia de corte = 1.0  
% Rp = rizo pasa-banda = 1 dB  
%  
% *****  
  
clear all;  
  
% Definiciones *****  
  
omega = 0:0.05:10; % Barrido lineal de frecuencia  
  
Rp = 1;  
Wn = 1;  
  
N3 = 3;  
N5 = 5;  
N8 = 8;  
  
% Función de Transferencia en (s) del sistema.  
  
[num3,den3] = cheby1(N3,Rp,Wn,'s');  
[num5,den5] = cheby1(N5,Rp,Wn,'s');  
[num8,den8] = cheby1(N8,Rp,Wn,'s');  
  
% Graficación de la magnitud (escala lineal) *****  
  
H3 = freqs(num3,den3,omega);  
H5 = freqs(num5,den5,omega);  
H8 = freqs(num8,den8,omega);  
  
plot(omega,abs(H3),'k-',omega,abs(H5),'k--',omega,abs(H8),'k:');  
title('Filtros Chebyshev Tipo I');  
ylabel('Magnitud');  
xlabel('Frecuencia omega (rad/seg)');  
legend('N = 3','N = 5','N = 8');  
axis([0 3 0 1.2]);  
grid on;  
  
% Graficación de la magnitud (escala logarítmica) *****  
  
omega = logspace(-1,1); % Barrido log de la frecuencia  
  
H3 = freqs(num3,den3,omega);  
H5 = freqs(num5,den5,omega);  
H8 = freqs(num8,den8,omega);
```

```
H3dB = 20*log10(abs(H3));
H5dB = 20*log10(abs(H5));
H8dB = 20*log10(abs(H8));

figure(2);

semilogx(omega,H3dB,'k-',omega,H5dB,'k--',omega,H8dB,'k:');
title('Filtros Chebyshev Tipo I');
ylabel('Magnitud en dB');
xlabel('Frecuencia omega (rad/seg)');
legend('N = 3','N = 5','N = 8');
axis([0.1 10 -80 5]);
grid on;
```

## PROGRAMA 13.4.2

Capítulo 13  
Sección 13.4  
Figuras 13.17 y 13.18

```
% *****
%
% Programa en Matlab para obtener la respuesta en
% frecuencia (magnitud) de filtros analógicos
% pasa-bajas Chebyshev Tipo II de orden N = 3, 5 y 8
% Wn = frecuencia de corte = 1.0
% Rs = atenuación mínima en la banda-rechazo = 20 dB
%
% *****

clear all;

% Definiciones *****

omega = 0:0.05:10; % Barrido lineal de frecuencia

Rs = 20;
Wn = 1;

N3 = 3;
N5 = 5;
N8 = 8;

% Función de Transferencia en (s) del sistema.

[num3,den3] = cheby2(N3,Rs,Wn,'s');
[num5,den5] = cheby2(N5,Rs,Wn,'s');
[num8,den8] = cheby2(N8,Rs,Wn,'s');

% Graficación de la magnitud (escala lineal) *****

H3 = freqs(num3,den3,omega);
H5 = freqs(num5,den5,omega);
H8 = freqs(num8,den8,omega);

plot(omega,abs(H3),'k-',omega,abs(H5),'k--',omega,abs(H8),'k:');
title('Filtros Chebyshev Tipo II');
ylabel('Magnitud');
xlabel('Frecuencia omega (rad/seg)');
legend('N = 3','N = 5','N = 8');
axis([0 3 0 1.2]);
grid on;

% Graficación de la magnitud (escala logarítmica) *****

omega = logspace(-1,1); % Barrido log de la frecuencia

H3 = freqs(num3,den3,omega);
H5 = freqs(num5,den5,omega);
H8 = freqs(num8,den8,omega);
```

```

H3dB = 20*log10(abs(H3));
H5dB = 20*log10(abs(H5));
H8dB = 20*log10(abs(H8));

figure(2);

semilogx(omega,H3dB,'k-',omega,H5dB,'k--',omega,H8dB,'k:');
title('Filtros Chebyshev Tipo II');
ylabel('Magnitud en dB');
xlabel('Frecuencia omega (rad/seg)');
legend('N = 3','N = 5','N = 8');
axis([0.1 10 -60 5]);
grid on;

```

## PROGRAMA 13.5.1

Capítulo 13  
Sección 13.5  
Figuras 13.19 y 13.20

```
% *****
%
% Programa en Matlab para obtener la respuesta en
% frecuencia (magnitud) de filtros analógicos
% pasa-bajas Bessel de orden "N" = 3, 5 y 8
% Wn = frecuencia de corte = 1.0
%
% *****

clear all;

% Definiciones *****

omega = 0:0.05:10; % Barrido lineal de frecuencia

Wn = 1;

N3 = 3;
N5 = 5;
N8 = 8;

% Función de Transferencia en (s) del sistema.

[num3,den3] = besself(N3,Wn);
[num5,den5] = besself(N5,Wn);
[num8,den8] = besself(N8,Wn);

% Graficación de la magnitud (escala lineal) *****

H3 = freqs(num3,den3,omega);
H5 = freqs(num5,den5,omega);
H8 = freqs(num8,den8,omega);

plot(omega,abs(H3),'k-',omega,abs(H5),'k--',omega,abs(H8),'k:');
title('Filtros Bessel');
ylabel('Magnitud');
xlabel('Frecuencia omega (rad/seg)');
legend('N = 3','N = 5','N = 8');
axis([0 3 0 1.2]);
grid on;

% Graficación de la magnitud (escala logarítmica) *****

omega = logspace(-1,1); % Barrido log de la frecuencia

H3 = freqs(num3,den3,omega);
H5 = freqs(num5,den5,omega);
H8 = freqs(num8,den8,omega);
```

```
H3dB = 20*log10(abs(H3));
H5dB = 20*log10(abs(H5));
H8dB = 20*log10(abs(H8));

figure(2);

semilogx(omega, H3dB, 'k-', omega, H5dB, 'k--', omega, H8dB, 'k:');
title('Filtros Bessel');
ylabel('Magnitud en dB');
xlabel('Frecuencia omega (rad)seg');
legend('N = 3', 'N = 5', 'N = 8');
axis([0.1 10 -160 10]);
grid on;
```

## PROGRAMA 13.6.1

Capítulo 13  
Sección 13.6  
Figuras 13.21 y 13.22

```
% *****  
%  
% Programa en Matlab para obtener la respuesta en  
% frecuencia (magnitud) de filtros analógicos  
% pasa-bajas Elípticos de orden N = 3 y 8  
% Wn = frecuencia de corte = 1.0  
% Rp = rizo en la banda de paso = 1 dB  
% Rs = atenuación mínima en la banda-rechazo = 10 dB  
%  
% *****  
  
clear all;  
  
% Definiciones *****  
  
omega = 0:0.05:10; % Barrido lineal de frecuencia  
  
Rp = 1;  
Rs = 20;  
Wn = 1;  
  
N3 = 3;  
N8 = 8;  
  
% Función de Transferencia en (s) del sistema.  
  
[num3,den3] = ellip(N3,Rp,Rs,Wn,'s');  
[num8,den8] = ellip(N8,Rp,Rs,Wn,'s');  
  
% Graficación de la magnitud (escala lineal) *****  
  
H3 = freqs(num3,den3,omega);  
H8 = freqs(num8,den8,omega);  
  
plot(omega,abs(H3),'k-',omega,abs(H8),'k--');  
title('Filtros de Función Elíptica');  
ylabel('Magnitud');  
xlabel('Frecuencia omega (rad/seg)');  
legend('N = 3','N = 8');  
axis([0 3 0 1.2]);  
grid on;  
  
% Graficación de la magnitud (escala logarítmica) *****  
  
omega = logspace(-1,1); % Barrido log de la frecuencia  
  
H3 = freqs(num3,den3,omega);  
H8 = freqs(num8,den8,omega);
```

```
H3dB = 20*log10(abs(H3));
H8dB = 20*log10(abs(H8));

figure(2);

semilogx(omega, H3dB, 'k-', omega, H8dB, 'k--');
title('Filtros de Función Elíptica');
ylabel('Magnitud en dB');
xlabel('Frecuencia omega (rad/seg)');
legend('N = 3', 'N = 8');
axis([0.1 10 -45 5]);
grid on;
```



## PROGRAMA 14.1.1

Capítulo 14  
Sección 14.1  
Figuras 14.3 a 14.6

```
% *****  
%  
% Grafica de supericie (3D) de la  
% Transformada Discreta de Fourier  
%  
% *****  
  
% Definición de variables *****  
  
delta = .1;  
liminf = -2;  
limsup = abs(liminf);  
Rezmax=2*(limsup/delta)+1;  
Imzmax = Rezmax;  
  
% Definición del plano de graficación *****  
  
[x,y] = meshgrid(liminf:delta:limsup);  
  
% Conversión del plano de graficación al plano  
% complejo Z *****  
  
z = y.*j;  
z = z+x; % Plano complejo Z  
  
% Evaluación de la Función de Transferencia:  
%  
%  $H(z) = 3 / [(z+0.5+j)(z+0.5-j)]$   
%  
% en todo el plano Z  
% *****  
  
Hz = zeros(Rezmax,Imzmax);  
  
for Rez = 1:Rezmax  
    for Imz = 1:Imzmax  
        Hz(Rez,Imz)=3/((z(Rez,Imz)+0.5+j)*(z(Rez,Imz)+0.5-j));  
    end  
end  
  
% Calculo de la magnitud y fase de H(z) *****  
  
MagnitudHz = abs(Hz);  
FaseHz = angle(Hz)*180/pi; % Fase en grados
```

```

% Graficación de la Magnitud |H(z)| *****

surfl(x,y,MagnitudHz);
shading interp;
colormap(gray);
rotate3d;
axis([liminf limsup liminf limsup -2 8]);

xlabel('Re[z]');
ylabel('Im[z]');
zlabel('Magnitud de |H(z)|');
title('H(z) = 3/(z+0.5+j)(z+0.5-j)');

% *****
% Muestreo de H(z) en el círculo unitario exp(jw)
% *****

% Inicialización de la matriz de muestras de H(z)

SampleHz = zeros(Rezmax,Imzmax);

N = 8;      % Número de muestras

% Inicialización de la matriz de muestras de H(k)

MagnitudHk = zeros(1,N);
FaseHk = zeros(1,N);

ws = zeros(1,N); % Vector de frecuencias de muestreo

k = 1;      % Inicialización del índice de H(k)

% Evaluación de las muestras *****

for Rez = 1:Rezmax
    for Imz = 1:Imzmax
        for w = 0:2*pi/N:2*pi % Recorrido del círculo
            e = exp(j*w);      % unitario exp(jw)
            % Búsqueda de puntos de coincidencia entre
            % el círculo unitario exp(-jw) y el plano de
            % graficación x-y
            dif = abs(e-z(Rez,Imz));
            if dif < 0.02
                % Cálculo de las muestras de H(z)
                SampleHz(Rez,Imz)=3/((z(Rez,Imz)+0.5+j)*(z(Rez,Imz)+0.5-j));
                MagnitudHk(k) = abs(SampleHz(Rez,Imz));
                FaseHk(k) = angle(SampleHz(Rez,Imz))*180/pi;
                ws(k) = w/pi;
                k = k+1;
            end
        end
    end
end
end
end

```

```

% Calculo de la magnitud y fase de las muestras
% de H(z) en el círculo unitario *****

MagnitudSampleHz = abs(SampleHz);
FaseSampleHz = angle(SampleHz)*180/pi;

% Graficación de las muestras de la Magnitud |H(z)|

hold on;
stem3(x,y,MagnitudSampleHz);
axis([liminf limsup liminf limsup -2 8]);
grid on;

% Graficación de la Fase de H(z) y de sus
% muestras en el círculo unitario *****

figure(2);

surfl(x,y,FaseHz);
shading interp;
colormap(gray);
rotate3d;
axis([-2 2 -2 2 -100 100]);
xlabel('Re[z]');
ylabel('Im[z]');
zlabel('Fase en grados de [H(z)]');
title('H(z) = 3/(z+0.5+j)(z+0.5-j)');

hold on;
stem3(x,y,FaseSampleHz);
grid on;

% Graficación de las muestras de la magnitud y
% fase de la función G(z)

figure(3);

subplot(211);
stem(ws,MagnitudHk);
title('Transformada Discreta de Fourier de la función: H(z) =
3/(z+0.5+j)(z+0.5-j)');
ylabel('Magnitud: |H(k)|');
xlabel('Frecuencia Normalizada: w/pi');
grid on;

subplot(212);
stem(ws,FaseHk);
ylabel('Fase en grados');
xlabel('Frecuencia Normalizada: w/pi');
grid on;

```

```

% *****
% Cálculo de la respuesta en frecuencia de H(z)
%
%
%          -2
%          3z
% H(z) = ----- = -----
%          -1      -2      (z+0.5+j)(z+0.5-j)
%          1 + z   +1.25z
%
% *****

B = [0 0 3];
A = [1 1 1.25];
[HHz,w] = freqz(B,A,256,'whole');

% Graficación de la respuesta en frecuencia de H(z)
% y de sus muestras en el círculo unitario

figure(4);

subplot(211);
plot(w/pi,abs(HHz));
hold on;
stem(ws,MagnitudHk);
title('Respuesta en frecuencia de la función: H(z) = 3/(z+0.5+j)(z+0.5-j)');
ylabel('Magnitud: |H(k)|');
xlabel('Frecuencia Normalizada: w/pi');
grid on;

subplot(212);
plot(w/pi,angle(HHz)*180/pi);
hold on;
stem(ws,FaseHk);
ylabel('Fase en grados');
xlabel('Frecuencia Normalizada: w/pi');
grid on;

```

## PROGRAMA 14.5.1

Capítulo 14  
Sección 14.5  
Ejemplo 14.5.1  
Figuras 14.10 a 14.12

```
% *****
%
% Programa en Matlab para obtener la convolucion lineal
% de dos secuencias de duracion finita, mediante el uso
% de la TDF:
%
% y(n) = x(n) .convolucion. h(n)
%
% con:      x(n) = [1 2 0 1]; h(n) = [2 2 1 1]
%
% *****

clear all;

% Definición de las secuencias *****

xn = [1 2 0 1];          % Primera secuencia de ejemplo
hn = [2 2 1 1];          % Segunda secuencia de ejemplo

% Cálculo de la longitud de la secuencia resultante *****

L = length(xn)+length(hn)-1;

% Cálculo de las TDF's aumentando con ceros *****

XE = fft(xn,L);
HE = fft(hn,L);

% Cálculo de la TDFI del producto de las secuencias *****
% (Cálculo de la Convolución Lineal a través de la
% Convolución Circular)

ylcn = ifft(XE.*HE);

% Gráfica de la secuencias x(n) y h(n)

n = 0:3;

figure(1);

subplot (211);
stem(n,xn);
axis([0 6 0 2.5]);
title('x(n)');
grid on;
```

```

subplot (212);
stem(n,hn);
axis([0 6 0 2.5]);
title('h(n)');
xlabel('Indice de tiempo: n');
grid on;

% Gráfica de la secuencia ylc(n) generada por convolución
% circular

ne = 0:L-1;

figure(2);

subplot(211);
stem(ne,abs(ylcn));
axis([0 6 0 7]);
title('Magnitud de la Convolución Lineal ylc(n) calculada a través de la
Convolución Circular');
ylabel('Amplitud');
grid on;

% Cálculo de la Convolución Lineal Directa

yldn = conv(xn,hn);

% Gráfica de la secuencia yld(n) generada por Convolución
% Lineal Directa

subplot(212);
stem(ne,abs(yldn));
axis([0 6 0 7]);
title('Magnitud de la Convolución Lineal Directa: yld(n) =
conv[x(n),h(n)]');
xlabel('Indice de tiempo: n');
ylabel('Amplitud');
grid on;

% Cálculo y gráfica del error entre los dos cálculos de la
% convolución

figure(3);

error = abs(ylcn)-abs(yldn);

stem(ne,error);
title('Magnitud del error: ylc(n)-yld(n)');
xlabel('Indice de tiempo: n');
ylabel('Amplitud');
grid on;

```