

CURSO: PROCESAMIENTO DIGITAL DE SEÑALES

TEMA: APLICACIÓN DE FILTROS DIGITALES-SIMULACIÓN EN MATLAB

PROFESOR: JORGE ANTONIO POLANÍA P

1. SEÑALES ECG

EJEMPLO 1:

```
%SEÑAL ECG FILTRADA CON IIR
clear all

%Carga de la señal ecg
x = load ('ecg1.txt');
%graficamos la señal de muestra
N=length(x);
%N=1000
dt=0.01;
Ejet=0:dt:(N-1)*dt;
% t=0 0.01 0.02 .....9.99
figure(1)
plot(Ejet,x(:,2));
%axis([0 1 min(x) max(x)]);
xlabel('t(sg) ');
ylabel('y(mV) ');
title('Señal ECG')

%Espectro de la señal ecg
df=N*dt;
Ejef=[0:1/df:(N-1)/df];
spectr=abs(fft(x));
spectr=spectr/max(spectr);
figure(2)
plot(Ejef,spectr);
%axis([0 10 0 max(spectr)]);
xlabel('frecuencia(Hz) ');
title('Espectro de la señal ECG')

%DISEÑO DEL FILTRO PASA BANDA IIR BUTTERWORTH
fs=1000;
fN=fs/2;
wp=[100 250]/fN;
ws=[10 300]/fN;
%Atenuaciones en banda paso y banda stop
Rp=3;
Rs=30;
[n,wn]=buttord(wp,ws,Rp,Rs);
[b,a]=butter(n,wn);
%Respuesta en frecuencia del filtro
Hd = dfilt.df1(b,a)
```

```

%Salida de la señal filtrada mediante convolución usando
filter
y=filter(Hd,x(:,2));
%subplot(414);
figure(3)
plot(Ejet,y)
xlabel('Tiempo(sg)');
title('Senal de salida');

%axis([0 1 min(y) max(y)]);

%Espectro de frecuencia de la salida
figure(4)
plot(Ejef,abs(fft(y)));
%axis([0 100 0 500]);
xlabel('Frecuencia (Hz)');
title('Espectro de la salida');

```

EJEMPLO 2:

```

%FILTRADO DE SEÑAL ECG CON FILTROS FIR
clear all
%Carga de la señal ecg
ecg = load ('ecg2.txt');
%graficamos la señal de muestra
x=ecg(:,2);
N=length(x);
%N=1000
dt=0.01;
t=0:dt:(N-1)*dt;
% t=0 0.01 0.02 .....9.99
figure(1)
plot(t,x);
%axis([0 1 min(x) max(x)]);
xlabel('t(sg)');
ylabel('y(mV)')
title('Señal ECG')

%Transformada de Fourier de ECG
Espect=fft(x);
%como los coeficientes de fft son simétricos, se toman solo
la mitad
%mitadEspect=Espect(1:N/2);
df=N*dt;
%Ejef=0:1/df:(N/2-1)/df;
Ejef=0:1/df:(N-1)/df;
figure(2)
%plot(Ejef,abs(mitadEspect))
plot(Ejef,abs(Espect))

```

```

%xlim([0 1])
xlabel('f (Hz)');
ylabel('I Magnitud I');
title('Espectro ECG')

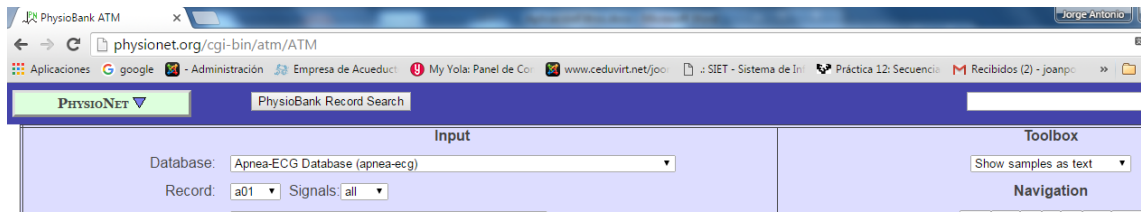
%filtro Notch FIR a 50 Hz
fs=1000;
fN=fs/2;
w1=48/fN;
w2=52/fN;
h50=fir1(50,[w1 w2],'stop');
Hd=dfilt.dffir(h50);
y50=filter(Hd,x);
figure(3)
plot(t,y50,'r')
xlabel('t(sg)');
ylabel('mV')
title('ECG sin ruido en 50Hz')

%Filtro pasa bajo FIR para eliminar frecuencias menores a 1
Hz,
fs=1000;
fN=fs/2;
wn=1/fN;
hDC=fir1(50,wn,'high');
yDC=filter(hDC,1,y50);
figure(4)
plot(t,yDC,'r')
xlabel('t(sg)');
ylabel('mV')
title('ECG sin componente DC')

%Espectro final
Espect50=fft(yDC);
%como los coeficientes de fft son simétricos, se toman solo
la mitad
%mitadEspect50=Espect50(1:N/2);
df=N*dt;
%Ejef=0:1/df:(N/2-1)/df;
Ejef=0:1/df:(N-1)/df;
figure(5)
plot(Ejef,abs(Espect50))
xlabel('f (Hz)');
ylabel('I Espect I')
title('Espectro final')

```

En esta página pueden obtener señales ECG:



2. SEÑALES AUDIO

EJEMPLO 3:

```
%FILTRAR UN AUDIO CON FILTRO IIR
% Crear archivo WAV.
load handel.mat
audio1 = 'handel.wav';
%wavwrite(x, Fs, hfile)

% Leer archivo y escuchar audio.
[x, Fs, nbits, readinfo] = wavread(audio1);
sound(x, Fs);

%Filtro pasa bajo IIR elíptico
fN=Fs/2;
wp = 2000/fN;
ws = 4000/fN;
Rp = 1;
Rs = 80;
[n,wn]=ellipord(wp,ws,Rp,Rs);
[b,a]=ellip(n,Rp,Rs,wn);
Hd=dfilt.df1(b,a);
y=filter(Hd,x);
sound(y, Fs)
```

EJEMPLO 4:

```
%ECUALIZADOR CON FILTROS FIR
% Crear archivo WAV.
load handel.mat
audio1 = 'handel.wav';
% Leer archivo y escuchar audio.
[x, Fs, nbits, readinfo] = wavread(audio1);
sound(x, Fs);

%Diseño de filtros
Wn = 0.2; %Frecuencia de corte
n = 62; %orden del filtro
%Ganancia en las bandas
gLP = 0.4; %Pasa bajo
gBP = 1.5; %Pasa banda
```

```

gHP = 1.5; %pasa alto
%Filtros FIR
LP = fir1(n,Wn);
Wn1 = [0.2, 0.5];
BP = fir1(n,Wn1);
Wn2 = 0.5;
HP = fir1(n,Wn2,'high');
y1 = filter(LP,1,x);
y2 = filter(BP,1,x);
y3 = filter(HP,1,x);
yA= gHP * y3;
wavwrite(yA,Fs,'Ecualiz3');
yB= gLP * y1;
wavwrite(yB,Fs,'Ecualiz1');
yC= gBP * y2;
wavwrite(yC,Fs,'Ecualiz2');
yD = yA + yB + yC;
wavwrite(yD,Fs,'Ecualiz4');
sound(yD,Fs)

```

3. SEÑALES SÍSMICAS

EJEMPLO 5:

```

%ELIMINAR RUIDO DE LA SEÑAL
sismo
delt = 1/200;
t = delt*(1:length(e))';

%Grafica de la señal sísmica
yrange = [-250 250];
limits = [0 50 yrange];
y=e;
subplot(311)
plot(t,y,'b')
grid
axis(limits)
title('Señal sísmica')

%Espectro de frecuencia
N = length(y); % longitud del vector y
Fs = 20; % frecuencia de muestreo
Y = fft(y);
mag= Y.*conj(Y) / N;
f = Fs*(0:N/2-1) / N;
subplot(312)
plot(f,mag(1:N/2),'g')
grid
title('Espectro de frecuencia')

```

```

%Filtración de ruido filtro pasa alto
fcorte=8;    %elimina debajo de 8 Hz
[b, a] = butter (5,fcorte/(Fs/2), 'high');
yf = filtfilt (b,a,y);
subplot(313)
plot (t,y, 'red')
title('Señal de salida filtrada')
grid
Hd=dfilt.df1(b,a);

```

EJEMPLO 6:

```

% CÁLCULO DE LAS FASES, PERIODO Y LA AMPLITUD DE UNA ONDA
SÍSMICA
%señal sísmica
sismo
y=e;
Ny = length(y);    % longitud del vector de entrada
% Ny=10001

%Grafica de la señal sísmica
t = 1/200*(1:Ny)';    %vector tiempo
subplot(221)
plot (t,y), grid on
title ('Grafica de la señal sísmica')
Fs = 20;    %frecuencia de muestreo
T = 1/Fs;    % periodo de muestreo

%cálculo de la envolvente
y1 = hilbert(y);
y2 = sqrt(y.*y+y1.*conj(y1));
A2 = max(y2);
y2 = y2/A2;    % envolvente
comp = mean(y2(1:60));
%comp=0.0013
[A,I]=max(abs(y));    % I = posición del máximo
%A=416.3 I=3161
for n = 10:I
if y2(n) > 5.9*comp
n;
break
end
end
for k = 1:n
if y2(n-k)-y2(n-k-1)<0 & y2(n-k)<0.015
Ip = n-k;
tp = (Ip)*T;
break
end
end
end

```

```

fprintf('%s%8.6f\n', ' Tiempo P = ',tp);
%tp=78.1

Ix = round((Ip+I)/2);
for n = Ix:I
if y2(n) > 0.45
n;
break
end
end
% Calculo de la fase S
for k = Ix:n+Ix
if (y2(n+Ix-k)-y2(n+Ix-k-1)<0)&y2(n+Ix-k)<0.09
Is = n+Ix-k;
ts = (Is)*T;
break
end
end
fprintf('%s%8.6f\n', ' Tiempo S = ',ts);
%ts=126.3

% Plotear las gráficas

fprintf('%s%8.6f\n', ' Amplitud = ',A);
Y = fft(y,Ny);
Pyy = Y.*conj(Y) / Ny;
f = (1/T)*(0:Ny/2) / Ny;
subplot(2,2,3), plot(f,Pyy(1:Ny/2+1))
%axis([0 1000 0 30]), grid on
title('Espectro de frecuencias de la señal')
xlabel('Frecuencia (Hz)')

N = 128; % N = número de puntos DTF
for k = 1:N
y11(k) = y(I-N/4+k);
end
subplot(2,2,2), plot (t(1:N),y11), grid on
title ('Zoom en el cambio de fase P-S')

Y = fft(y11,N);
Pyy = Y.*conj(Y) / N;
f = (1/T)*(0:N/2) / N;
delta_f = f(2)-f(1);
subplot(2,2,4), plot(f,Pyy(1:N/2+1))
%axis([0 1000 0 30]), grid on
title('Espectro de frecuencias del zoom')
xlabel('Frecuencia (Hz)')

% Calculo del periodo
[amp_espectro,J] = max (abs(Pyy));
fs = delta_f*(J-1);

```

```

periodo = 1/fs;
fprintf('%s%8.6f\n', ' Periodo = ', periodo);
%periodo=0.1563

% Calcular la duracion
prom_f = sum(abs(y(N-20:N)))/21;
%prom_f=0.49
for k = Ny:-1:I
    if y(k) > 4*prom_f
        %if y(k) > 4*prom_f
        k;
        %td = k*T+ss;
        td=k*T;
        break
    end
end
D = round(td-tp);
fprintf('%s%7.2f\n', ' Duracion = ', D);
%Duración=421

```