Luis Fernando Rodriguez Gtz ie705694

Sistemas de Comunicaciones Digitales
02/24/20

TAREA 04

EXERCISE 1

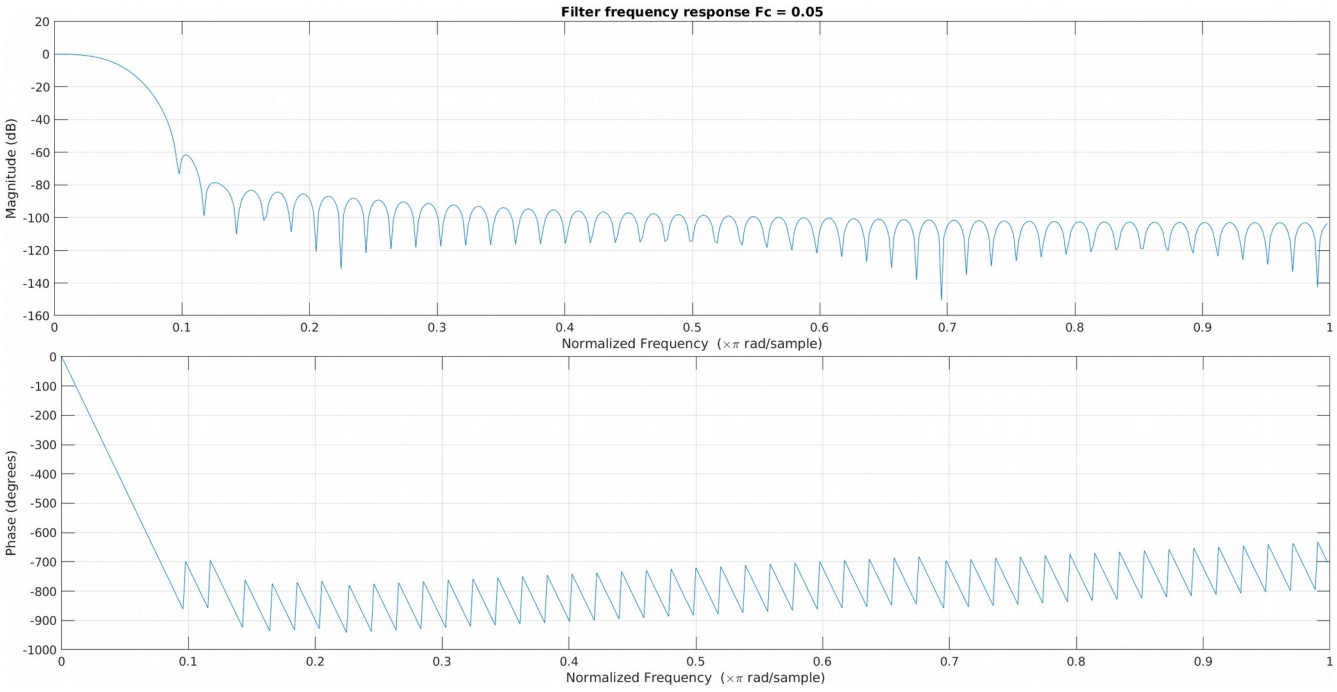
Matlab Code:

```
%% Low pass filter

    %Frequency band edges
f = [0, fc_05, fc_05, 1];
    %Amplitudes
m = [1, 1, 0, 0];
    %Filter order
n = 100;
fir = fir2(n,f,m);

%fir_temp = firls(n,f,m);
%fvtool(fir,1,fir_temp,1);
%phasez(fir);

    %Better way of getting frequency response.
freqz(fir,1);   title('Filter frequency response Fc
= 0.05');
```



Luis Fernando Rodriguez Gtz ie705694

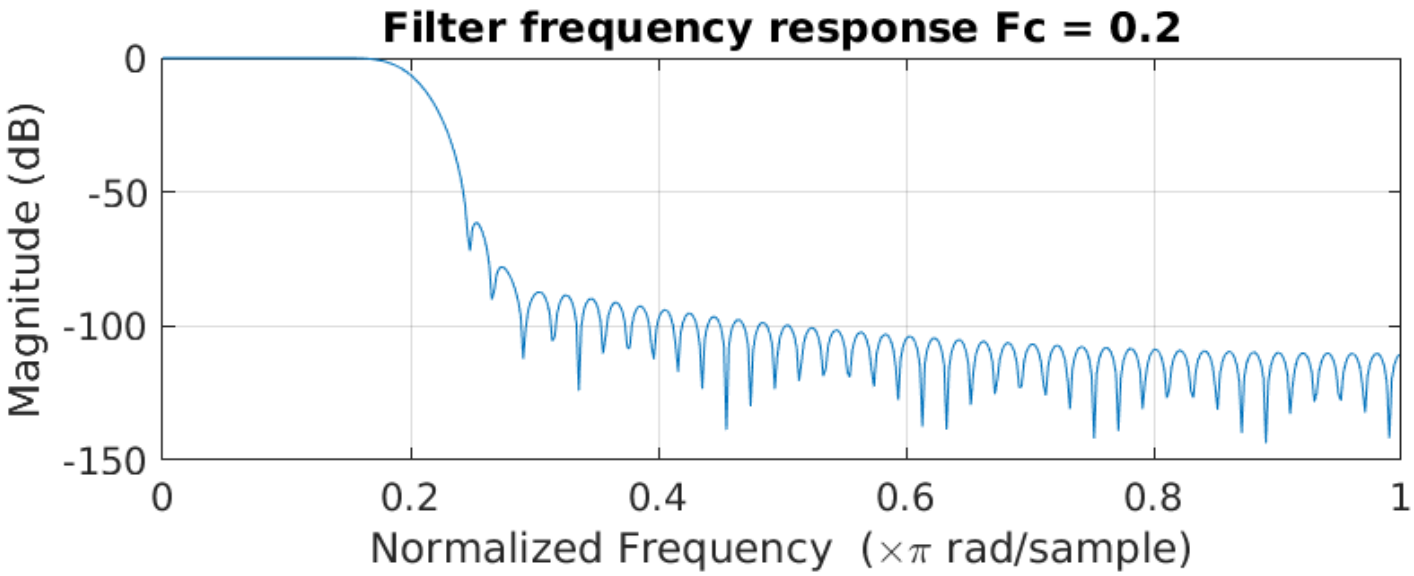
Sistemas de Comunicaciones Digitales
02/24/20

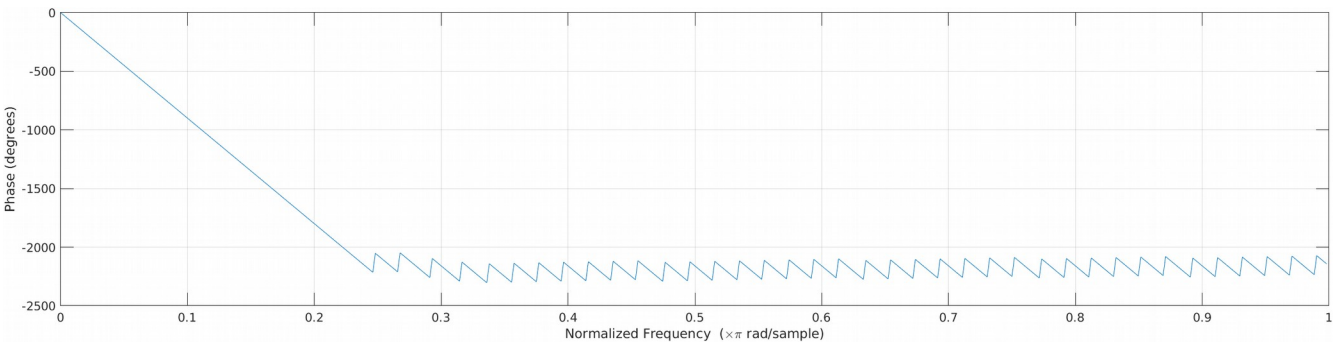
```
%% Low pass filter

    %Frequency band edges
f = [0, fc_2, fc_2, 1];
    %Amplitudes
m = [1, 1, 0, 0];
    %Filter order
n = 100;
fir = fir2(n,f,m);

%fir_temp = firls(n,f,m);
%fvtool(fir,1,fir_temp,1);
%phasez(fir);

    %Better way of getting frequency response.
freqz(fir,1);   title('Filter frequency response Fc
= 0.2');
```





Luis Fernando Rodriguez Gtz ie705694

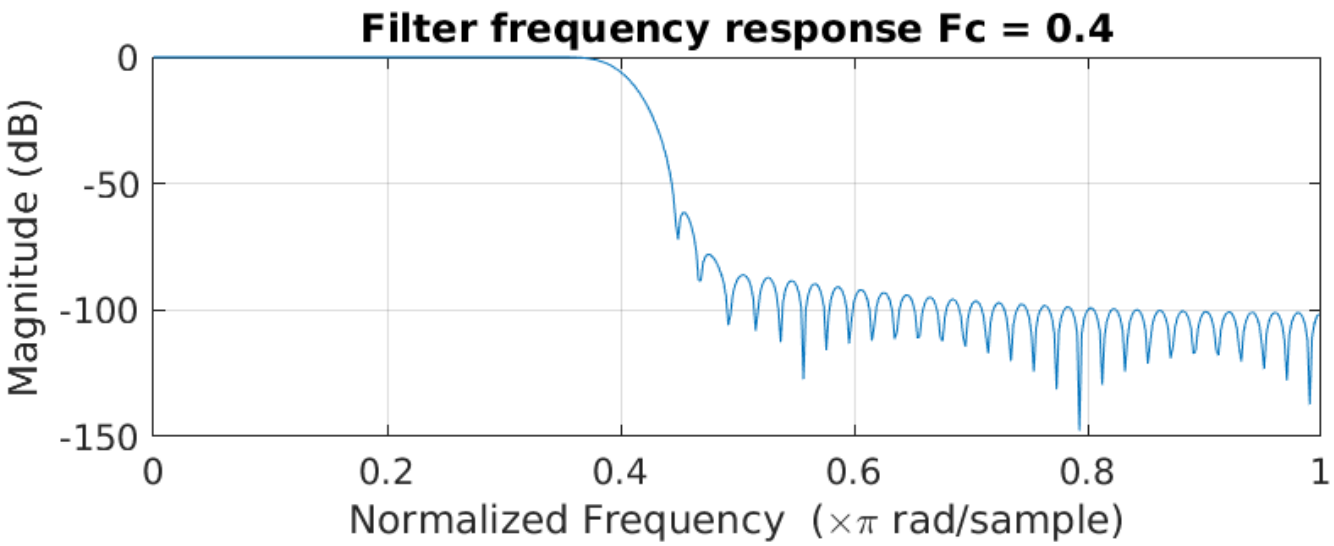
Sistemas de Comunicaciones Digitales
02/24/20

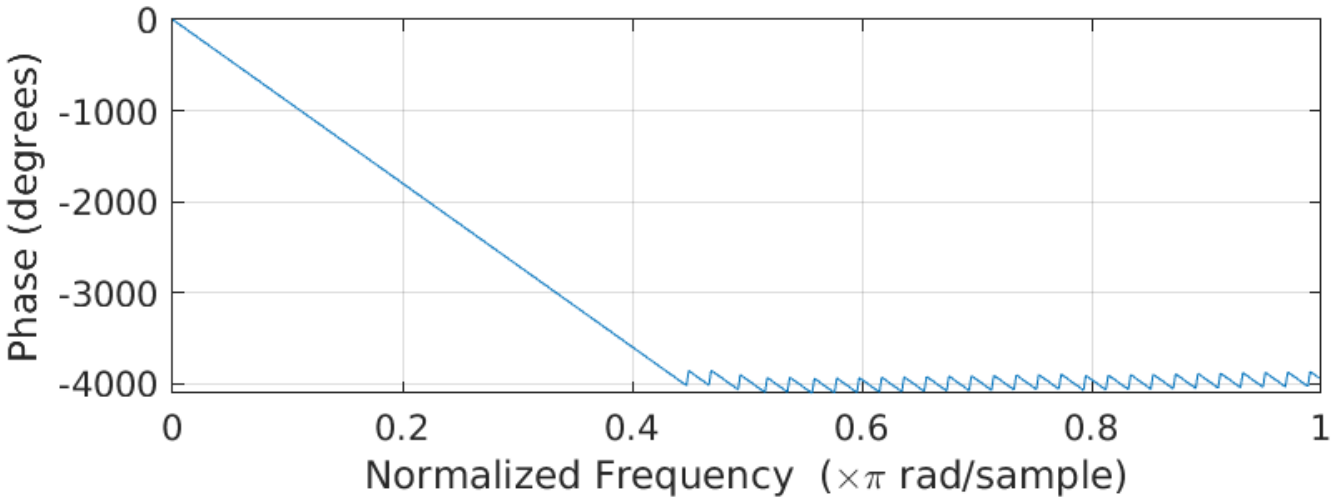
```
%% Low pass filter

    %Frequency band edges
f = [0, fc_4, fc_4, 1];
    %Amplitudes
m = [1, 1, 0, 0];
    %Filter order
n = 100;
fir = fir2(n,f,m);

%fir_temp = firls(n,f,m);
%fvtool(fir,1,fir_temp,1);
%phasez(fir);

    %Better way of getting frequency response.
freqz(fir,1);   title('Filter frequency response Fc
= 0.4');
```



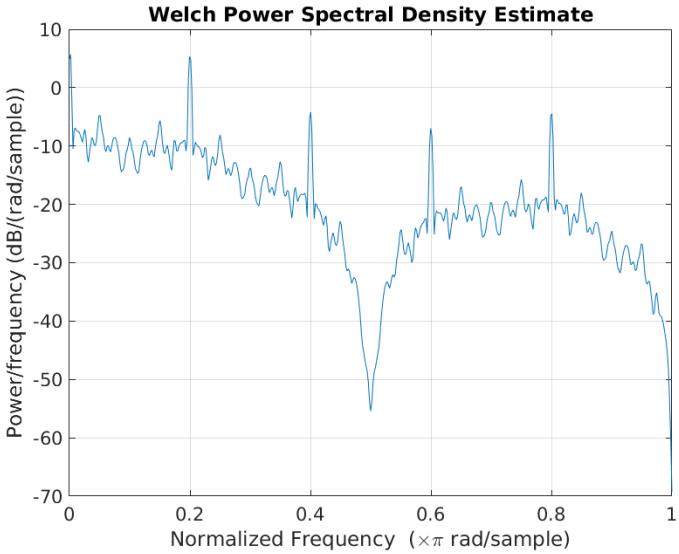


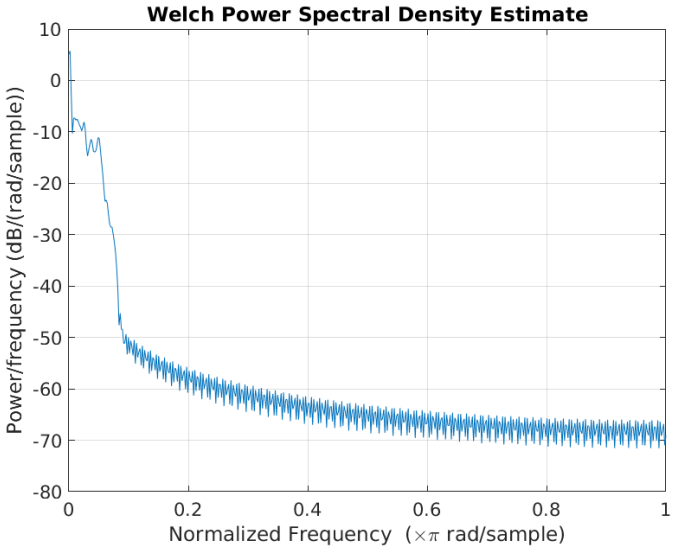
Luis Fernando Rodriguez Gtz ie705694

Sistemas de Comunicaciones Digitales
02/24/20

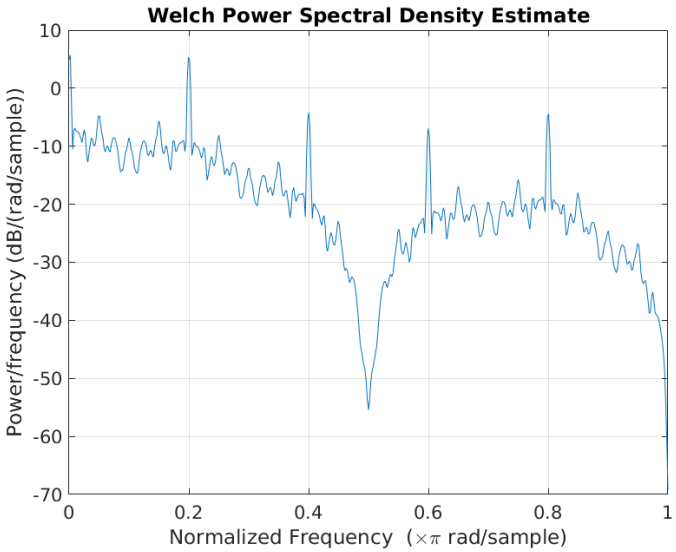
Specter Uni-polar.

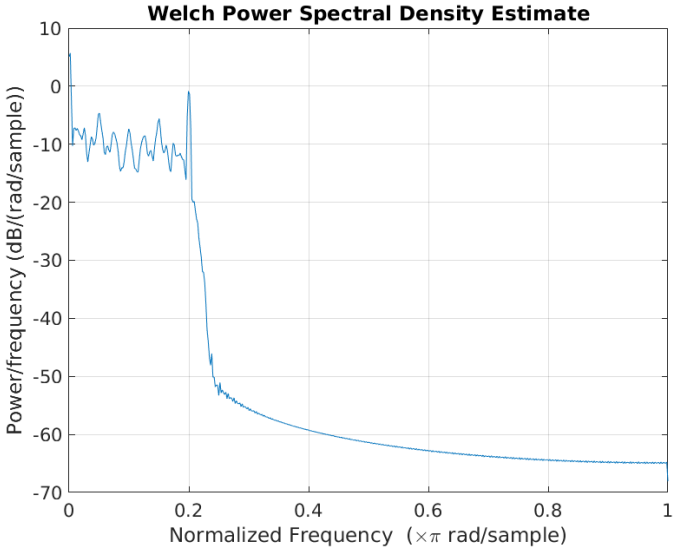
Fc = 0.05





Fc = 0.2

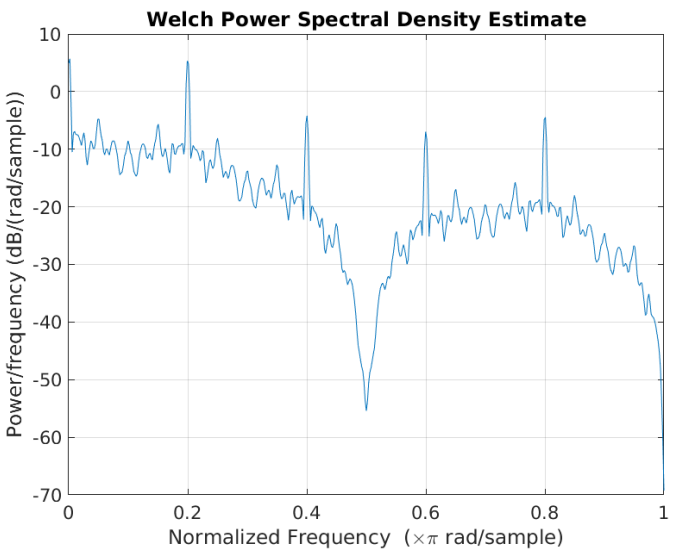


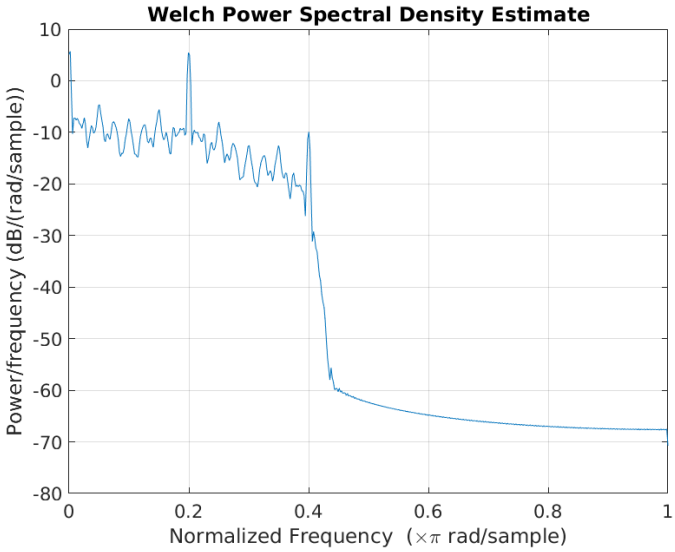


Luis Fernando Rodriguez Gtz ie705694

Sistemas de Comunicaciones Digitales
02/24/20

Fc = 0.4



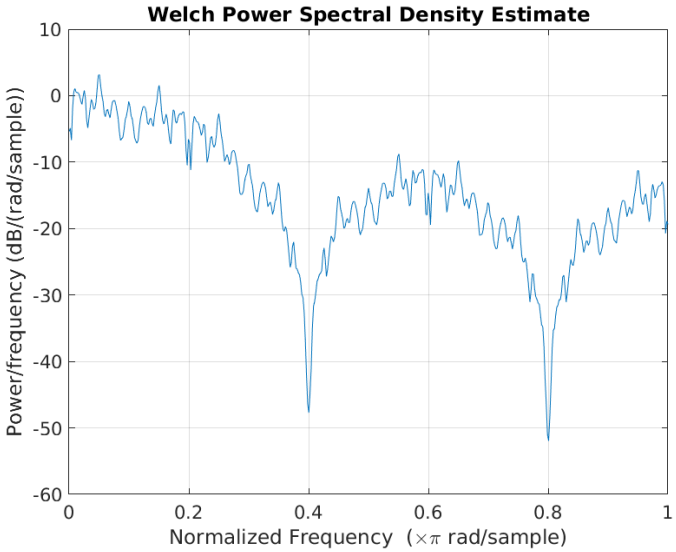


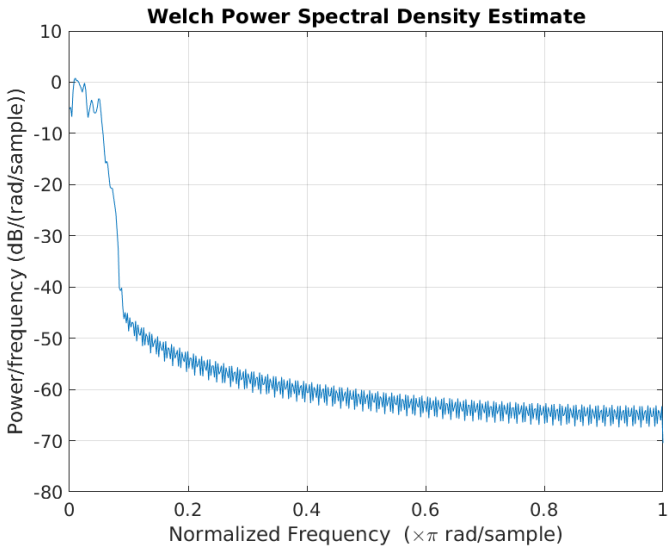
Luis Fernando Rodriguez Gtz ie705694

Sistemas de Comunicaciones Digitales
02/24/20

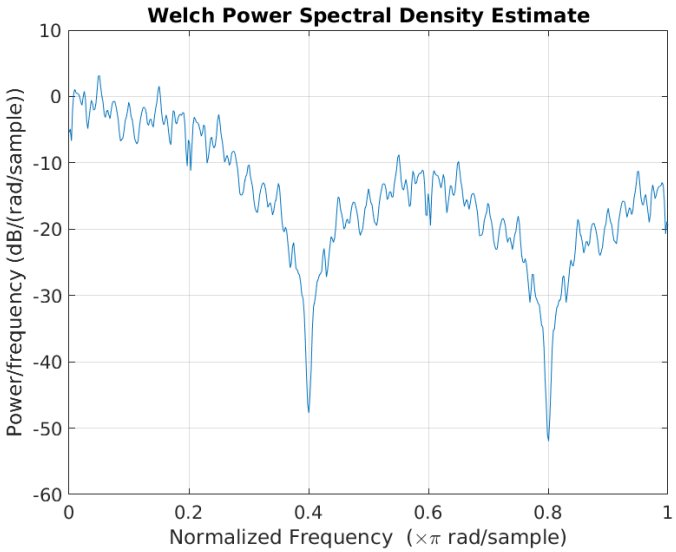
Specter Polar

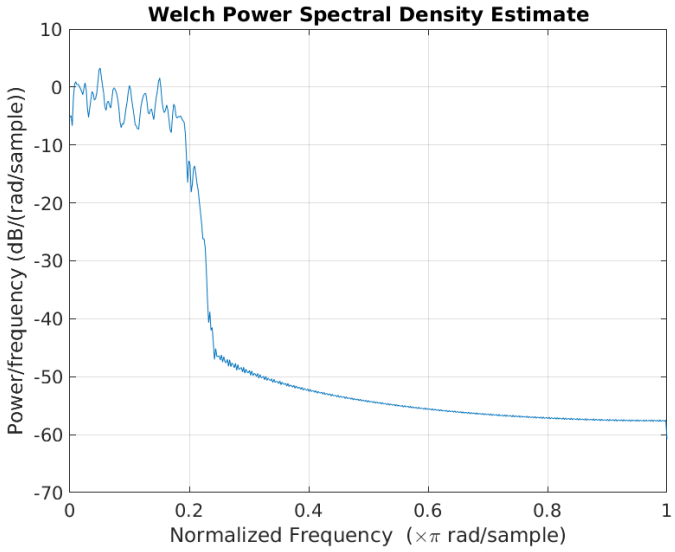
Fc = 0.05





Fc = 0.2

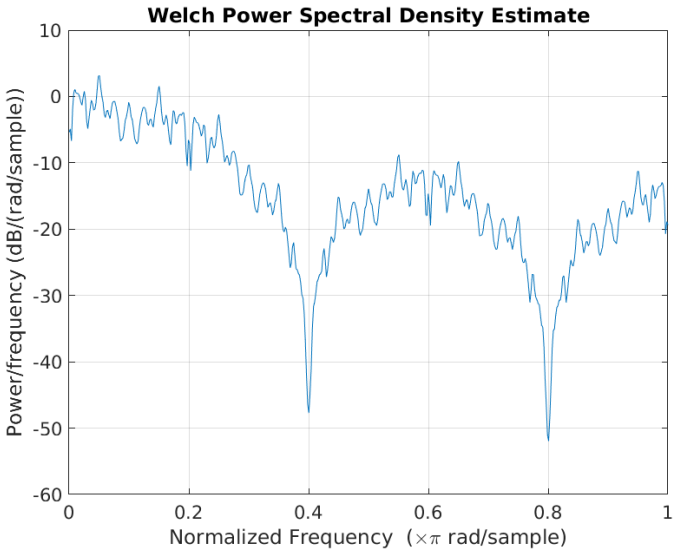


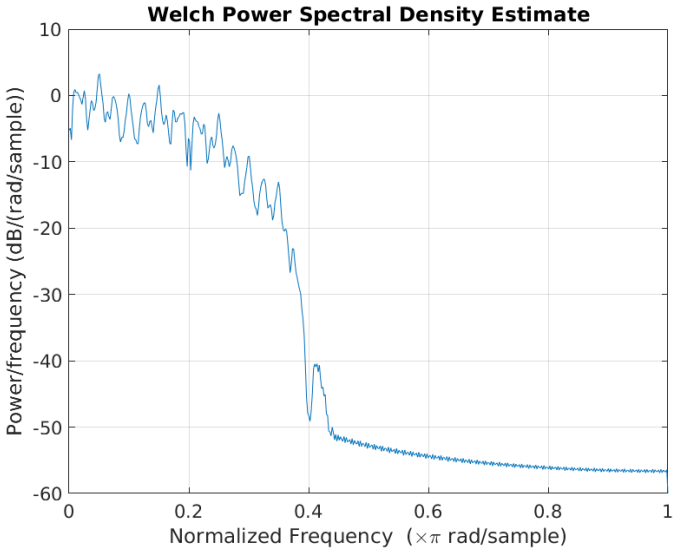


Luis Fernando Rodriguez Gtz ie705694

Sistemas de Comunicaciones Digitales
02/24/20

Fc = 0.4



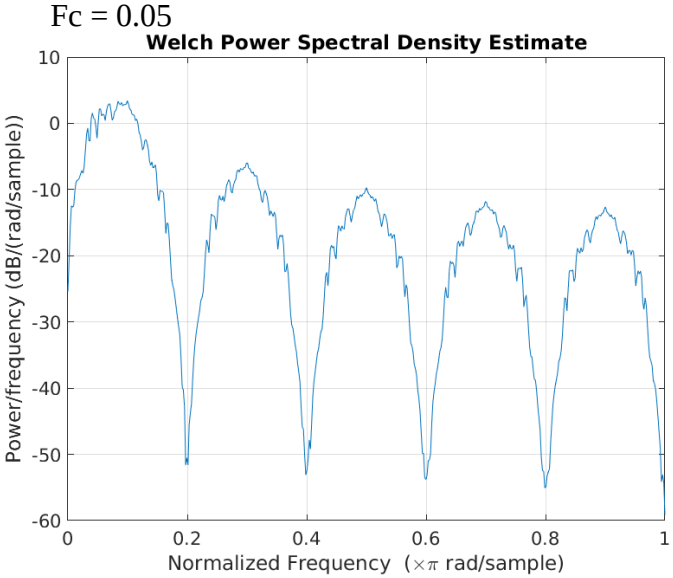


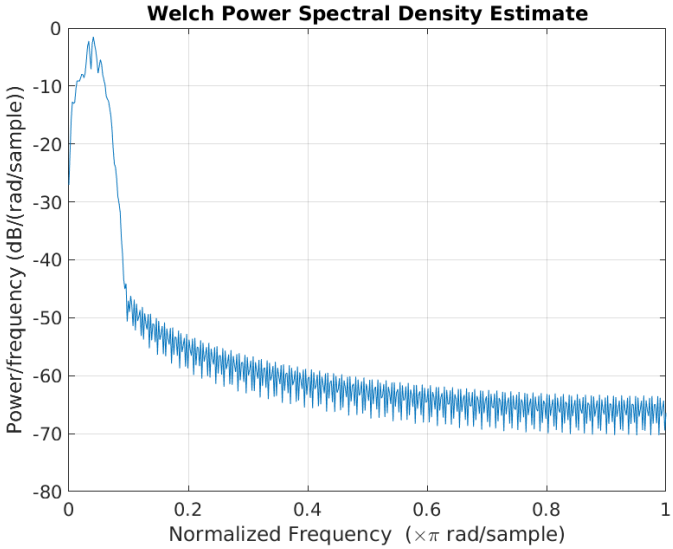
Luis Fernando Rodriguez Gtz ie705694

Sistemas de Comunicaciones Digitales
02/24/20

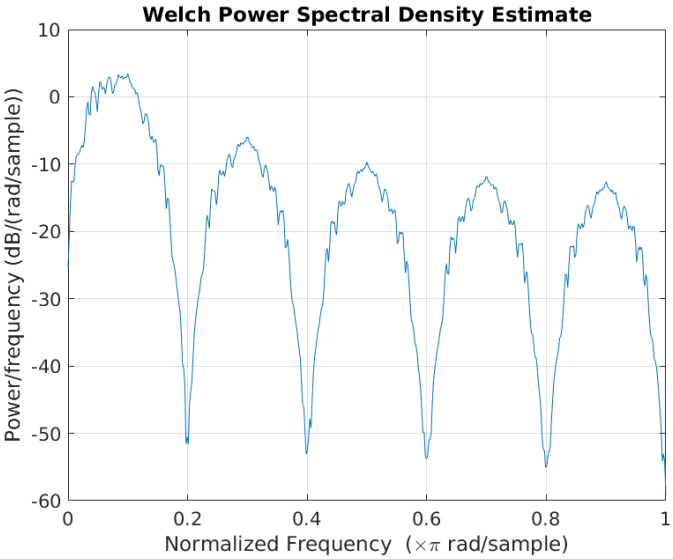
Specter Bi-Polar

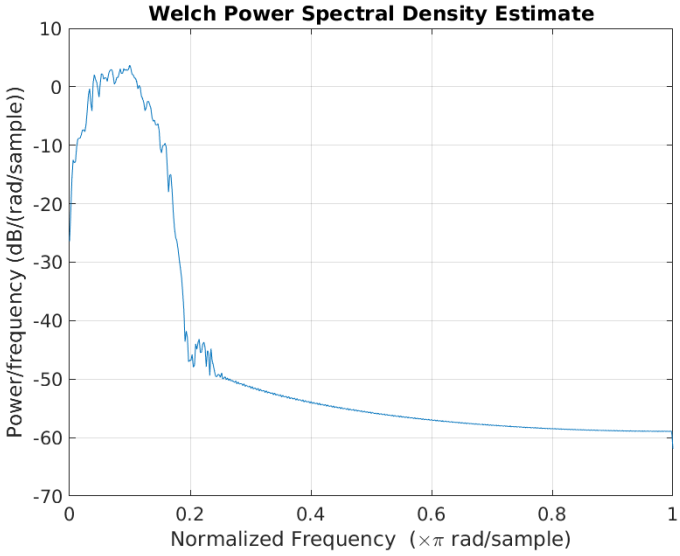
Fc = 0.05





Fc = 0.2

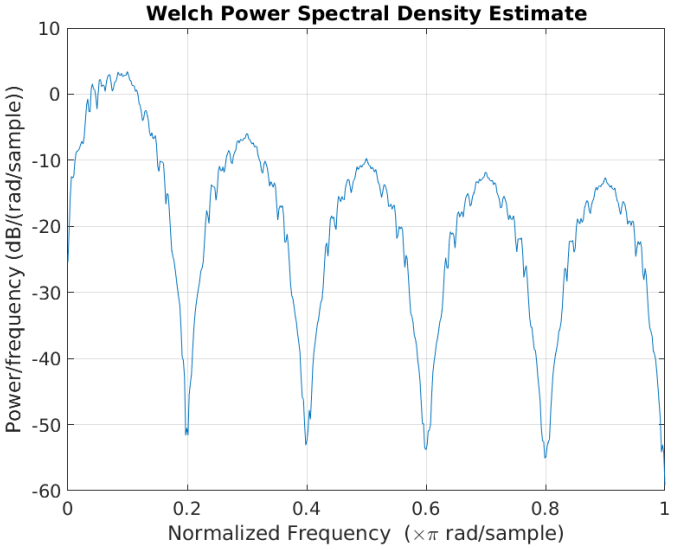


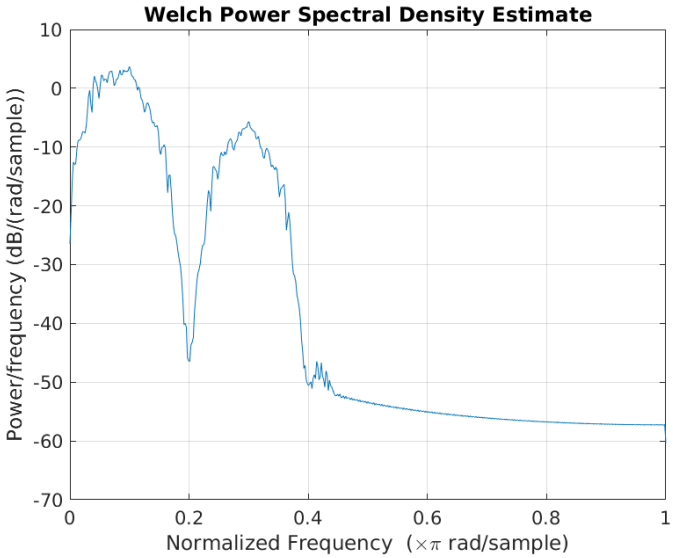


Luis Fernando Rodriguez Gtz ie705694

Sistemas de Comunicaciones Digitales
02/24/20

Fc = 0.4



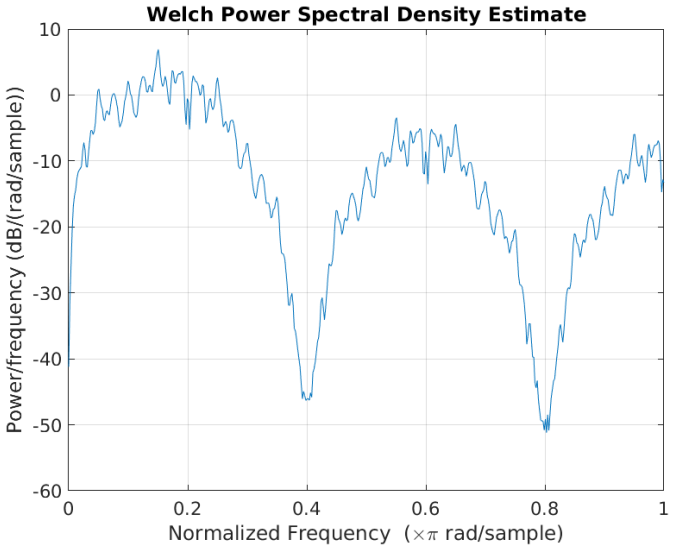


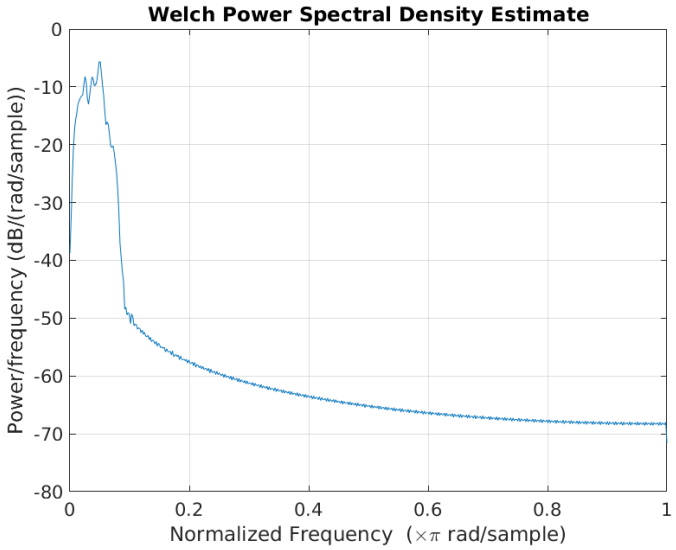
Luis Fernando Rodriguez Gtz ie705694

Sistemas de Comunicaciones Digitales
02/24/20

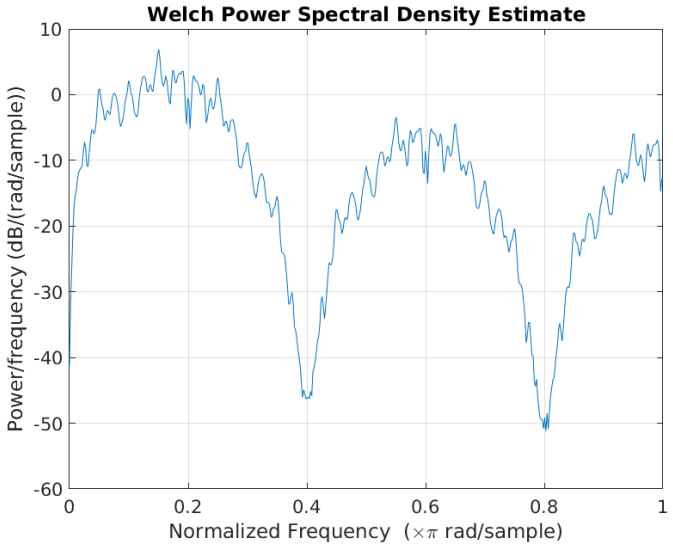
Specter Manchester

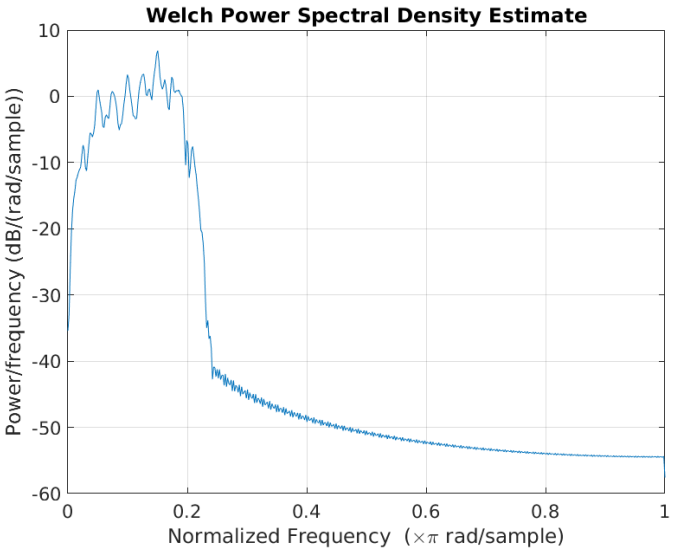
Fc = 0.05





Fc = 0.2

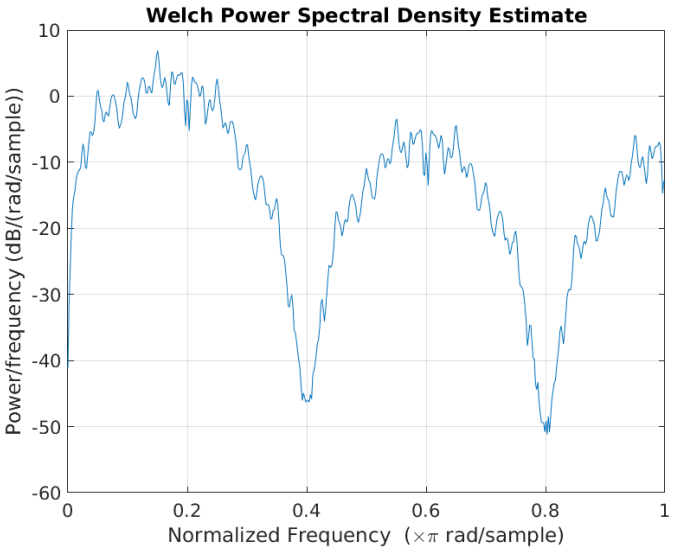


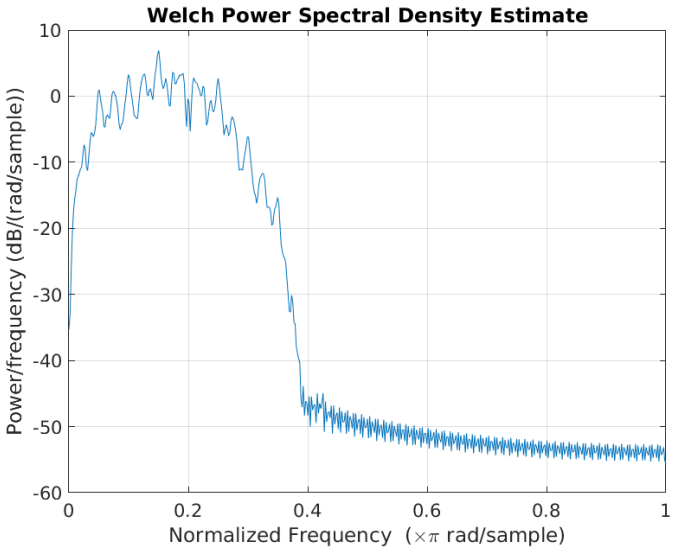


Luis Fernando Rodriguez Gtz ie705694

Sistemas de Comunicaciones Digitales
02/24/20

Fc = 0.4





Uni-polar Code Matlab

```
%% Low pass filter

    %Frequency band edges
f = [0, fc_4, fc_4, 1];
    %Amplitudes
m = [1, 1, 0, 0];
    %Filter order
n = 100;
fir = fir2(n,f,m);

%fir_temp = firls(n,f,m);
%fvtool(fir,1,fir_temp,1);
%phasez(fir);

    %Better way of getting frequency response.
freqz(fir,1);   title('Filter frequency response Fc = 0.4');

%% Lena
load lena512.mat;

figure;
imshow(uint8(lena512));
title('Original Img');

lenarec=lena512(252:284,318:350);

figure;
imshow(uint8(lenarec));
title('Cutted Img');

b = de2bi(lenarec,8); %For Default it is the 'right-msb'
b = b'; %Transponse operation
bits_rx = b(:); %Concatena el resto de bits, para que sea un solo vector

lena_size = size(lenarec);
y = bits_rx(1:256);

mp = 10;
t = 0:0.1:1;
    %Use numel because it is less heavy operation
    %We follow the examples made in class
x = zeros(1,numel(t));
x( find( (t>=0.4) & (t<=0.7) ) ) = 1;
s1 = y;
s1(s1 == 0) = 0;

s = zeros(1,numel(y)*mp);
s(1:mp:end) = s1;
x = conv(s,x);
figure;
pwelch(x);

    %Filer
%Low Pass filter fc = 0.4
LPF = conv(x,fir);
pwelch(LPF);
```

Polar Code Matlab

```
%% Low pass filter

    %Frequency band edges
f = [0, fc_4, fc_4, 1];
    %Amplitudes
m = [1, 1, 0, 0];
    %Filter order
n = 100;
fir = fir2(n,f,m);

%fir_temp = firls(n,f,m);
%fvtool(fir,1,fir_temp,1);
%phasez(fir);

    %Better way of getting frequency response.
freqz(fir,1);   title('Filter frequency response Fc = 0.4');

%% Lena
load lena512.mat;

figure;
imshow(uint8(lena512));
title('Original Img');

lenarec=lena512(252:284,318:350);

figure;
imshow(uint8(lenarec));
title('Cutted Img');

b = de2bi(lenarec,8); %For Default it is the 'right-msb'
b = b'; %Transponse operation
bits_rx = b(:); %Concatena el resto de bits, para que sea un solo vector

lena_size = size(lenarec);
y = bits_rx(1:256);

mp = 10;
t = 0:0.1:1;
    %Use numel because it is less heavy operation
    %We follow the examples made in class
x = zeros(1,numel(t));
x(find((t>0)&(t<=0.5)))=1;
x(find((t>0.5)&(t<1)))=0;
s1 = y;
s1(s1 == 0) = -1;

s = zeros(1,numel(y)*mp);
s(1:mp:end) = s1;
x = conv(s,x);
figure;
pwelch(x);

    %Filer
%Low Pass filter fc = 0.4
LPF = conv(x,fir);
pwelch(LPF);
```

Luis Fernando Rodriguez Gtz ie705694

Sistemas de Comunicaciones Digitales
02/24/20

Bipolar Code Matlab

```
%% Low pass filter

    %Frequency band edges
f = [0, fc_4, fc_4, 1];
    %Amplitudes
m = [1, 1, 0, 0];
    %Filter order
n = 100;
fir = fir2(n,f,m);

%fir_temp = firls(n,f,m);
%fvtool(fir,1,fir_temp,1);
%phasez(fir);

    %Better way of getting frequency response.
freqz(fir,1);   title('Filter frequency response Fc = 0.4');

%% Lena
load lena512.mat;

figure;
imshow(uint8(lena512));
title('Original Img');

lenarec=lena512(252:284,318:350);

figure;
imshow(uint8(lenarec));
title('Cutted Img');

b = de2bi(lenarec,8); %For Default it is the 'right-msb'
b = b'; %Transponse operation
bits_rx = b(:); %Concatena el resto de bits, para que sea un solo vector

lena_size = size(lenarec);
y = bits_rx(1:256);

mp = 10;
t = 0:0.1:1;
    %Use numel because it is less heavy operation
    %We follow the examples made in class
x = zeros(1,numel(t));
x( find( (t>=0.1) & (t<=1.0) ) ) = 1;
s1 = y;
flag = 1;
for i = 1:numel(s1)
    if(s1(i) == 1)
        s1(i) = flag;
        flag = flag*-1;
    end
end
s = zeros(1,numel(y)*mp);
s(1:mp:end) = s1;
x = conv(s,x);
figure;
pwelch(x);
```

Manchester Code Matlab

```
%% Low pass filter
    %Frequency band edges
f = [0, fc_4, fc_4, 1];
    %Amplitudes
m = [1, 1, 0, 0];
    %Filter order
n = 100;
fir = fir2(n,f,m);

%fir_temp = firls(n,f,m);
%fvtool(fir,1,fir_temp,1);
%phasez(fir);

    %Better way of getting frequency response.
freqz(fir,1);   title('Filter frequency response Fc = 0.4');

%% Lena
load lena512.mat;

figure;
imshow(uint8(lena512));
title('Original Img');

lenarec=lena512(252:284,318:350);

figure;
imshow(uint8(lenarec));
title('Cutted Img');

b = de2bi(lenarec,8); %For Default it is the 'right-msb'
b = b'; %Transponse operation
bits_rx = b(:); %Concatena el resto de bits, para que sea un solo vector

lena_size = size(lenarec);
y = bits_rx(1:256);

mp = 10;
t = 0:0.1:1;
    %Use numel because it is less heavy operation
    %We follow the examples made in class
x = zeros(1,numel(t));
x(find((t>=0)&(t<0.5)))=1;
x(find((t>=0.5)&(t<1)))=-1;
s1 = y;

s1(s1==0) = -1;

s = zeros(1,numel(y)*mp);
s(1:mp:end) = s1;
x = conv(s,x);
figure;
pwelch(x);

    %Filer
%Low Pass filter fc = 0.4
LPF = conv(x,fir);
pwelch(LPF);
```

Luis Fernando Rodriguez Gtz ie705694

Sistemas de Comunicaciones Digitales
02/24/20

EXERCISE 2

- Un monitor LCD de resolución 1920x1200 cuantifica el valor de cada pixel con 24 bits, y refresca la imagen a 60 veces por segundo. ¿Cuál es la tasa de bit en el cable entre el monitor y la computadora?

  f = 60 Hz.
  Res * bits = 1920x1200x24 = 55,296 k bits.

  Rb = 60*Res*bits = 3, 317, 760 bits/s.

  Rb = 3.317 G bits/s

- ¿Cuál es la tasa de bit si consideramos un monitor del tipo 4K?
  Considerando las mismas operaciones anteriores pero ahora con resolucion 4k

  4k equivale a 3840x2160
  Rb = 60*Res*bits = 11.94 G bits/s
-

  12 Mega pixels
  4k = 3840x2160
  60 fps

  Rb = 24*3840*2160*60 = 11.94 G bits/s

- Cual es la tasa de bits que soporta USB 3.1 / HDMI / PCIe 4.0
  - USB 3.1
    - HBR2-. Tasa de Bits Alta 2 = 5.4 Gbps
    - HBR-. Tasa de Bits Alta = 2.7 Gbps
    - RBR-. Tasa de Bits Reducida = 1.62 Gbps
  - HDMI
    - 4.9 Gbps
  - PCIe 4.0
    - 16 GT (GigaTransfer per Second)
      - 16 * (1 – (2/130)) / 8 = 1.969 GB / s por línea (31.51GB/s x16).

Sources Used:
https://www.mathworks.com/help/signal/ug/fir-filter-design.html
https://www.mathworks.com/help/signal/ref/phasez.html
https://www.mathworks.com/help/signal/ref/freqz.html