Aula 08 – filtro I - FIR

Prof. Dr. Thiago Martini Pereira
Processamentos de sinais

Calendário final

Data	Atividade
08/11	Aula Filtro FIR
15/11	Feriado
22/11	Projeto sobre filtro
29/11	Aula de IRR e projeto sobre filtro
06/12	Apresentação dos projetos

frequência normalizada

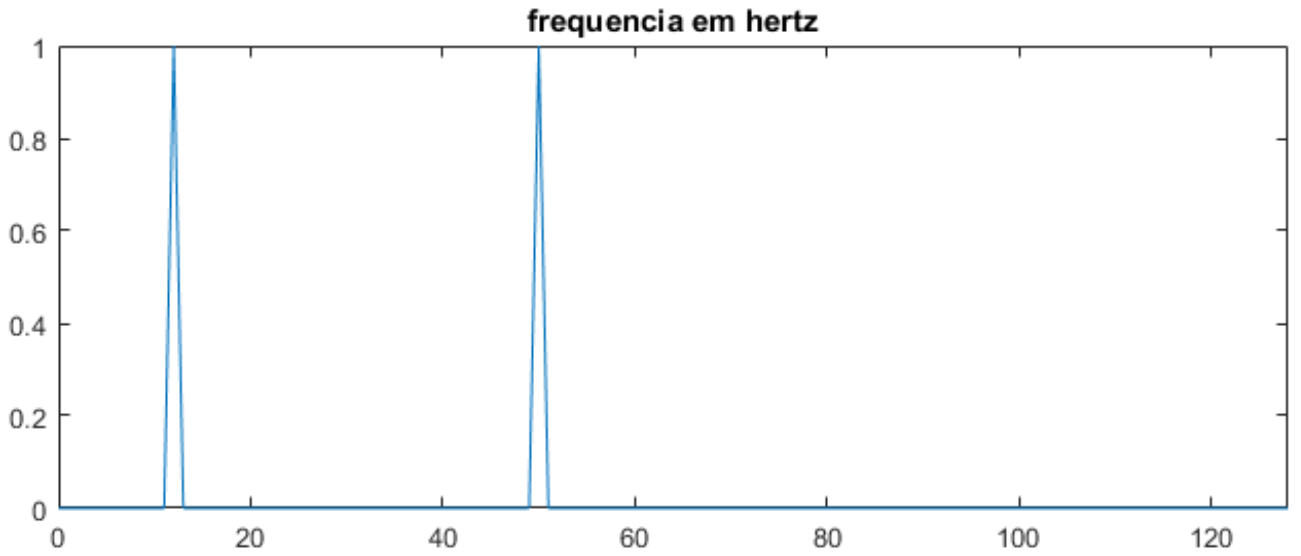
$$\omega = \frac{2\pi f}{f_s}$$

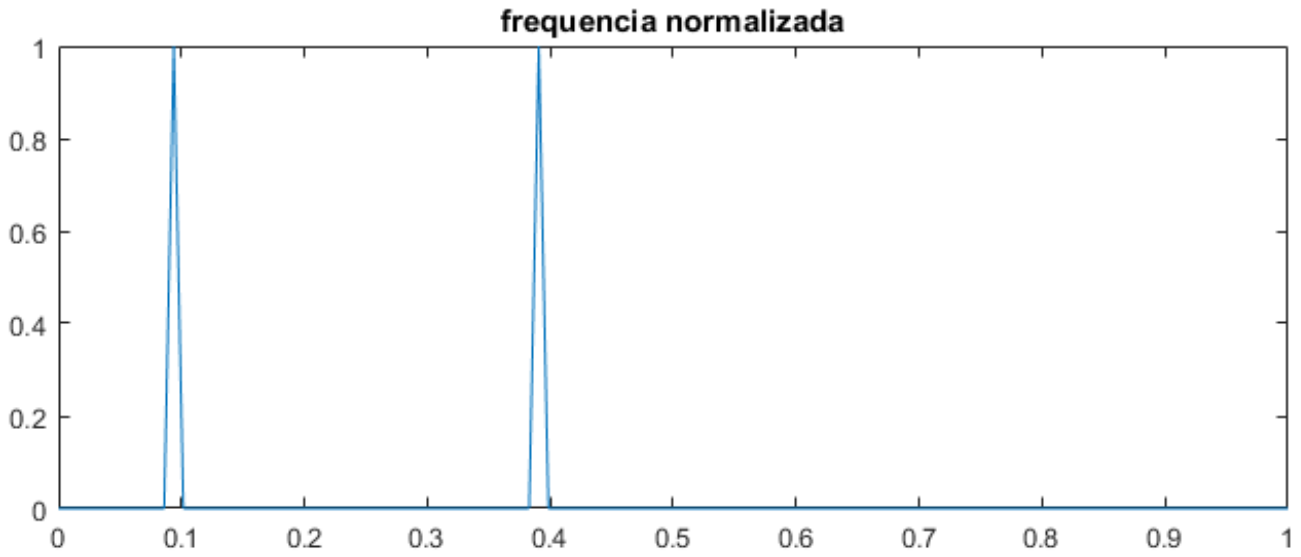
```
% gerando o sinal
fs =256;f = [12 50];
t = 0:1/fs:1-1/fs
y = sin(2*pi*t*f(1)) +
sin(2*pi*t*f(2));

%fazendo a fft

x = fft(y);
n = numel(x);
freq = (0:n-1).*fs/n;
subplot(2,1,1);plot(freq,2*abs(x)./n);
xlim([0 fs/2])
title('frequencia em hertz')

freqnorm = linspace(0,2,n)
subplot(2,1,2);
plot(freqnorm,2*abs(x)./n);
xlim([0 1])
title('frequencia normalizada')
```





Filtros

Funções principais dos filtros:

– separação de sinais
Exemplo: monitorar o sinal de ECG do feto ainda dentro do útero da mãe.

– recuperação de sinais
Exemplo: recuperação de gravações de áudio e melhoramento de imagens borradas.

Filtros de resposta a impulso finita (FIR)

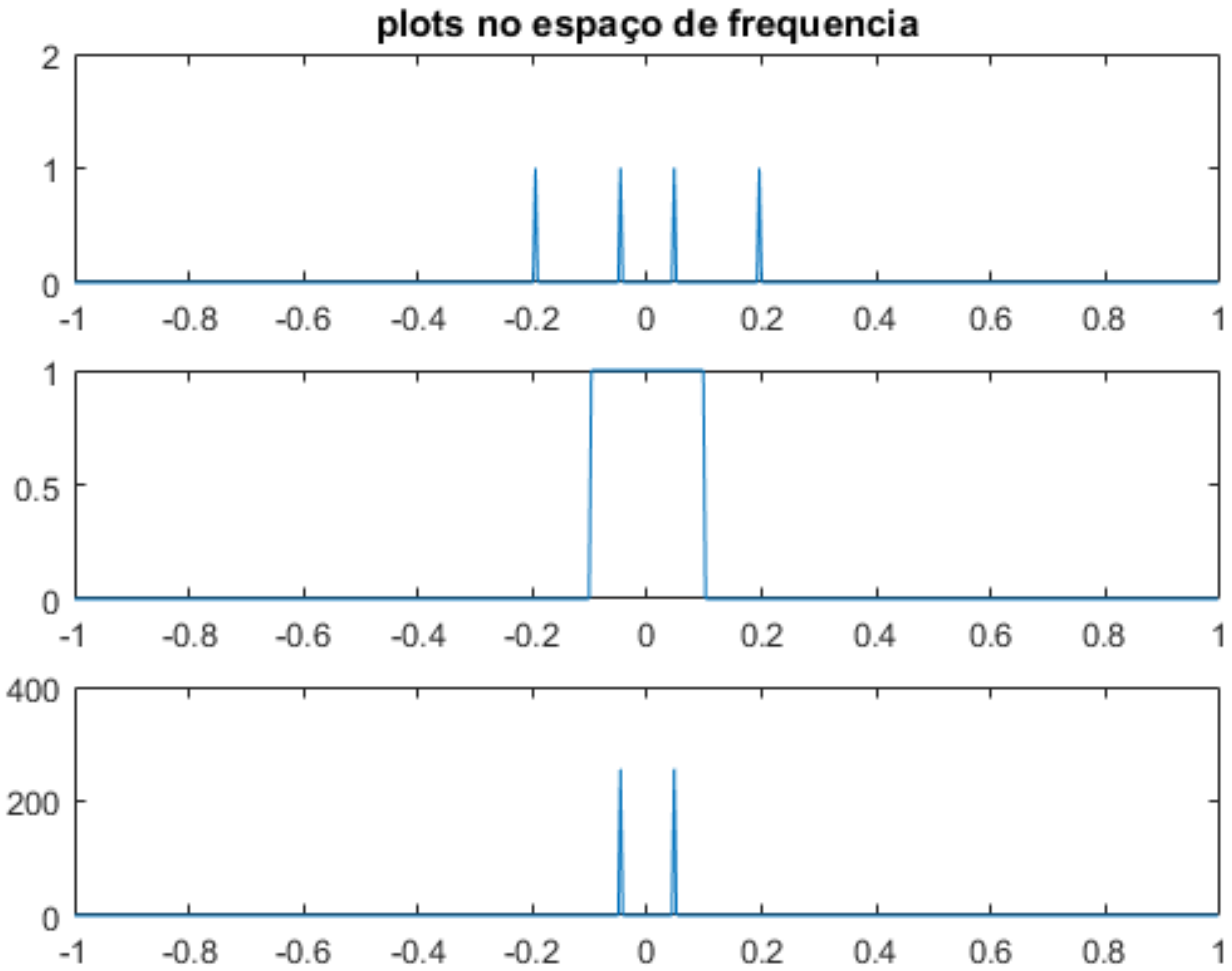
– operam por convolução da resposta a impulso (kernel) com o sinal

– todos os filtros lineares possíveis podem ser implementados desta maneira

– possuem desempenho impressionante, mas podem ser lentos, dependendo do comprimento de seu kernel

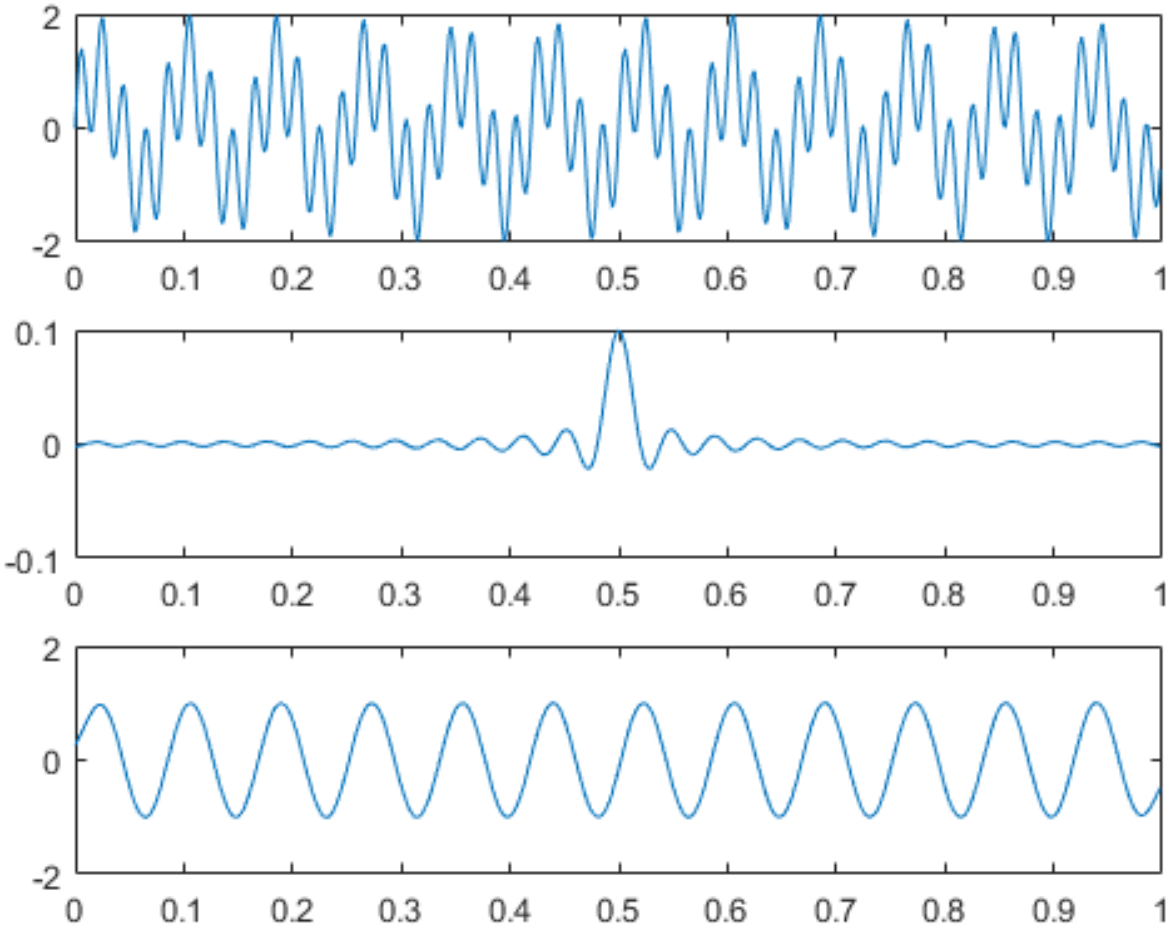
construindo um filtro passa baixa ideal

```
fs =512;f = [12 50];
t = 0:1/fs:1-1/fs
% gerando o sinal
y = sin(2*pi*t*f(1)) +
sin(2*pi*t*f(2));
x = fftshift(fft(y));
n = numel(x);
freqnorm = linspace(-1,1,n);
% criando um fç caixa para filtragem.
box = zeros(size(y));
box(abs(freqnorm) < 0.1) = 1;
% plotando no espaço de frequencia
figure
subplot(3,1,1);plot(freqnorm,2*abs(x).
/n);xlim([-1 1])
title('plots no espaço de frequencia')
subplot(3,1,2);plot(freqnorm,box)
subplot(3,1,3);plot(freqnorm,box.*abs(
x))
```

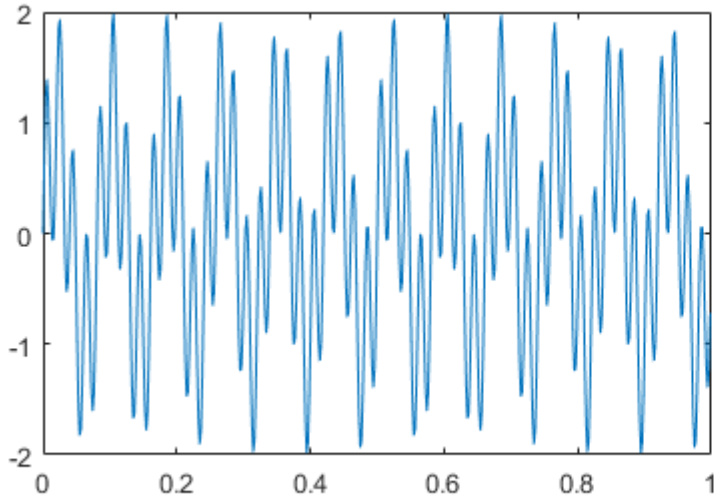


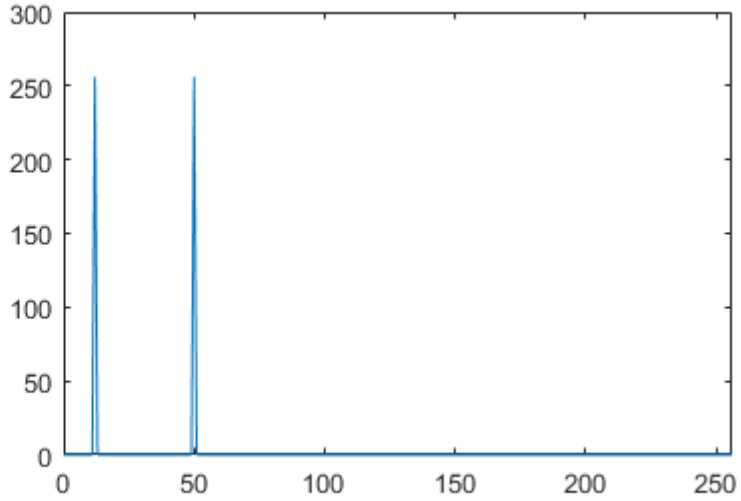
construindo um filtro passa baixa ideal

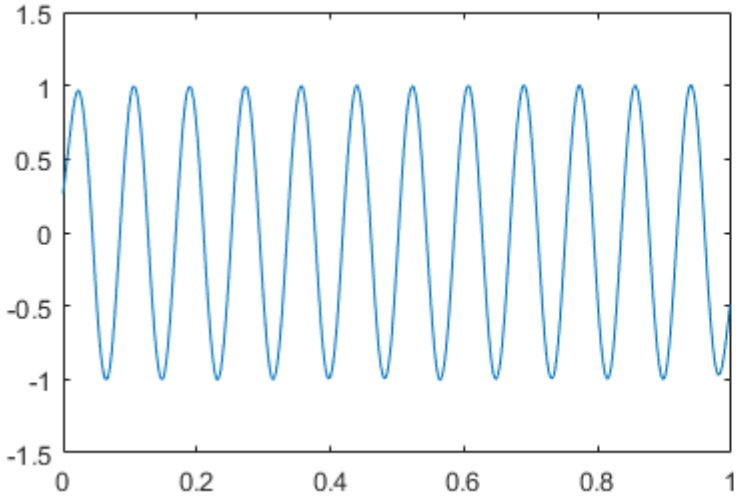
```
% aplicando a ifft no função box
box = ifftshift(box);
ibox = ifft(box);
ibox = fftshift(ibox);
figure
subplot(3,1,1);plot(t,y)
subplot(3,1,2);plot(t,ibox)
filtered = conv(y,ibox);
n = numel(box);
subplot(3,1,3);
plot(t,filtered(n/2:end-n/2))
%plot(t,-1))
```

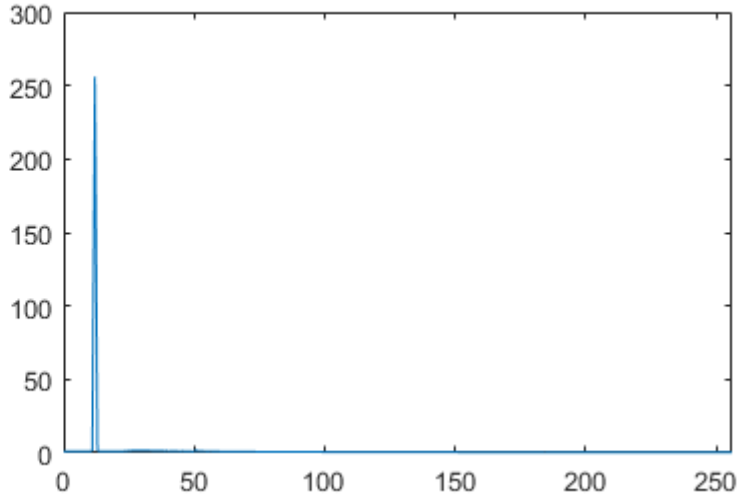


construindo um filtro passa baixa ideal



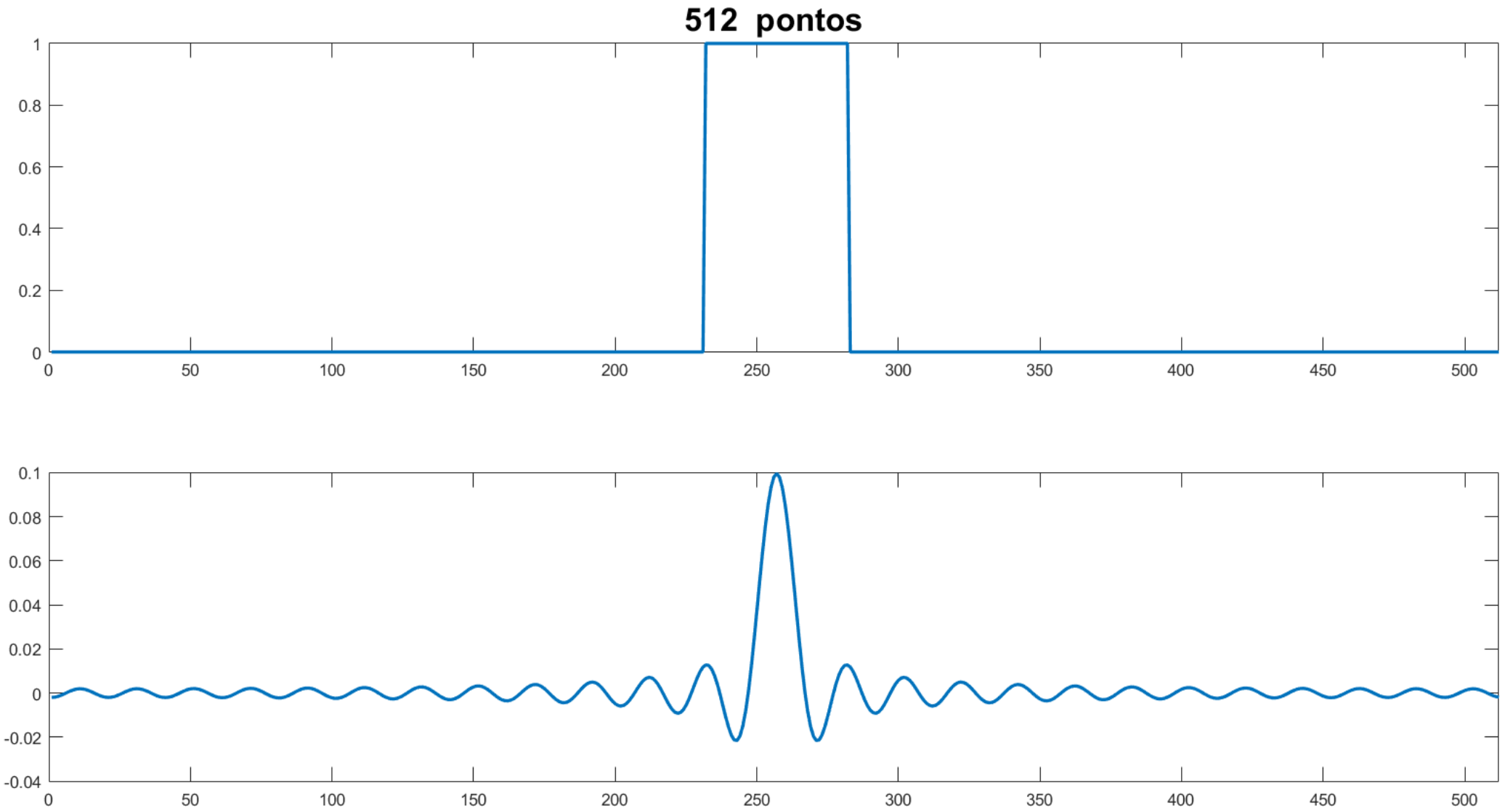




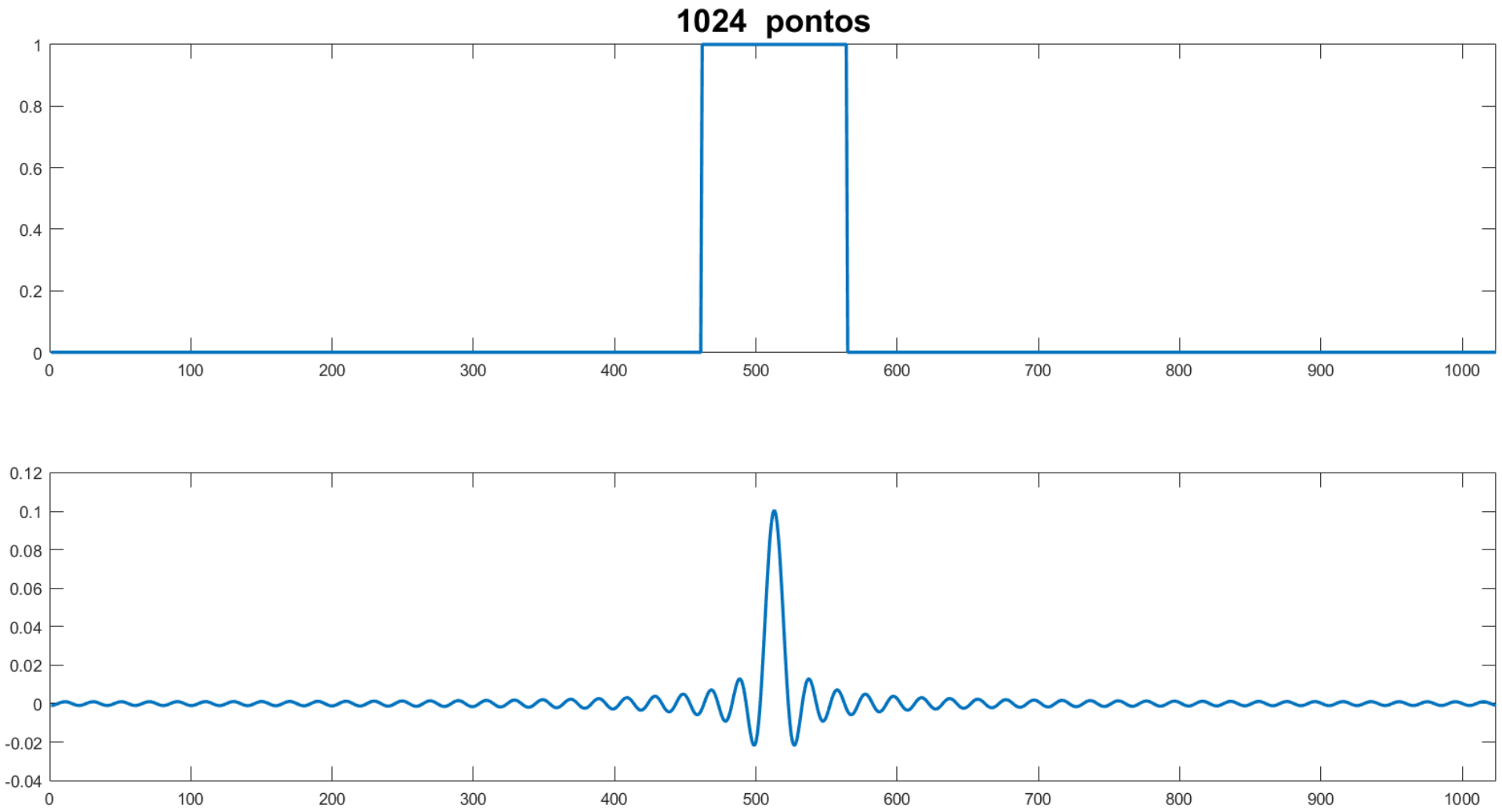


Janelamento em filtros FIR

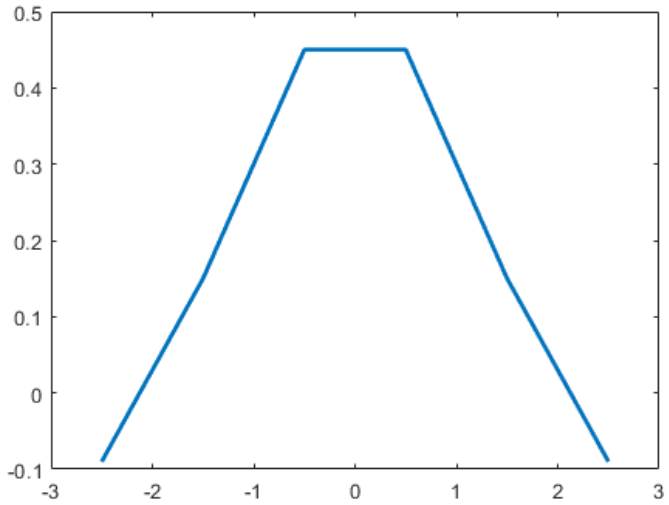
Janelamento

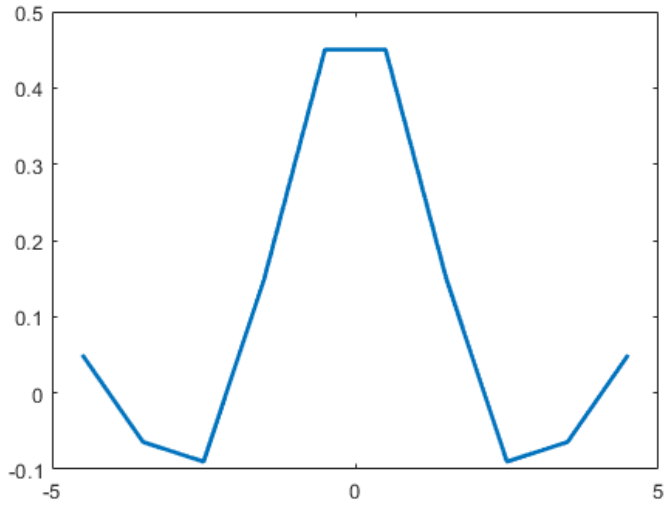


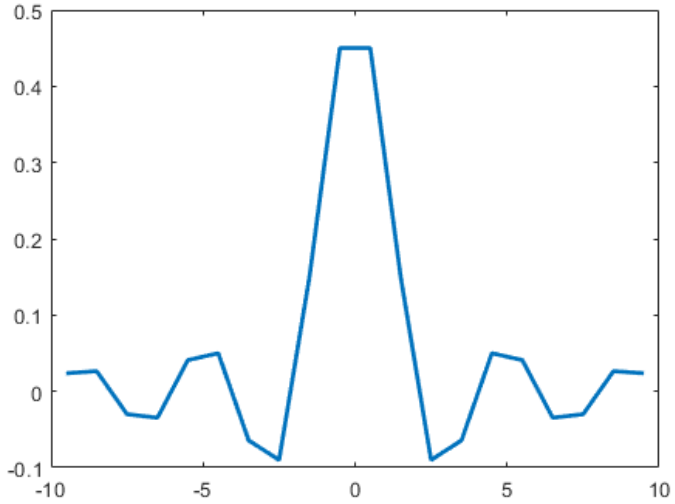
Janelamento

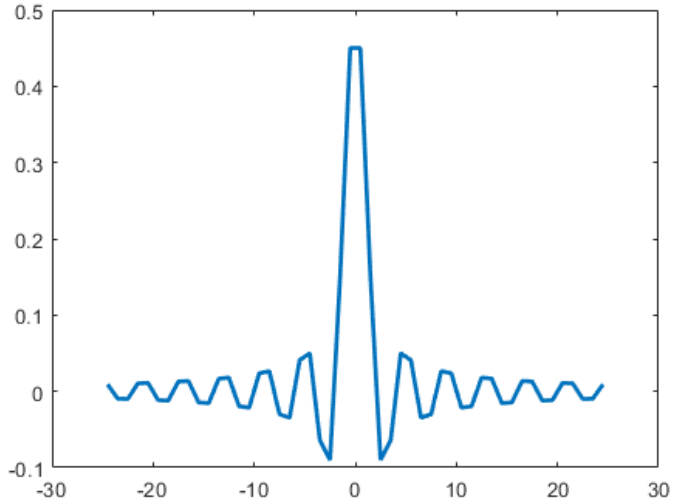


Corte

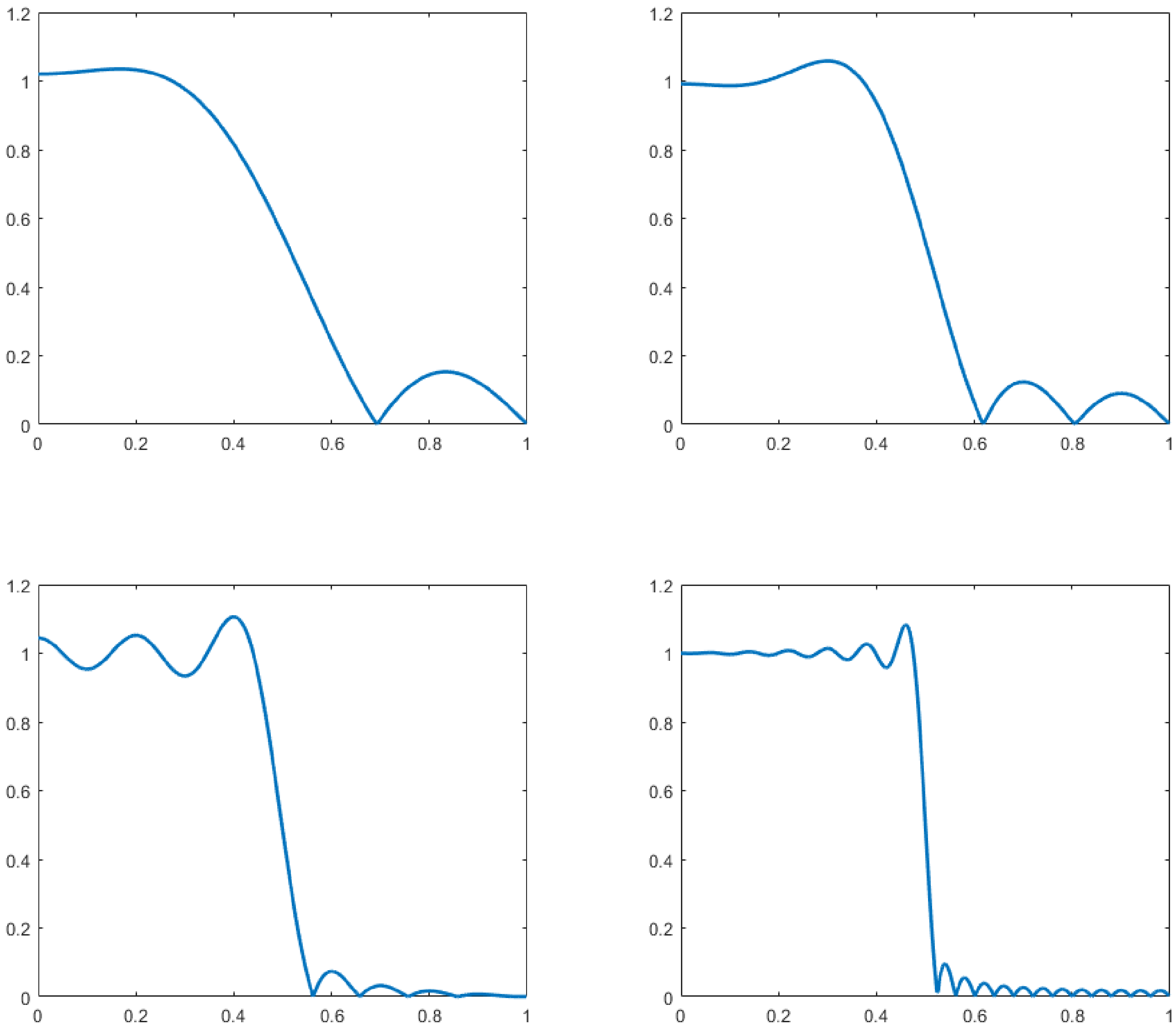

0.5
0.4
0.3
0.2
0.1
0
-0.1
-3
-2
-1
0
1
2
3


0.5
0.4
0.3
0.2
0.1
0
-0.1
-5
0
5

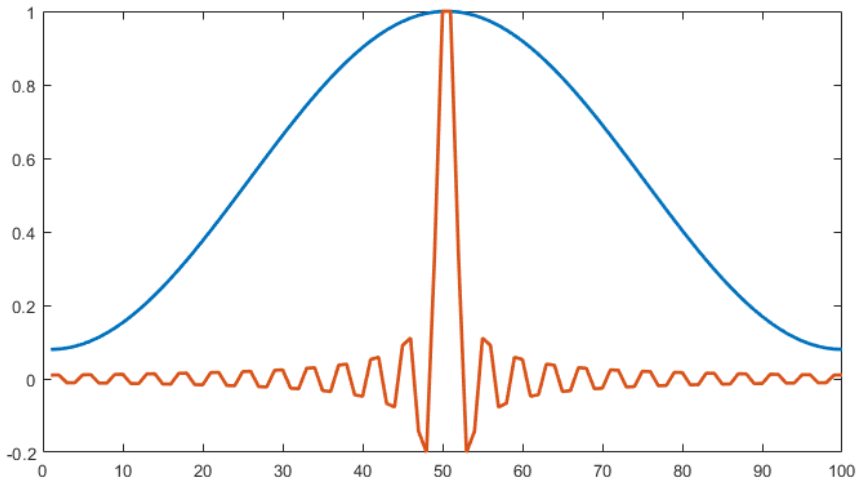

0.5
0.4
0.3
0.2
0.1
0
-0.1
-10
-5
0
5
10

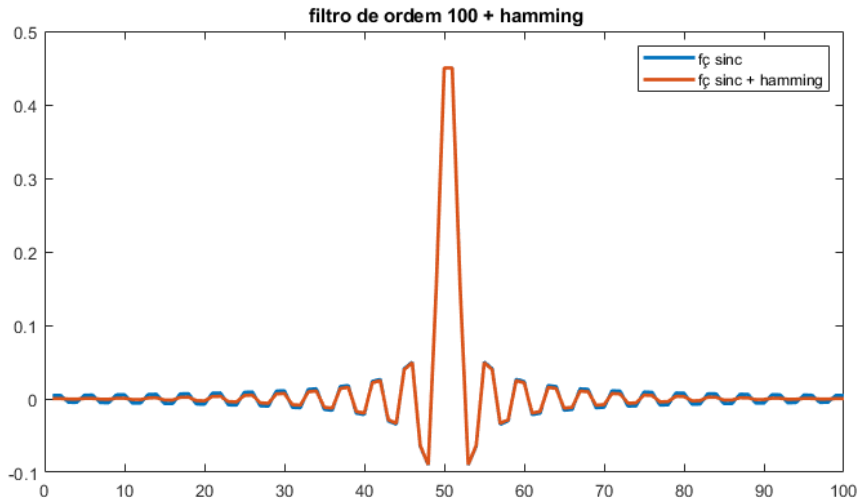

0.5
0.4
0.3
0.2
0.1
0
-0.1
-30
-20
-10
0
10
20
30

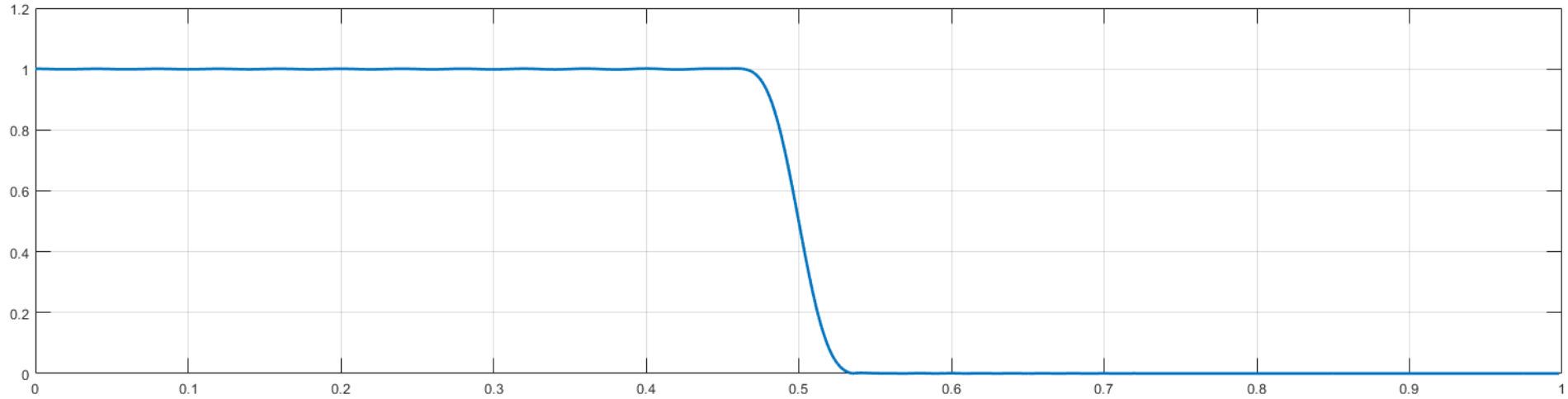
fc = 0.5



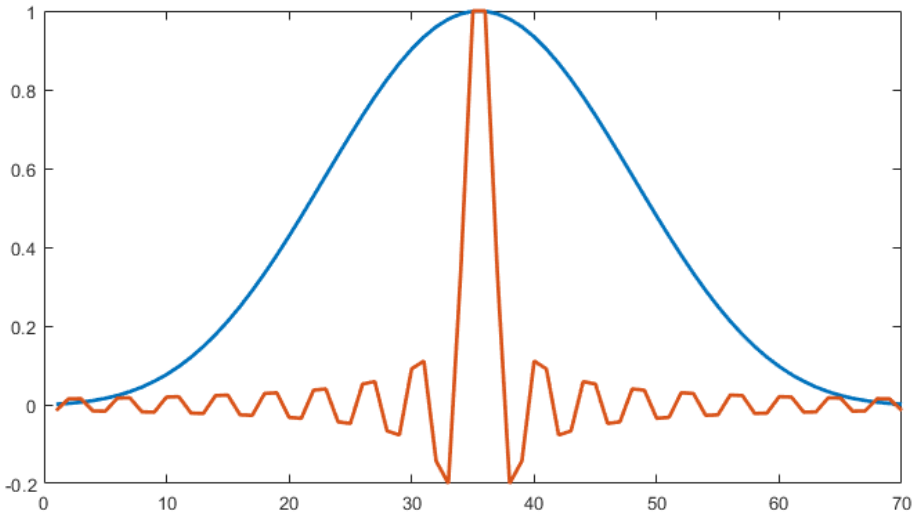
Janelamento – hamming

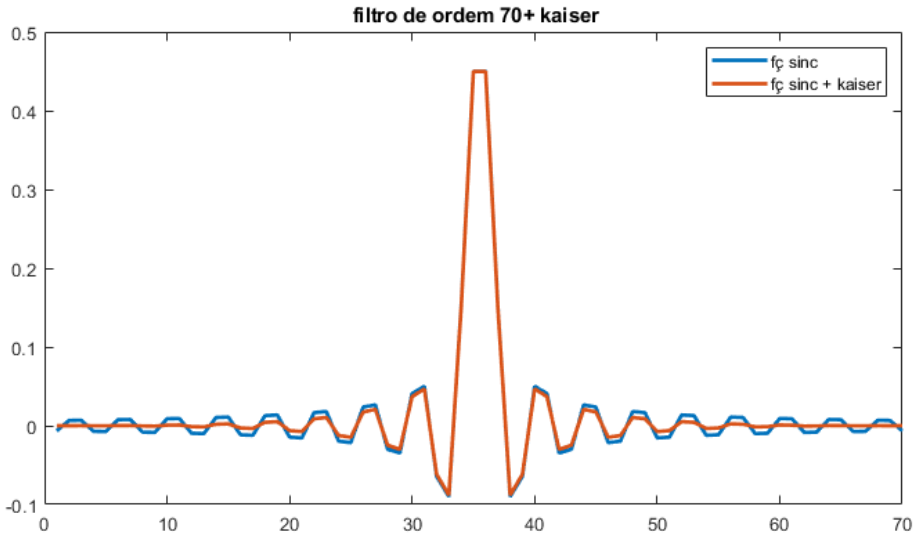


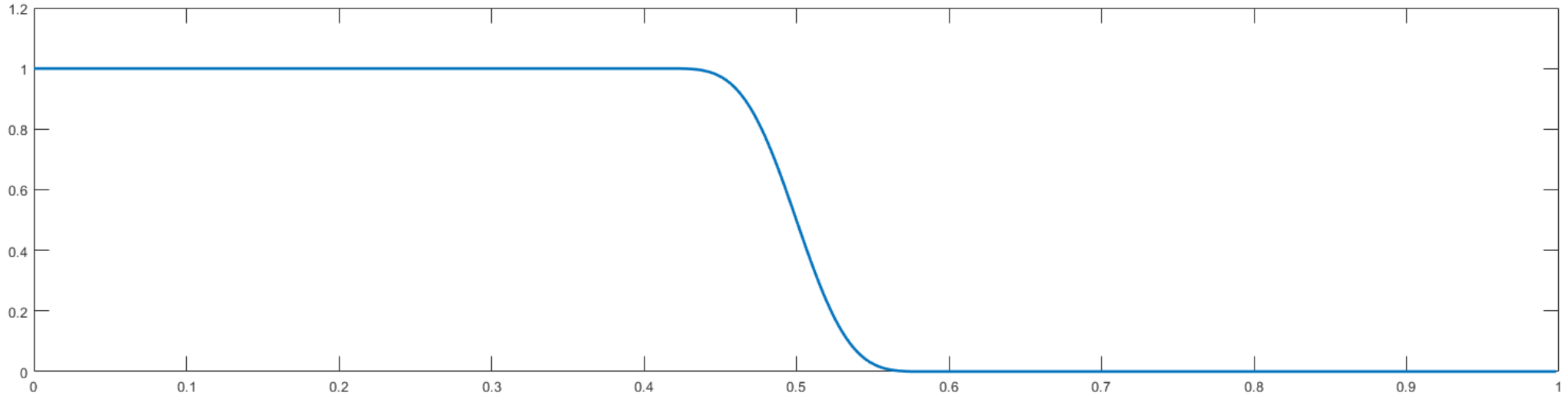




Janelamento – kaiser



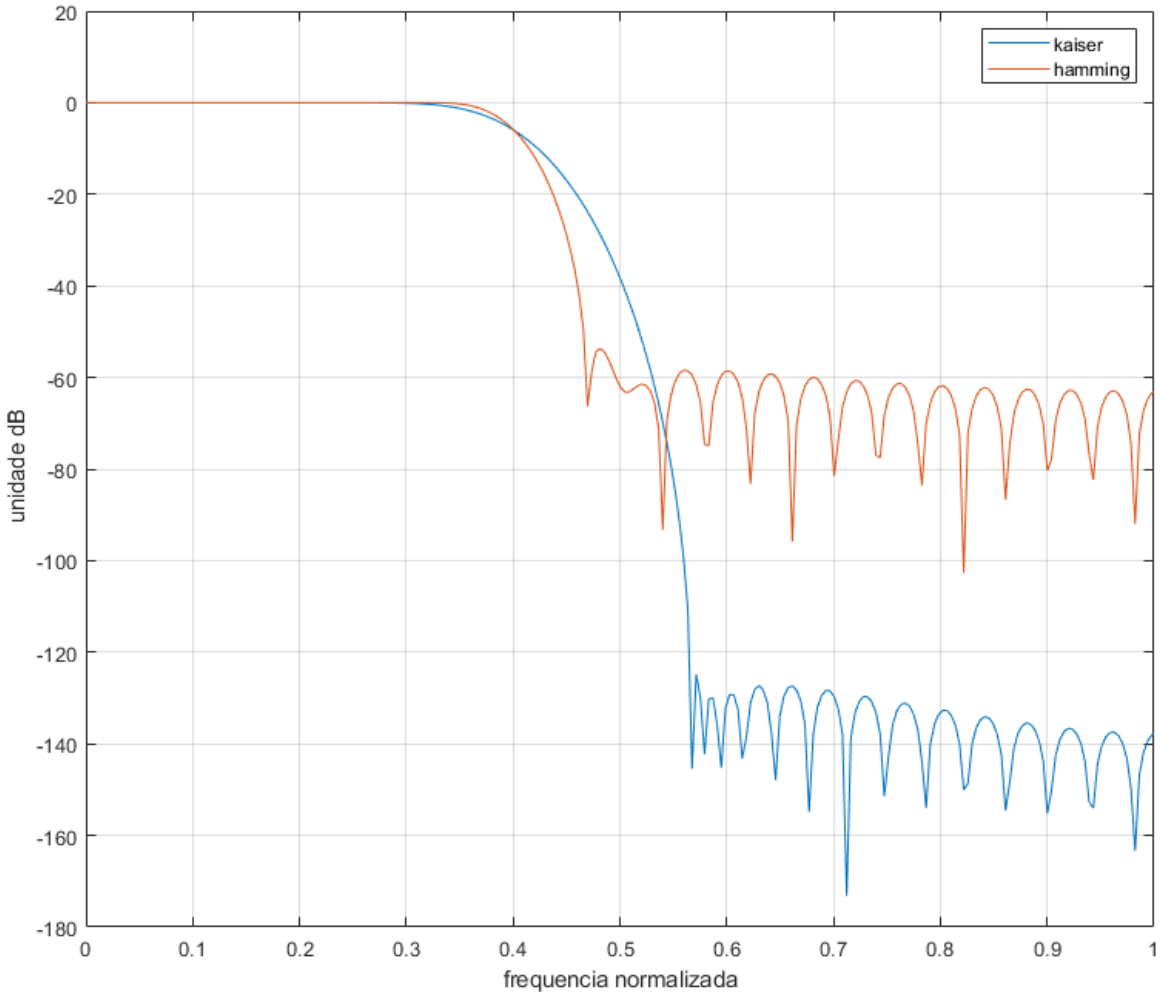




Janelamento – resp em freq. – escala Db

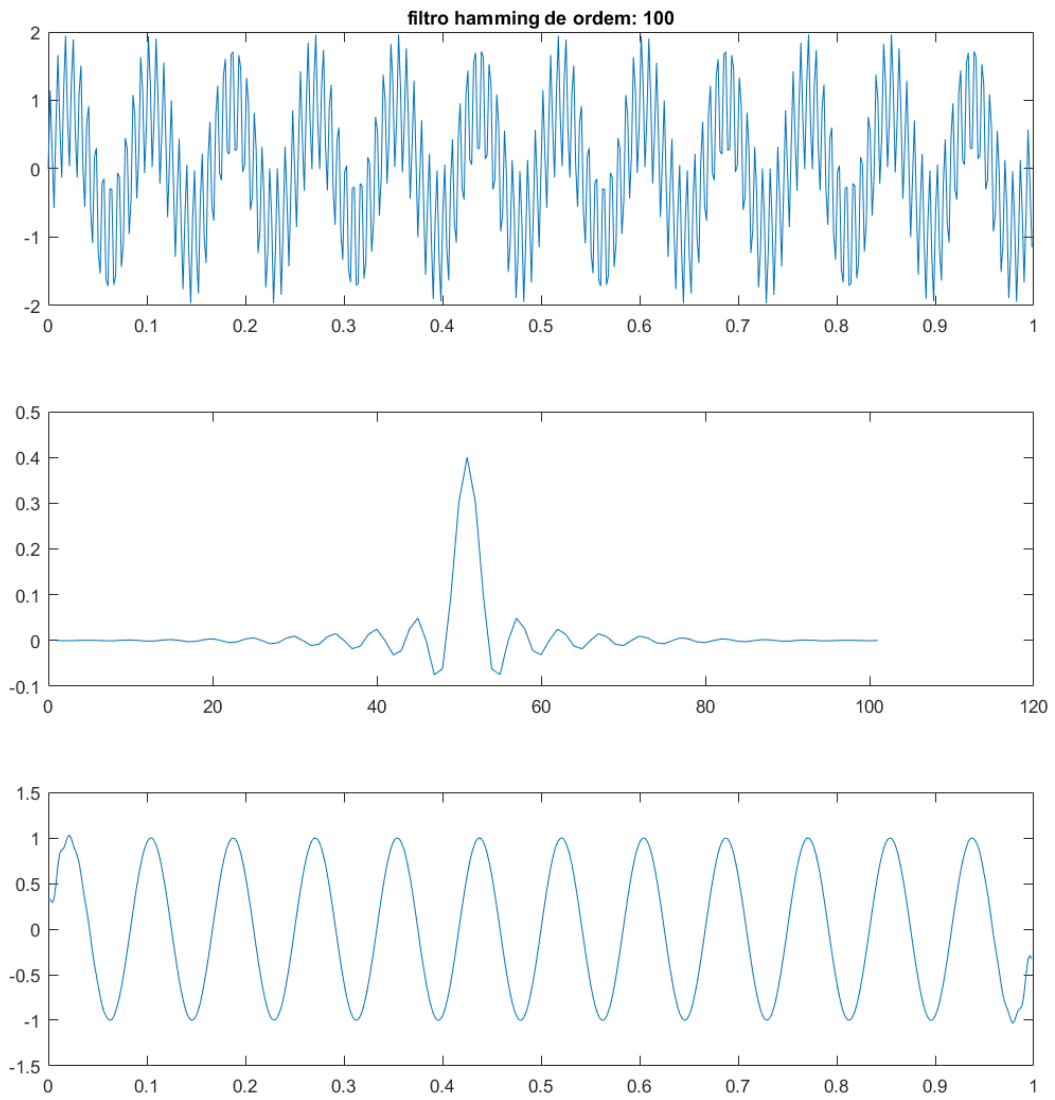
```
%% escala db e função Fvtool
clear
clc
order = 50
win_hamming = hamming(order+1);
h_hamming = fir1(order,0.4,'low',win_hamming);
win_kaiser = kaiser(order+1,13);
h_kaiser = fir1(order,0.4,'low',win_kaiser);
 h_kaiserfft = fft(h_kaiser,512);
h_hammingfft = fft(h_hamming,512);
fn = linspace(0,2,512);
%plot em Db
plot(fn,20*log10(abs(h_kaiserfft)),fn,20*log10(
abs(h_hammingfft)))
grid on; legend(strvcat('kaiser','hamming'))
xlabel('frequencia normalizada')
ylabel(' unidade dB');xlim([0 1]);
fvtool(h_kaiser,1,h_hamming,1)
```

$$Db = 10 * log10(abs(fft(h)))$$



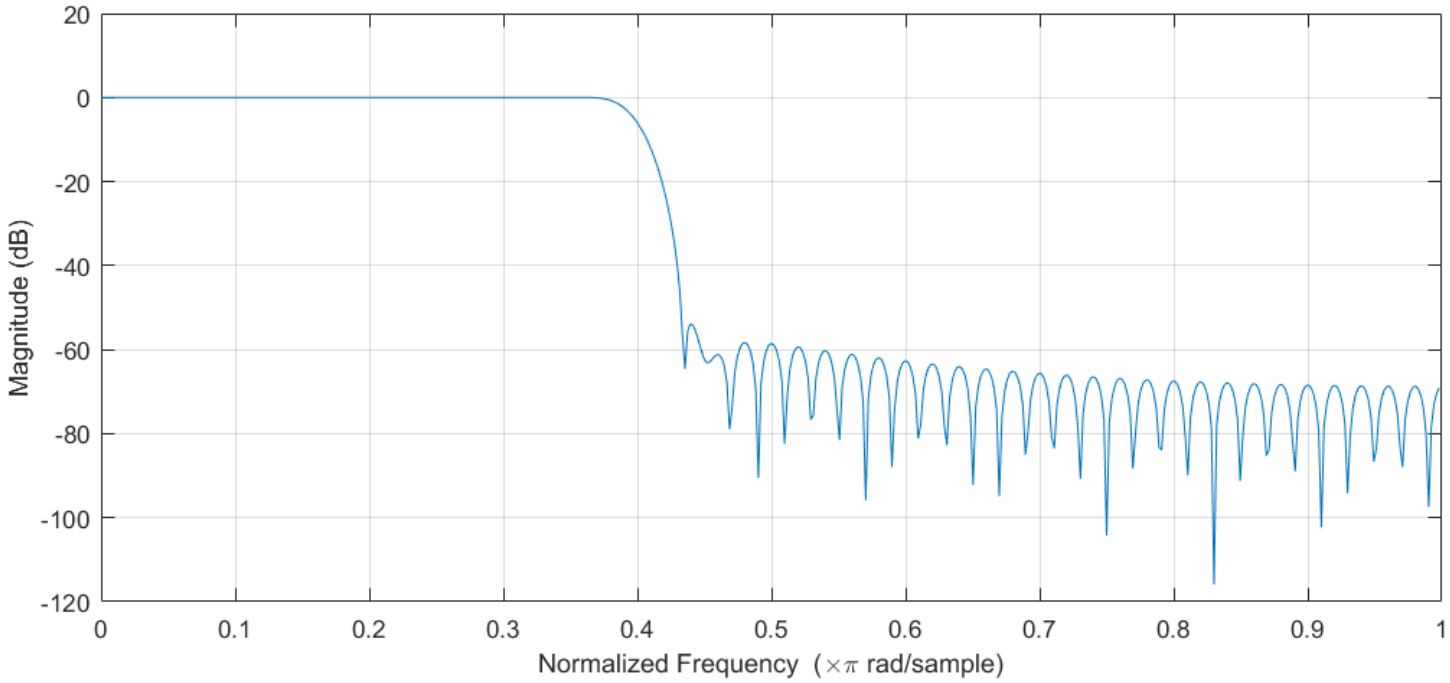
Matlab – função fir1 – passa baixa

```
order = 100
h = fir1(order,[0.4],'low');
freqz(h)
figure
subplot(3,1,1);plot(t,y)
title(['filtro hamming de ordem:
' num2str(order)])
subplot(3,1,2);plot(h)
filtered = conv(y,h,'same');
subplot(3,1,3);plot(t,filtered)
```



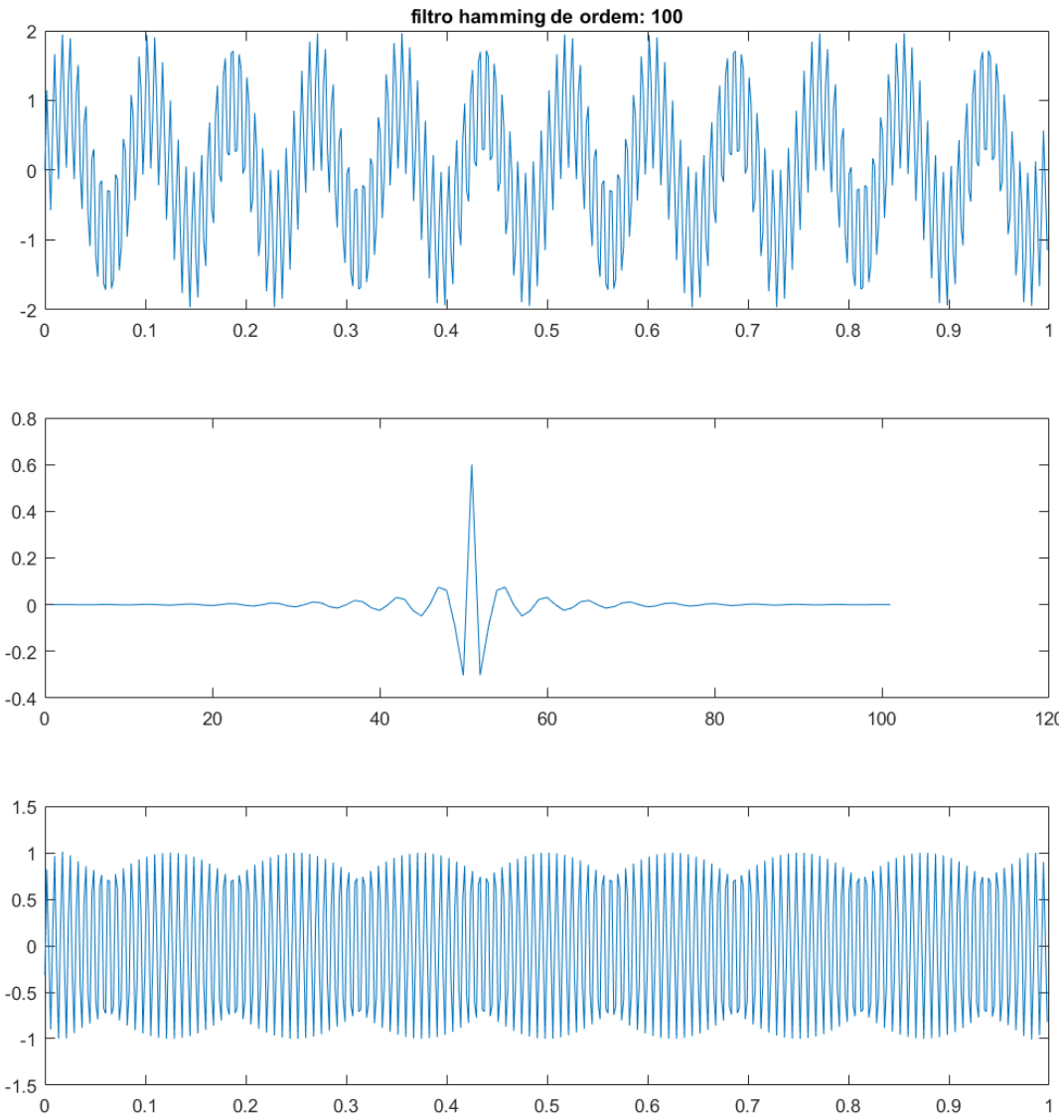
Matlab – função fir1 – passa baixa

```
order = 100
h = fir1(order,[0.4],'low');
freqz(h)
figure
subplot(3,1,1);plot(t,y)
title(['filtro hamming de ordem:
' num2str(order)])
subplot(3,1,2);plot(h)
filtered = conv(y,h,'same');
subplot(3,1,3);plot(t,filtered)
```



Matlab – função fir1 – passa alta

```
figure(2)
order = 100
h = fir1(order,[0.4],'high');
freqz(h)
figure
subplot(3,1,1);plot(t,y)
title(['filtro hamming de ordem:
' num2str(order)])
subplot(3,1,2);plot(h)
filtered = conv(y,h,'same');
subplot(3,1,3);plot(t,filtered)
```



Matlab – função fir1 – passa alta

```
figure(2)
order = 100
h = fir1(order,[0.4],'high');
fvtool(h)
figure
subplot(3,1,1);plot(t,y)
title(['filtro hamming de ordem:
' num2str(order)])
subplot(3,1,2);plot(h)
filtered = conv(y,h,'same');
subplot(3,1,3);plot(t,filtered)
```