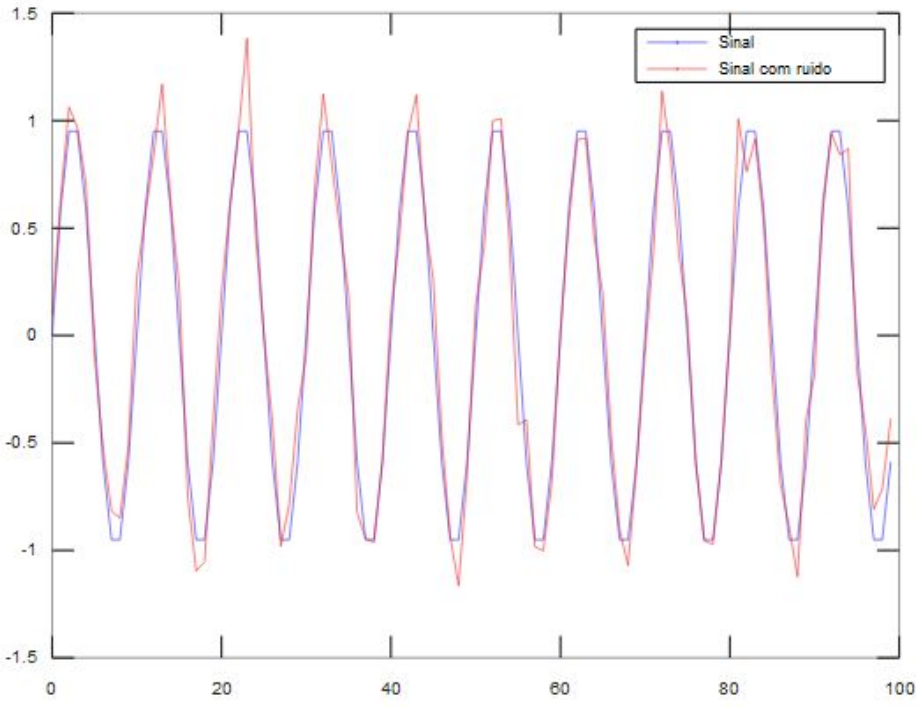
Provinha 3

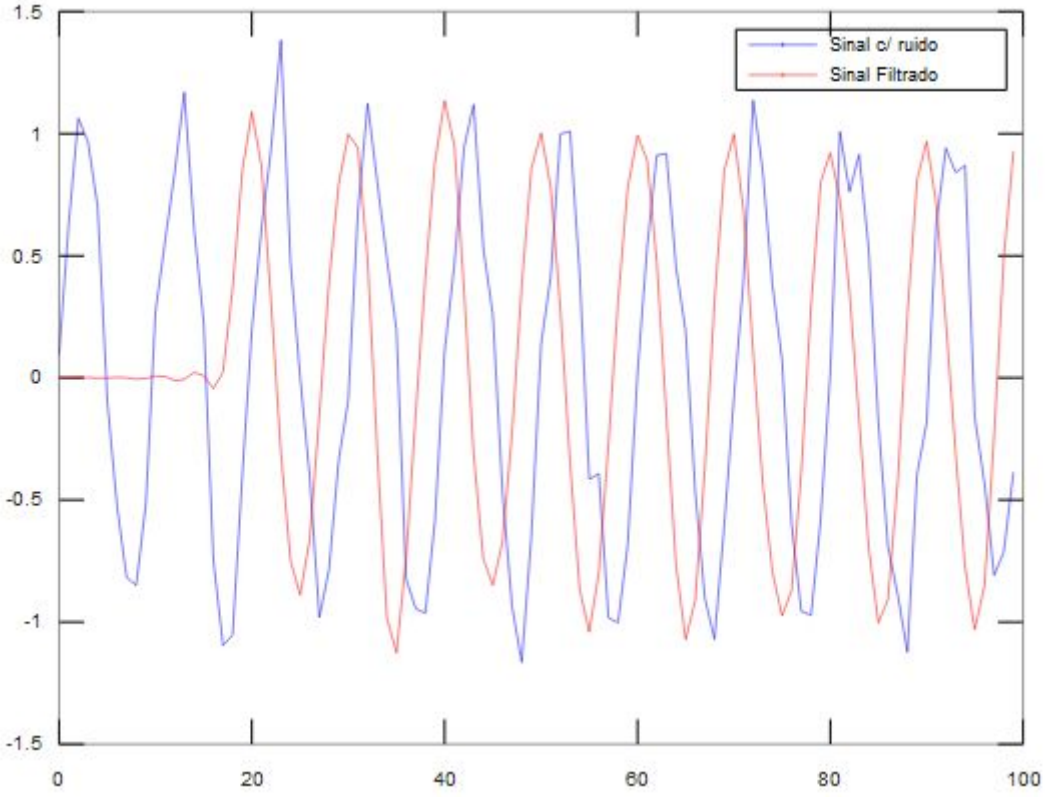
1 a - Utilizaria 12.5 Hz. Já tive contato com sistemas de aquisição embarcados em que a regra do filtro passa baixa era 1/4 da frequência de amostragem.

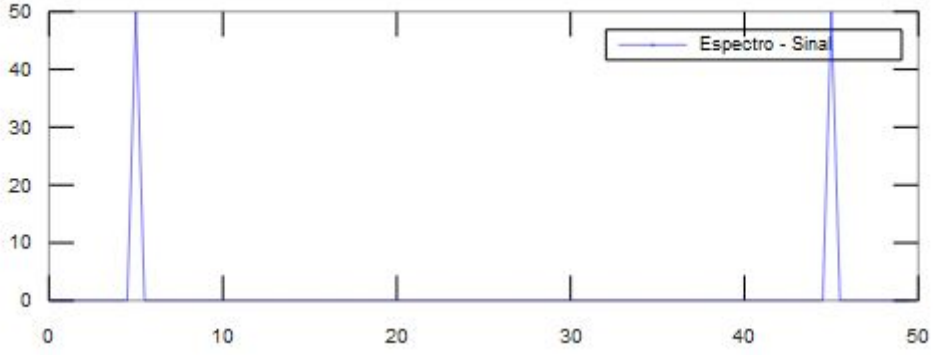
1 b - 28+7 = 35 coeficientes

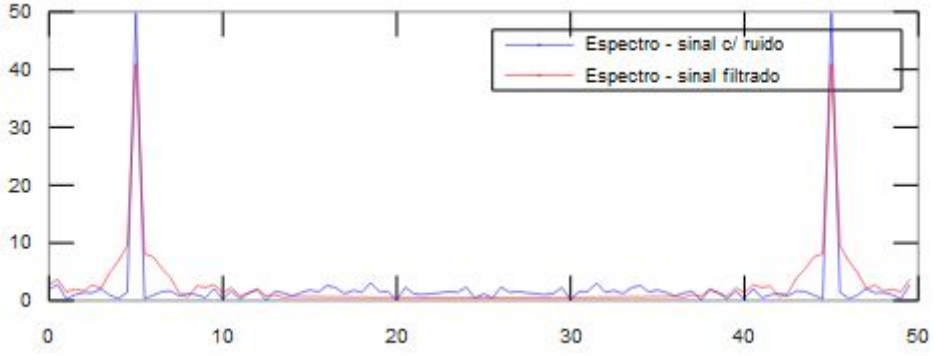
i) coeficientes do filtro fir.

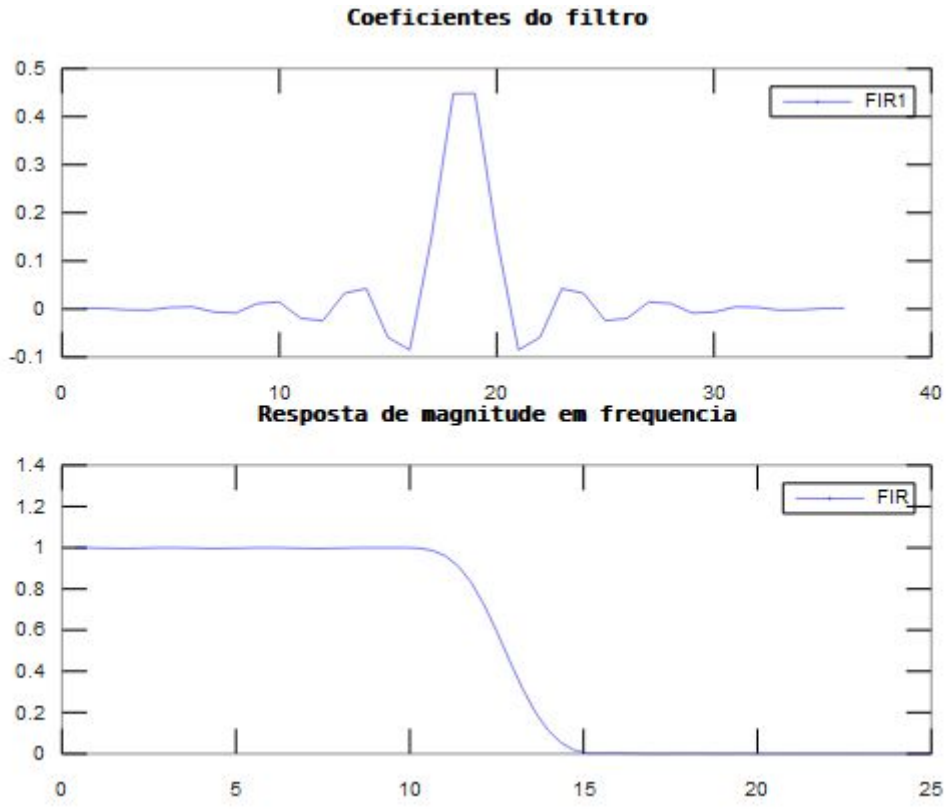
```
b_fir1 =[
   1.0811e-03   1.1286e-03  -1.6588e-03  -2.1468e-03   3.3530e-03   4.3737e-03
  -6.5099e-03  -8.2349e-03   1.1636e-02   1.4427e-02  -1.9680e-02  -2.4411e-02
   3.2951e-02   4.2234e-02  -5.9173e-02  -8.5107e-02   1.4796e-01   4.4778e-01
   4.4778e-01   1.4796e-01  -8.5107e-02  -5.9173e-02   4.2234e-02   3.2951e-02
  -2.4411e-02  -1.9680e-02   1.4427e-02   1.1636e-02  -8.2349e-03  -6.5099e-03
   4.3737e-03   3.3530e-03  -2.1468e-03  -1.6588e-03   1.1286e-03   1.0811e-03 ];
```




Sinal c/ ruido
Sinal Filtrado
1.5
1
0.5
0
-0.5
-1
-1.5
0
20
40
60
80
100

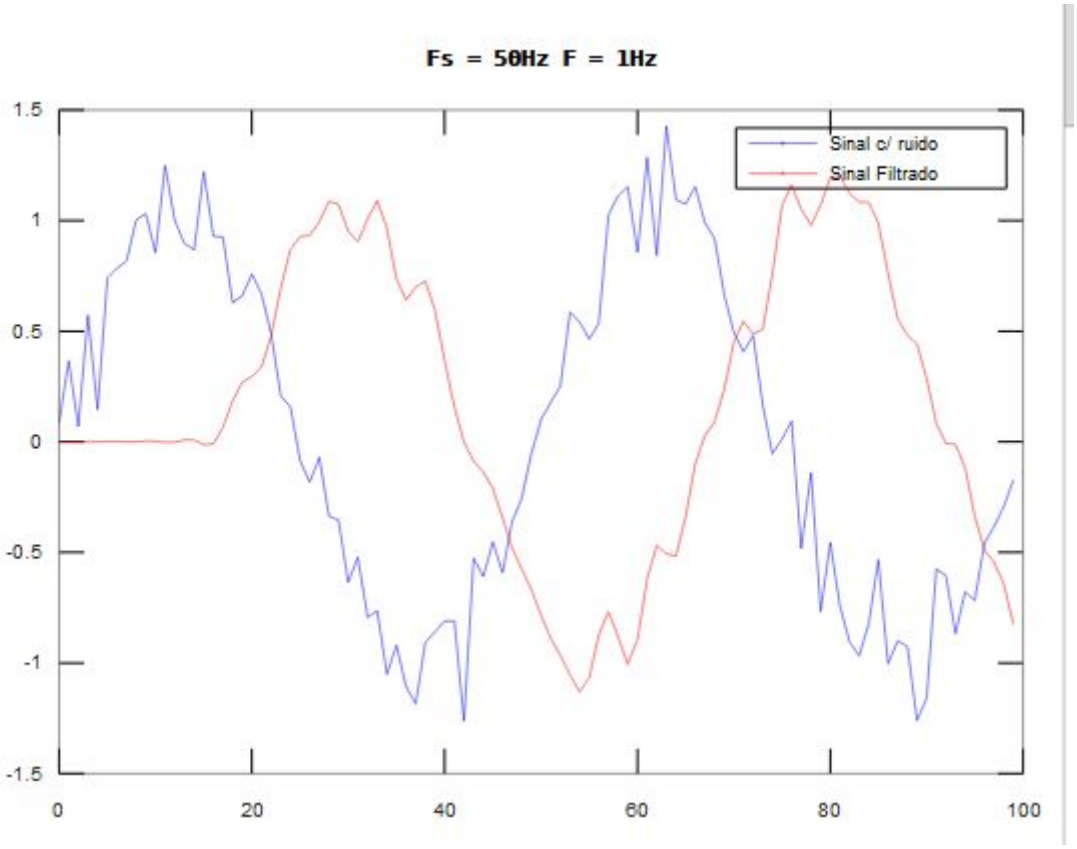

Espectro - Sinal
50
40
30
20
10
0
0
10
20
30
40
50

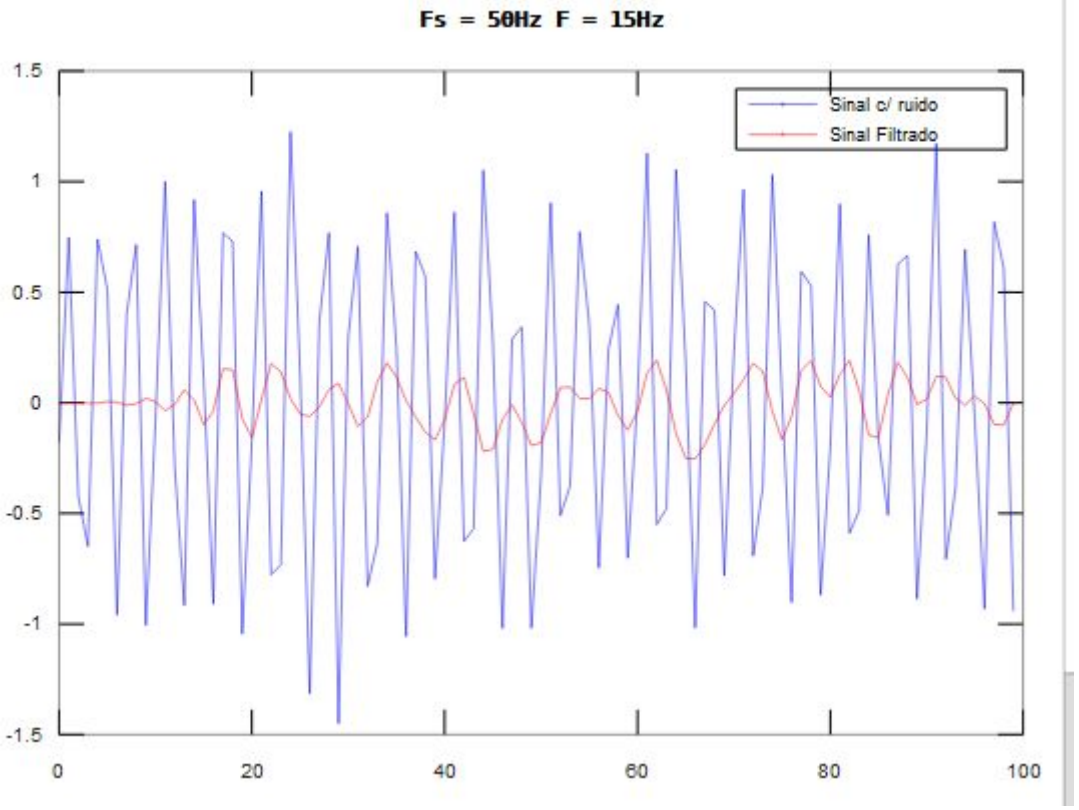

Espectro - sinal c/ ruido
Espectro - sinal filtrado
50
40
30
20
10
0
0
10
20
30
40
50

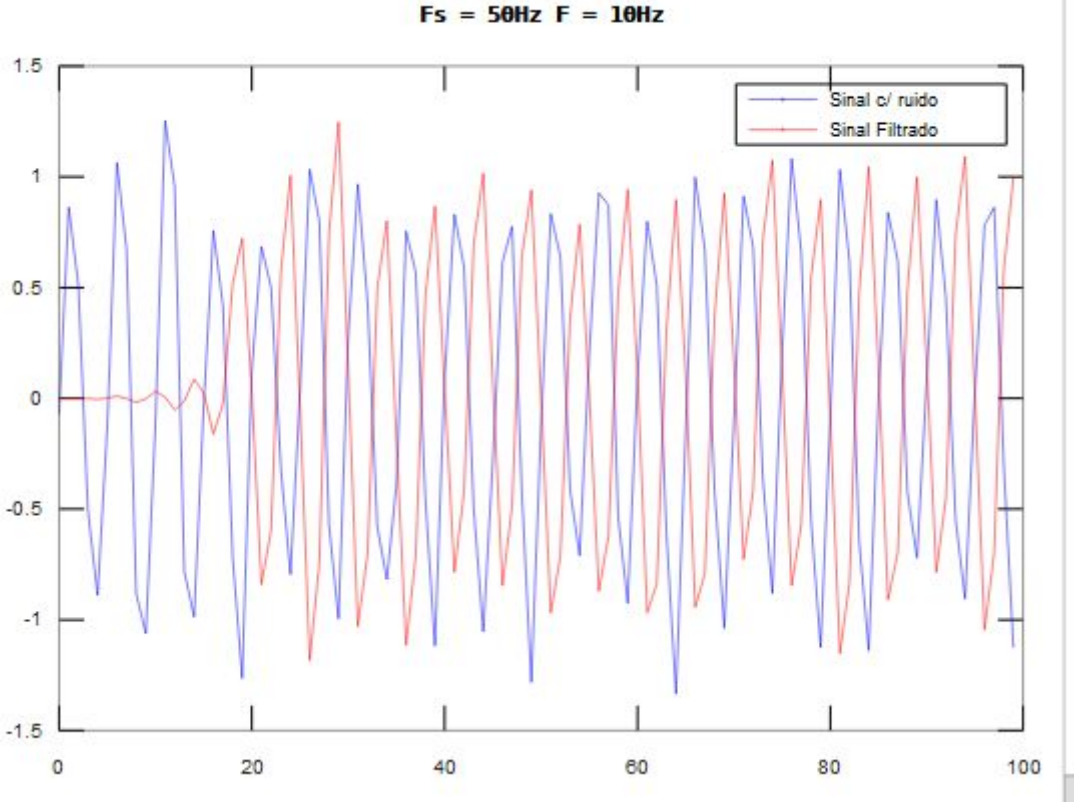


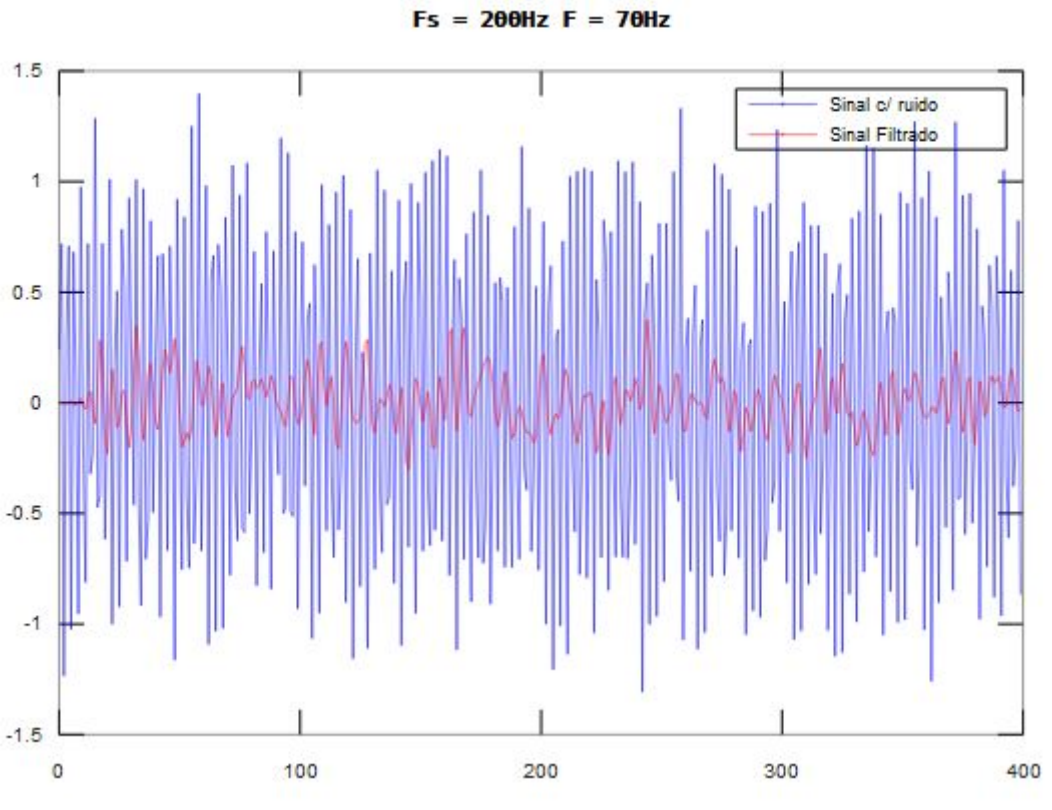
1 b ii)

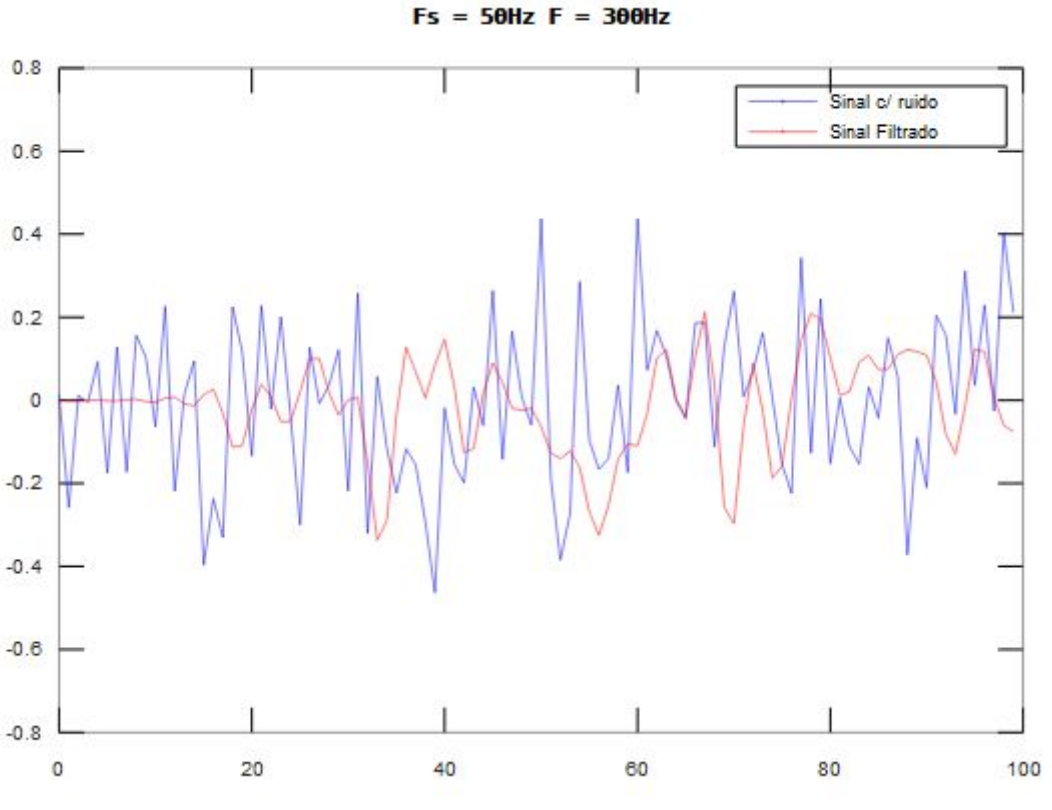
simulações do filtro:




Fs = 50Hz F = 15Hz
Sinal c/ ruido
Sinal Filtrado
1.5
1
0.5
0
-0.5
-1
-1.5
0
20
40
60
80
100


Fs = 50Hz F = 10Hz
Sinal c/ ruido
Sinal Filtrado
1.5
1
0.5
0
-0.5
-1
-1.5
0
20
40
60
80
100


Fs = 200Hz F = 70Hz
Sinal c/ ruido
Sinal Filtrado
1.5
1
0.5
0
-0.5
-1
-1.5
0
100
200
300
400


Fs = 50Hz F = 300Hz
Sinal c/ ruido
Sinal Filtrado
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
0
20
40
60
80
100

1 c - primeiro faz-se o zero padding para deixar os vetores do mesmo tamanho
o tamanho do vetor resultado da convolução é tamanho do kernel + tamanho do vetor de dados menos 1 ,então num loop de 1 ao tamanho do vetor de dados, computa-se um elemento do vetor de saída.

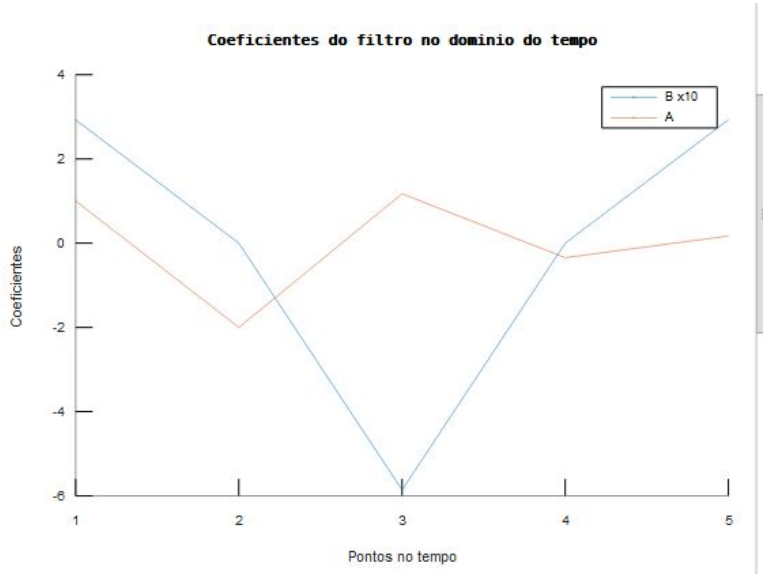
```
function Y = convolution(kernel,data)

m=length(kernel);
n=length(data);

%zero padding, deixar ambos os vetores do mesmo tamanho
X=[kernel,zeros(1,m)]
H=[data,zeros(1,n)]

   for i=1:n+m-1 % tamanho final do vetor de saida n+m-1 ou 2*length(X)-1
      Y(i)=0; % inicia com zero o elemento i do vetor de saida.
      for j=1:length(H)
         if(i-j+1>0) % posicao nao pode ser menor ou igual a zero
            Y(i)=Y(i)+X(j)*H(i-j+1); %computa a convolucao
         else
         end
      end
   end

endfunction
```

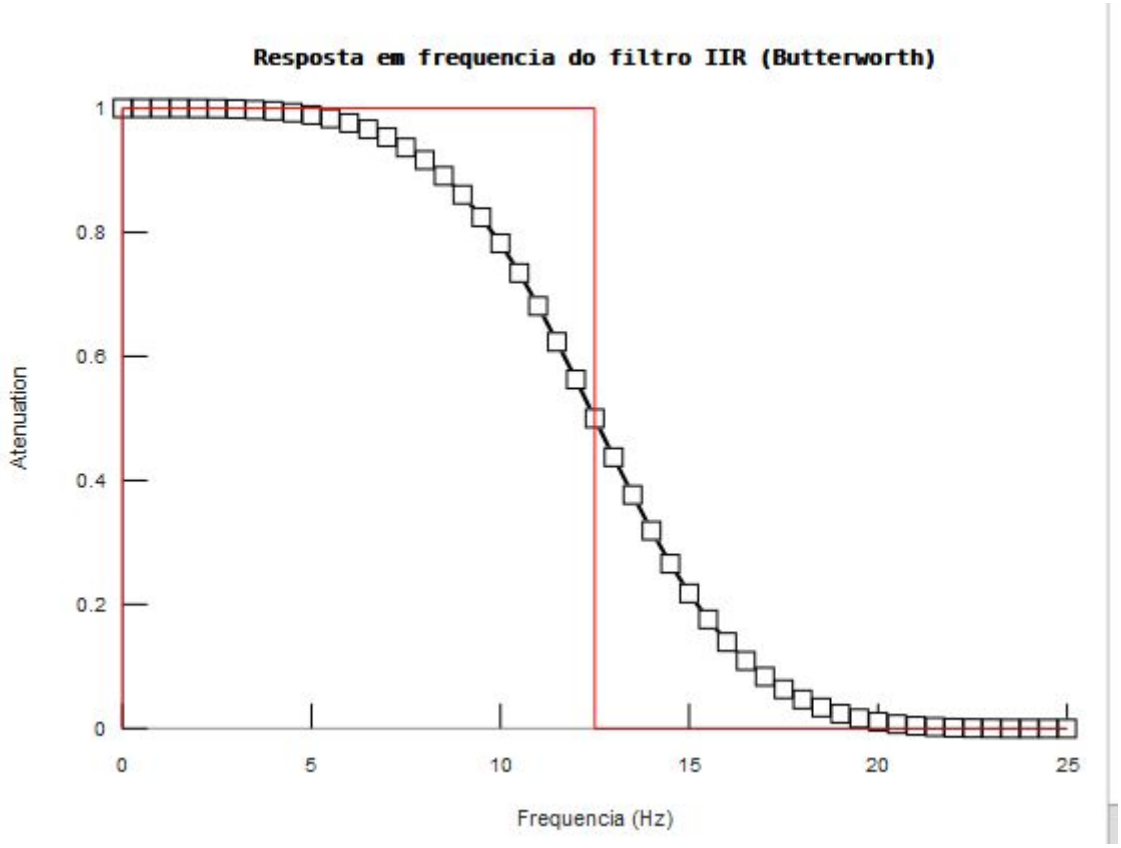
https://en.wikipedia.org/wiki/Convolution

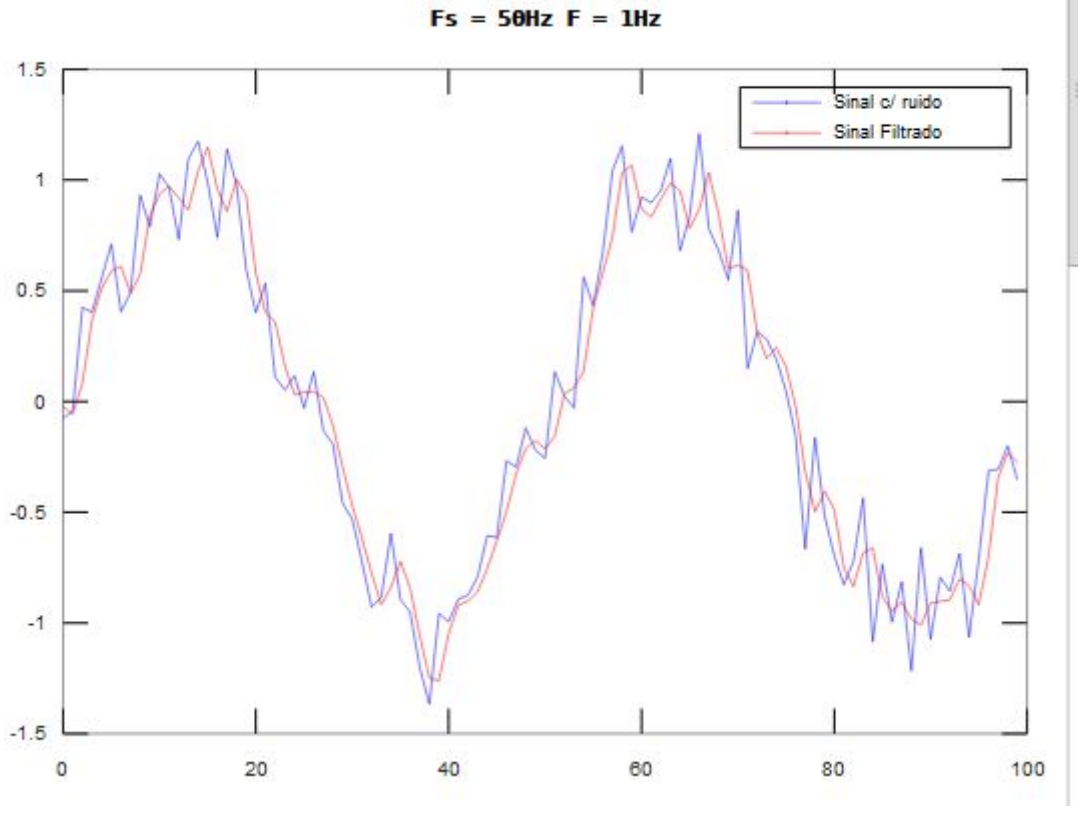
2 a-

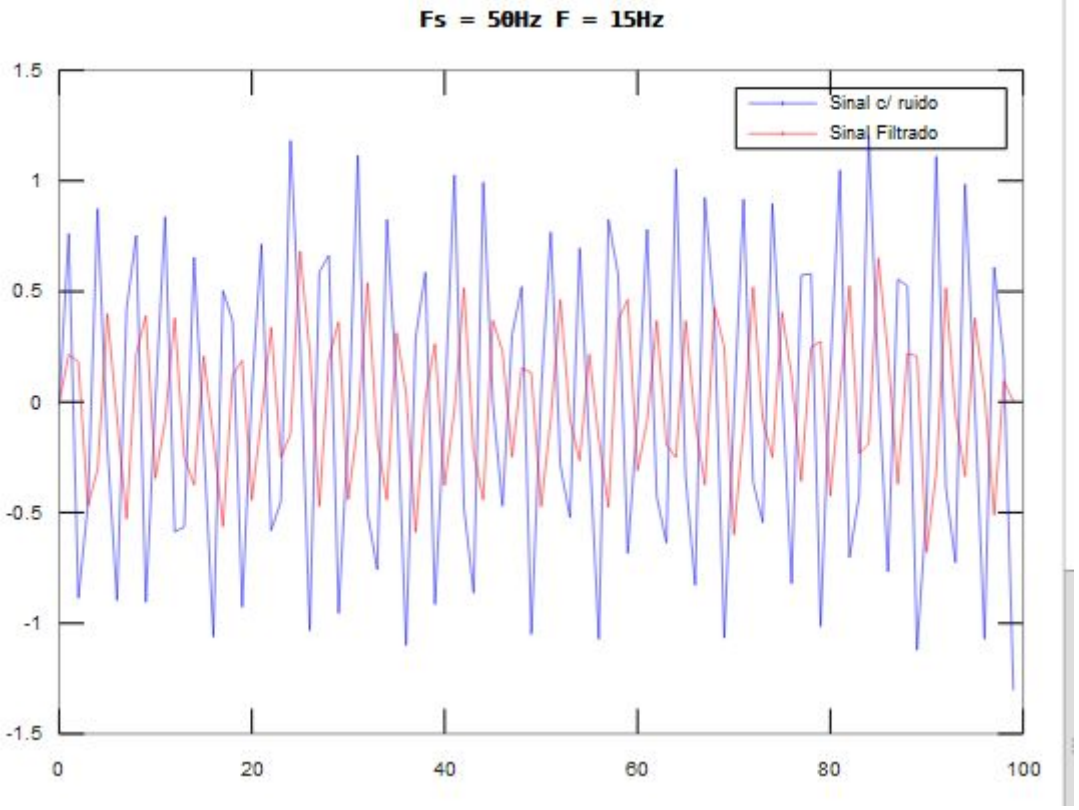
kernelA = [1,-2,1.1716,-0.34315,0.17157]; kernelB = [0.29289,0,-0.58579,0,0.29289];

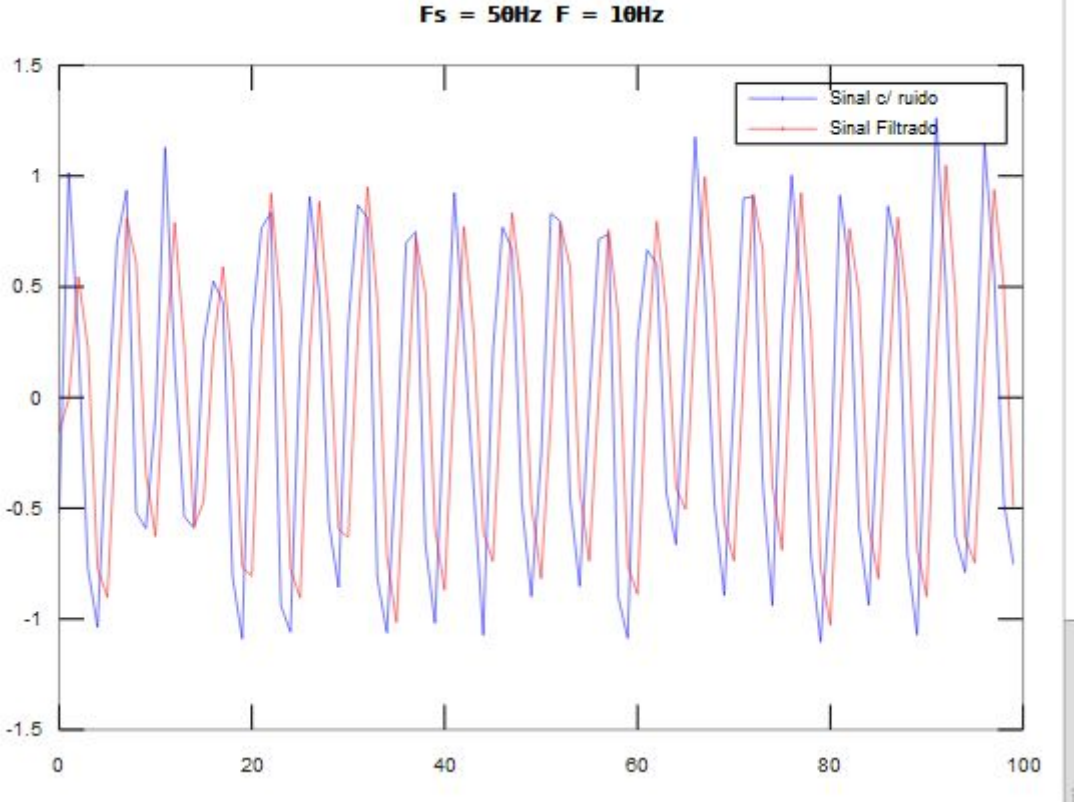
para ter uma resposta parecida com o filtro FIR um filtro IIR de ordem 2 foi suficiente.




Resposta em frequencia do filtro IIR (Butterworth)
Atenuation
Frequencia (Hz)
0
0.2
0.4
0.6
0.8
1
0
5
10
15
20
25


Fs = 50Hz F = 1Hz
Sinal c/ ruido
Sinal Filtrado
1.5
1
0.5
0
-0.5
-1
-1.5
0
20
40
60
80
100


Fs = 50Hz F = 15Hz
Sinal c/ ruido
Sinal Filtrado
1.5
1
0.5
0
-0.5
-1
-1.5
0
20
40
60
80
100


Fs = 50Hz F = 10Hz
Sinal c/ ruido
Sinal Filtrado
1.5
1
0.5
0
-0.5
-1
-1.5
0
20
40
60
80
100

2 b - 8 coeficientes/ordem. A partir de 9 já observa-se overshooting na resposta do impulso unitário.