

Aluno: Jedson Jhones Barbosa Alves

Usei o sinal completo do respirations.mat, sendo assim os gráficos ficam melhor de visualizar no arquivo de MATLAB.

Utilizei prints neste PDF, sendo assim os códigos podem ser obtidos (copiados) no arquivo de MATLAB, ou de HTML, que se encontram na mesma pasta que este PDF.

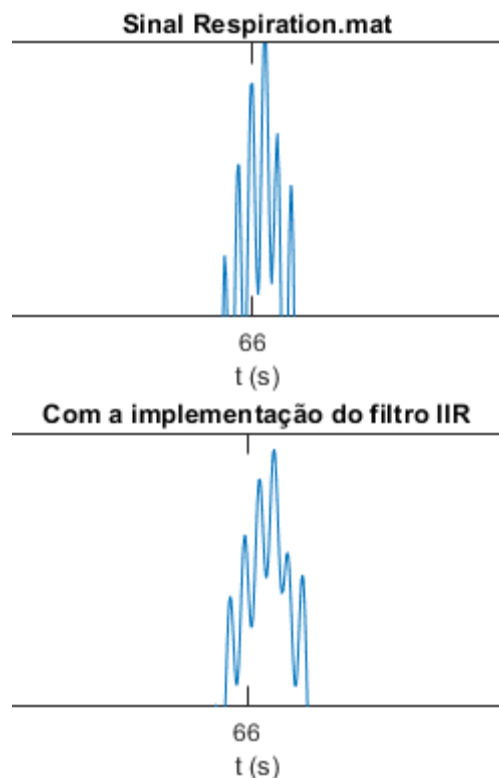
Exercício 1 – Filtro de média móvel

Usando o conteúdo apresentado em sala de aula, implemente filtros passa-baixas que permitam remover o ruído da rede elétrica do sinal contido no arquivo "**respiration.mat**".

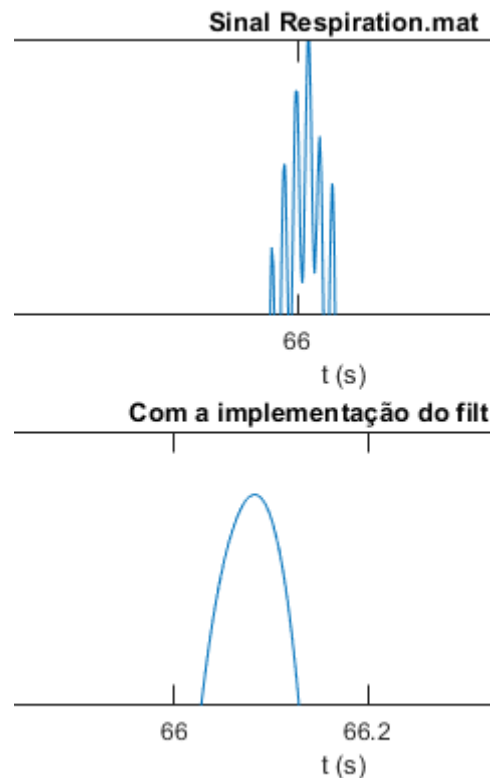
- a) Fixe uma frequência de corte f_c apropriada para a aplicação, justifique a sua escolha.

A frequência de corte f_c escolhida foi de 10 hz.

A escolha da frequência de corte foi feita depois de implementado os filtros. Inicialmente foi escolhida uma frequência de corte de 60 hz, mas como podemos ver nas imagens abaixo, ainda existiam ruídos no sinal obtido.



Então foi escolhida uma frequência de corte de 10 hz, onde assim o sinal ficou sem ruídos, que é como podemos perceber a partir da mesma partir do sinal, na figura abaixo.



- b) Implemente versões IIR e FIR do filtro passa baixas. Para obter os coeficientes dos filtros, siga as recomendações apresentadas na Figura 1.

Para a implementação do filtro IIR.

```
% Abrindo sinal respiration.mat
MatFile = matfile('respiration.mat');
x = MatFile.respiration;
fs = 1000; % sample rate in Hz
ts = 1/fs; % sample period in sec
NSamples = length(x);
% Criando um vetor de tempo
t = (0:NSamples-1)/fs;
%% Plotando o sinal de respiration.mat
fig = figure('Position',[10 10 900 300],'color','w');
ax = subplot(2,1,1);
lin1 = plot(t,x);
xlabel('t (s)')
ylabel('Amp')
title('Sinal Respiration.mat');
% Frequencia de corte
fc = 10; % Cut-off frequency in Hz
%% Obtendo os coeficientes do filtro IIR
wc = fc/(fs/2); % Normalized Cut-off frequency, fs/2 -> 1
```

```

N = 7; % filter order
[b,a] = butter(N,wc);
% Implementação do filtro IIR, e plotagem do sinal
x1b = filter(b,a,x);
ax = subplot(2,1,2);
plot(t,x1b);
xlabel('t (s)');
ylabel('Amp');
title('x_{1b}');
title('Com a implementação do filtro IIR');
ylim([-10 -4])

```

```

1      % Abrindo sinal respiration.mat
2 -    MatFile = matfile('respiration.mat');
3 -    x = MatFile.respiration;
4 -    fs = 1000; % sample rate in Hz
5 -    ts = 1/fs; % sample period in sec
6 -    NSamples = length(x);
7      % Criando um vetor de tempo
8 -    t = (0:NSamples-1)/fs;
9      %% Plotando o sinal de respiration.mat
10 -   fig = figure('Position',[10 10 900 300],'color','w');
11 -   ax = subplot(2,1,1);
12 -   lin1 = plot(t,x);
13 -   xlabel('t (s)');
14 -   ylabel('Amp');
15 -   title('Sinal Respiration.mat');|
16      % Frequencia de corte
17 -   fc = 10; % Cut-off frequency in Hz
18      %% Obtendo os coeficientes do filtro IIR
19 -   wc = fc/(fs/2); % Normalized Cut-off frequency, fs/2 -> 1
20 -   N = 7; % filter order
21 -   [b,a] = butter(N,wc);
22      % Implementação do filtro IIR, e plotagem do sinal
23 -   x1b = filter(b,a,x);
24 -   ax = subplot(2,1,2);
25 -   plot(t,x1b);
26 -   xlabel('t (s)');
27 -   ylabel('Amp');
28 -   title('x_{1b}');
29 -   title('Com a implementação do filtro IIR');
30 -   ylim([-10 -4])

```

Para a implementação do filtro FIR:

```

% Abrindo sinal respiration.mat
MatFile = matfile('respiration.mat');
x = MatFile.respiration;
fs = 1000; % sample rate in Hz

```

```

ts = 1/fs; % sample period in sec
NSamples = length(x);
% Criando um vetor de tempo
t = (0:NSamples-1)/fs;
%% Plotando o sinal de respiration.mat
fig = figure('Position',[10 10 900 300],'color','w');
ax = subplot(2,1,1);
lin1 = plot(t,x);
xlabel('t (s)')
ylabel('Amp')
title('Sinal Respiration.mat');
% Frequencia de corte
fc = 10; % Cut-off frequency in Hz
%% Obtendo os coeficiente do filtro FIR
wc = fc/(fs/2); % Cut-off frequency normalized, fs/2 -> 1
N = 51; % filter order
[b,a] = fir1(N,wc);
% Implementação do filtro FIR, e plotagem do sinal
x1b = filter(b,a,x);
ax = subplot(2,1,2);
plot(t,x1b);
xlabel('t (s)');
ylabel('Amp');
title('x_{1b}');
title('Com a implementação do filtro FIR');
ylim([-10 -4])

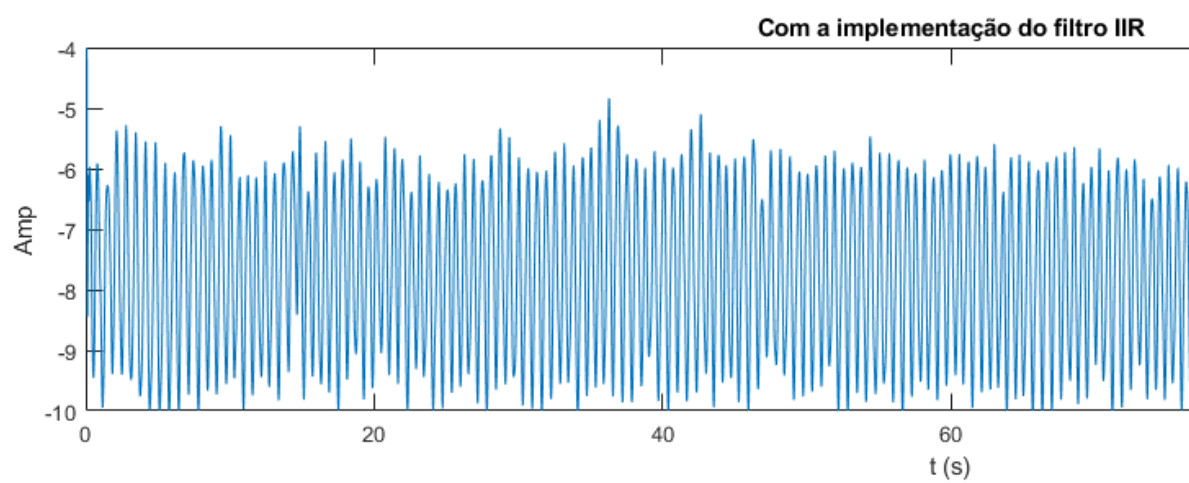
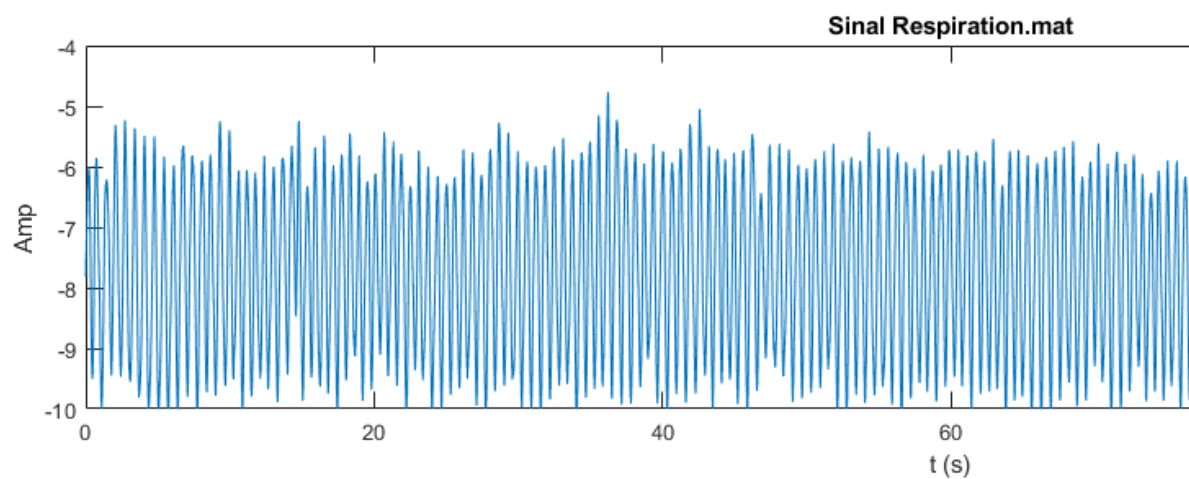
```

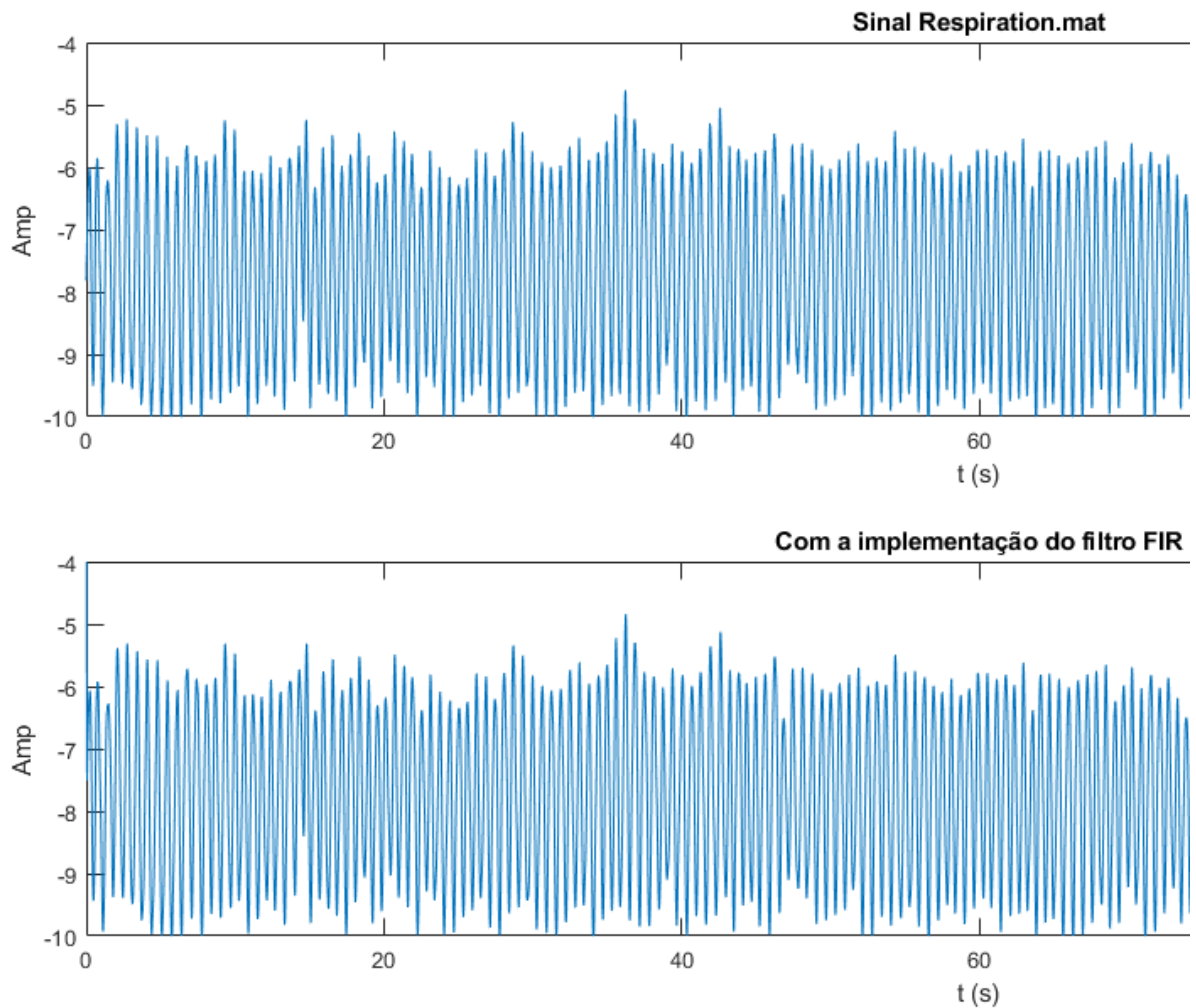
```

1 | % Abrindo sinal respiration.mat
2 - MatFile = matfile('respiration.mat');
3 - x = MatFile.respiration;
4 - fs = 1000; % sample rate in Hz
5 - ts = 1/fs; % sample period in sec
6 - NSamples = length(x);
7 | % Criando um vetor de tempo
8 - t = (0:NSamples-1)/fs;
9 | %% Plotando o sinal de respiration.mat
10 - fig = figure('Position',[10 10 900 300],'color','w');
11 - ax = subplot(2,1,1);
12 - lin1 = plot(t,x);
13 - xlabel('t (s)')
14 - ylabel('Amp')
15 - title('Sinal Respiration.mat');
16 | % Frequencia de corte
17 - fc = 10; % Cut-off frequency in Hz
18 | %% Obtendo os coeficiente do filtro FIR
19 - wc = fc/(fs/2); % Cut-off frequency normalized, fs/2 -> 1
20 - N = 51; % filter order
21 - [b,a] = firl(N,wc);
22 | % Implementação do filtro FIR, e plotagem do sinal
23 - xlb = filter(b,a,x);
24 - ax = subplot(2,1,2);
25 - plot(t,xlb);
26 - xlabel('t (s)');
27 - ylabel('Amp');
28 - title('x_{lb}');
29 - title('Com a implementação do filtro FIR');
30 - ylim([-10 -4])

```

- c) Plote no domínio tempo: 1) o sinal original e 2) os sinais resultantes das filtrações aplicadas no passo anterior.





d) Obtenha e plote o espectro do sinal original e dos sinais resultantes das filtrações aplicadas no passo anterior.

Para obter e plotar o espectro do sinal original, e dos sinais resultando da filtragem, foi adicionado o seguinte código ao código do filtro FIR e IIR.

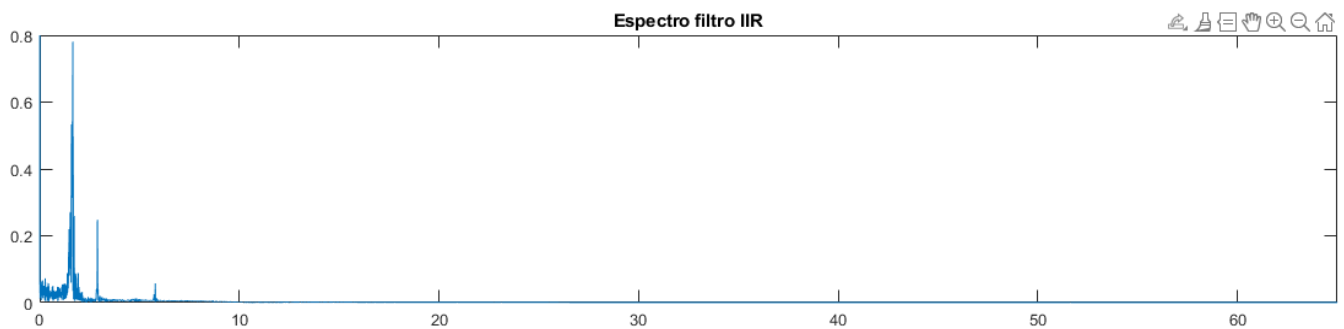
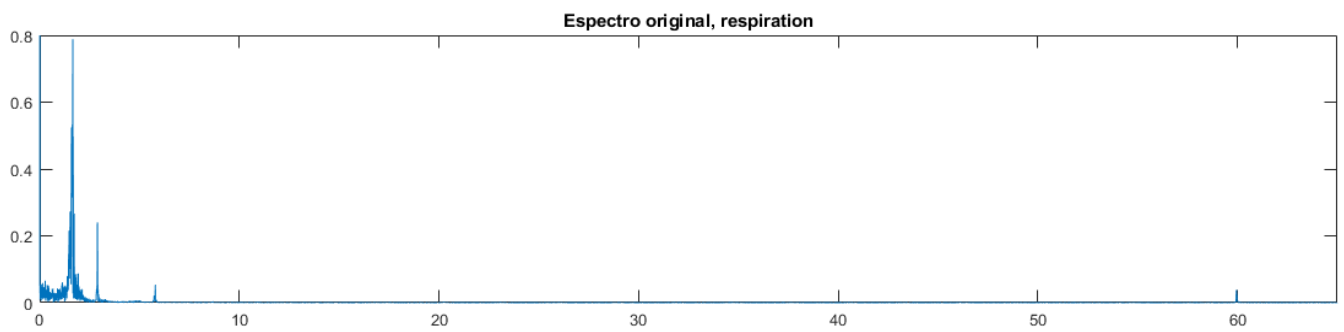
```
%%Calculando o espectro para o sinal respiration.mat
nfft = NSamples;
Y1 = fft(x);
Y1 = abs(Y1)/nfft;
Y1 = 2*Y1(1:nfft/2+1);
f = linspace(0,fs/2,length(Y1));
% Plotando o espectro do respiration.mat
fig = figure('Position',[10 10 900 300],'color','w');
ax2 = subplot(2,1,1);
plot(f,Y1)
title(['Espectro original, respiration'])
xlim([0 65])
ylim([0 0.8])
```

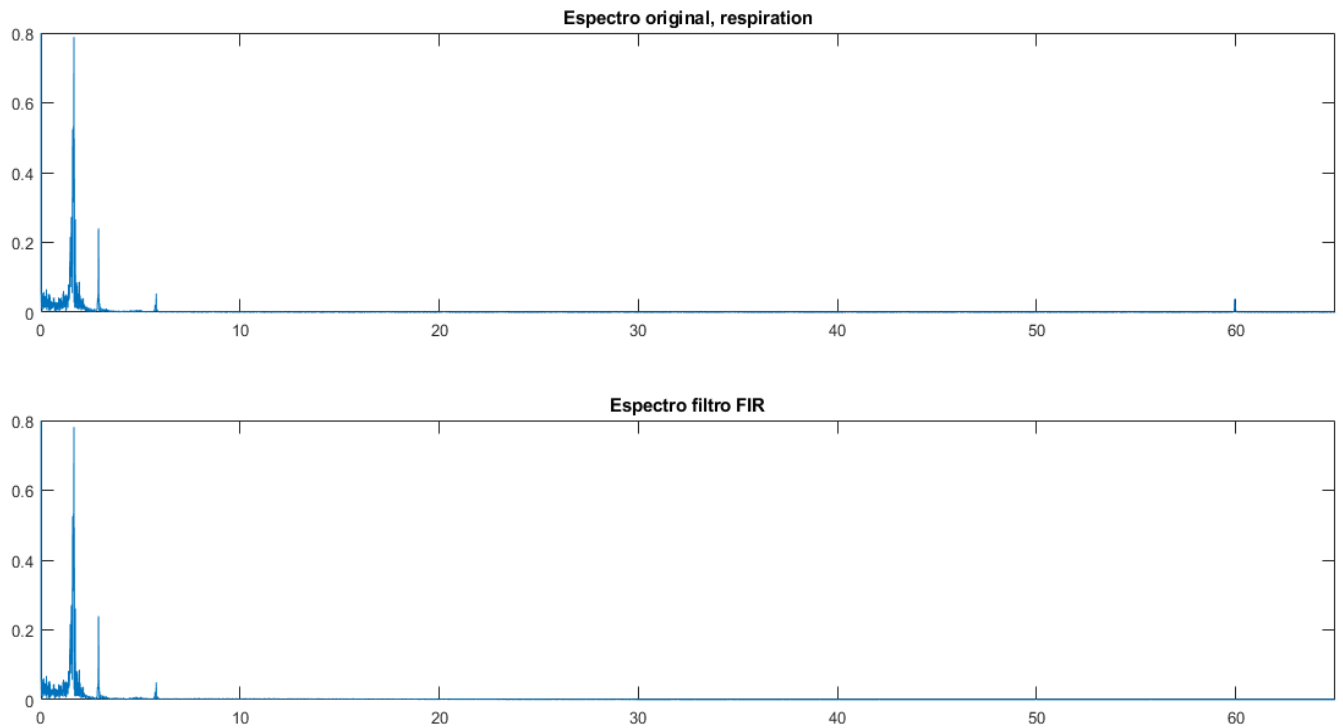
```

%%Calculando o espectro, filtro IIR
nfft = NSamples;
Y2 = fft(x1b);
Y2 = abs(Y2)/nfft;
Y2 = 2*Y2(1:nfft/2+1);
f = linspace(0,fs/2,length(Y2));
% Plotando o espectro do sinal com a filtragem IIR
ax3 = subplot(2,1,2);
plot(f,Y2)
title(['Espectro filtro IIR'])
xlim([0 65])
ylim([0 0.8])

```

Obtendo se os gráficos:





e) Discuta os resultados obtidos nos passos anteriores

Através dos Filtros IIR e FIR fomos capazes de retirar os ruídos da rede elétrica presente no sinal "respiration", ao fazer a filtragem para uma determinada frequência, dessa forma tendo um sinal mais limpo e fácil de ser analisado. Também fomos capazes de plotar os espectros dos sinais para serem analisados.