

Contents

- [Base de dados 1](#)
- [Leitura do arquivo .MAT](#)
- [Mostrando um dos sinais](#)
- [Criação de filtros para retirar componentes da alimentação AC](#)
- [Filtrando o sinal](#)
- [Usando um filtro passa baixas](#)
- [Usando o baseline](#)

Base de dados 1

Esta script lê a base de dados 1, disponível em: [<https://sites.google.com/site/projectbci/>]

```
clear all; close all; clc
```

Leitura do arquivo .MAT

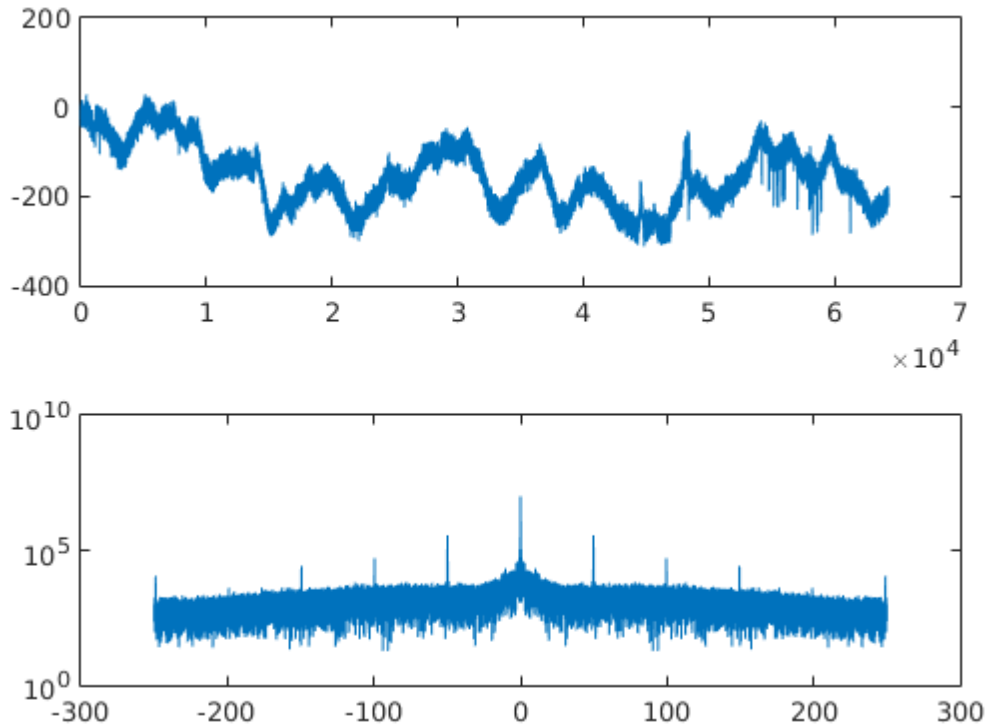
```
load Subject1_1D.mat
```

Mostrando um dos sinais

A base de dados é dividida em 3 variáveis: * baseline * right * left

Iremos verificar o funcionamento dos sinais

```
% Cálculo do espectro de frequências  
espectro_right = fftshift(abs(fft(right(1,:))));  
  
figure  
subplot(2,1,1); plot(right(1,:))  
subplot(2,1,2); semilogy(linspace(-250,250,64300),espectro_right)
```



Criação de filtros para retirar componentes da alimentação AC

O site fala que a alimentação é em 50 Hz. Criaremos filtros para retirar sua interferência.

Os filtros podem ser criados no FDATool, mas para este caso será usada a função FIR1. Criaremos filtros rejeita faixa para tirar as harmônicas.

```
filtro50 = fir1(200,[45 55]/500,'stop');
filtro100 = fir1(200,[95 105]/500,'stop');
filtro150 = fir1(200,[145 155]/500,'stop');
filtro200 = fir1(200,[195 205]/500,'stop');
filtro250 = fir1(200,[245 255]/500,'stop');
```

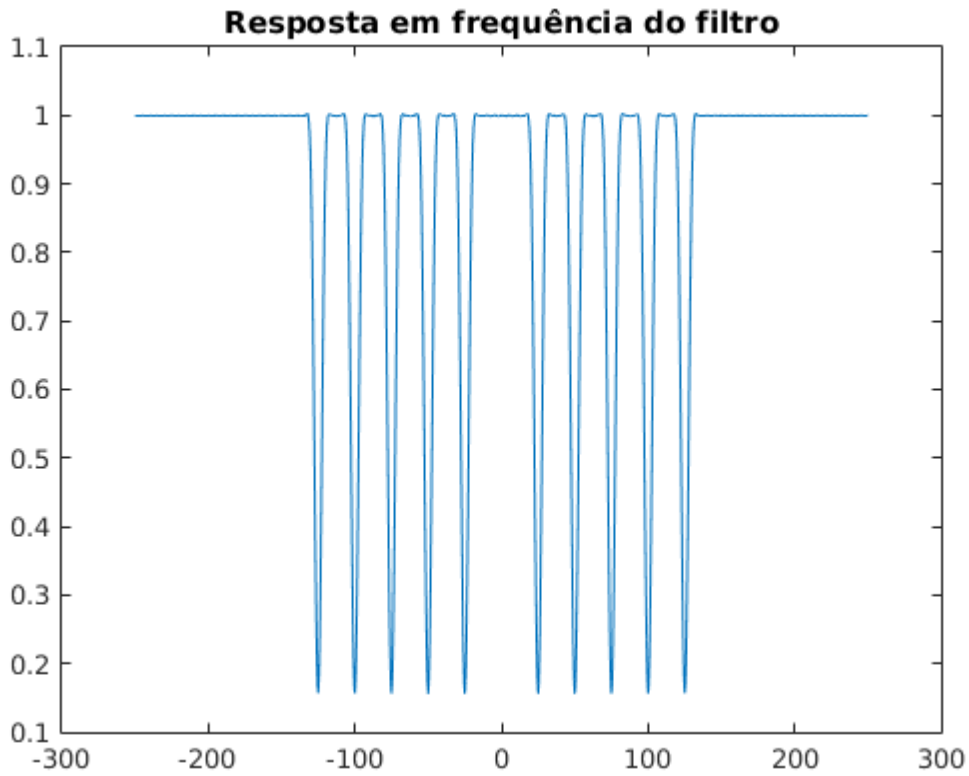
A resposta final é obtida a partir da convolução das respostas ao impulso

```
filtro = conv(filtro50,filtro100);
filtro = conv(filtro, filtro150);
filtro = conv(filtro, filtro200);
filtro = conv(filtro, filtro250);
```

A resposta em frequência do filtro é calculada

```
F = fftshift(abs(fft(filtro)));

figure
plot(linspace(-250,250,length(filtro)),F)
title('Resposta em frequência do filtro')
```



Filtrando o sinal

O sinal observado anteriormente será filtrado. Para isso, usaremos a função `FILTER`

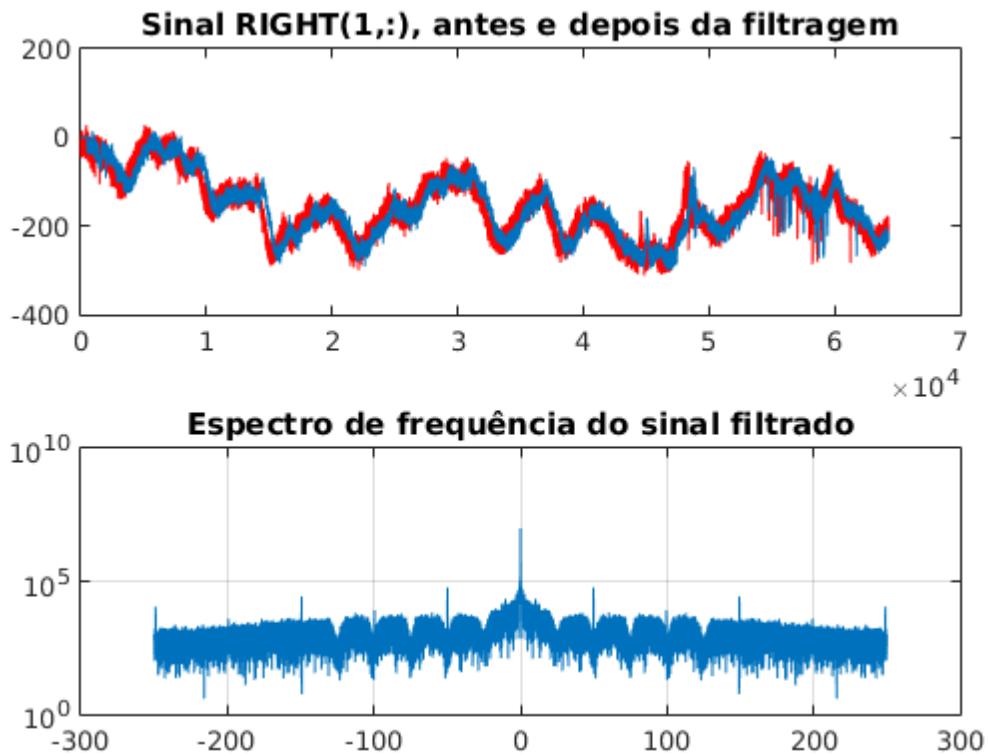
```
saida = filter(filtro,1,right(1,:));
```

Calculando o espectro de frequência

```
espectro_saida = fftshift(abs(fft(saida)));
```

Mostrando os resultados

```
figure  
subplot(2,1,1); plot(right(1,:), 'r'); hold on; plot(saida)  
title('Sinal RIGHT(1,:), antes e depois da filtragem')  
subplot(2,1,2); semilogy(linspace(-250,250,length(espectro_saida)),espectro_saida)  
grid on; title('Espectro de frequência do sinal filtrado')
```



Usando um filtro passa baixas

Para testar realmente a base de dados, iremos usar um filtro passa baixas, criado da mesma forma anterior.

```
filtroPB = fir1(200,30/500);
```

Filtrando o sinal

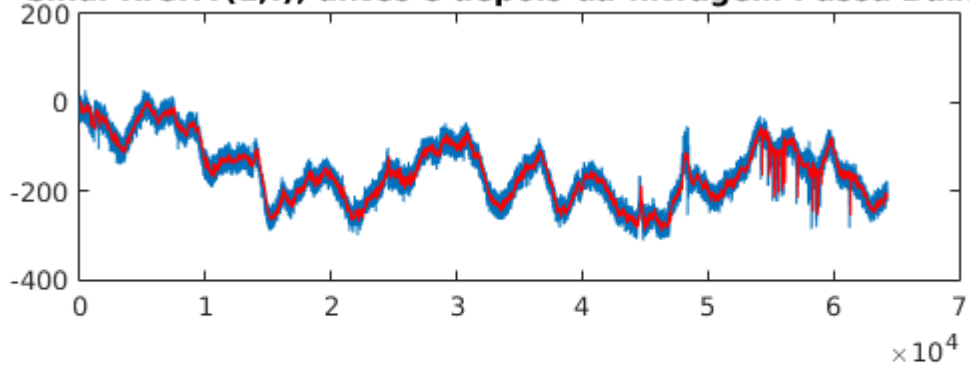
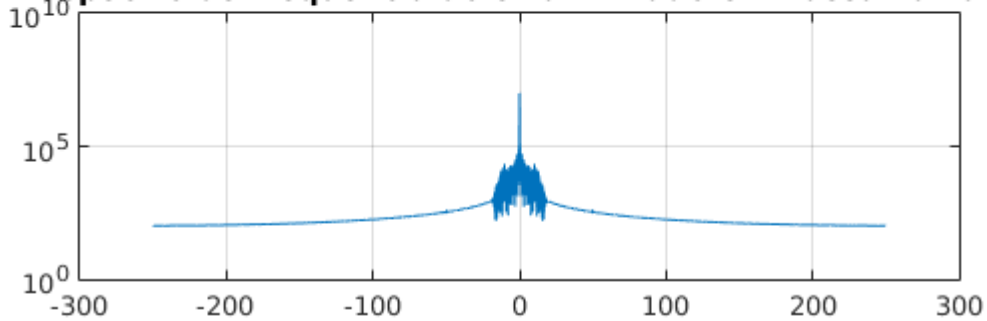
```
saidaPB = filter(filtroPB,1,right(1,:));
```

Calculando o espectro de frequências

```
espectroPB = fftshift(abs(fft(saidaPB)));
```

Mostrando os resultados

```
figure
subplot(2,1,1); plot(right(1,:)); hold on; plot(saidaPB,'r')
title('Sinal RIGHT(1,:), antes e depois da filtragem Passa Baixas')
subplot(2,1,2); semilogy(linspace(-250,250,length(espectroPB)),espectroPB)
grid on; title('Espectro de frequência do sinal filtrado em Passa Baixas')
```

Sinal RIGHT(1,:), antes e depois da filtragem Passa Baixas**Espectro de frequência do sinal filtrado em Passa Baixas**

Usando o baseline

O mesmo teste anterior será feito com o sinal ***baseline***

Filtragem do sinal

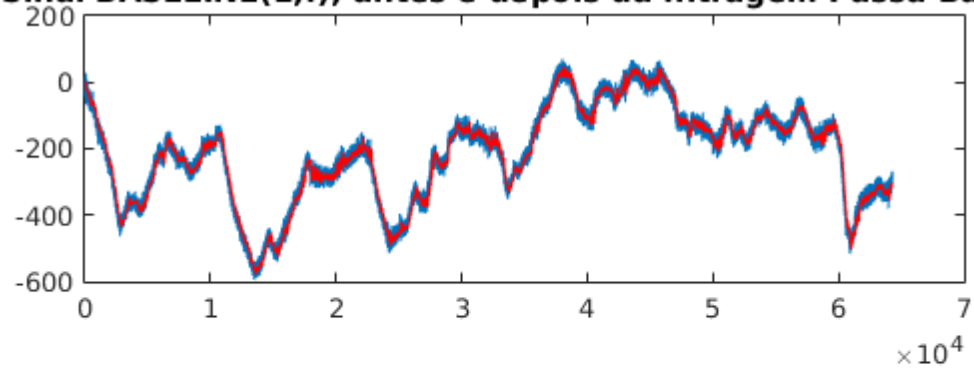
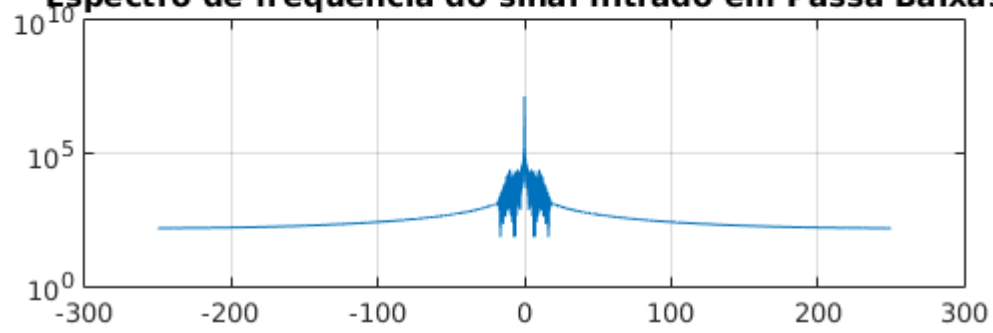
```
saida_baseline = filter(filtroPB,1,baseline(1,:));
```

Espectro de frequências

```
espectro_baseline = fftshift(abs(fft(saida_baseline)));
```

Mostrando os resultados

```
figure
subplot(2,1,1); plot(baseline(1,:)); hold on; plot(saida_baseline,'r')
title('Sinal BASELINE(1,:), antes e depois da filtragem Passa Baixas')
subplot(2,1,2); semilogy(linspace(-250,250,length(espectro_baseline)),espectro_baseline)
grid on; title('Espectro de frequência do sinal filtrado em Passa Baixas')
```

Sinal BASELINE(1,:), antes e depois da filtragem Passa Baixas**Espectro de frequência do sinal filtrado em Passa Baixas**

Published with MATLAB® R2015a