



# Filtro

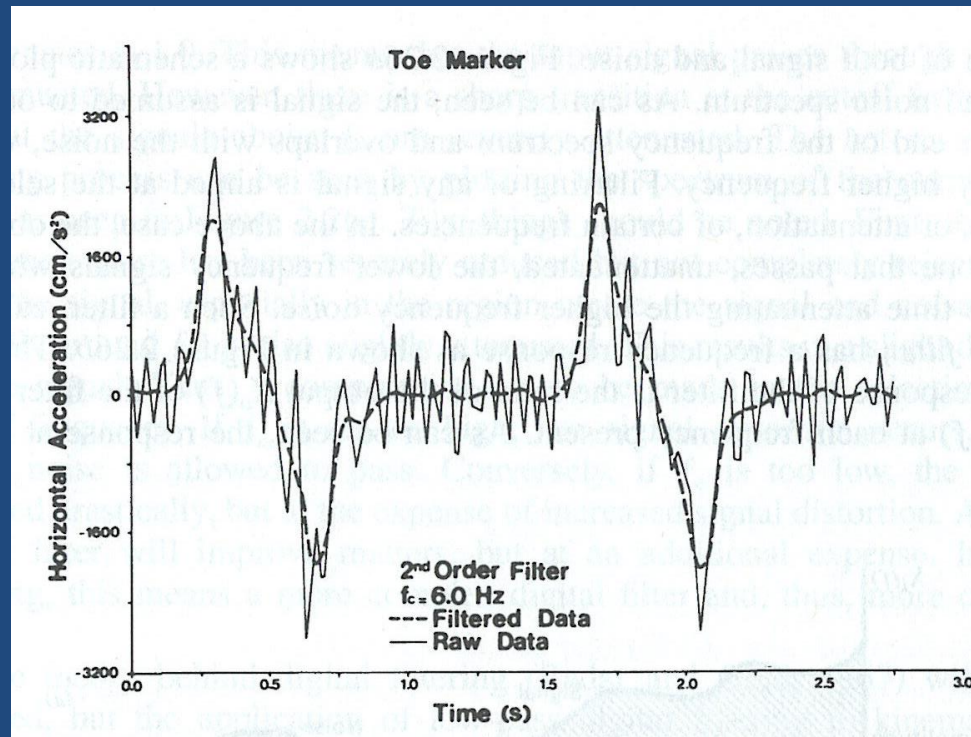
*Processamento de Sinais e Instrumentação  
para a Análise do Movimento Humano*

Prof. Dr. Renato de Moraes

# Ruído

- Termo usado para descrever os componentes de um sinal final que não são devidos ao processo em si (i.e., movimento analisado)
- Fontes de ruído:
  - Eletrônico
  - Movimento da pele
  - Erro humano
- Erro aleatório
  - Usualmente é um componente de alta frequência

# Dado bruto e dado filtrado



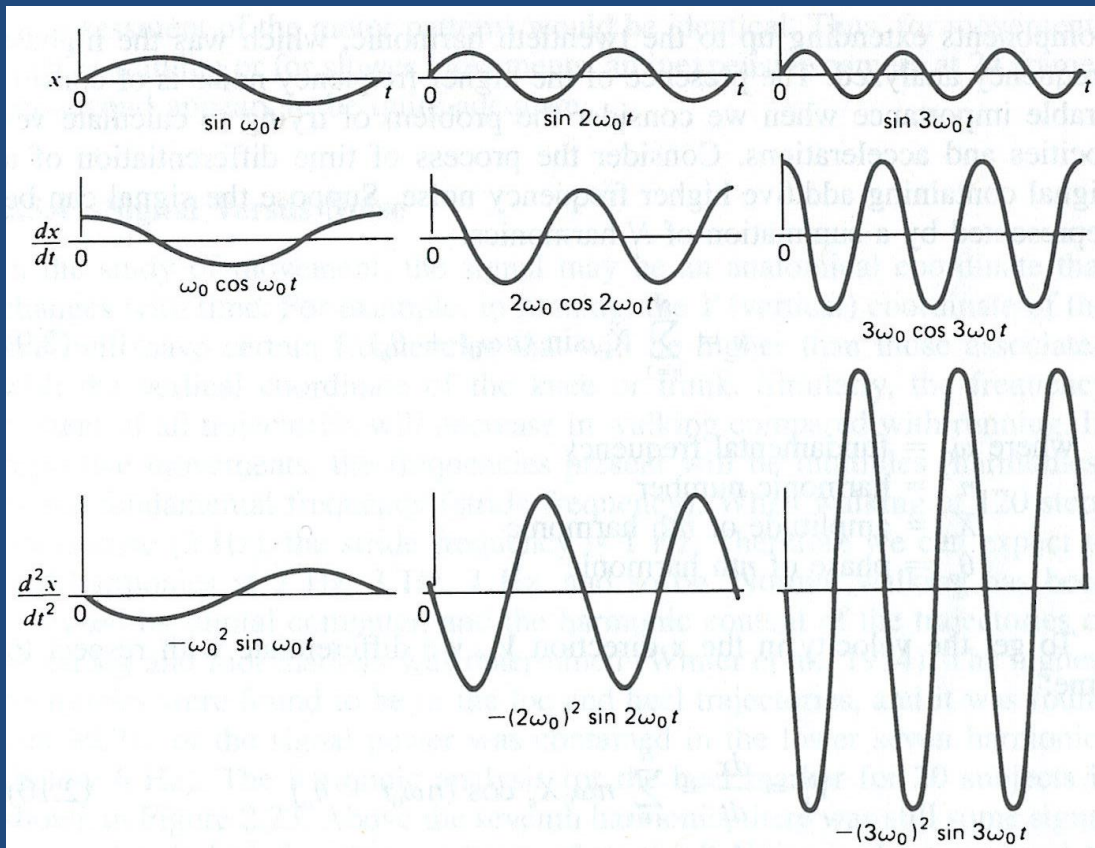
**Dado bruto: sinal coletado e armazenado sem ter sido processado**

# Ruído e cálculo de velocidade e aceleração

Frequência Fundamental

2º Harmônico

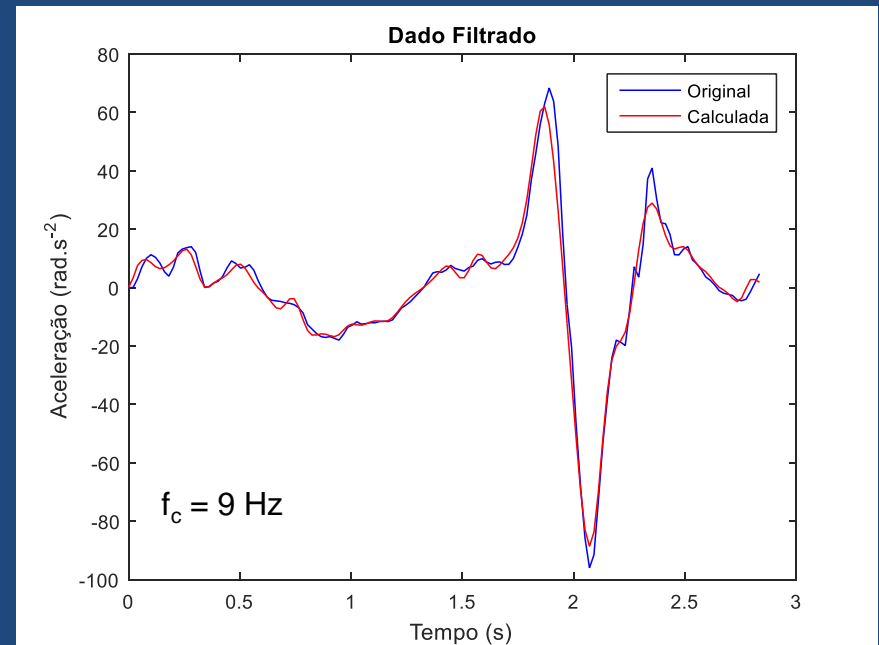
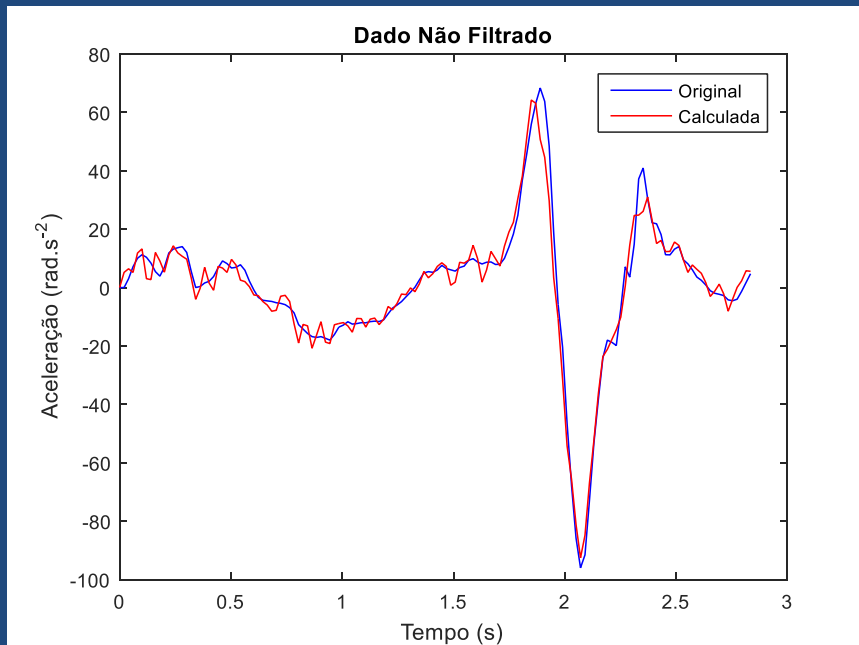
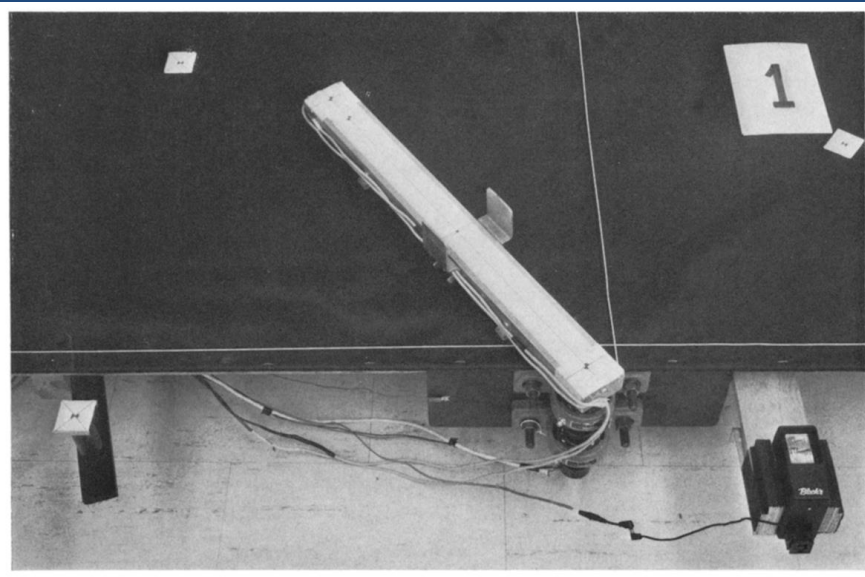
3º Harmônico



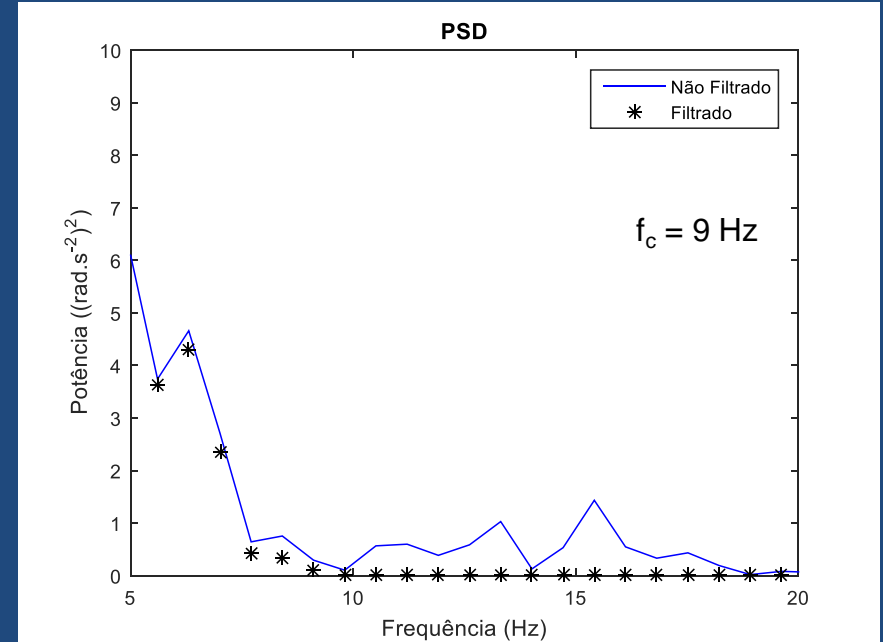
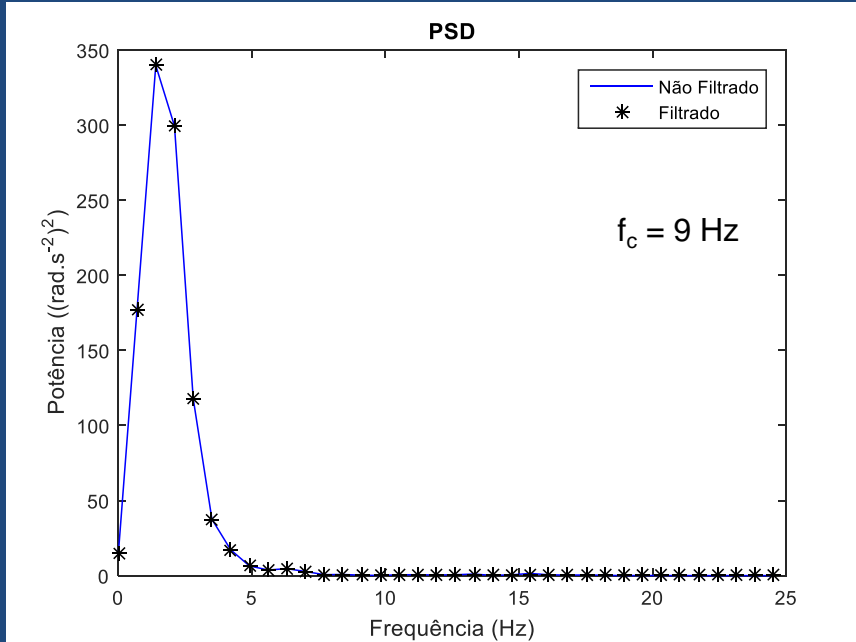
O aumento da frequência, aumenta a amplitude da velocidade e da aceleração

Velocidade: linear  
Aceleração: exponencial

# Efeito do filtro no cálculo da aceleração



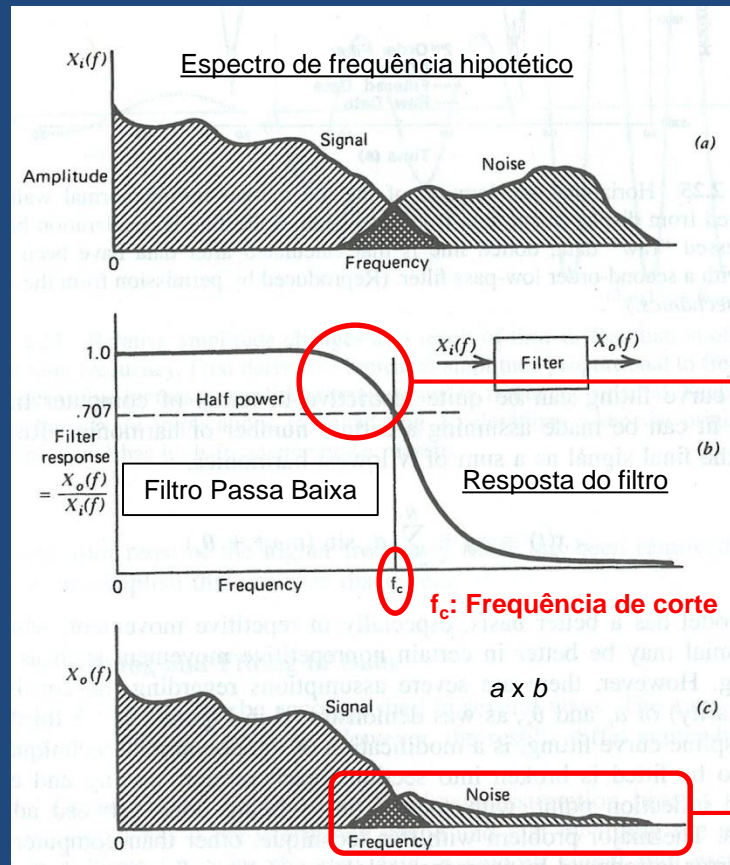
# PSD – Sinal filtrado e não filtrado



# Técnicas para remoção de ruído

- Ajuste de curva
  - Ajuste de um polinômio
  - Reconstrução do sinal usando harmônicos
  - Spline
- Filtros Digitais
  - Butterworth
  - Chebyshev

# Relação sinal/ruído



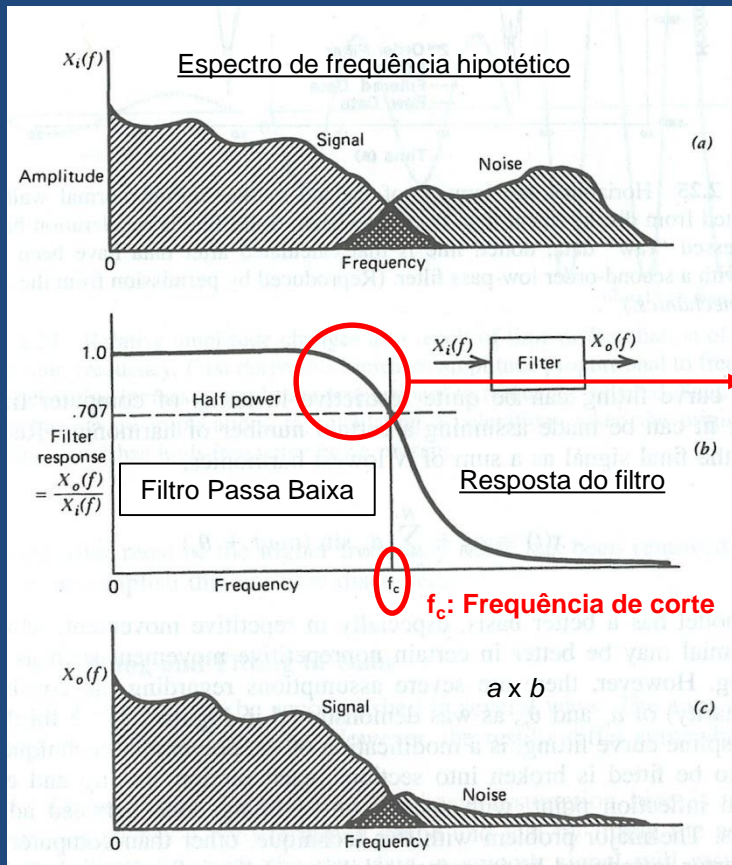
$X_i(f)$ : input para o filtro  
 $X_o(f)$ : output do filtro

**Distorção do sinal**

**Ruído de alta frequência foi severamente reduzido, mas não completamente rejeitado**



# Relação sinal/ruído

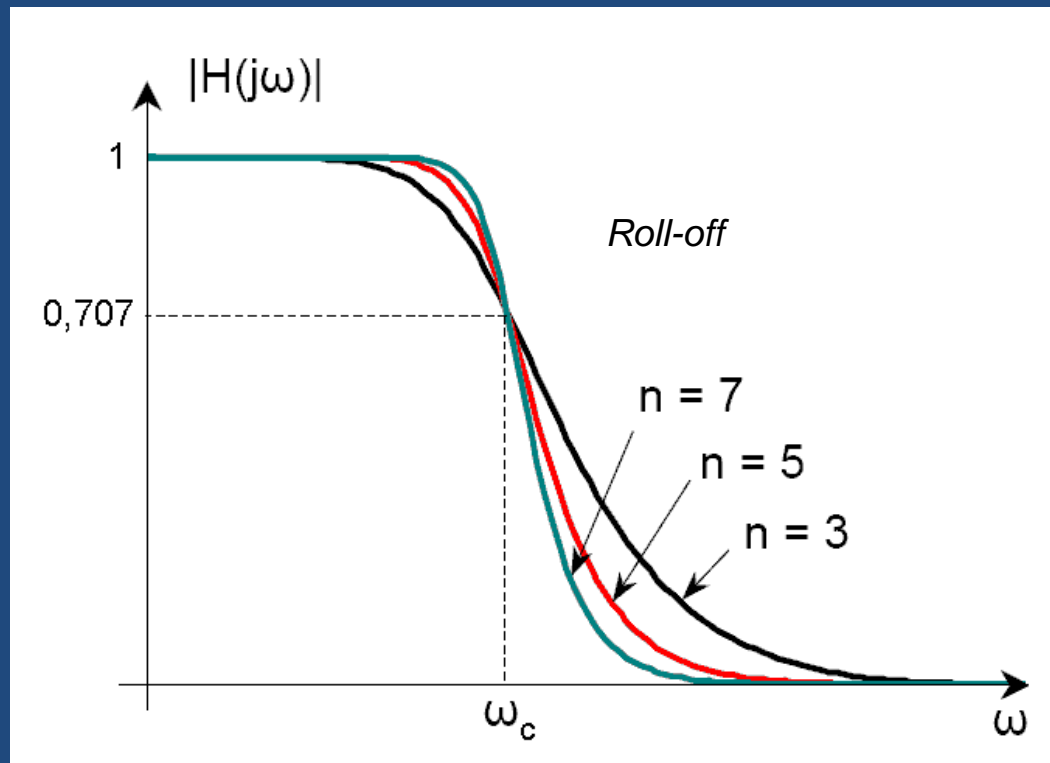


## Distorção do sinal

Se  $f_c$  for muito alta  $\rightarrow$   $\downarrow$  distorção do sinal, mas passa muito ruído.

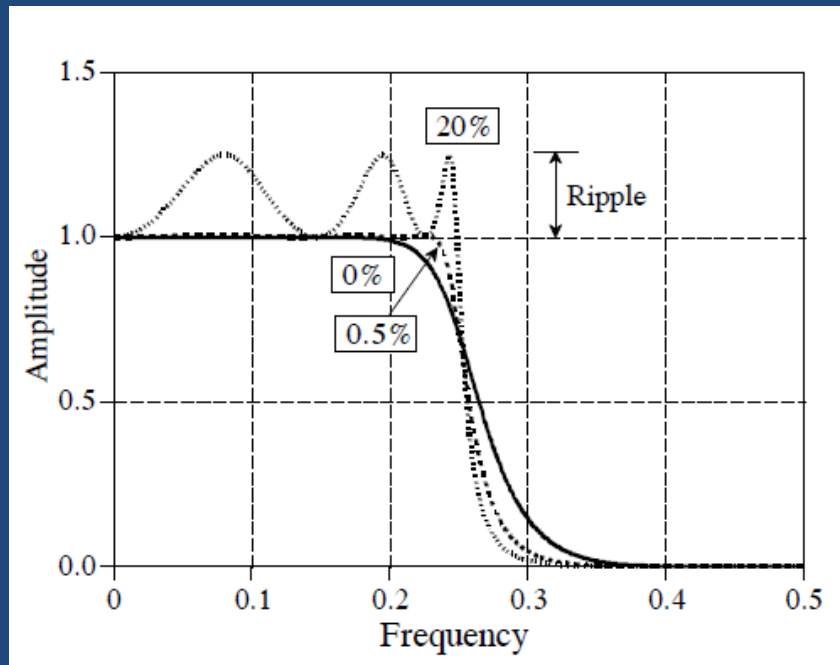
Se  $f_c$  for muito baixa  $\rightarrow$   $\uparrow$  distorção do sinal, mas passa pouco ruído.

# Resposta do filtro – Ordem do filtro



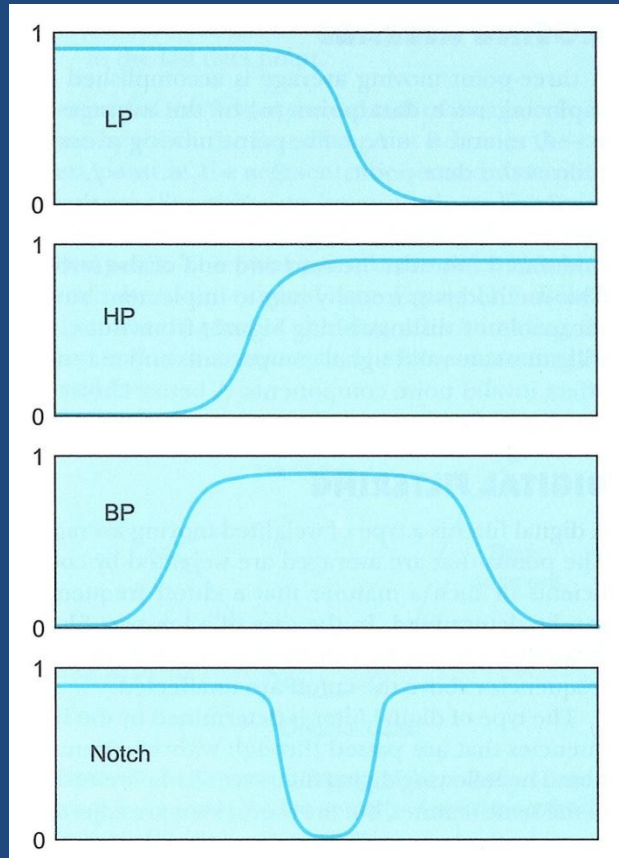
**Roll-off:** razão de atenuação acima da frequência de corte

# Resposta do filtro



Chebyshev: *roll-off* mais rápido, porém produz ondulação (*ripple*)

# Tipos de Filtro



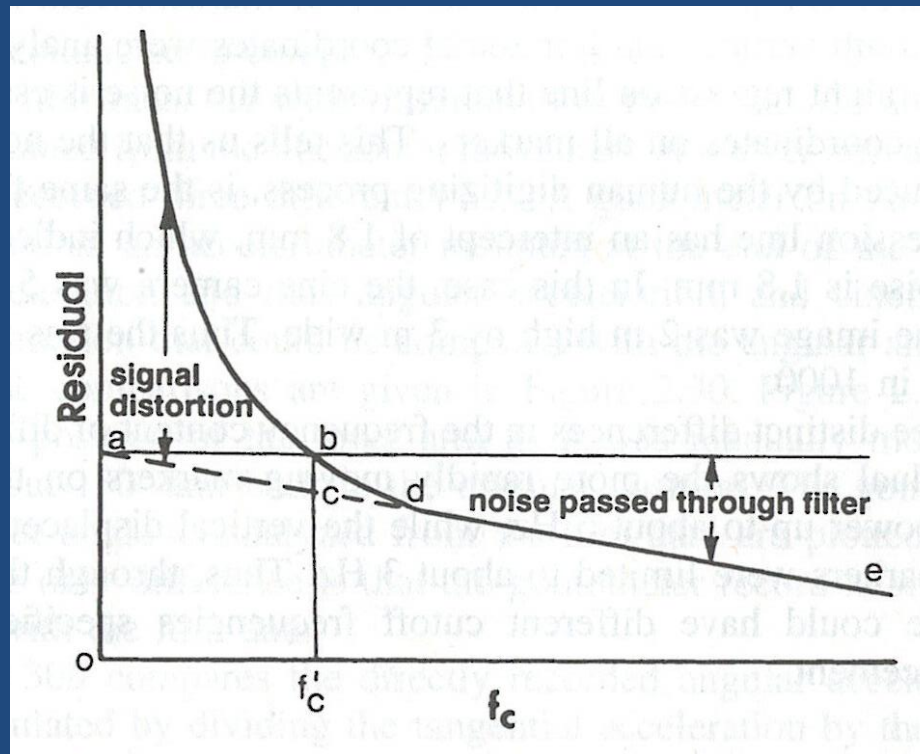
Low-pass

High-pass

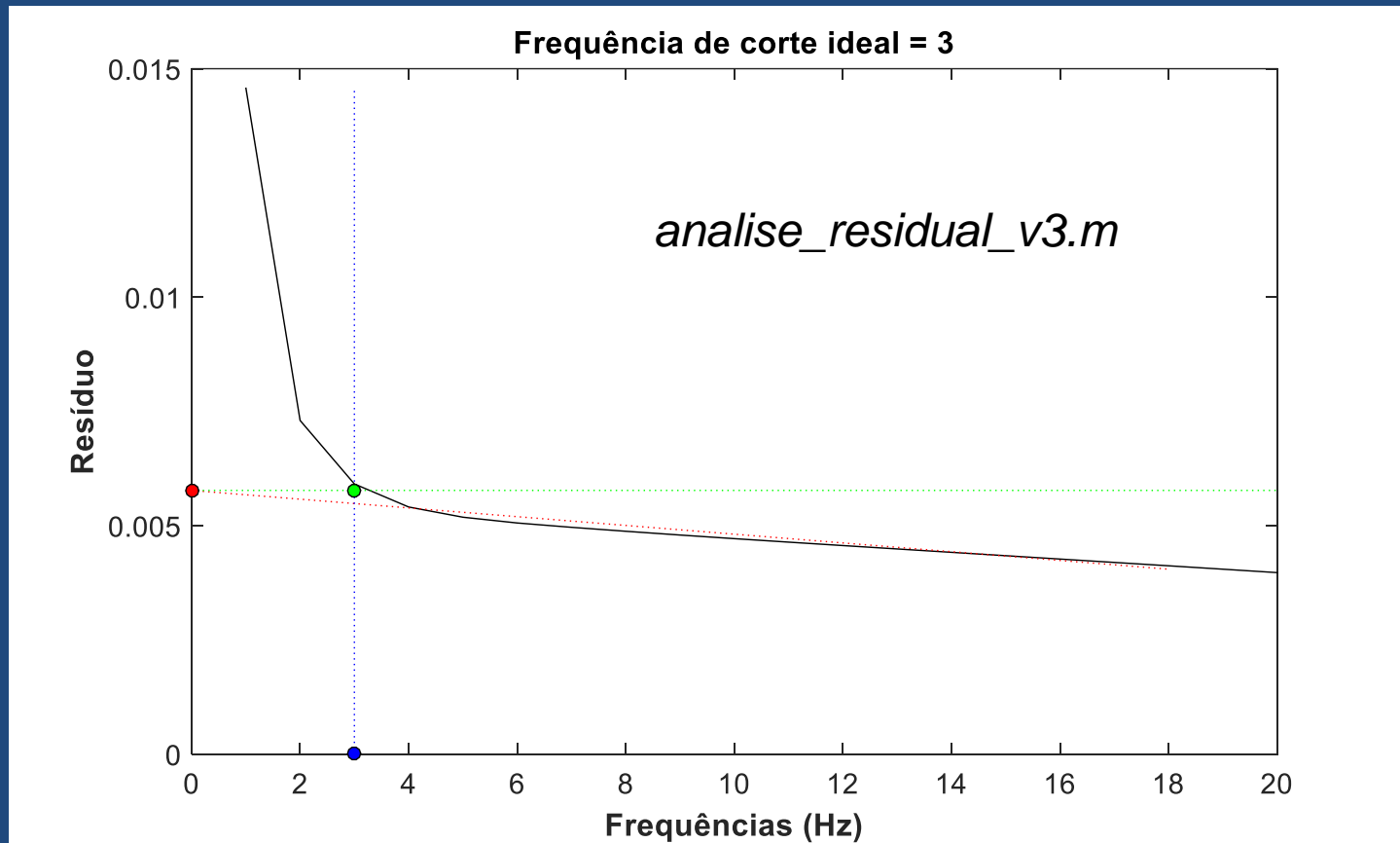
Band-pass

# Análise de resíduo para determinar a frequência de corte ( $f_c$ )

Análise residual da diferença entre sinais filtrados e não filtrados em uma ampla faixa de frequências de corte



# Exemplo – Centro de Pressão



# Comandos Matlab

# butter.m e filtfilt.m

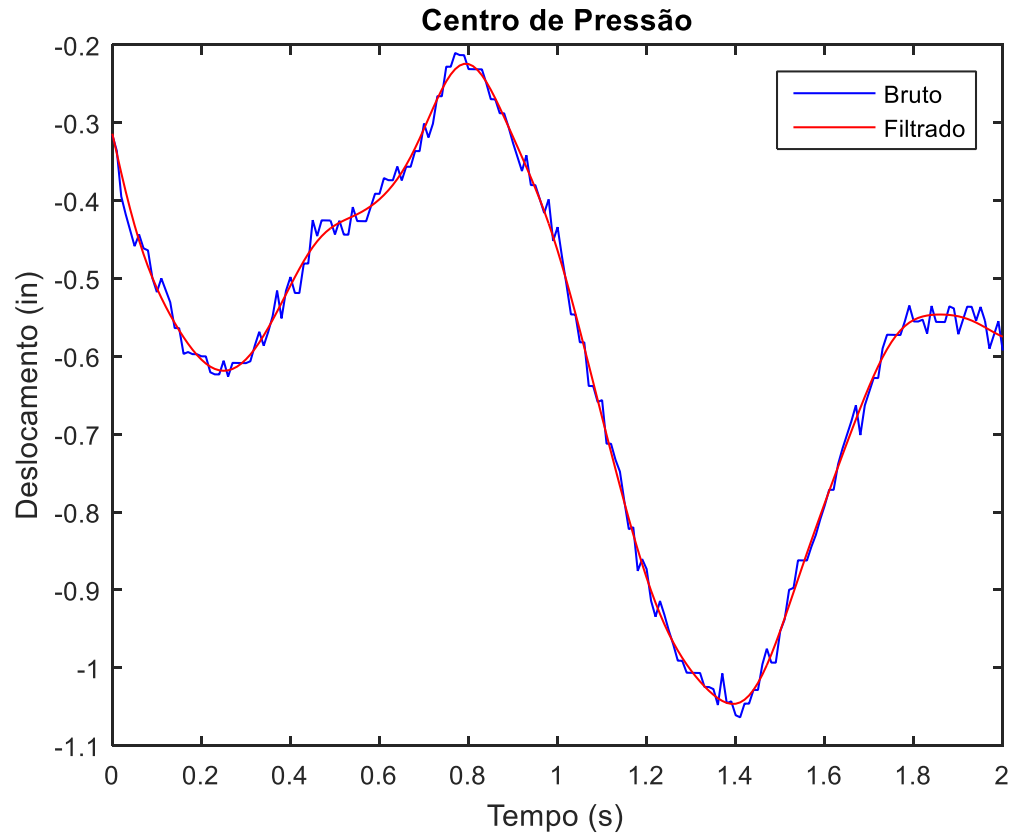
- butter.m
  - Determina os coeficientes do filtro
- filtfilt.m
  - Filtra o sinal com base nos coeficientes obtidos na função butter.m.



# Implementando um filtro passa baixa

```
n = 4; % ordem do filtro
Wn = 5/(freq/2); % Fc = 5 Hz
[b,a] = butter(n,Wn); % Coeficientes do filtro
data = filtfilt(b,a,data);
```

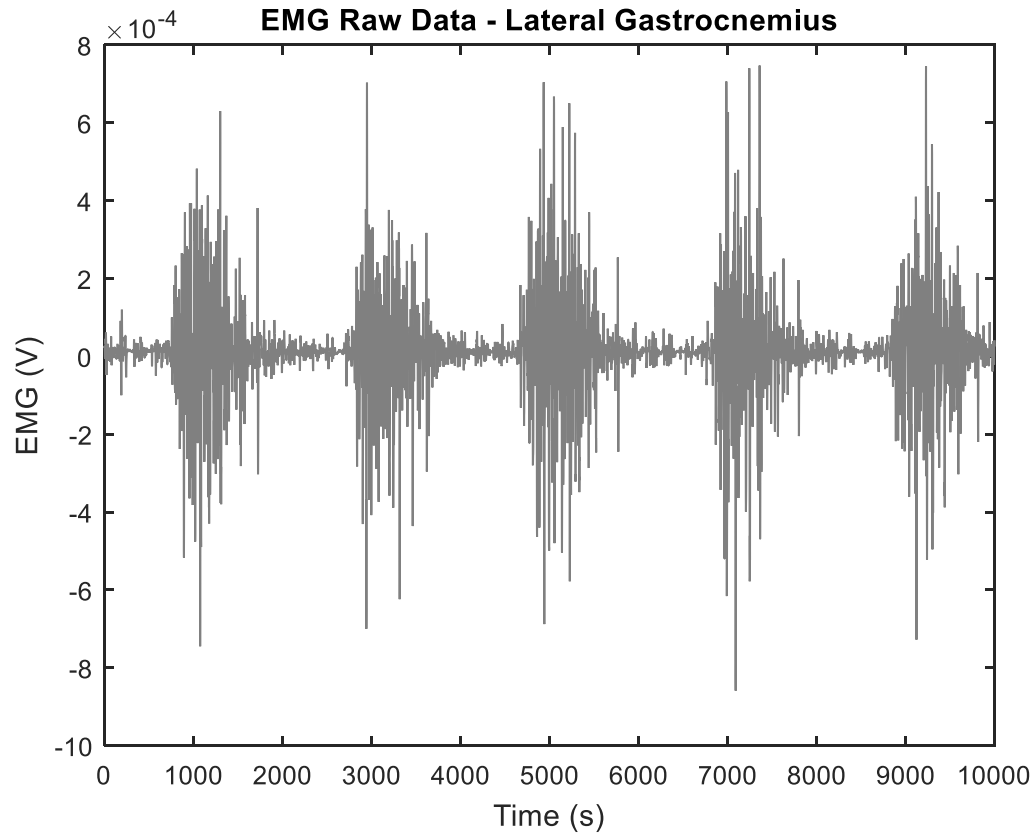
# cop.txt



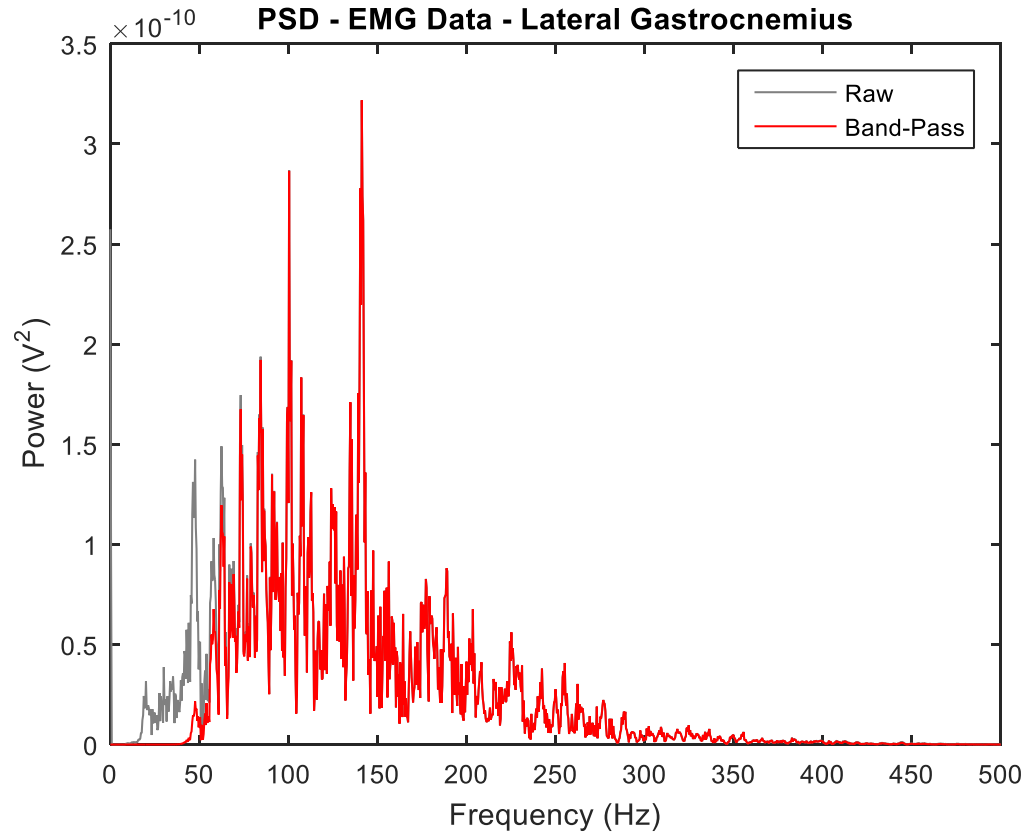
# Implementando um filtro passa banda

```
n = 4; % ordem do filtro
Wn = [50 500] / (freq/2);
[b,a] = butter(n,Wn,'bandpass');
EMG = filtfilt(b,a,data);
```

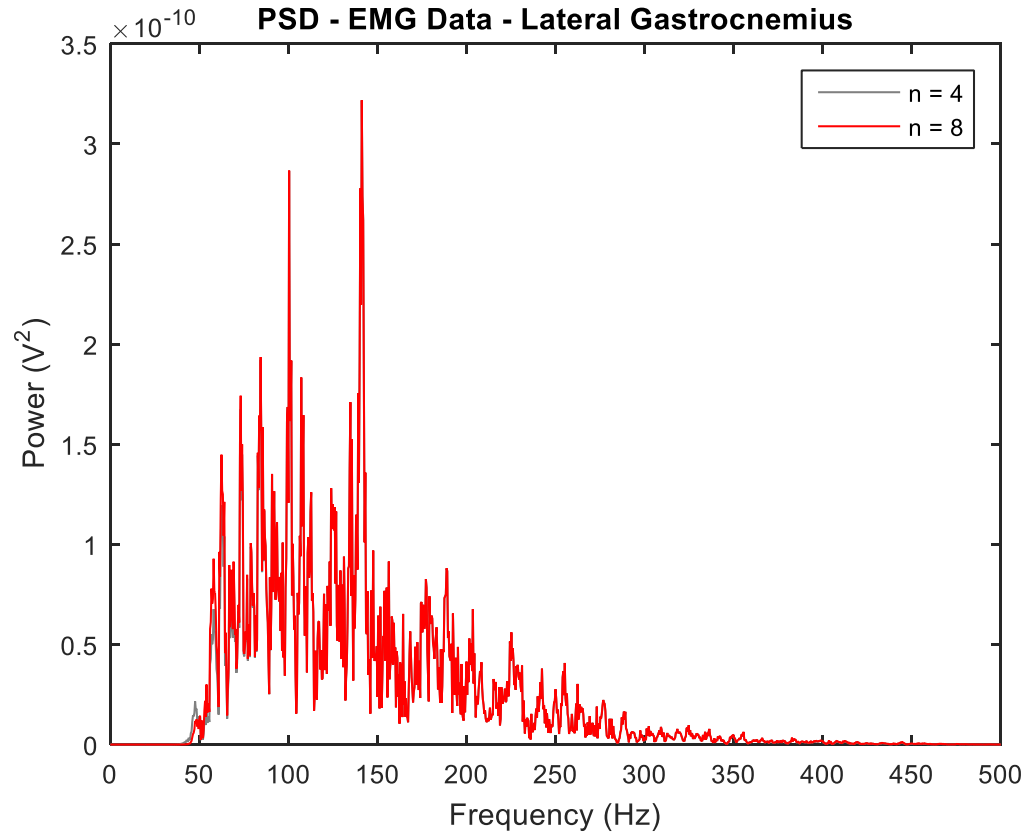
# EMG.txt



# Filtro Passa-Banda

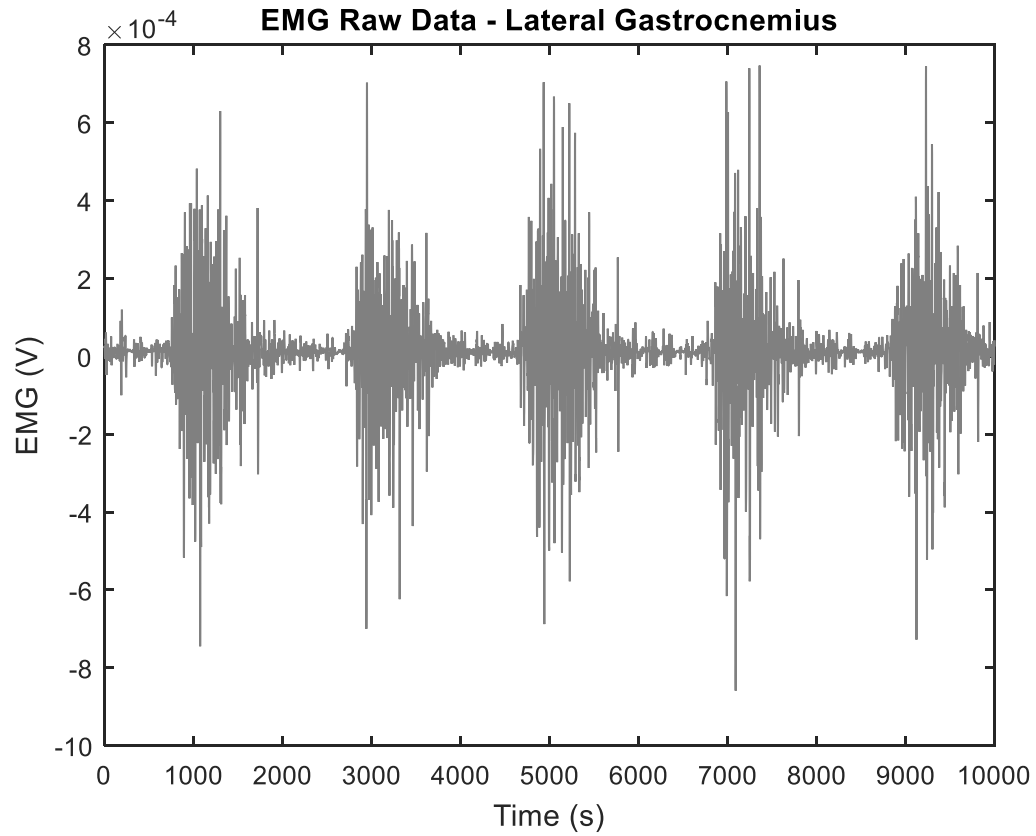


# Ordem do filtro



Envelope linear - EMG

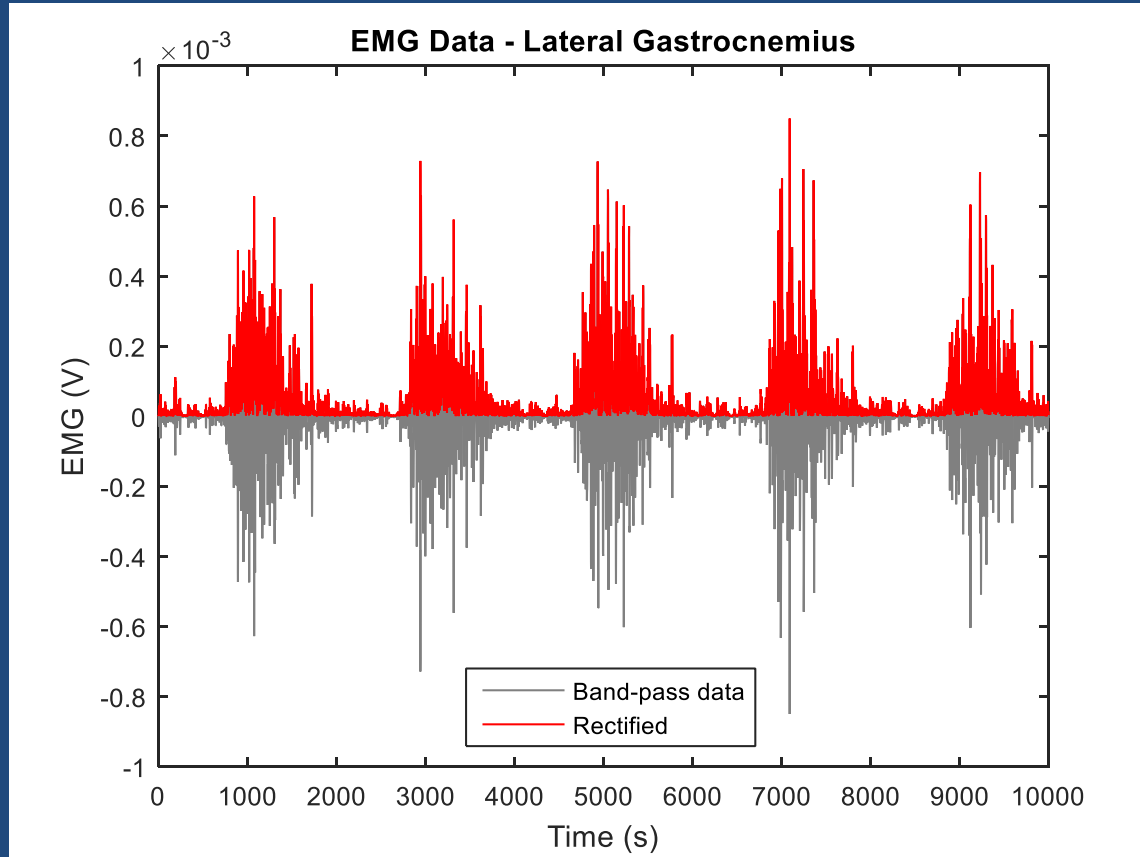
# EMG.txt





# Envelope Linear – Passo 1

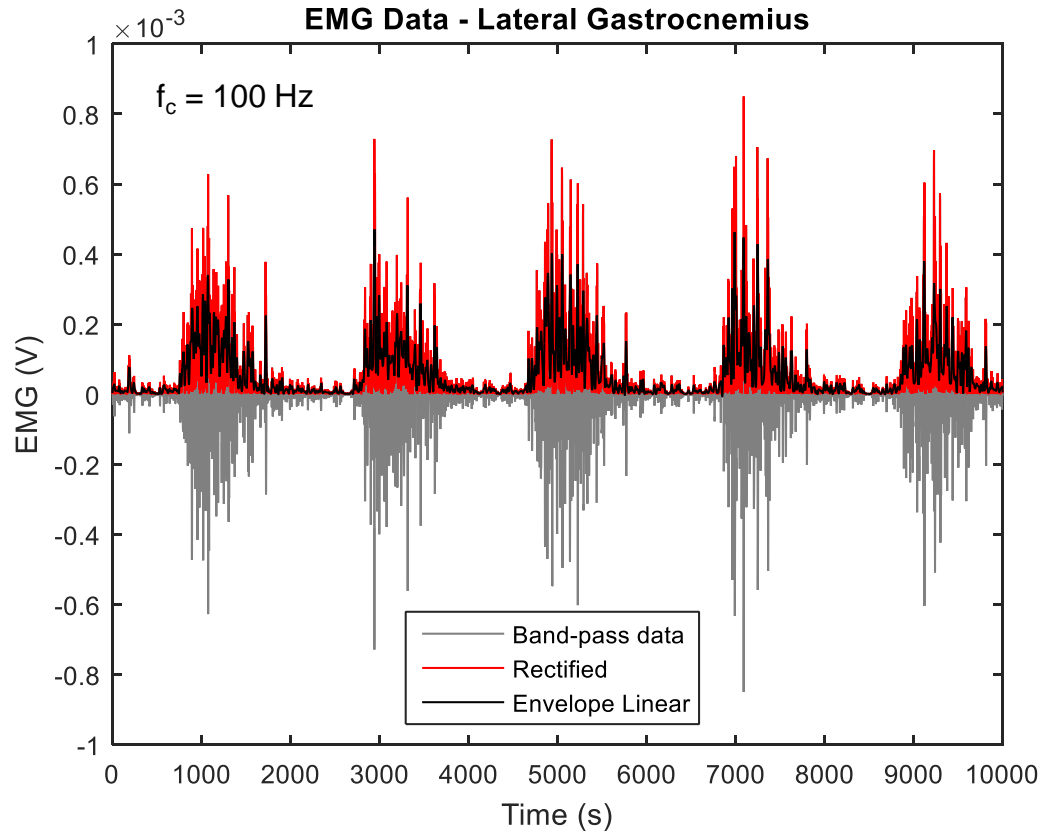
$$\text{EMG} = \text{abs}(\text{EMG})$$



# Envelope Linear – Passo 2

```
% Filtro Passa Baixa  
n = 4; % ordem do filtro  
Wn = 100/(freq/2); % Fc = 100 Hz  
[b,a] = butter(n,Wn,'low'); % Coeficientes do filtro  
data = filtfilt(b,a,data);
```

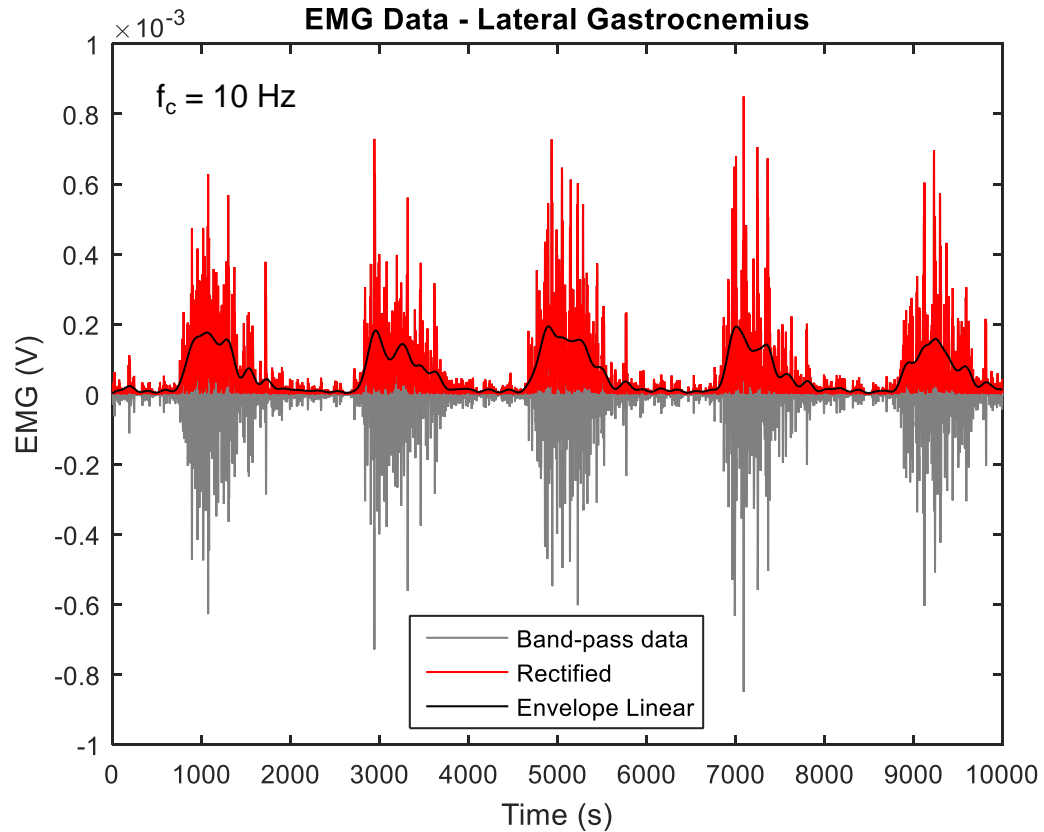
# Envelope Linear – Passo 2



# Envelope Linear – Passo 2

```
% Filtro Passa Baixa
n = 4; % ordem do filtro
Wn = 10/(freq/2); % Fc = 10 Hz
[b,a] = butter(n,Wn,'low'); % Coeficientes do filtro
data = filtfilt(b,a,data);
```

# Envelope Linear – Passo 2



# Integral EMG (iEMG)

# iEMG

```
time = linspace(1/freq,length(data)/freq,length(data));  
iEMG = cumtrapz(time,data);  
iEMG = iEMG(end)
```