

Trabalho Computacional 1

Linguagem utilizada: Python

Questão 1. Análise espectral do ECG

- (a) O sinal foi carregado a partir do arquivo *signal.txt*. A frequência de amostragem $f_s = 500\text{Hz}$ foi utilizada para calcular o tamanho do passo temporal: $dt = 1/f_s = 0.002$ segundos. A partir de dt e do número de amostras $N = 1000$, foi criado um vetor tempo, indo de 0.000 a 19.998 segundos. O sinal no tempo pode ser visto na figura 1, abaixo.

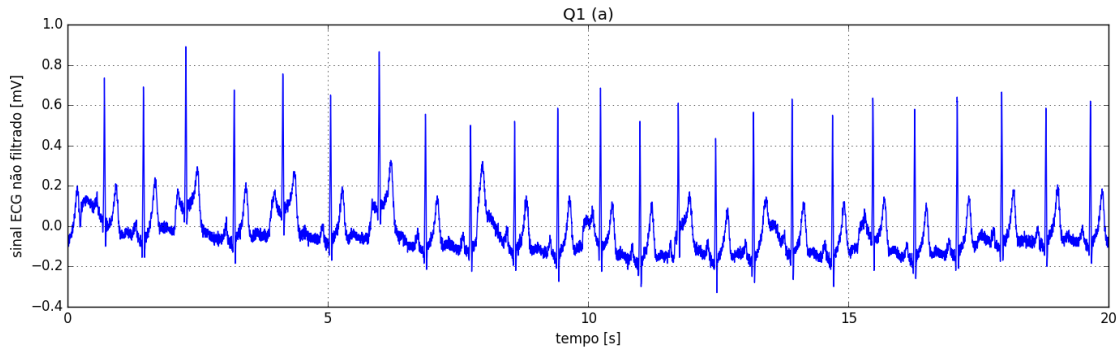


Figura 1: Sinal de ECG em função do tempo.

- (b) Foi utilizado o comando `np.fft.rfft` para obter a FFT do sinal (figura 2). Outra maneira seria utilizar a FFT complexa (`np.fft.fft`) e utilizar apenas a parte real da resposta. O comando `np.fft.rfftfreq` fornece as frequências de amostragem, porém em unidade adimensional. Para calibrar o eixo das abscissas em Hertz foi calculada a resolução espectral como o inverso do tempo de amostragem, isto é, $df = 1/(dt * N) = 0.05\text{Hz}$, e utilizado o número de amostras N .

$$f_{Hz} = f_{adim} * df * N = \frac{f_{adim}}{dt * N} * N = \frac{f_{adim}}{dt}$$

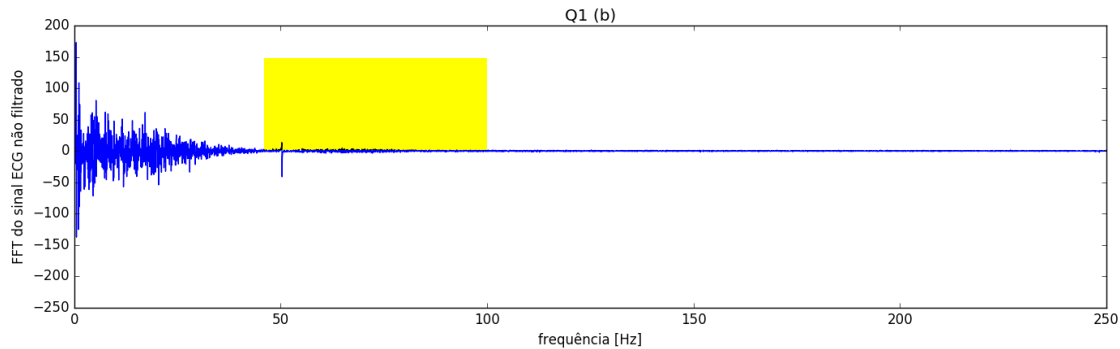


Figura 2: Espectro do sinal de ECG.

- (c) O espectro possui componentes próximas a 50Hz, possivelmente devido à interferência da rede elétrica do local. São observadas, também, componentes em baixas frequências que podem estar relacionadas à respiração ou outros movimentos do indivíduo durante a aquisição do sinal e componentes em alta frequência, que podem estar relacionadas à detecção de sinais de EMG e ruídos causados pelos fios e pela interface entre a pele e os eletrodos.

Questão 2. Filtragem digital

Para criar os filtros foram utilizadas funções da biblioteca `scipy.signal`.

- (a) Filtro passa-baixas.

(a.1)(a.2) O filtro passa-baixas foi projetado com as seguintes restrições: atenuação em 50Hz de 40dB (ou seja, faixa de rejeição iniciando em 50Hz), ordem máxima do filtro igual a 10 (para evitar instabilidade do filtro) e atenuação máxima de 3dB na faixa de passagem. Com estas restrições a faixa de passagem foi um resultado do projeto do filtro. Iterativamente, foi utilizada a função `buttord` fixando-se a frequência da faixa de rejeição (f_{stop}) em 50Hz, a atenuação na faixa de passagem em 3dB e a atenuação na faixa de rejeição em 40dB. Desta maneira a frequência de corte máxima para obter um filtro de ordem máxima igual a 10 foi de $f_{pass} = 32Hz$.

Então, foi utilizada a função `butter` para gerar o filtro ButterWorth passa-baixas de ordem 10 e frequência de corte de 32Hz.

- (a.3) Função de transferência do filtro passa-baixas:

$$\frac{Y(z)}{X(z)} = H(z) = \frac{\sum_{k=0}^{10} b_k z^{-k}}{\sum_{l=1}^{10} a_l z^{-l}}$$

Com os valores dos coeficientes b_k e a_l abaixo:

b_0	3.39e-08	a_0	1.00
b_1	3.39e-07	a_1	-7.43
b_2	1.53e-06	a_2	25.10
b_3	4.07e-06	a_3	-50.73
b_4	7.12e-06	a_4	67.86
b_5	8.54e-06	a_5	-62.73
b_6	7.12e-06	a_6	40.57
b_7	4.07e-06	a_7	-18.12
b_8	1.53e-06	a_8	5.34
b_9	3.39e-07	a_9	-0.94

Abaixo, a resposta em frequência do filtro:

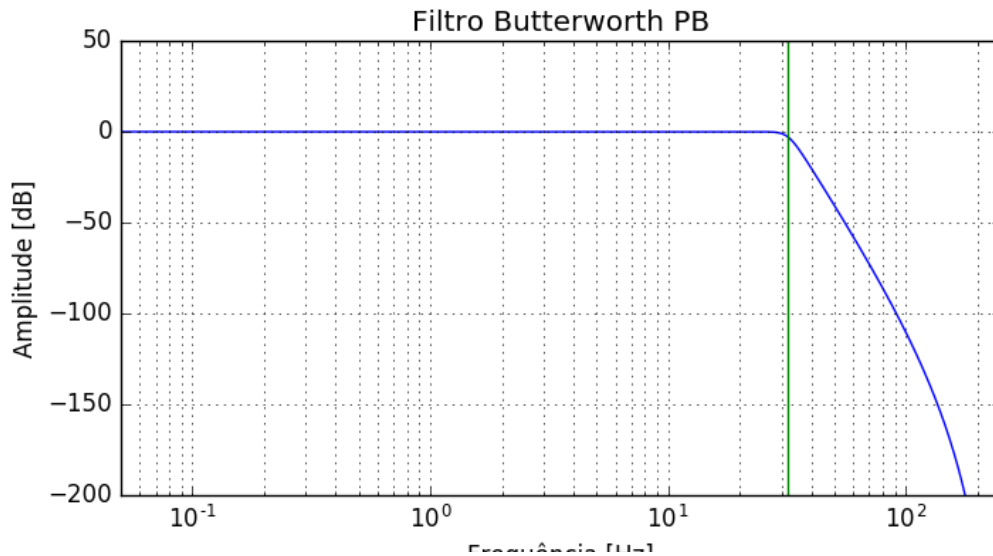


Figura 3: Resposta em frequência do filtro passa-baixas.

Abaixo estão apresentados o espectro do sinal após o filtro passa-baixas (figura 4) e um comparativo do sinal antes e depois do filtro (figura 5), mostrando a atenuação das oscilações próximas a 50Hz.

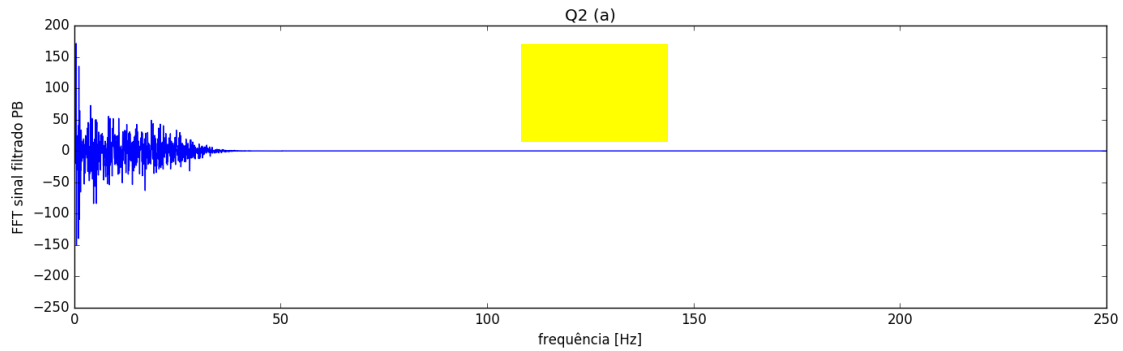


Figura 4: Espectro do sinal de ECG após filtro passa-baixas (freq. corte = 32Hz).

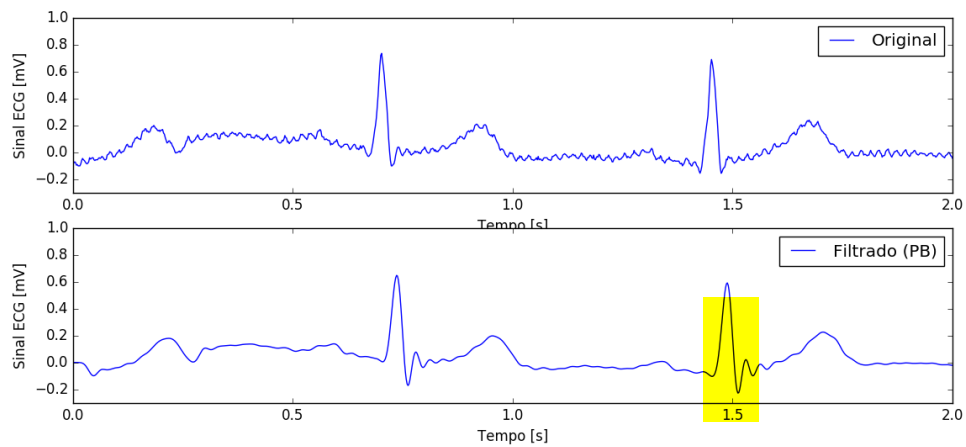


Figura 5: Sinal ECG antes e depois do filtro passa-baixas (freq. corte = 32Hz).

(a.4) Caso a frequência de corte escolhida para o filtro seja muito grande, as componentes perto de $50Hz$ serão menos atenuadas e para frequências de corte ainda menores que $32Hz$, aumenta-se a quantidade de informação que está sendo atenuada mas que poderia ser desejada para caracterizar o sinal de ECG.

Abaixo, na figura 6, mostra-se o sinal filtrado em três frequências de corte: $22Hz$, $32Hz$ (do filtro original) e $42Hz$.

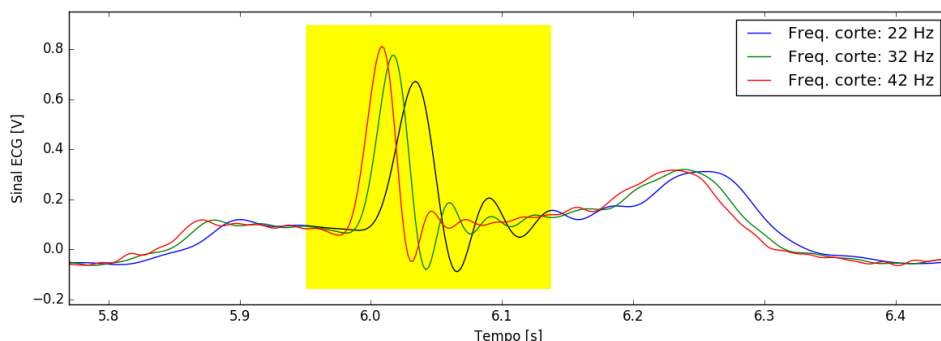


Figura 6: Sinais filtrado em três frequências de corte distintas.

(b) Filtro passa-altas.

(b.1)(b.2) Para o filtro passa-altas foi escolhida a frequência de corte de $0,5Hz$ para filtrar as flutuações da linha de base que possam estar relacionadas a respiração ou movimentações do sujeito durante a aquisição do sinal. O filtro foi projetado com atenuação mínima de $40dB$ na faixa de rejeição e de no máximo $3dB$ na faixa de passagem (acima de $0,5Hz$). A ordem do filtro foi definida como 4, pois, com ordens a partir de 5 o filtro apresentava descontinuidades na resposta em frequência abaixo de $0,1Hz$.

Abaixo, a resposta em frequência do filtro passa-altas.

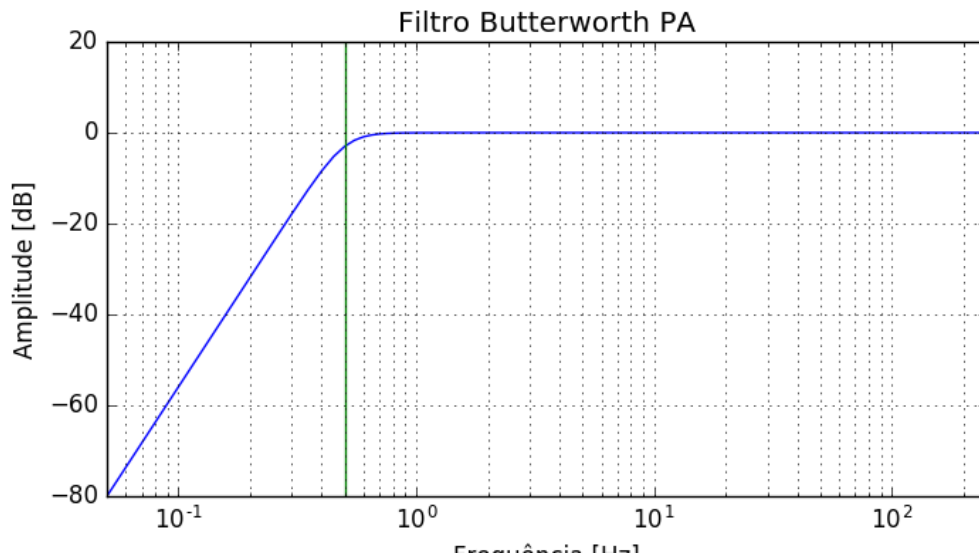


Figura 7: Resposta em frequência do filtro passa-altas (freq. corte = 0.5Hz).

(b.3) Função de transferência do filtro passa-altas:

$$\frac{Y(z)}{X(z)} = H(z) = \frac{\sum_{k=0}^4 b_k z^{-k}}{\sum_{l=1}^4 a_l z^{-l}}$$

Com os valores dos coeficientes b_k e a_l abaixo:

b_0	0.9918	a_0	1
b_1	-3.9672	a_1	-3.9835
b_2	5.9509	a_2	5.9508
b_3	-3.9672	a_3	-3.9510
b_4	0.9918	a_4	0.9837

Nas figuras 8 e 9, abaixo, estão apresentados o espectro do sinal após o filtro passa-altas e um comparativo do sinal antes e depois do filtro. É possível observar a atenuação em baixíssimas frequências e o ajuste das flutuações na linha-base.

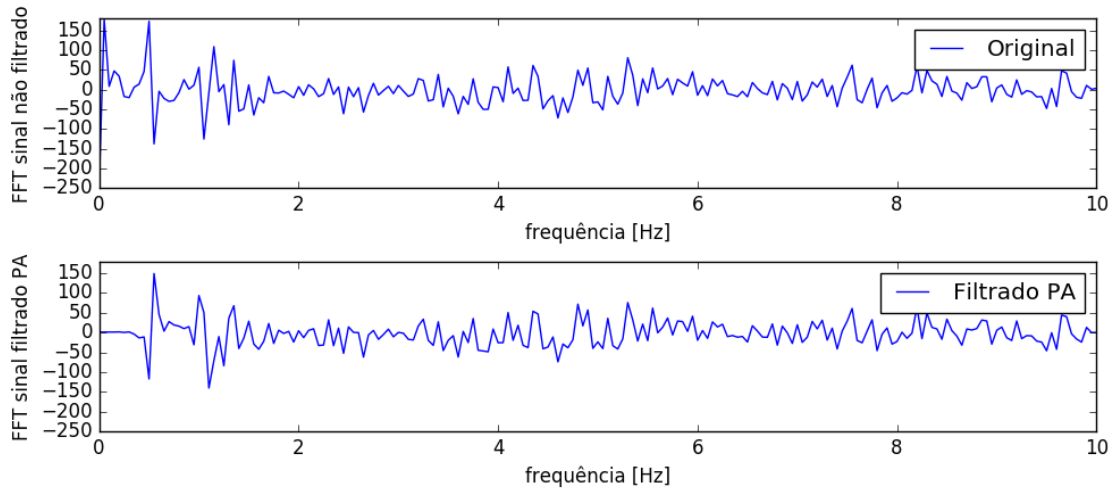


Figura 8: Espectro do sinal de ECG após filtro passa-altas (freq. corte = 0.5Hz).

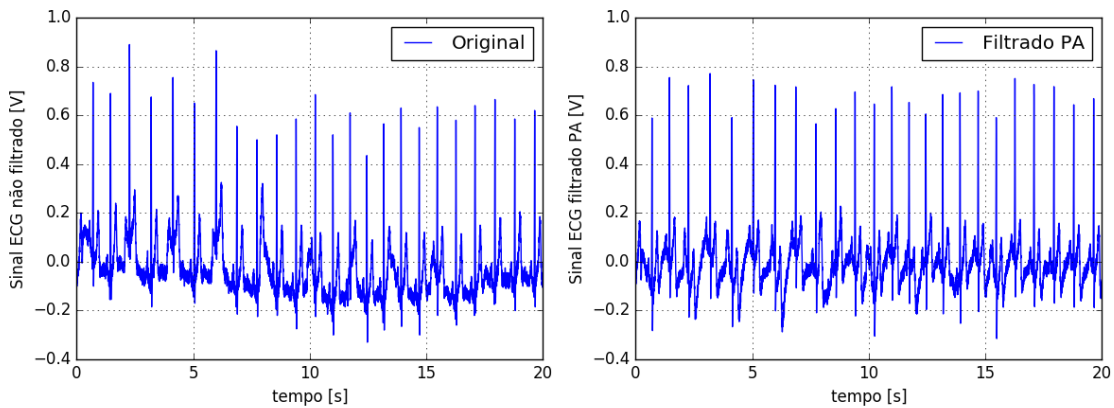


Figura 9: Comparativo do sinal ECG antes e depois do filtro passa-altas (freq. corte = 0.5Hz).

(b.4) Caso a frequência de corte fosse menor, parte da flutuação da linha de base não seria removida, porém, caso a frequência de corte fosse mais que 0.5, haveria uma distorção ainda maior no espectro de baixas frequências, podendo haver perda significativa de informação.

Abaixo, mostra-se o sinal filtrado em três frequências de corte: 0.1Hz , 0.5Hz (do filtro original) e 5.0Hz . Observa-se que um filtro com frequência de corte muito baixa como 0.1Hz ainda deixa presente algumas flutuações de baixa frequência. O sinal filtrado com passa-altas a 5Hz , por sua vez, causa uma deformação no sinal, o que não é observado no sinal filtrado a 0.5Hz .

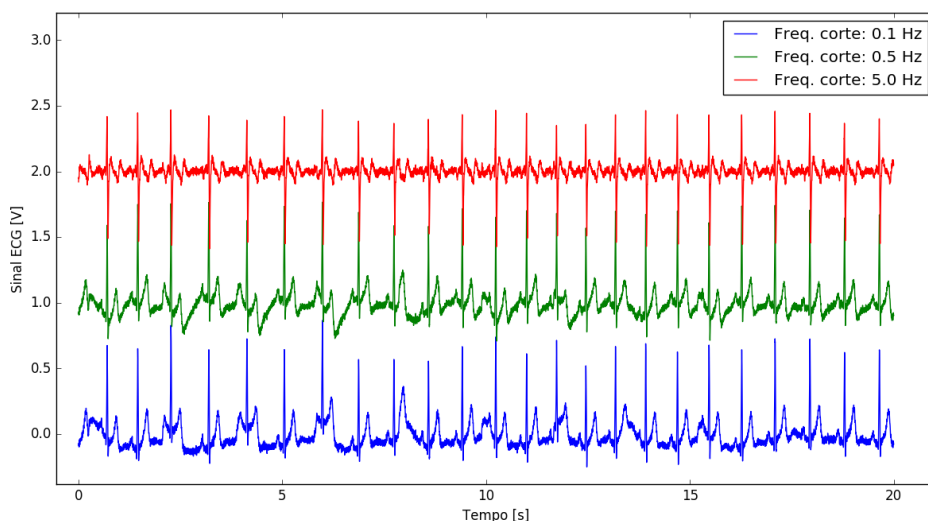


Figura 10: Sinais filtrado em três frequências de corte distintas. Os sinais filtrados a 0.5Hz (linha verde) e a 0.1Hz (linha vermelha) estão deslocados em 1V e 2V , respectivamente, para melhor observação de seus efeitos.

(c) Filtro Notch.

Para implementar o filtro Notch foi utilizada a função `iirnotch`, também da biblioteca `scipy.signal`.

Para utilizar esta função entra-se com a frequência de filtragem e o fator de qualidade. Um favor de qualidade de 25 foi suficiente para ter atenuação de 40dB na frequência de 50.3Hz .

Abaixo, a resposta em frequência do filtro Notch projetado.

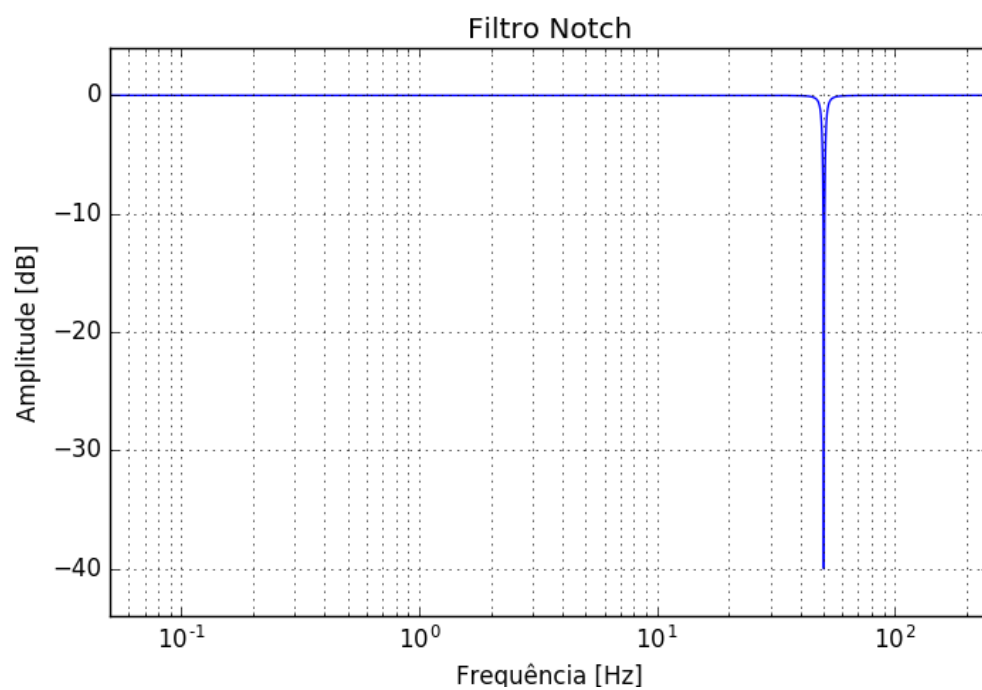


Figura 11: Resposta em frequência do filtro Notch.

Na figura 12, abaixo, é possível verificar o efeito do filtro Notch no sinal original.

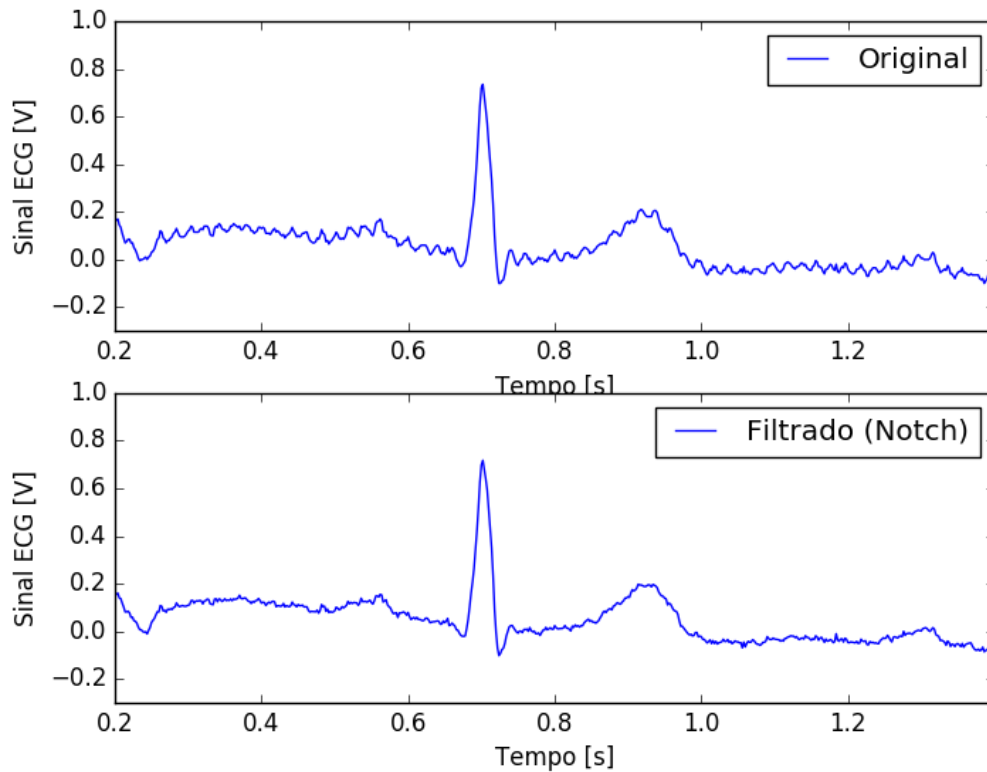


Figura 12: Sinal original comparado com o sinal após filtragem do tipo Notch.

(d) Combinação de filtros.

Foi decidido manter o filtro passa-altas para remover as flutuações em baixa frequência. Para eliminar as componentes excessivas próximas de 50Hz sem perder informações do sinal em frequências altas, foi utilizado o filtro Notch em 50.3Hz. Então, foi incluído um filtro passa-baixas com faixa de transição entre 80Hz e 120Hz, para filtrar componentes de frequências muito altas.

Abaixo, nas figuras 13, 14 e 15 é possível observar o efeito do filtro sobre o sinal no tempo e na frequência. Houve diminuição da flutuação da linha de base, remoção das frequências próximas a 50Hz e atenuação de frequências altíssimas.

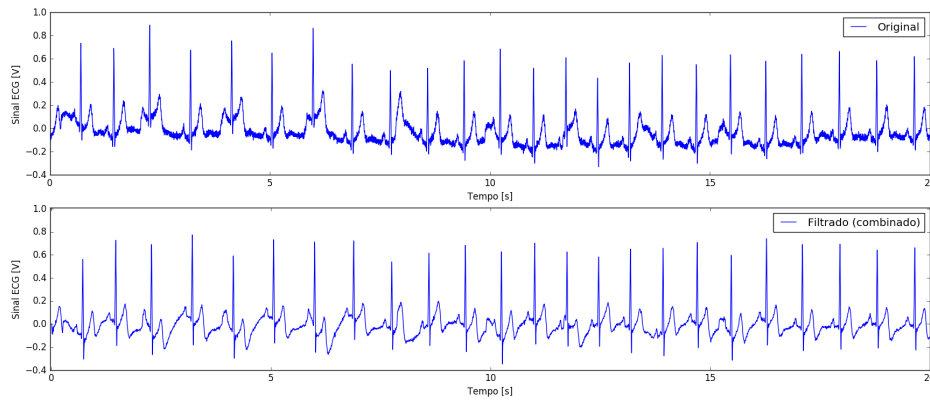


Figura 13: Sinal original comparado com o sinal após filtragem combinada.

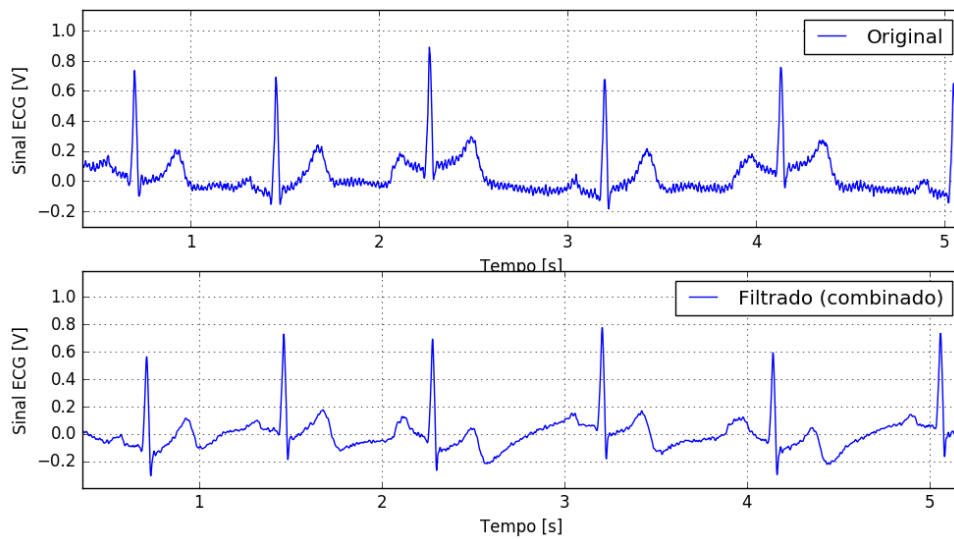


Figura 14: Sinal original comparado com o sinal após filtragem combinada, entre 1 e 5 segundos.

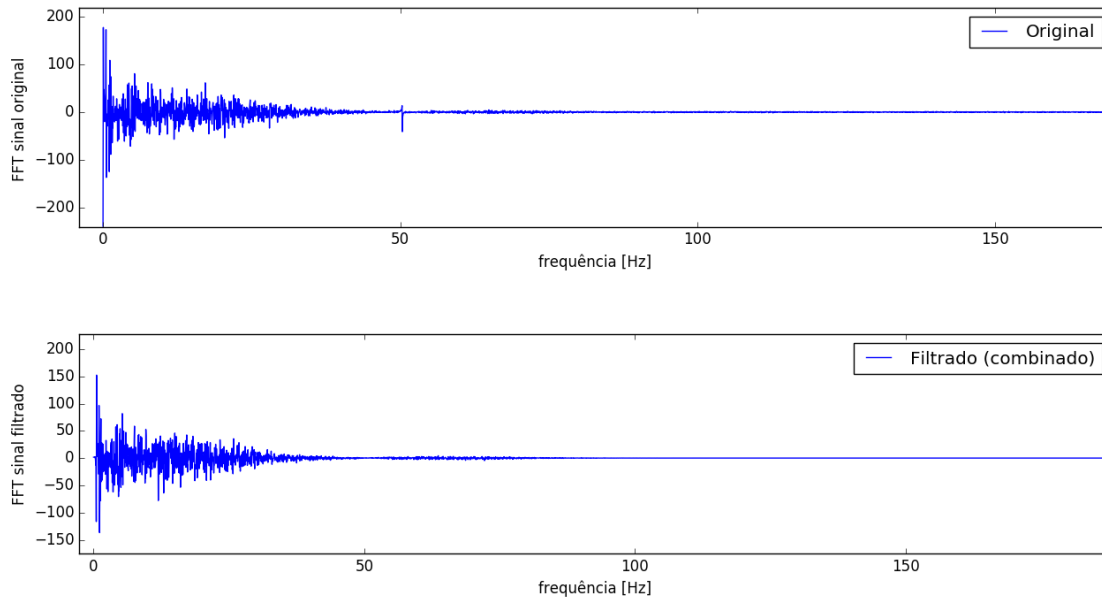


Figura 15: Espectro do sinal original comparado com o do sinal após filtragem combinada.

Questão 3. Estimar frequência instantânea.

- (a) Para identificar os picos do sinal de ECG filtrado foram verificados os pontos no sinal maiores que 0.3V que fossem, simultaneamente, maiores que o ponto imediatamente antes e maiores que o ponto imediatamente depois. Isto é:

$$\text{signal}[n] \text{ é um pico se } \text{signal}[n-1] < \text{signal}[n] > \text{signal}[n+1], n = 2, 3, \dots, N-1$$

Com N sendo o número de amostras do sinal. Abaixo, o sinal com os picos identificados.

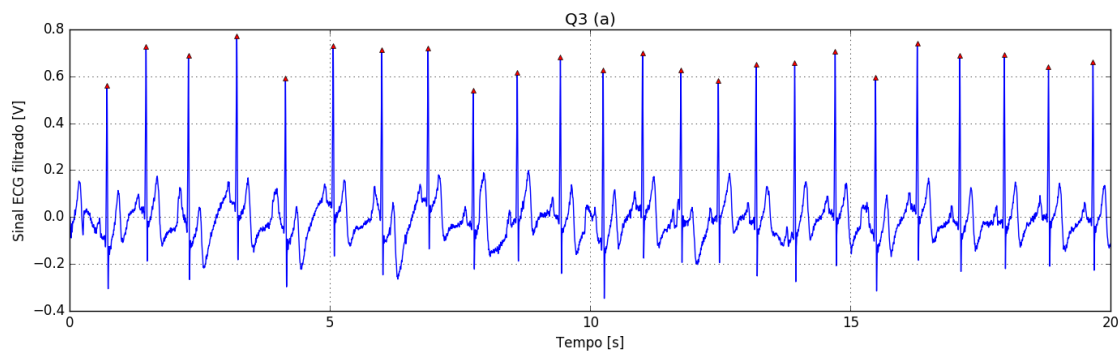


Figura 16: Sinal filtrado com os picos identificados (triângulos vermelhos).

- (b) O trem de impulsos foi criado a partir de um vetor de zeros de mesmo tamanho que o sinal para que a frequência de amostragem seja igual à do sinal. Então, nos instantes onde ocorrem os picos, os zeros foram substituídos pelo valor $1/dt$, representando o impulso discreto.

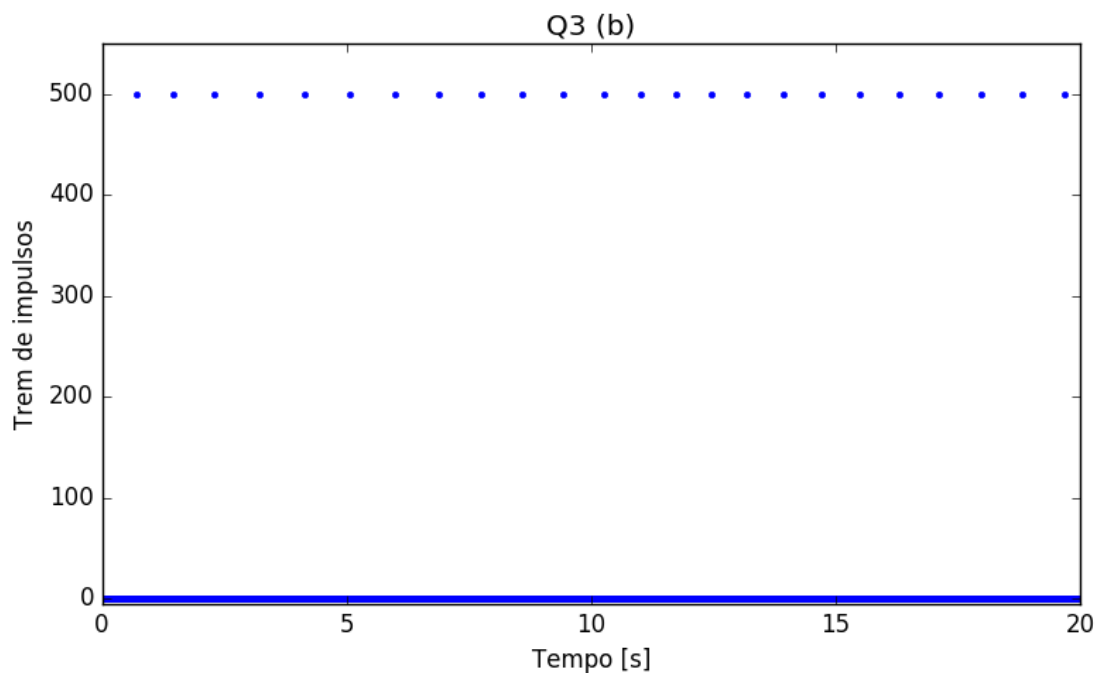


Figura 17: Trem de impulsos.

- (c) A janela foi criada como um vetor com w valores iguais a $1/w$. A convolução foi feita usando a função `np.convolve`.

Para uma janela muito pequena, a convolução irá mostrar pulsos nos instantes dos impulsos, como é o caso de $w = 100$, visto na figura 18. Por outro lado, caso a janela seja muito grande, como $w = 5000$, a convolução irá apresentar um perfil quase triangular, como pode ser visto na figura 19. No entanto, uma janela bem ajustada resulta em uma convolução que com poucos pontos passa a oscilar em torno do que seria a frequência média estimada, como pode ser visto na figura 20, com $w = 1200$.

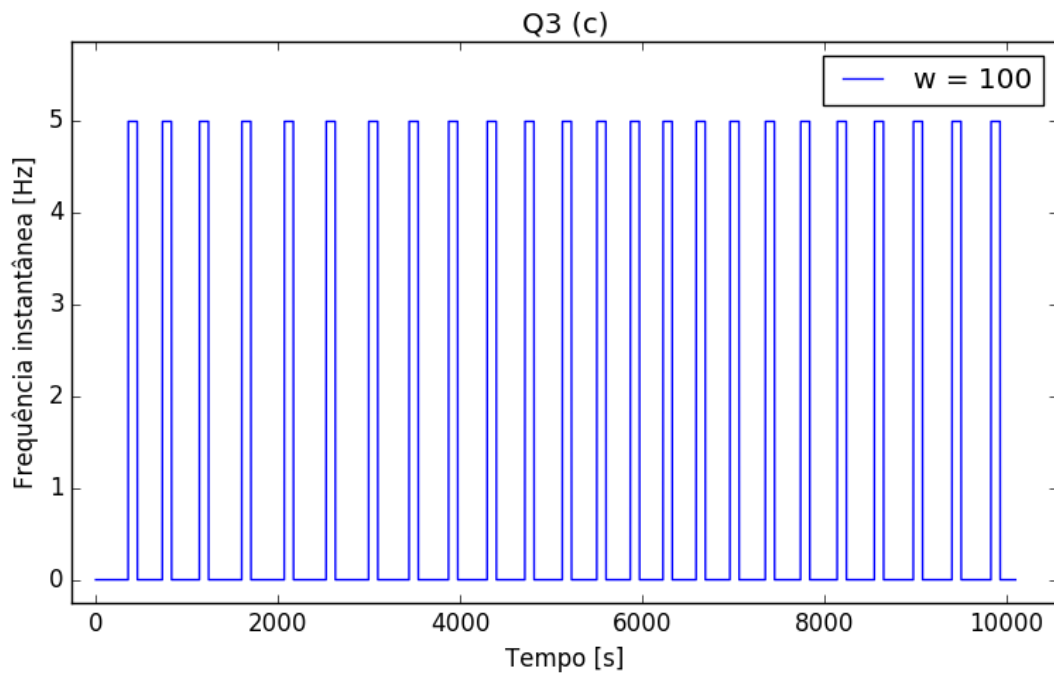


Figura 18: Estimação de frequência instantânea por convolução com janela muito pequena.

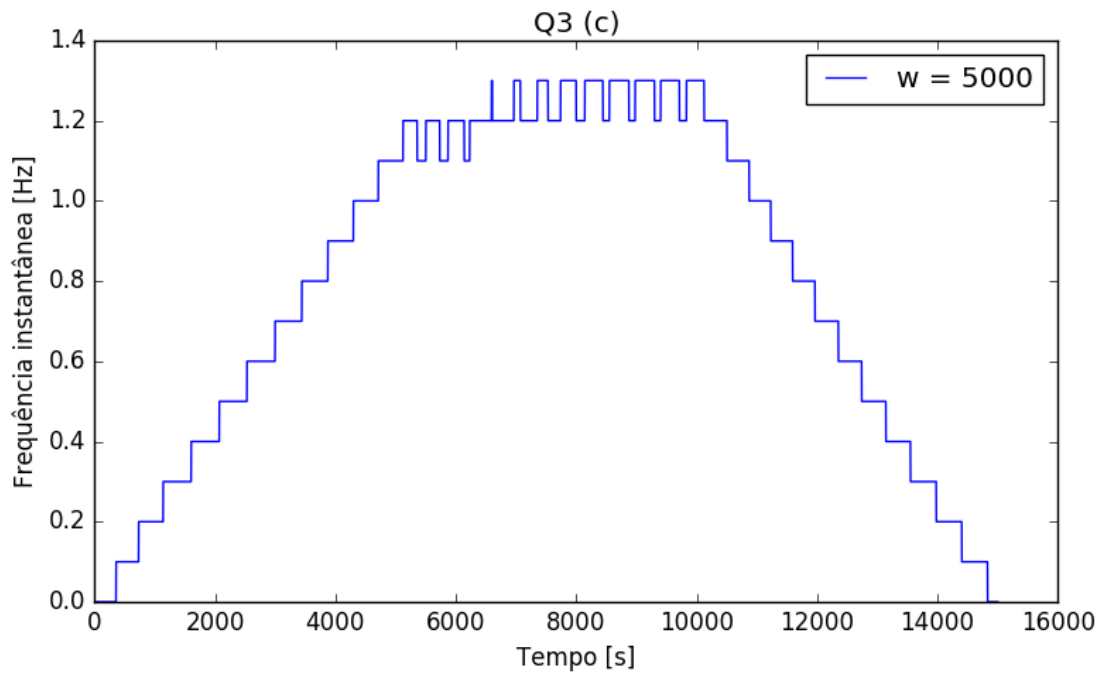


Figura 19: Estimação de frequência instantânea por convolução com janela muito grande.

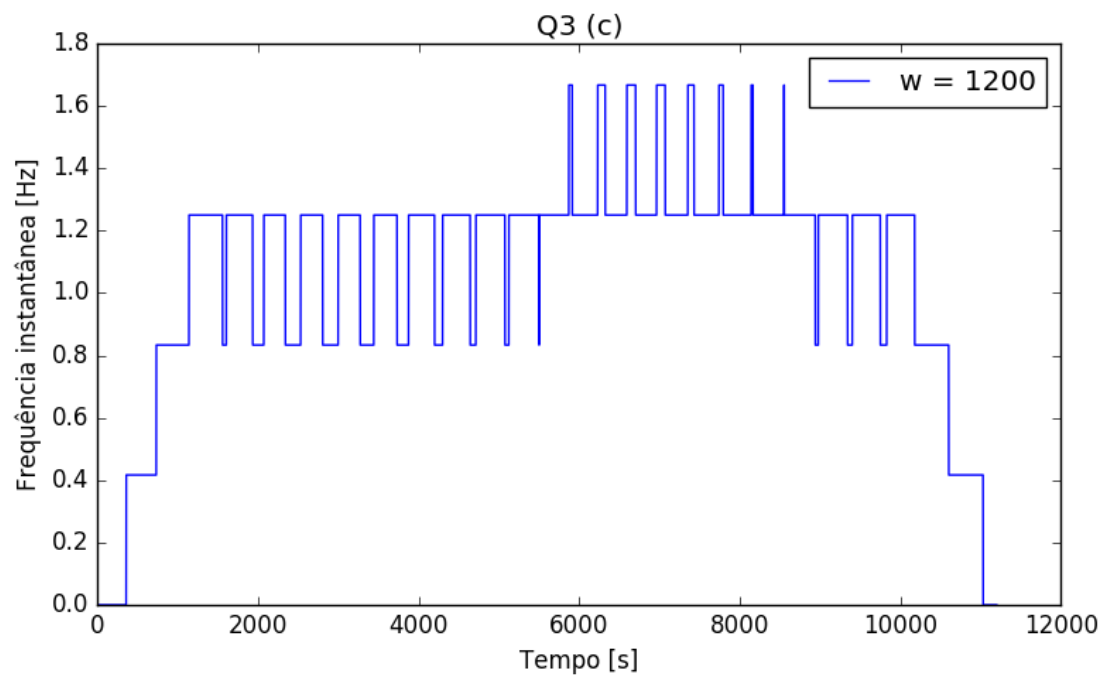


Figura 20: Estimação de frequência instantânea por convolução com janela adequada.