

FILTROS FIR E IIR

DOUGLAS FROTA BRANDÃO E KARINE ROSA FERREIRA PIRES

Resumo

Este projeto trata-se de projetos de filtros digitais FIR e IIR para remoção de ruídos.

Palavras-chave: eletrocardiograma, ruído, filtros, iir, fir

1. Introdução

Filtros são uma classe particularmente importante de sistemas LIT. Estritamente falando, o termo filtro seletivo em frequência sugere um sistema que passa certos componentes de frequência de um sinal de entrada e rejeita totalmente todos os outros. O projeto de filtros IIR implica a obtenção de uma função de transferência aproximada que seja uma função racional de z , enquanto o projeto de filtros FIR leva a uma aproximação polinomial. Neste projeto, projetaremos uma ampla gama de métodos tanto com filtros FIR quanto IIR e discutiremos as vantagens e desvantagens de cada método. (OPPENHEIM, SCHAFER; 2013)

2. Objetivo

Este trabalho tem por objetivo testar os conhecimentos de Filtros FIR e IIR aprendidos durante a disciplina de Processamento Digitais de Sinais e projetar diferentes filtros para remoção de ruído 60Hz em um Eletrocardiograma.

3. Fundamentação teórica

Filtro digitais:

$$Hd(z) = \frac{Yd(z)}{Xd(z)} = \frac{\sum_{k=0}^M b_k z^{-k}}{\sum_{k=0}^M a_k z^{-k}} \Rightarrow \sum_{k=0}^N a_k x[n-k]$$

Os filtros digitais são uns dos grandes campos de aplicação de processamento digital de sinal. Projetar um filtro digital é determinar um algoritmo que satisfaça as especificações pretendidas, e assim implementar este algoritmo seja em software ou em hardware, atendendo as implicações do uso de precisão infinita na representação dos coeficientes e dos sinais de entrada e de saída e na realização das operações aritméticas respectivas.

Filtros digitais do tipo FIR:

$$y(n) = \sum_{k=0}^{N-1} h(k).x[n-k]$$

Não têm contrapartida no mundo analógico, na medida em que implementam diretamente um sistema discreto a partir da sua resposta ao impulso $h(n)$, finita. Estes filtros podem ser realizados com fase exatamente linear, correspondente a um atraso fixo da saída relativamente à entrada, permitindo que no seu projeto a atenção do projetista se concentre na resposta em amplitude, e, além disso, são estáveis quando implementados não recursivamente. Os filtros digitais do tipo FIR são também muito usados na realização de sistemas adaptativos.

Filtros digitais do tipo IIR:

$$y(n) = \sum_{k=1}^{N-1} b_k \cdot y[n-k] - \sum_{k=1}^{N-1} a_k \cdot x[n-k]$$

Podem ser facilmente relacionados com filtros analógicos e são normalmente menos exigentes em termos do número de operações necessárias para realizar uma determinada resposta em frequência. Contudo, há situações como, por exemplo, a dizimação e a interpolação em que esta vantagem é apenas aparente. Os parâmetros do projeto de um filtro digital são normalmente os limites das bandas de passagem e de rejeição e o ripple máximo admitido em cada uma destas bandas.

Para o desenvolvimento dos filtros propostos, serão utilizados os métodos:

Butterworth

O Butterworth é o único filtro que mantém o mesmo formato para ordens mais elevadas (porém com uma inclinação mais íngreme na banda atenuada) enquanto outras variedades de filtros (Bessel, Chebyshev, elíptico) possuem formatos diferentes para ordens mais elevadas.

Comparado com um filtro Chebyshev do Tipo I/Tipo II ou com um filtro elíptico, o filtro Butterworth possui uma queda relativamente mais lenta, e portanto irá requerer uma ordem maior para implementar uma especificação de banda rejeitada particular. Entretanto, o filtro Butterworth apresentará uma resposta em fase mais linear na banda passante do que os filtros Chebyshev do Tipo I/Tipo II ou elípticos. Dada uma função de transferência, o filtro Butterworth pode ser implementado utilizando a forma Cauer - 1:

O elemento k é dado por:

$$C_k = 2 \sin \left[\frac{2k-1}{2n} \pi \right] \quad L_k = 2 \sin \left[\frac{2k-1}{2n} \pi \right]$$

Chebyshev Tipo I e Tipo II

Filtros Chebyshev são filtros analógicos ou digitais que possuem um aumento na atenuação (roll-off) mais íngreme e uma maior ondulação (ripple) na banda passante que os Filtros Butterworth. Os filtros Chebyshev possuem a propriedade de minimizarem o erro entre as características do filtro idealizado e o atual com relação à faixa do filtro, porém com ripples na banda passante. Este tipo de filtro recebeu seu nome em honra a Pafnuty Chebyshev, devido a suas características matemáticas serem derivadas dos polinômios de Chebyshev.

Tipo I

Estes são o tipo mais comum dos filtros Chebyshev. A sua característica da amplitude em frequência de ordem n pode ser descrita matematicamente como:

$$G_n(\omega) = |H_n(j\omega)| = \frac{1}{\sqrt{1 + \epsilon^2 T_n^2 \left(\frac{\omega}{\omega_0} \right)}}$$

Tipo II

Também conhecidos como Chebyshev invertidos, este tipo é menos comum pois ele não apresenta um roll off tão acentuado quanto o tipo I, e requer uma maior quantidade de componentes. Ele não possui ripple em sua banda passante, porém possui ripple na sua banda atenuada. Sua função de transferência é:

$$|H(\Omega)|^2 = \frac{1}{\sqrt{1 + \epsilon^2 T_n^2\left(\frac{\omega}{\omega_0}\right)}}$$

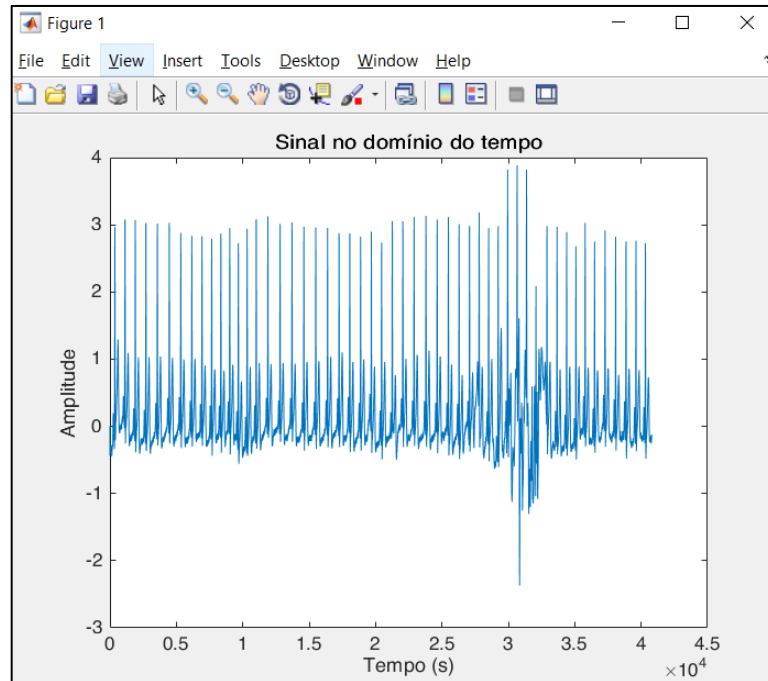
Elíptico

Um filtro elíptico (também conhecido como filtro de Cauer) é um filtro com ondulações (ripple) na banda passante e na banda rejeitada. Isto significa que ele minimiza o erro máximo em ambas as bandas, ao contrário do filtro Chebyshev, que apresenta ripple apenas na banda passante, ou no caso do Chebyshev inverso, na banda rejeitada. A magnitude da resposta em frequência de um filtro passa-baixas elíptico é dada por:

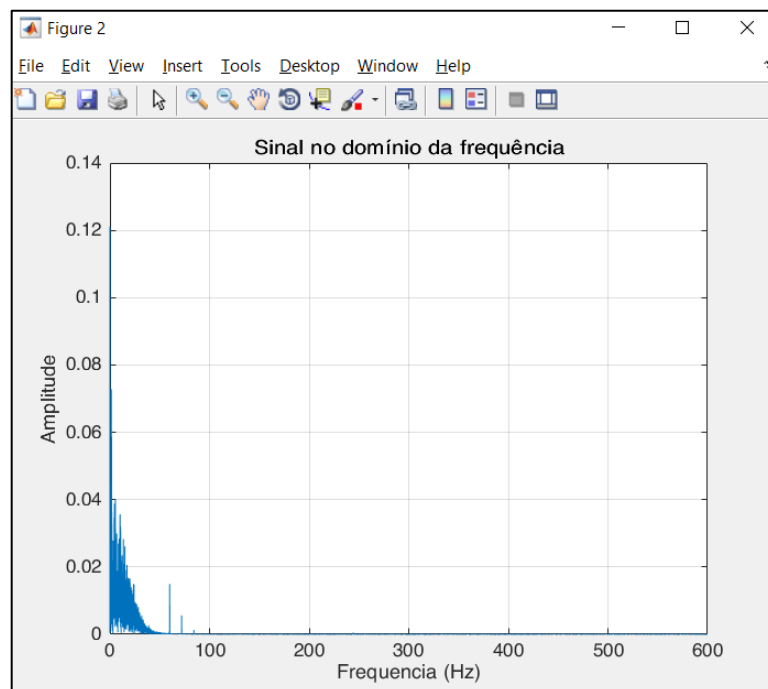
$$G_n(\omega) = |H_n(j\omega)| = \frac{1}{\sqrt{1 + \epsilon^2 R_n^2(\omega)}}$$

4. Resultados

Questão I – a) Plotar o sinal no domínio do tempo e o espectro de frequência do sinal de ECG. Considerar, neste exercício, a frequência de amostragem (F_s) igual a 1200 Hz.

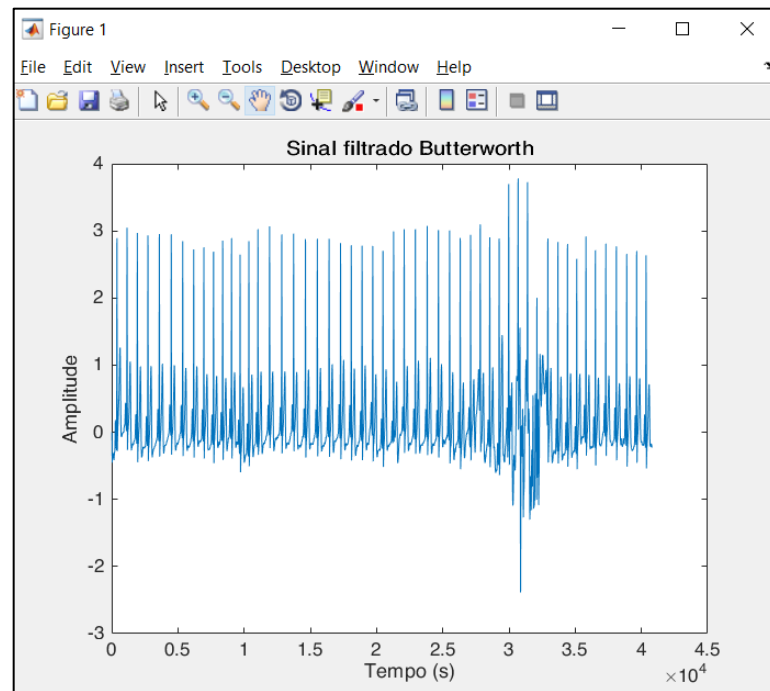


Sinal plotado no domínio do tempo - Figura 1

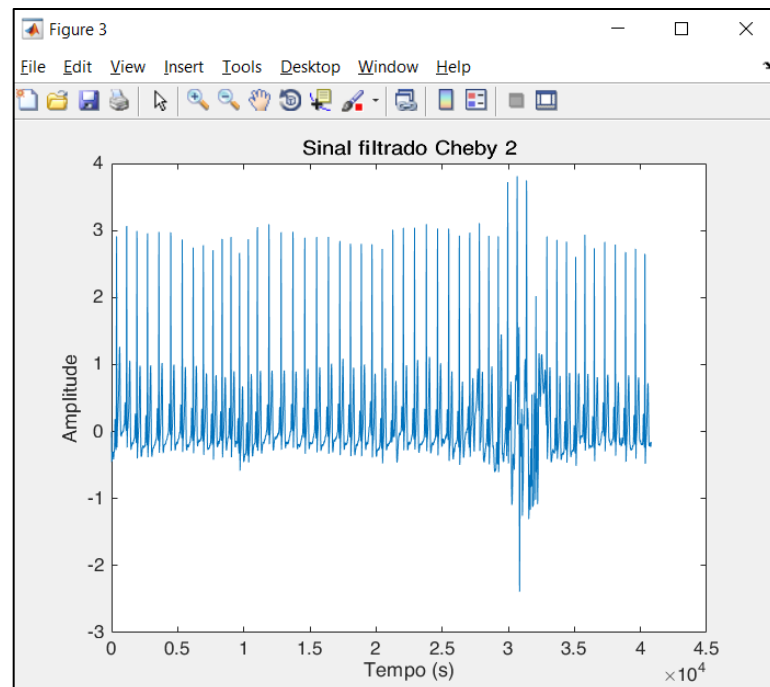


Sinal plotado no domínio da frequência - Figura 2

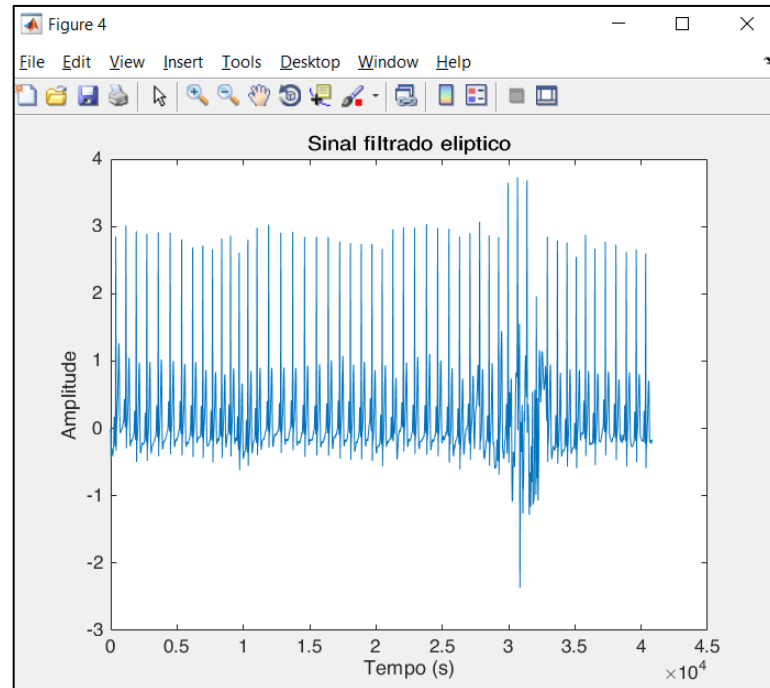
Questão I) – b) Projete um filtro IIR para remoção do ruído de 60Hz presente no sinal de ECG. Testar e comparar os diferentes métodos de design de filtros IIR: Butterworth, Chebyshev e Elíptico.



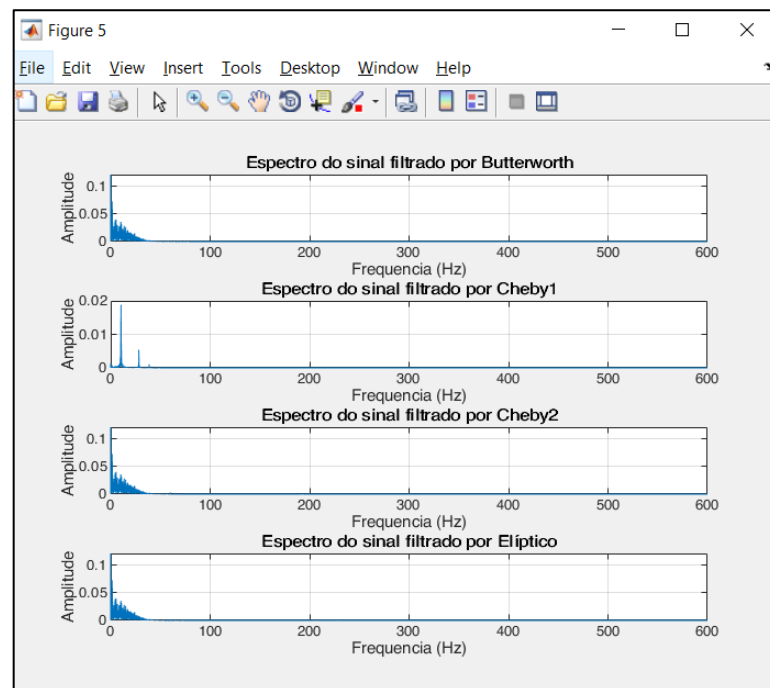
Sinal filtrado com Butterworth - Figura 3



Sinal filtrado com Cheby 2 - Figura 4

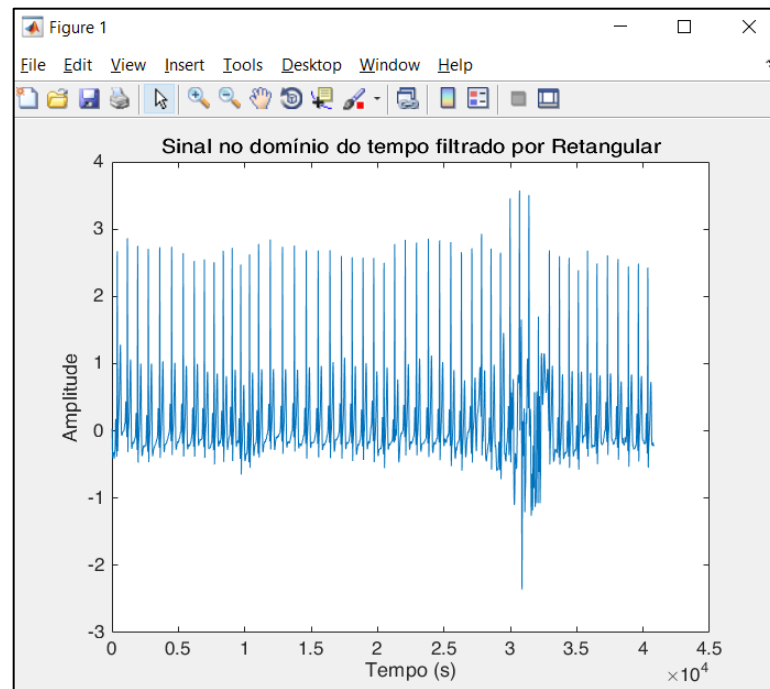


Sinal filtrado com Elíptico - Figura 5

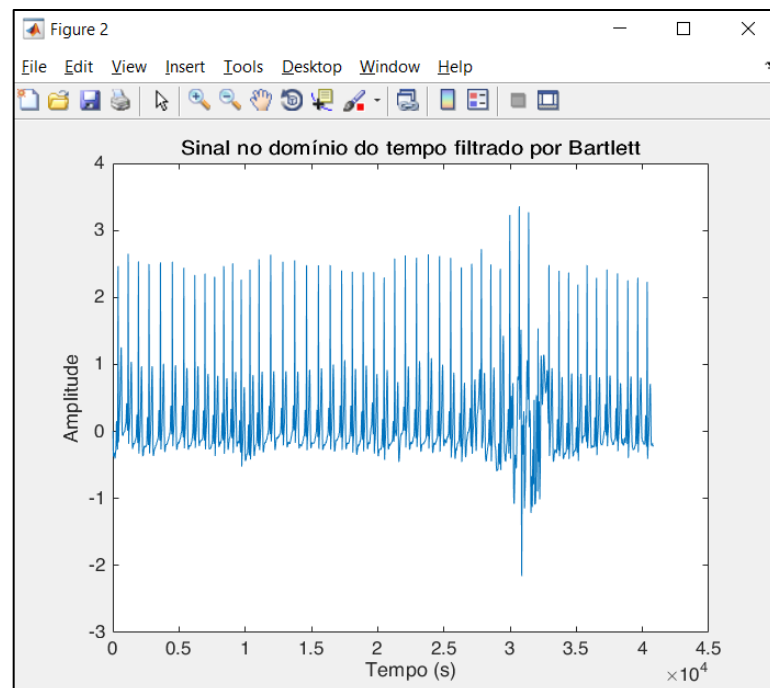


Espectros de frequência dos sinais filtrados - Figura 6

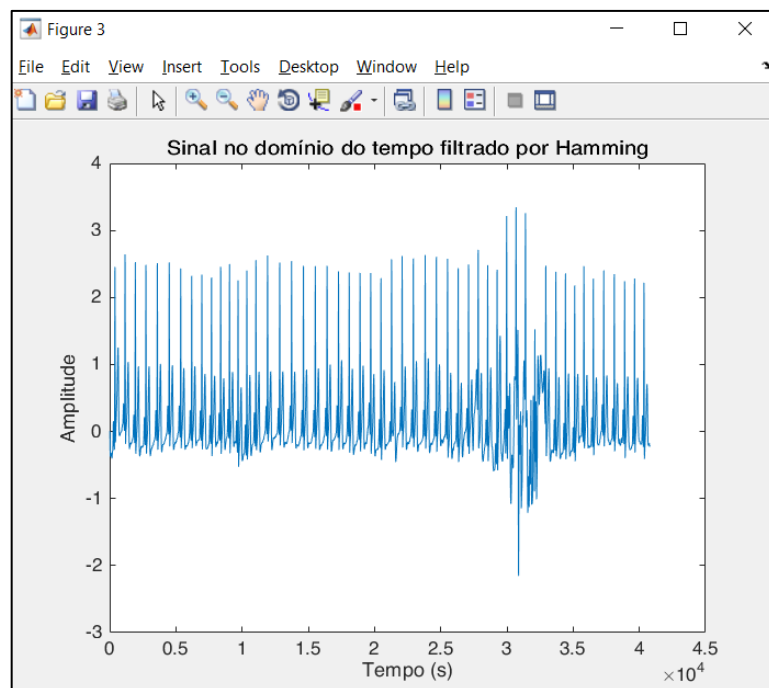
Questão I) – c) Projetar um filtro FIR para a remoção do ruído de 60Hz do sinal de ECG. Testar e comparar as diferentes janelas: Retangular, Bartlett, Hamming, Hann, Blackman e Kaiser.



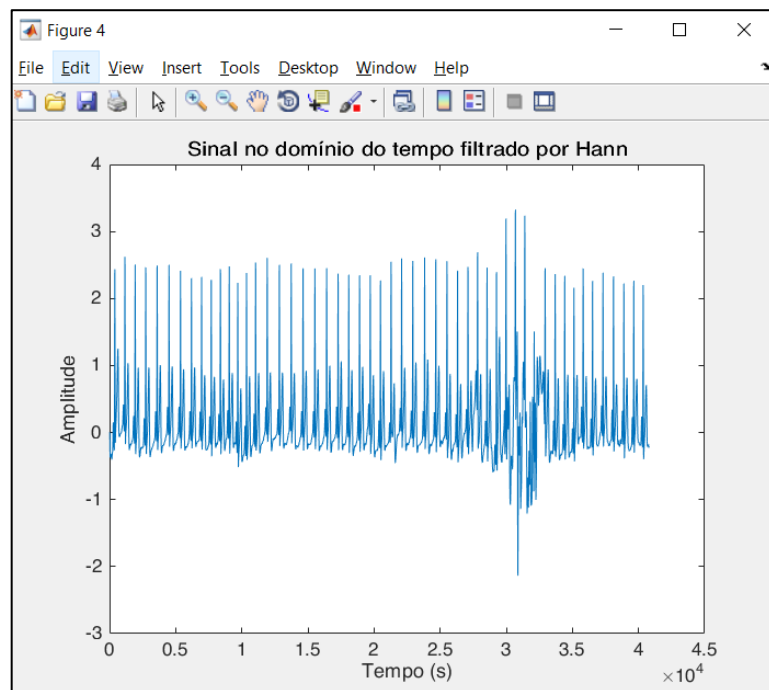
Sinal filtrado com Retangular - Figura 7



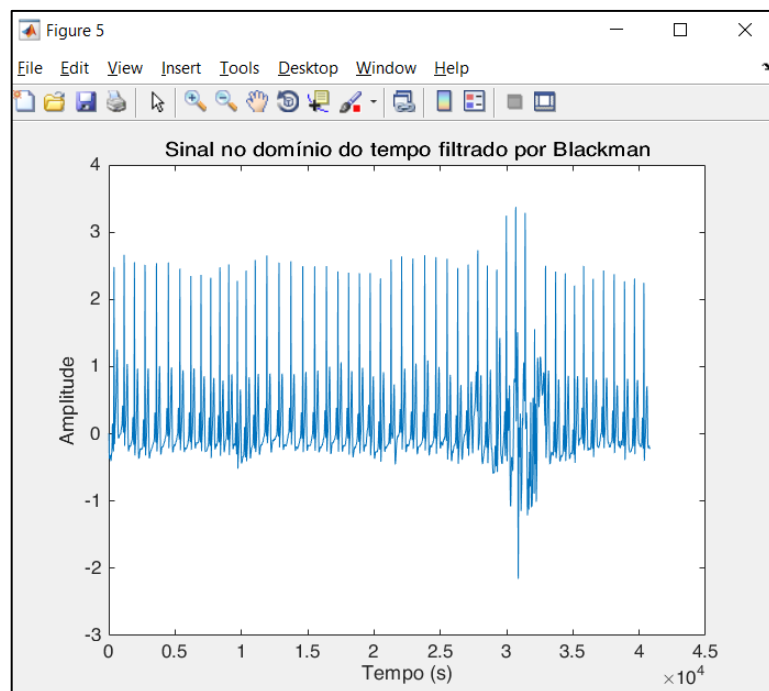
Sinal filtrado com Bartlett - Figura 8



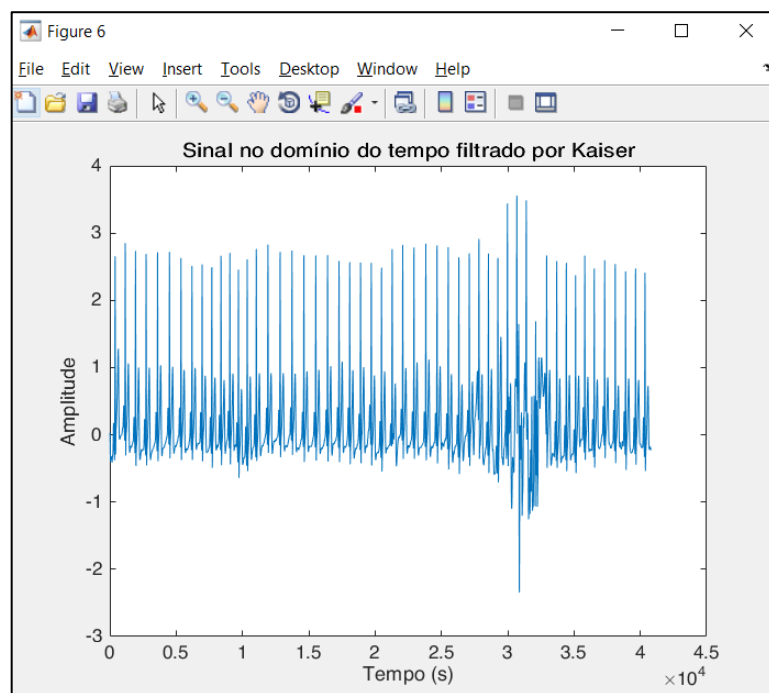
Sinal filtrado com Hamming - Figura 9



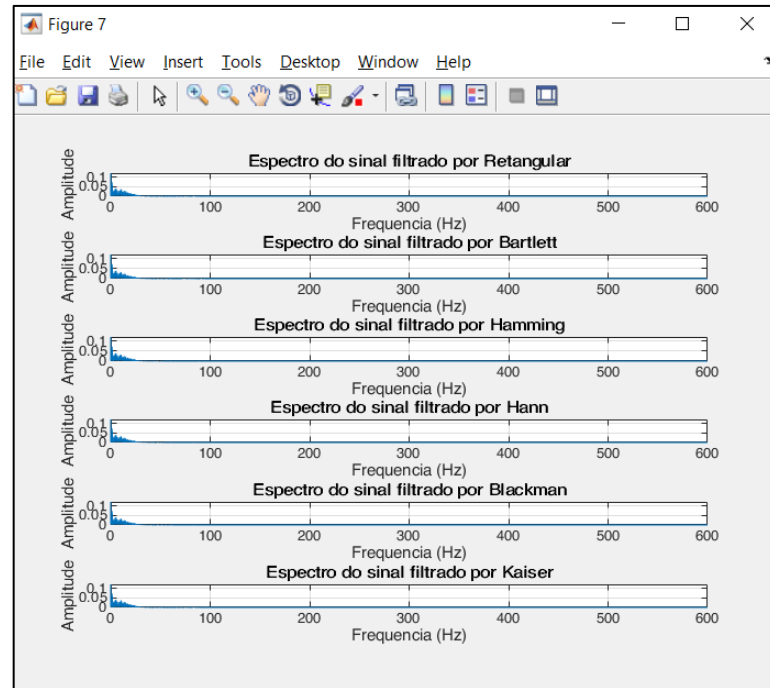
Sinal filtrado com Hann - Figura 10



Sinal filtrado com Blackman - Figura 11

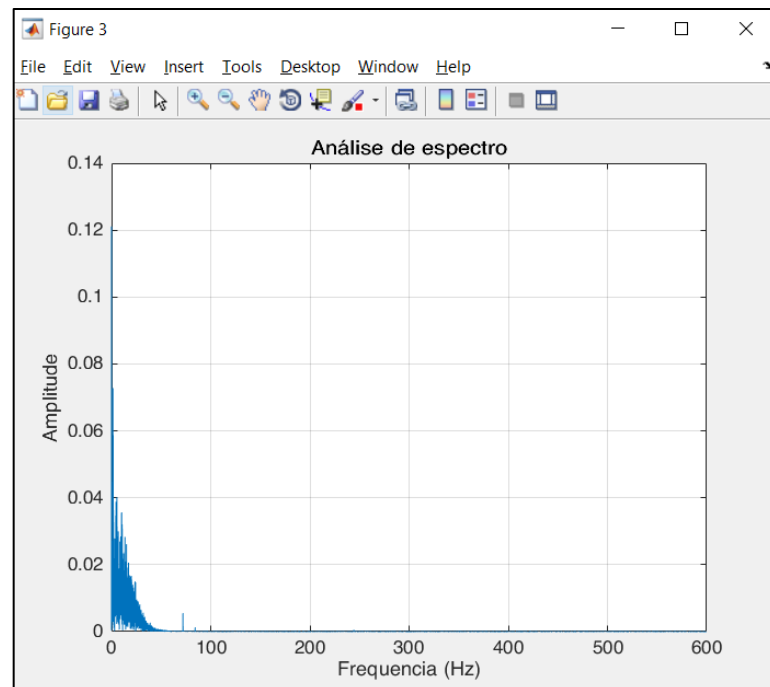


Sinal filtrado com Kaiser - Figura 12

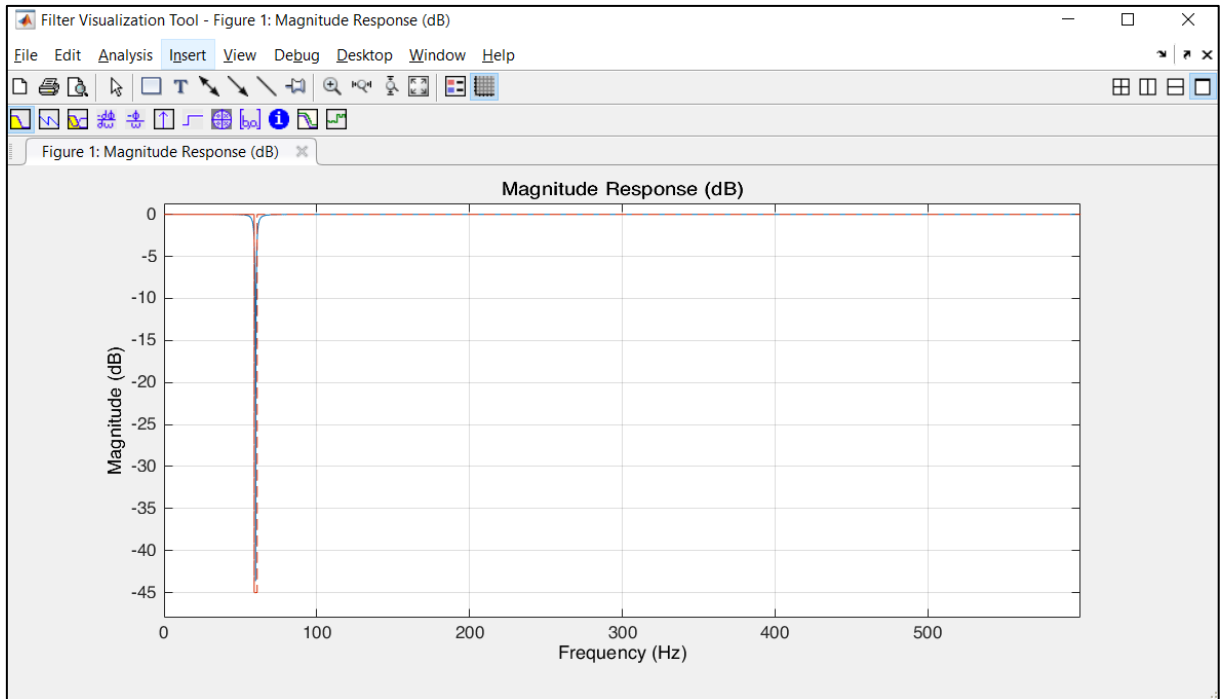


Espectros de frequência dos sinais filtrados - Figura 13

Questão I) – d) O filtro Notch é considerado um filtro rejeita faixa de banda estreita que é bastante útil para remoção de frequência única, como para o caso da de 60Hz do ECG. Assim, projetar um filtro Notch para remover o 60Hz do sinal de ECG.



Análise do espectro com o Filtro Notch - Figura 14



Filtro Notch Aplicado no ECG - Figura 15

Questão I) – e) Desenvolver um algoritmo para detecção da frequência cardíaca do ECG.

```
close all % fechar todas figuras
clear all % limpar todas variáveis
clc % limpar janela de comando

ecg = load('ECG_noise.txt');
fs = 1200;

% filtro para retirar os 60hz do ecg

[n3, wn3] = cheb2ord(40/(fs/2), 72/(fs/2), 0.1, 40);
[num3, den3] = cheby2(n3, 40, wn3);
filtro_cheby2 = filter(num3, den3, ecg);
fileID = fopen('ecg-filtrado.txt', 'w');
fprintf(fileID, '%f\r\n', filtro_cheby2);
fclose(fileID);

% lê o ecg filtrado novamente

ecg = load('ecg-filtrado.txt');

% declaração variáveis

batidas = 0;

% determinar bpm de um eletrocardiograma
% faz a contagem dos picos dominantes do sinal.
% os picos são definidos como amostras maiores que 2 e maiores que
os dois
% picos que estão ao seu lado (esquerda e direita)
```

```

for i = 2 : length(ecg)-1 % inicia em 2 pois iremos comparar com
valores anteriores (indice 0 matlab)
    if(ecg(i) > ecg(i-1) & ecg(i) > ecg(i+1) & ecg(i) > 2.5)
        batidas = batidas + 1;
    end
end

% divide as batidas encontradas pela duração do sinal em minutos

n = length(ecg);
duracao_segundos = n/fs;
duracao_minutos = duracao_segundos/60;
bpm = batidas/duracao_minutos;

% imprimir batidas e bpm no Command Window

batidas
bpm

```



```

Command Window

batidas =

    51

bpm =

    90

fx >>

```

Valores obtidos e mostrados no console do MATLAB - Figura 15

5. Conclusão

O desenvolvimento dos exercícios fixou os conteúdos aprendidos em sala de aula e os principais benefícios e desvantagens do uso de alguns deles. O uso de algumas funções do MATLAB acabaram por se tornar um pouco difícil o aprendizado devido a falta de prática com a ferramenta, porém após a leitura de algumas documentações no próprio site da ferramenta ou com o parâmetro help facilitaram o desenvolvimento dos algoritmos e foi conseguido obter resultados satisfatórios para solucionar os problemas propostos pelo Professor.

Os códigos desenvolvidos estão disponíveis no Github:
<https://www.github.com/douglasbrandao>

6. Referências Bibliográficas

ALAN V. OPPENHEIM, R. W. S. **Processamento em tempo discreto de sinais**. [S.l.]: [s.n.], 2013.
MATLAB Documentation. Disponível em: <<https://www.mathworks.com/help/matlab/>>.
Acesso em: 09 Junho 2019.

WR Kits. Disponível em:

<<https://www.youtube.com/channel/UCazAvTtoRIOrFDWDJDB2DKQ>>. Acesso em: 12 Junho 2019.