INSTITUTO FEDERAL DO CEARÁ
ENGENHARIA DE TELECOMUNICAÇÕES
PROCESSAMENTO DIGITAL DE SINAIS

FILTRAGEM DE SINAL DE ÁUDIO ATRAVÉS DA IMPLEMENTAÇÃO DE FILTROS FIR E IIR

Beatriz Cristina Pereira de Assis

FORTALEZA/CE

2021

SUMÁRIO

1. INTRODUÇÃO……..……..... 02.
2. FUNDAMENTAÇÃO TEÓRICA…..……...…….. 02.
   2.1. FILTROS…….…….. 02.
   2.2. FILTROS DIGITAIS……..……... 03.
   2.3. FILTRO FIR……..…..……..... 03.
      2.3.1. MÉTODO DAS JANELAS…….……..... 04.
         a. JANELA RETANGULAR…….……..... 04.
         b. JANELA DE BARTLETT……..……..… 05.
         c. JANELA DE HANNING……..….. 05.
         d. JANELA DE HAMMING…….…… 05.
         e. JANELA DE BLACKMAN……..….. 05.
   2.4. FILTRO IIR……..……..……..…….….... 07.
      2.4.1. MÉTODO DA TRANSFORMAÇÃO BILINEAR……..... 08.
      2.4.2. FILTRO DE BUTTERWORTH…….……..….....…..... 08.
      2.4.3. FILTRO DE CHEBYSHEV..... 09.
      2.4.4. FILTRO ELÍPTICO…….……...…….... 09.
      2.4.5. FILTRO NOTCH…….……..……..... 10.
   2.4. MATLAB……..…..... 11.
3. METODOLOGIA…...……..... 11.
   3.1. PRIMEIRA PARTE……..…… 14.
   3.2. SEGUNDA PARTE……..…. 15.
4. RESULTADOS E DISCUSSÕES……..……..……..... 16.
   4.1. PRIMEIRA PARTE….……...……..…… 16.
   4.2. SEGUNDA PARTE..……..……..…… 19.
5. CONCLUSÃO……....……..…….…….. 21.
6. REFERENCIAS…….……..…..……... 21.
7. ANEXOS……..…… 22.

1. INTRODUÇÃO

Os sinais estão presentes em diversas situações do cotidiano dos humanos, como o sinal de voz que permite a comunicação pessoalmente ou por um canal de transmissão. Um sinal pode ser definido como uma função na qual contém informações sobre o estado ou comportamento de um fenômeno físico denotado no domínio adequado. Quando o objetivo é a transmissão de informações, é imprescindível que o sinal chegue ao receptor da forma mais fiel possível. (OPPENHEIM; SCHAFER, 1998).

Em um sinal de áudio, a forma como o mesmo é processado pode implicar em problemas de ruídos e interferências. Tanto o ruído quanto a interferência são sinais indesejáveis capazes de degradar a clareza da informação no sinal original, em que dependendo das intensidades, pode até torná-lo incompreensível. (ROBERTS, 2009). A solução para problemas com ruídos e interferência, baseia-se na utilização de técnicas de filtragem de sinais. A filtragem de sinais é realizada por um sistema denominado filtro. Um filtro de forma geral, apresenta a capacidade de selecionar uma parte específica de frequência e descartar o restante, permitindo separar, recuperar, atenuar e maximizar sinais (MARQUES, 2014).

Neste projeto serão implementados filtros digitais do tipo FIR e IIR, que serão melhor descritos na fundamentação teórica, para retirar interferências sobre o som da fala humana de um sinal de áudio, com a intenção de comparar os resultados obtidos quando se usa cada uma dessas estruturas.

2. FUNDAMENTAÇÃO TEÓRICA

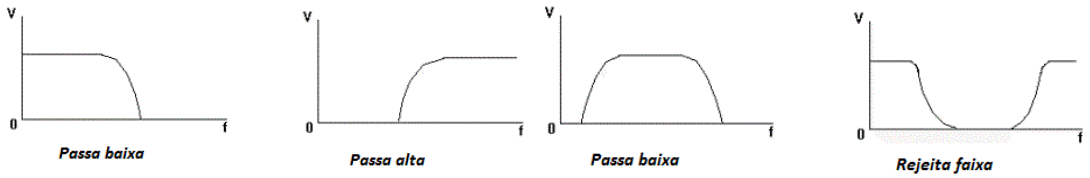
2.1. FILTROS

Filtros são estruturas construídas de forma a eliminar ou acentuar determinados aspectos de um sinal. Por exemplo, um aparelho de som que reforça os componentes graves de uma música possui um filtro que tenha esse comportamento. É comum a necessidade de se eliminar o ruído causado pela rede elétrica em sinais médicos, sendo necessário também um filtro para isso. Receptores de rádio (e qualquer outro dispositivo que receba informação por radiofreqüência) usam filtros para selecionar apenas um canal dentre os vários sendo recebidos. Em processamento de imagens há filtros para, entre outras aplicações, atenuar as distorções causadas pela exibição em pixels, chamados de filtros anti-aliasing. Estes mesmos filtros são utilizados em quase todas as aplicações para diminuir distorções causadas pela sub-amostragem (MARQUES, 2019).

Existem filtros de diversos tipos, de acordo com sua implementação. Podem ser ativos ou passivos, dependendo da necessidade de alimentação externa; digitais, se a representação do sinal for discreta, ou analógicos, se ela for contínua, isto é, poder expressar infinitos níveis diferentes. Esta propriedade do sinal é extrapolada para o filtro, sendo o mesmo também "discreto"ou "contínuo"de acordo com o tipo do sinal por ele processado. Além disso, os filtros são classificados entre FIR e IIR de acordo com a duração do sinal na sua saída ao ser submetido a uma entrada impulsiva, sendo finito e infinito respectivamente.

A funcionalidade ou comportamento do filtro também é uma das formas de classificá-lo. Na figura 1 é mostrado todos os comportamentos que um filtro pode apresentar.

Figura 1. Tipos de comportamentos de filtros.



Fonte: *Marques*, 2014.

Em que,

- Filtro passa-baixa: Rejeita componentes de alta frequência do sinal.Filtro passa-alta:
- Rejeita componentes de baixa frequência do sinal.
- Filtro passa-banda: Passa uma faixa limitada de frequências.
- Filtro rejeita-banda: Passa somente frequências abaixo e acima de uma faixa limitada, rejeitando as componentes pertencentes à faixa.

De uma forma generalizada, a banda de passagem é o conjunto de frequências que o filtro não altera consideravelmente, e a banda de rejeição o conjunto de frequências que o filtro rejeita. A frequência de corte pode ser definida como aquela em que a energia do sinal é reduzida à metade. Ela está presente na faixa denominada "zona de transição", entre a banda de passagem e a banda de rejeição.

2.2. FILTROS DIGITAIS

Um filtro digital é um sistema temporal discreto projetado para passar o conteúdo espectral de um sinal de entrada em uma determinada banda de frequências, isto é, a função de transferência do filtro forma uma janela espectral através da qual somente é permitida a passagem da parte desejada do espectro de entrada (LATHI, 2006).

Os filtros digitais podem ser do tipo FIR (Finite Impulse Response), filtros não recursivos (não requerem realimentação) cuja resposta impulsional tem tamanho finito, ou do tipo IIR (Infinite Impulse Response), filtros recursivos (usam realimentação) cuja resposta impulsional tem tamanho infinito. Os filtros IIR projetados com auxílio de mapeamento matemático entre uma função analógica e uma função discreta são o equivalente digital dos filtros analógicos.

2.3. FILTRO FIR

Um filtro FIR (*Finite Impulse Response*) possui resposta impulsional de tamanho finito (L coeficientes). Ou seja, a resposta ao impulso retorna a zero no tempo n = L. A sua ordem é dada por M = L-1.

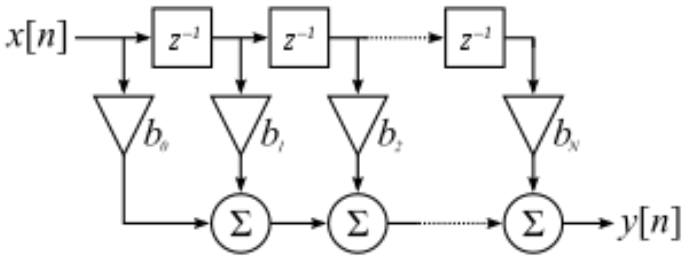
A função de transferência desse tipo de filtro tem o seguinte formato:

$$H(z) = B(z) = \sum_{k=0}^{M} b_k z^{-k},$$

em que B(z) é um polinômio em z de grau M e bk são os coeficientes do filtro. O filtro FIR não possui pólos.

Ao implementar sua realização em diagrama de blocos, conforme a Figura 2, fica perceptível que sua saída depende apenas da entrada atual e entrada anteriores.

Figura 2. Realização de Filtro FIR - Forma direta I.



Fonte: *Silvério*, 2011.

Existem dois problemas de implementação com a equação apresentada, isto é, a função de transferência representa um filtro digital não causal de duração infinita. Um filtro causal de duração finita pode ser obtido truncando-se a resposta impulsiva de duração infinita, isto é, multiplicando-a por uma janela retangular e, posteriormente, tornando-a causal. Esta modificação não altera a resposta em amplitude do filtro. Entretanto, o abrupto truncamento da série de Fourier resulta em oscilações na banda de passagem e de rejeição, chamadas de efeito de Gibbs (SILVÉRIO,2011).

2.3.1. MÉTODO DAS JANELAS

Para reduzir as oscilações devidas ao efeito Gibbs em filtros FIR com uma série de coeficientes de Fourier finita, é usada uma classe particular de funções de ponderação para modificar os seus coeficientes. Estas funções de ponderação no domínio do tempo são geralmente chamadas de funções de janelas. O filtro será obtido através do seguinte produto:

$$h[n] = h_d[n] \cdot w[n]$$

Em que hd[n] é a resposta impulsiva do filtro, a depender do tipo implementado, e w[n] é a função janela.

Ao fazer essa multiplicação no domínio do tempo, o efeito no domínio da frequência é a convolução da resposta do filtro ideal com a transformada de Fourier da função janela. Assim, quanto mais estreito o lóbulo principal da janela, mas acentuada é a transição no filtro. Da mesma forma, quanto menor for a amplitude dos lóbulos secundários, menos ondulações ocorrerão. Há vários tipos de janelas que podem ser utilizadas:

a. JANELA RETANGULAR

A função janela mais simples é a janela retangular. Ela é definida como:

$$w[n] = \begin{cases} 1 & 0 \leqslant n \leqslant M \\ 0 & \text{diferente.} \end{cases}$$

Essa função possui uma transição abrupta para o zero, e isso faz com que haja um comportamento oscilatório no filtro resultante. Os outros tipos de janela buscam amenizar essa descontinuidade. Sua resposta em frequência é apresentada na Figura 3-(a) e sua resposta no tempo na Figura 4.

b. JANELA DE BARTLETT

A janela de Bartlett, também chamada de janela triangular é definida como:

$$w[n] = \begin{cases} \frac{2n}{M} & 0 \leqslant n \leqslant \frac{M}{2} \\ 2 - \frac{2n}{M} & \frac{M}{2} \leqslant n \leqslant M \\ 0 & \text{diferente.} \end{cases}$$

A janela pode ser entendida como a convolução de duas janelas retangulares. A transição para o zero é linear, não mais descontínua. Dessa forma, o comportamento oscilatório da janela retangular é reduzido. Sua resposta em frequência é apresentada na Figura 3-(b) e sua resposta no tempo na Figura 4.

c. JANELA DE HANNING

A janela de Hanning é definida como:

$$w[n] = \begin{cases} 0.5 - 0.5cos(\frac{2\pi n}{M}), & 0 \leqslant n \leqslant M. \\ 0, & \text{diferente.} \end{cases}$$

Esta janela apresenta uma boa atenuação na banda de rejeição, se comparada às janelas anteriores. Sua resposta em frequência é apresentada na Figura 3-(c) e sua resposta no tempo na Figura 4.

d. JANELA DE HAMMING

A janela de Hamming é definida como**:**

$$w[n] = \begin{cases} 0.54 - 0.46cos(\frac{2\pi n}{M}), & 0 \leqslant n \leqslant M. \\ 0, & \text{diferente.} \end{cases}$$

A janela foi projetada de forma a minimizar o maior lóbulo secundário da janela de Hann, distribuindo sua energia para os demais lóbulos secundários. Sua resposta em frequência é apresentada na Figura 3-(d) e sua resposta no tempo na Figura 4.

e. JANELA DE BLACKMAN

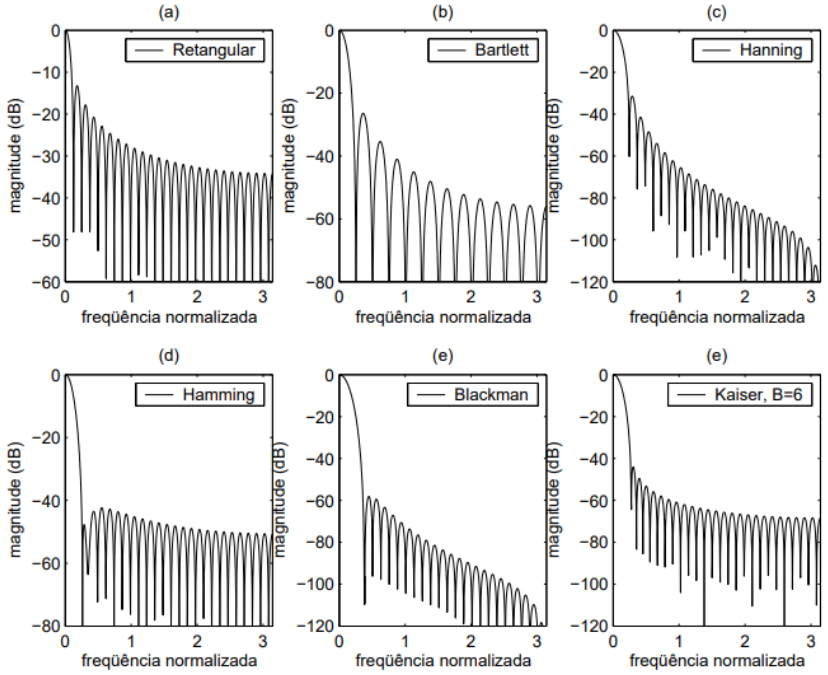
A janela de Blackman é definida como:

$$w[n] = \begin{cases} 0.42 - 0.5cos(\frac{2\pi n}{M-1}) + 0.08cos(\frac{4\pi n}{M-1}), & 0 \leqslant n \leqslant M. \\ 0, & \text{diferente.} \end{cases}$$

Essa janela possui uma atenuação maior na banda de rejeição, se comparado com todas as janelas anteriores. Por outro lado, a zona de transição se alarga. Sua resposta em frequência é apresentada na Figura 3-(e) e sua resposta no tempo na Figura 4.

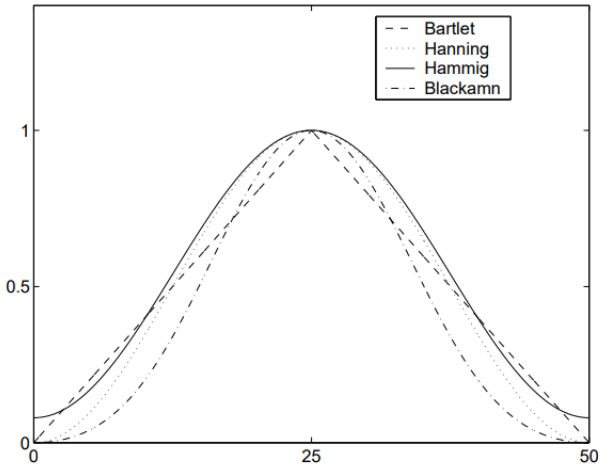
Há ainda outros tipos de janelas que podem ser implementadas com o objetivo de reduzir o efeito de Gibbs, como a janela de Kaiser, em que sua resposta em frequência é mostrada na Figura 3-(f).

Figura 3. Resposta em frequência das janelas.



Fonte: *Lopes*, 2008

Figura 4. Resposta no tempo das janelas.



Fonte: *Lopes*, 2008.

Para o projeto de um Filtro FIR é necessário realizar a escolha de uma janela que melhor se adeque ao objetivo final, de acordo com as especificações e características de cada janela. Outro parâmetro que se deve levar em consideração é a ordem do filtro que será implementado, ela definirá o quão seletivo ele será (LOPES, 2008).

2.4. FILTRO IIR

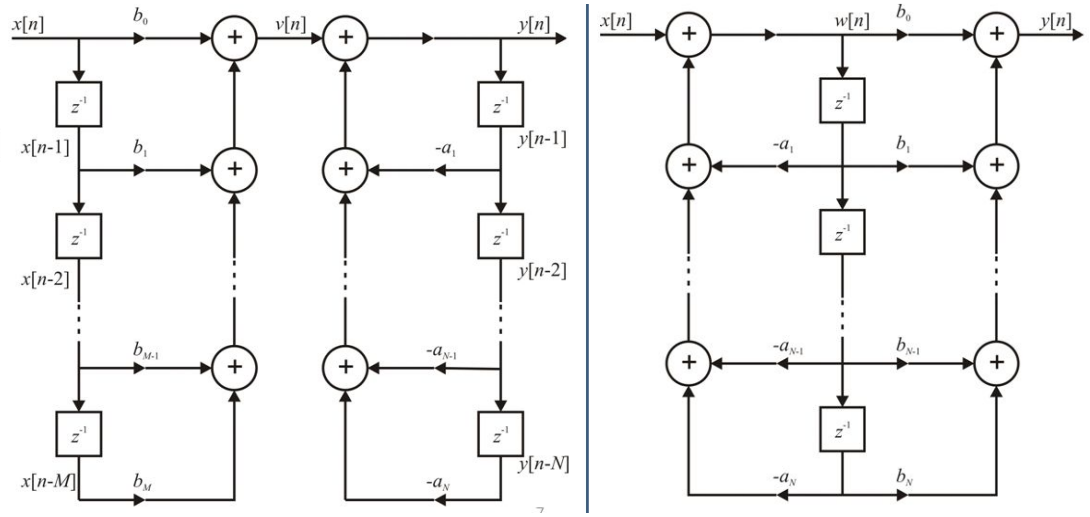
Esse tipo de filtro possui uma resposta impulsional que não retorna a zero depois de algum tempo. Um filtro IIR (*Infinite Impulse Response*) possui resposta impulsional de tamanho infinito. Geralmente um sistema com realimentação apresenta uma resposta IIR. A função de transferência desse tipo de filtro tem o seguinte formato:

$$H(z) = \frac{B(z)}{A(z)} = \frac{\sum_{k=0}^{N} b_k z^{-k}}{\sum_{k=1}^{M} a_k z^{-k}},$$

em que B(z) e A(z) são polinômios em z. A(z) é de ordem M, enquanto B(z) tem ordem igual ou inferior a M (filtros causais). A maior vantagem do filtro IIR é a necessidade de uma ordem menor para atender às mesmas restrições, se comparado com a de um filtro FIR. A maior desvantagem é a dificuldade em garantir a fase linear e uma possível instabilidade do filtro.

Sua realização na forma direta I e II é mostrada na Figura 5.

Figura 5. Diagrama de Blocos do Filtro IIR - Forma direta I e II, respectivamente.



Fonte: *Neto*, 2018.

No caso dos filtros IIR, o problema da aproximação para o projeto de filtros digitais não é conceitualmente diferente do problema para projeto de filtros analógicos. A abordagem para o projeto de filtros analógicos envolve uma aproximação analítica das especificações do filtro por uma função de transferência, a partir da qual projeta-se uma rede analógica que implemente esta função.

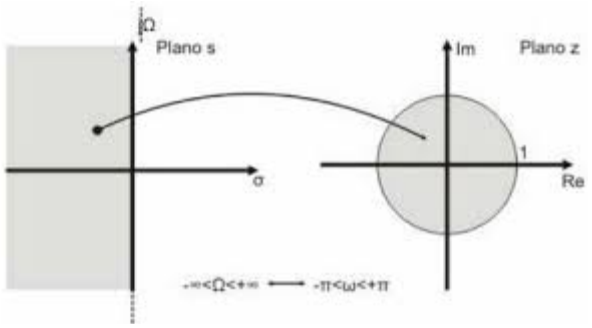
Uma função de transferência realizável é uma das características de uma rede linear estável e causal. Estas características podem ser obtidas fazendo com que a função de transferência seja uma função racional de S (Laplace) com coeficientes reais, que os pólos do filtro analógico estejam na metade esquerda do plano S e o grau do numerador seja igual ou menor que o grau do polinômio denominador (NETO, 2018).

O problema do projeto de filtros digitais requer a determinação dos coeficientes da equação diferencial para preencher as características desejadas para o filtro, como resposta em frequência. Como já existem abordagens clássicas para o projeto de filtros analógicos, foram desenvolvidas aproximações que mapeiam os pólos e zeros analógicos do plano S para o plano Z, de forma a alcançar as características desejadas do filtro digital. Um dos mapeamentos mais utilizados é a transformação bilinear.

2.4.1. MÉTODO DA TRANSFORMAÇÃO BILINEAR

No método da transformação bilinear cada ponto da resposta em frequência do filtro analógico corresponde a um ponto na resposta em frequência do filtro digital. A estabilidade é preservada, ou seja, o semi-plano esquerdo do plano S (Laplace) é mapeado no interior do círculo unitário no plano Z, conforme apresentado na Figura 6. Além disso, esse mapeamento não produz distorção devido ao fenômeno de aliasing.

Figura 6. Transformação bilinear de S para Z



Fonte: *Santana*, 2017.

O filtro digital é obtido a partir da seguinte mudança de variáveis:

$$z = \frac{1 + \frac{s}{2fs}}{1 - \frac{s}{2fs}},$$

onde fs é um parâmetro que pode variar.

O mapeamento entre as frequências digital (ω) e analógica (Ω) resultado da transformação acima é dado por:

$$\Omega = 2fs \cdot tan(\frac{\omega}{2}) \qquad \omega = 2 \cdot arctan(\frac{\Omega}{2fs})$$

A transformação bilinear tem a propriedade de conservar a magnitude da resposta em frequência do filtro analógico, porém introduzindo distorção de fase. Desta forma, caso se necessite resposta em frequência linear em fase e magnitude, então deve-se utilizar filtros FIR. Porém, como nem todas as aplicações necessitam de linearidade na fase da resposta em frequência, os filtros IIR também têm sua utilidade nestes casos (SANTANA, 2017).

2.4.2. FILTRO DE BUTTERWORTH

O filtro Butterworth é otimizado para apresentar a resposta mais plana na banda de passagem. Um filtro Butterworth passa-baixas é sempre decrescente (monotônico). Ele não possui oscilações, mas apresenta uma taxa de decaimento menor, se comparado com os filtros seguintes (Chebyshev e Elíptico), assim demandando uma ordem maior para que as mesmas especificações sejam atendidas. O filtro também apresenta uma resposta em fase mais linear.

O quadrado do módulo da sua resposta em frequência possui o seguinte formato:

$$|H(j\Omega)|^2 = \frac{1}{1 + \left(\frac{\Omega}{\Omega_c}\right)^{2M}},$$

em que Ωc é a frequência de corte do filtro e M sua ordem. Assim, tem-se:

$$H(s)H(-s) = \frac{(j\Omega_c)^{2M}}{s^{2M} + (j\Omega_c)^{2M}}$$

Os 2M pólos de H(s)H(-s) estão distribuídos no círculo de raio Ωc (separados por π M radianos). Os M pólos de H(s) são aqueles localizados no lado esquerdo do eixo imaginário (condição para filtro estável e causal). O filtro não possui zeros. Sua resposta em frequência é mostrada na Figura 6.

2.4.3. FILTRO DE CHEBYSHEV

As aproximações de Chebyshev se diferem da de Butterworth por possuírem ganho monotônico em apenas uma das faixas. O tipo I possui a banda de rejeição monotônica, enquanto o tipo II possui a banda de passagem monotônica. Esta perda de monotonicidade implica em ondulações no ganho, mais usualmente chamadas de *ripple*, como mostrado na Figuras 6.

O módulo da resposta em frequência do tipo I possui o seguinte formato:

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

Sendo n sua ordem, ∈ o fator de oscilação da banda de passagem e Tn(x) o polinômio Chebyshev de ordem n. Já o quadrado do módulo da resposta em frequência do tipo II tem o seguinte formato:

$$|H(j\omega)|^2 = \frac{1}{\sqrt{1 + \frac{1}{\epsilon^2 T_n^2(\omega_0/\omega)}}}$$

A quantidade de *ripple* pode ser diminuída em troca de aumento no número de coeficientes, ou piorada a taxa de velocidade da transição entre as bandas de passagem e rejeição (*roll-off)* . É possível demonstrar que quando o *ripple* tende a zero, o filtro de Chebyshev se transforma em um filtro de Butterworth. Esta capacidade de se manipular *ripple* e *roll-off* sem alterar o número de coeficientes do filtro torna a aproximação de Chebyshev muito mais flexível do que a de Butterworth. Porém, não é possível especificar todos os parâmetros ao mesmo tempo (YUKIO, 2018). O projetista deve escolher entre fixar o *ripple* ou fixar o número de coeficientes. Sua resposta em frequência é mostrada na Figura 6.

2.4.4. FILTRO ELÍPTICO

Filtros Elípticos possuem *ripple* tanto na banda de passagem quanto na banda de rejeição, assim sua banda de transição é a mais estreita, se comparada com os filtros Chebyshev

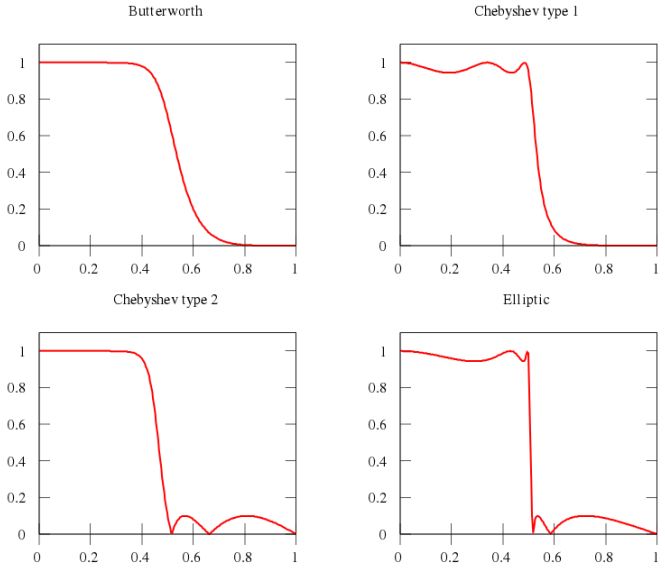
e Butterworth. Do mesmo modo, para atender às mesmas especificações, é necessária uma ordem menor para o filtro Elíptico. Além do mais, as oscilações na banda de passagem e banda de rejeição podem ser independentemente controladas.

O quadrado do módulo da sua resposta em frequência possui o seguinte formato:

$$|H(j\Omega)|^2 = \frac{1}{1 + \epsilon^2 U_M^2\left(\xi, \frac{\Omega}{\Omega_c}\right)}$$

Sendo M sua ordem, o fator de oscilação da banda de passagem, ξ o fator de seleção e Um(x) a função Jacobiana elíptica de ordem M. Sua resposta em frequência é mostrada na Figura 7.

Figura 7. Resposta em frequência: Passa-Baixa dos filtros Butterworth, Chebyshev I e II e Elíptico.



Fonte: *Yukio,* 2018.

2.4.5. FILTRO NOTCH

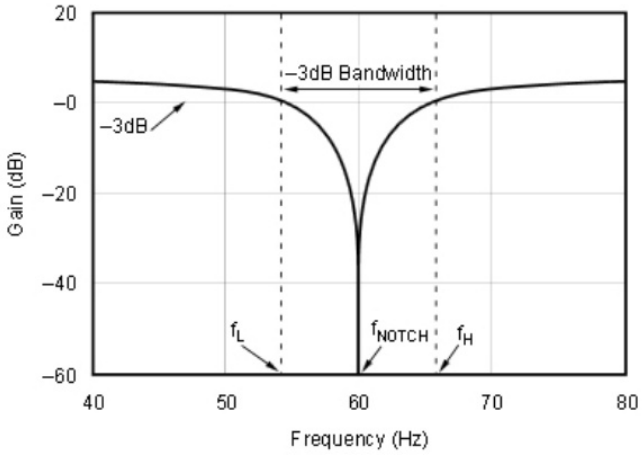
Esse tipo de filtro faz parte dos filtros rejeita-faixa e é utilizado para a eliminação de componentes específicas de frequência. O circuito do filtro é projetado para que a frequência de corte inferior seja ajustada como num filtro passa-baixa e a superior como em um filtro passa - alta. O filtro não efetua atenuações que realizam a eliminação das frequências abaixo de um ω1 e acima de um ω2 (FAMBRINI, 2013).

A função de transferência do filtro notch correspondente é dada por:

$$H(z) = \frac{E(z)}{U(z)} = \frac{1}{2}\left[1 + A(z)\right] = \frac{1 + \operatorname{sen}\tau}{2}\,\frac{1 + 2\operatorname{sen}\theta z^{-1} + z^{-2}}{1 + \operatorname{sen}\theta(1 + \operatorname{sen}\tau)z^{-1} + (\operatorname{sen}\tau)z^{-2}}.$$

Sua resposta em frequência é mostrada na Figura 8.

Figura 8. Resposta em frequência do Filtro Notch.



Fonte: Fambrini, 2013.

Ele é capaz de rejeitar uma faixa bastante estreita de frequências, atuando quase que exclusivamente na frequência selecionada, e bem pouco nas outras ao redor. Sua utilização é recomendada quando o sinal a ser atenuado é bem definido. Pelo fato de atuarem em faixas reduzidas de frequências, os filtros Notch interferem pouco na qualidade do sinal.

2.4. MATLAB

O MATLAB é uma *package* destinada ao estudo de problemas científicos e de engenharia, ou, de uma forma mais geral, ao estudo de quaisquer problemas em que haja um trabalho computacional significativo a realizar que possa (ou deva) utilizar matrizes. Este software foi produzido pela companhia americana The Math Works, Inc e o seu nome deriva do inglês *"matrix laboratory"*. Ele foi escolhido para o desenvolvimento deste projeto, pois oferece além da manipulação fácil de matrizes, o acesso a um número crescente de rotinas incorporadas, potencialidades gráficas a 2 dimensões e a possibilidade de ser configurado para a implementação de vários tipos de filtros digitais (MATHWORKS, 2017).

3. METODOLOGIA

O desenvolvimento deste trabalho será realizado utilizando o software MATLAB para a execução de duas etapas/abordagens da filtragem de um sinal de áudio com interferências. Inicialmente, o áudio será analisado no espectro de frequência para identificação das componentes indesejadas e do sinal de áudio. Na primeira parte, serão implementados filtros do tipo IIR a partir de uma função criada no MATLAB para eliminação das componentes de frequências indesejadas, serão analisadas ainda as respostas ao impulso de cada filtro implementado, e o áudio filtrado será gravado em arquivo .wav, cada gráfico gerado a partir das filtragens realizadas será analisado. Já na segunda parte, será implementado pelo menos um filtro FIR para filtragem do sinal de áudio, e as mesmas análises da primeira parte serão realizadas.

Para isso, foi realizada a criação de duas funções no Matlab para os filtros FIR e IIR, não será utilizada a função '*Filter*' da biblioteca do Matlab para implementação da filtragem do sinal.

Segue abaixo o código desenvolvido para a função do filtro FIR.

```
function y= filtro_fir(b,x)

M= length(x);                          %tamanho do vetor do sinal de entrada
N= length(b);                                  %tamanho do filtro FIR (Ordem)
y= zeros(1,M);          % gera o vetor para armazenar a saida do sistema
aux= zeros(1,N);       %gera vetor auxiliar para fazer a eq de diferenca
for i= 1:M
    aux= circshift(aux,1); %desloca circularmente os valores de aux em 1
    aux(1)=x(i);   %atualiza a 1 posição de aux com a amostra atual de x
    y(i)= b*aux';                                          %Saida do sistema
end
end
```

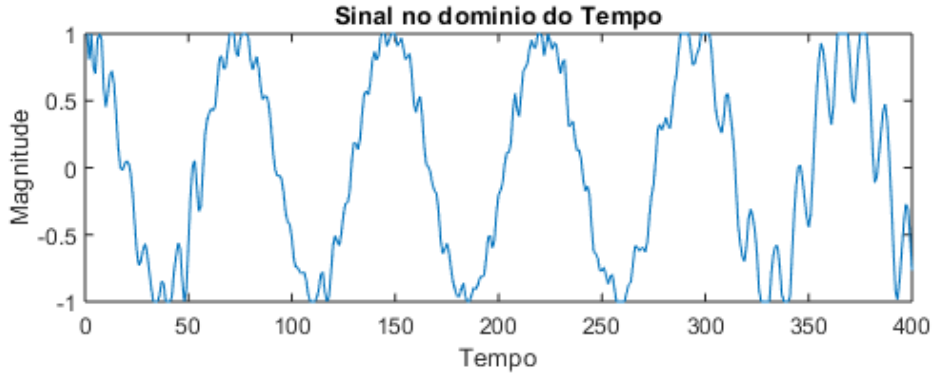
Já a função desenvolvida para o filtro IIR é a seguinte,

```
function y= filtro_iir(a,b,x)

  M = length(x);                          %tamanho do vetor do sinal de entrada
  N = length(a);                                  %tamanho do filtro IIR (Ordem)
  T = length(b);                 %tamanho do vetor do sinal de realimentacao

  y = zeros(1,M);              %gera vetor para armazenar a saida do sistema
  aux = zeros(N,1);      %gera vetor com os coeficientes da eq. dif. entrada
  aux1 = zeros(T-1,1);     %gera vetor com os coeficientes da eq. dif. saída

for i = 1:M
    aux = circshift(aux,1);   % desloca circularmente os valores de aux em 1
    aux(1) = x(i);     %atualiza a 1 posição de aux com a amostra atual de x

    y(i) = ((a*aux)-(b(2:T)*aux1));                          %saida do sistema

    aux1 = circshift(aux1,1); %desloca circularmente os valores de aux1 em 1
    aux1(1) = y(i);   %atualiza a 1 posição de aux1 com a amostra atual de y
end
end
```

Em seguida foi realizada a leitura e visualização do sinal de áudio no domínio do tempo, conforme código abaixo e Figura 9:

```
[x,fa]= audioread('fala_sirene_tml.wav'); %leitura do audio
subplot(2,1,1);                  %define a escala do gráfico
plot(x(1:400));             %plota as 400 primeiras amostras
title('Sinal no dominio do Tempo');      %Titulo do gráfico
xlabel('Tempo');                          %Titulo do eixo x
ylabel('Magnitude');                      %Titulo do eixo y
```

Figura 9. Sinal no domínio do tempo

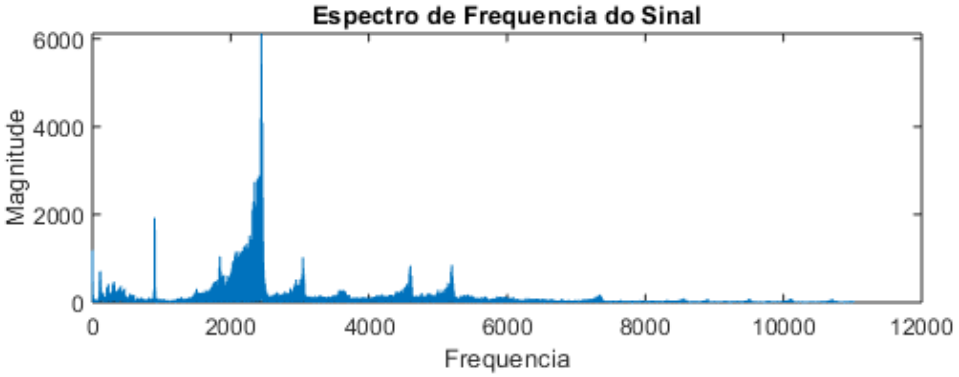


Fonte: Autoral.

Nota-se que há várias componentes distorcendo o sinal de áudio original, que deveria ser uma senóide limpa, ou seja, sem ondulações em seu formato. Após a visualização no domínio do tempo, realizou-se a análise do espectro de frequência através do seguinte código:

```
%Análise espectral da frequencia
figure;                                          %Novo grafico
X= fft(x);                                %Faz a transformada de fourier
N= length(X);                               %Pega a quantidade de pontos
X= X/(N/2);                                      %Normalizacao
f= [0: N-1]*fa/(N-1); %Vetor que contem as frequencias de f a fa
subplot(2,1,1);                              %define a escala do gráfico
plot(f(1:N/2),abs(X(1:N/2)));                 %Plota f sem espelhamento
title('Espectro de Frequencia do Sinal');     %Titulo do gráfico
xlabel('Frequencia');                          %Titulo do eixo x
ylabel('Magnitude');                           %Titulo do eixo y
```

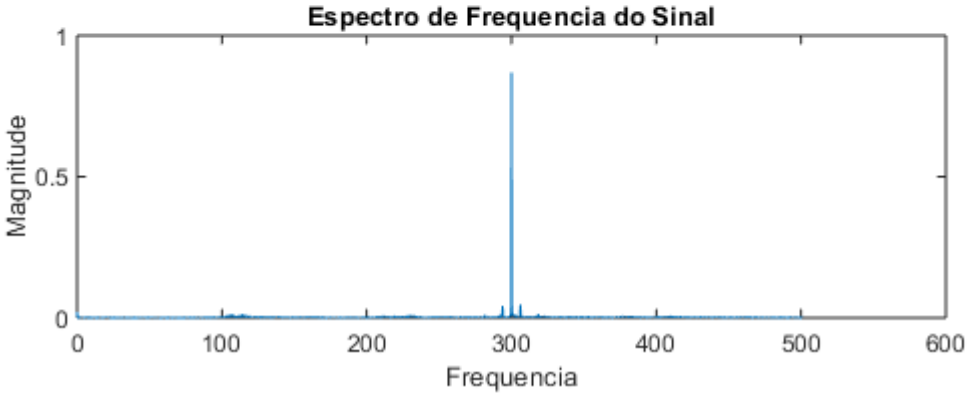
Figura 10. Espectro da frequência do sinal



Fonte: Autoral.

Analisando o espectro de frequências do sinal mostrado na Figura 10 é possível observar um pico na amplitude na frequência de 300Hz, essa frequência foi encontrada por inspeção do sinal, conforme Figura 11.

Figura 11. Componente de 300 Hz observada no espectro de frequências



Fonte: Autoral.

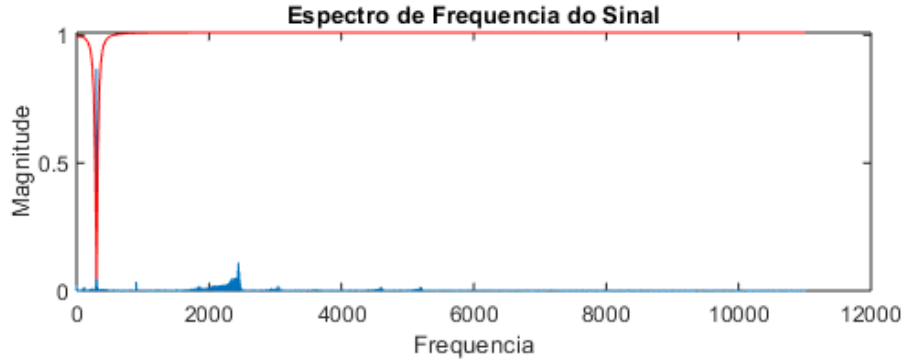
Os tratamentos desta componente e das demais que são consideradas interferências no sinal serão realizados na primeira parte através de filtros IIR e na segunda parte através de filtros FIR e IIR.

3.1. PRIMEIRA PARTE

A primeira filtragem será realizada por um filtro Notch, esse filtro foi escolhido por conta de sua seletividade e por ser recomendado para a eliminação de componentes específicas de frequência. O código abaixo foi implementado e o sinal de áudio, juntamente com a resposta em frequência do filtro foi plotado e pode ser observado na Figura 12.

```
%Primeira Filtragem - Notch
wc= 2*pi*300;                    %Frequência de corte do filtro Notch
r=0.99;                                    %Raio do filtro Notch
a=[1 -2*cos(wc/fa) 1];                   %Numerador do filtro Notch
b=[1 -2*r*cos(wc/fa) r^2];             %Denominador do filtro Notch
[H,W] = freqz(a,b,512,fa);  %Resposta em frequência do filtro Notch
hold on                                   %Mantém o gráfico anterior
plot(W,abs(H),'r'); %plota a resposta em frequencia do filtro Notch
```

Figura 12. Resposta em frequência do filtro plotada juntamente com a do sinal.



Fonte: Autoral.

A segunda filtragem foi realizada utilizando um filtro Chebyshev Tipo I de ordem 10 e passa-baixa, para eliminar frequências acima de 800 Hz, essa frequência de corte foi

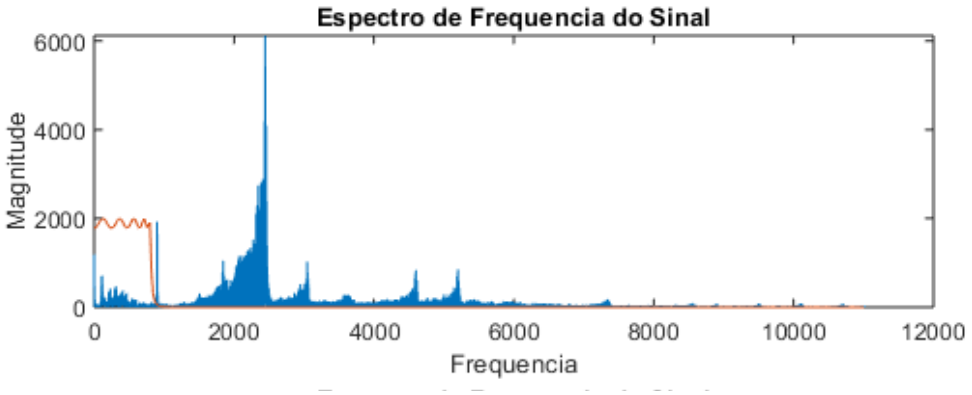
escolhida por tentativa e erro ao ouvir o áudio filtrado, ela foi a frequência em que se considerou-se o áudio o mais audível possível .

Esse filtro foi escolhido por ter uma resposta em frequência mais linear na região da frequência de corte, comparado ao butterworth, que apresenta uma resposta mais inclinada nesta região. O filtro elíptico não foi escolhido por apresentar ondulações após a frequência de corte, similar ao filtro Chebyshev tipo II, o que poderia ocasionar a passagem de parte das frequências indesejadas.

Abaixo é apresentado o código utilizado, bem como o sinal após a primeira filtragem com a aplicação do filtro de Chebyshev tipo I para a segunda filtragem, conforme Figura 13.

```
%Segunda Filtragem - Chebyshev Tipo I
wn = 800/(fa/2);                              %Frequencia de corte
[t,u]= cheby1(10,1,wn);                          %Filtro Chebyshev
[h,w] = freqz(t,u,512,fa);     %Resposta em frequência do filtro
hold on;                                  %Mantém o gráfico anterior
plot(w, 2000*abs(h)); %plota a resposta em frequencia do filtro
title('Espectro de Frequencia do Sinal');     %Titulo do gráfico
xlabel('Frequencia');                          %Titulo do eixo x
ylabel('Magnitude');                           %Titulo do eixo y
```

Figura 13. Resposta em frequência do filtro plotada juntamente com a do sinal.



Fonte: Autoral.

Por fim, o espectro de frequência do áudio filtrado foi obtido, sua legibilidade ouvida e o áudio foi gravado em um arquivo .wav, também foi gerada a resposta ao impulso de cada filtro.

3.2. SEGUNDA PARTE

Nesta parte será implementado pelo menos um filtro FIR em conjunto com filtro IIR, o que a diferencia da primeira parte, em que todos os filtros utilizados são IIR. Inicialmente, optou-se por realizar a mesma filtragem com filtro Notch realizada na parte 1, por se tratar de um filtro IIR de melhor seletividade. Do mesmo modo, eliminou-se a frequência de 300 Hz, conforme Figura 12.

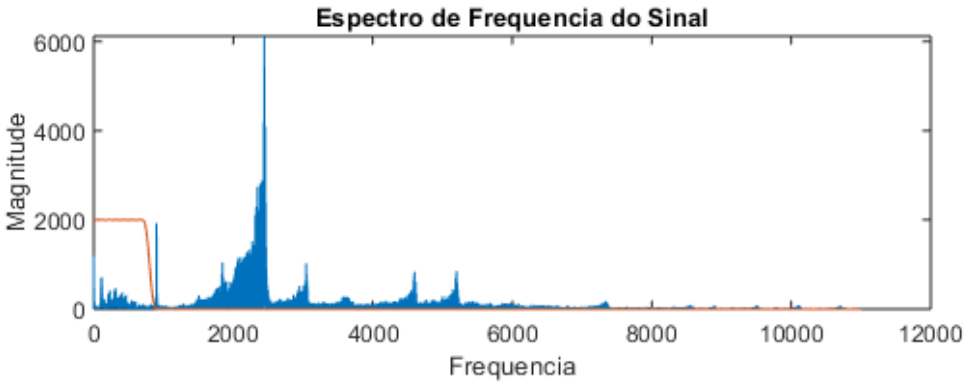
Realizada a primeira filtragem, a segunda foi feita por um filtro FIR passa-baixa utilizando a janela de Hamming, padrão do Matlab. Além da facilidade de implementação, essa janela foi escolhida por apresentar melhor atenuação na banda de rejeição comparada às

janelas: retangular, Bartlett e Hanning, perdendo apenas para a janela de Blackman, que possui maior atenuação na banda de rejeição, mas em troca sua zona de transição é mais larga.

O código é apresentado abaixo e na Figura 14 é mostrada a plotagem do filtro juntamente com o sinal de áudio filtrado pelo Notch.

```
%Segunda Filtragem - FILTRO FIR
wn = 2*(800/fa);                              %Frequencia de corte
q =fir1(400, wn);                                      %Filtro FIR
[h,w]=freqz(q,1,512,fa);       %Resposta em frequencia do filtro
hold on;                                %Mantém o gráfico anterior
plot(w,2000*abs(h));  %plota a resposta em frequencia do filtro
title('Espectro de Frequencia do Sinal');     %Titulo do gráfico
xlabel('Frequencia');                          %Titulo do eixo x
ylabel('Magnitude');                           %Titulo do eixo y
```

Figura 14. Resposta em frequência do filtro plotada juntamente com a do sinal.



Fonte: Autoral.

Assim como na primeira parte, o espectro de frequência do áudio filtrado foi obtido, sua legibilidade ouvida e o áudio foi gravado em um arquivo .wav, também foi gerada a resposta ao impulso de cada filtro.

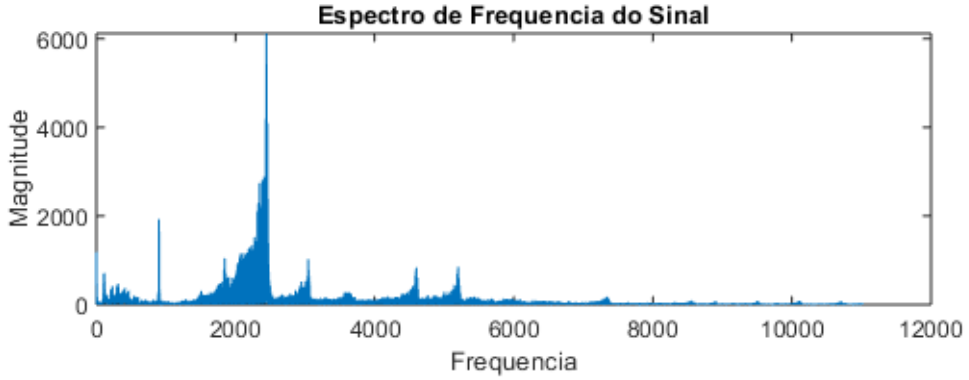
4. RESULTADOS E DISCUSSÕES

4.1. PRIMEIRA PARTE

Ao realizar a primeira filtragem do sinal de áudio com o filtro Notch através do código mostrado a seguir, foi obtido o espectro mostrado na Figura 15.

```
y = filtro_iir(a,b,x); % Aplicando o filtro Notch ao sinal original
Y = fft(y);                  %Calcula a transformada de Fourier de y
subplot(2,1,1);                %define a escala e posiçao do gráfico
plot(f(1:N/2),abs(Y(1:N/2)))     %plota o espectro do áudio filtrado
```

Figura 15. Espectro de frequência obtido após primeira filtragem



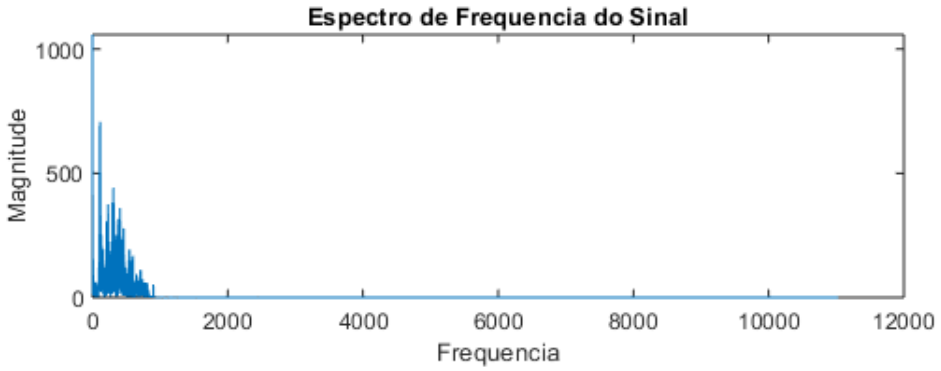
Fonte: Autoral.

Analisando a resposta em frequência obtida na primeira filtragem é possível inferir que o filtro Notch aplicado eliminou somente a componente de 300 Hz desejada, as demais frequências permanecem inalteradas.

Com a realização da segunda filtragem com o filtro de Chebyshev, no áudio já filtrado anteriormente com o filtro Notch a partir do código apresentado abaixo, obteve-se a resposta em frequência da Figura 16.

```
figure;                                                    %Novo grafico
p = filtro_iir(t,u,y);                  %implementação da função desenvolvida
P = fft(p);                           %Calcula a transformada de Fourier de P
subplot(2,1,1);                         %define a escala e posiçao do gráfico
plot(f(1:N/2),abs(P(1:N/2)))              %plota o espectro do áudio filtrado
title('Espectro de Frequencia do Sinal');                 %Titulo do gráfico
xlabel('Frequencia');                                      %Titulo do eixo x
ylabel('Magnitude');                                       %Titulo do eixo y
sound(p,fa)                                        %Reproduz o áudio filtrado
audiowrite('C:\Users\Windows\Downloads\PDS\audio_filtrado_1.wav',p,fa);
%Grava o áudio filtrado
```

Figura 16. Espectro de frequência obtido após segunda filtragem



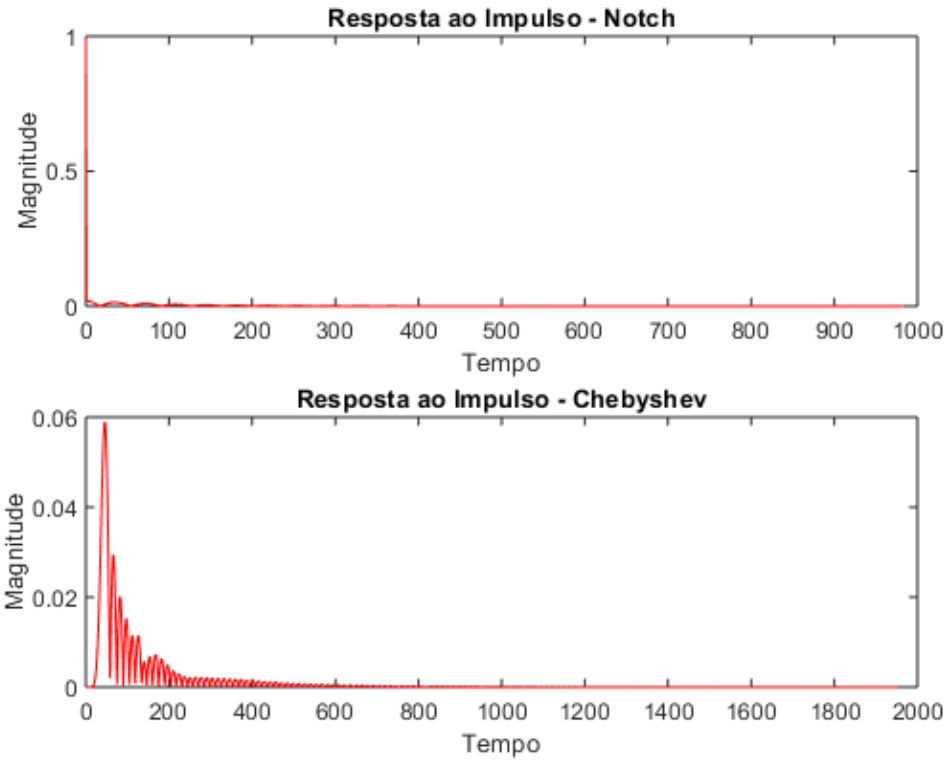
Fonte: Autoral.

Analisando a resposta em frequência final do áudio, pode-se observar que para frequências acima da frequência de corte de 800 Hz não há mais componentes de frequências, sendo elas a maior parte interferência e pequena parte sinal de áudio. Ao ouvir o áudio a voz humana masculina é distinguível e audível, no entanto, ela soa abafada, pois algumas de suas componentes não puderam ser separadas do ruído e foram eliminadas.

Já a resposta ao impulso dos filtros implementados foi gerada através do código mostrado abaixo e pode ser observada na Figura 17.

```
%Resposta ao impulso do Filtro Notch
figure;                                          %Novo grafico
[h1,w1] = impz(a,b);                  %coeficientes do filtro
subplot(2,1,1);         %define a escala e posiçao do gráfico
plot(w1,abs(h1), 'r');           %plota a resposta ao impulso
title('Resposta ao Impulso - Notch');      %Titulo do gráfico
xlabel('Tempo');                            %Titulo do eixo x
ylabel('Magnitude');                        %Titulo do eixo y

%Resposta ao impulso do Filtro Chebyshev
[h2,w2] = impz(t,u);                  %coeficientes do filtro
subplot(2,1,2);         %define a escala e posiçao do gráfico
plot(w2,abs(h2),'r');            %plota a resposta ao impulso
title('Resposta ao Impulso - Chebyshev'); %Titulo do gráfico
xlabel('Tempo');                            %Titulo do eixo x
ylabel('Magnitude');                        %Titulo do eixo y
```

Figura 17. Resposta ao impulso dos filtros utilizados



Fonte: Autoral.

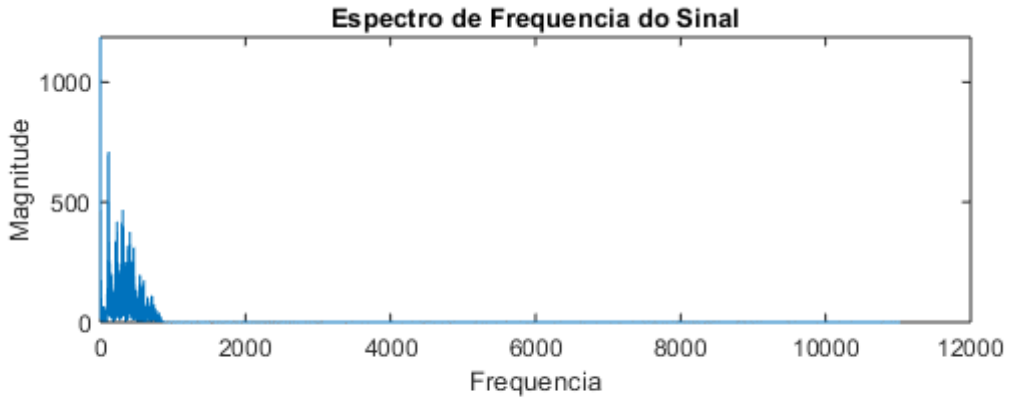
Observa-se que a resposta ao impulso do filtro Notch possui pequenas oscilações em torno do zero, enquanto a do filtro de Chebyshev produz uma resposta ao impulso na forma de uma onda senoidal com amplitude variando exponencialmente com efeito de amortecimento na oscilação.

4.2. SEGUNDA PARTE

Ao realizar a primeira filtragem do sinal de áudio com o filtro Notch, conforme implementado na primeira parte (vide Figura 15), a componente de 300 Hz foi eliminada. Ao realizar a segunda filtragem com o filtro FIR, a resposta em frequência da Figura 18 foi obtida. O seguinte código foi implementado:

```
figure;                                                    %Novo grafico
p= filtro_fir(q,y);                  %implementação da função desenvolvida
P = fft(p);                         %Calcula a transformada de Fourier de P
subplot(2,1,1);                       %define a escala e posiçao do gráfico
plot(f(1:N/2),abs(P(1:N/2)))            %plota o espectro do áudio filtrado
title('Espectro de Frequencia do Sinal');                %Titulo do gráfico
xlabel('Frequencia');                                     %Titulo do eixo x
ylabel('Magnitude');                                      %Titulo do eixo y
sound(p,fa)                                        %Reproduz o áudio filtrado
audiowrite('C:\Users\Windows\Downloads\PDS\audio_filtrado_2.wav',p,fa);
%Grava o áudio filtrado
```

Figura 18. Espectro de frequência obtido após segunda filtragem



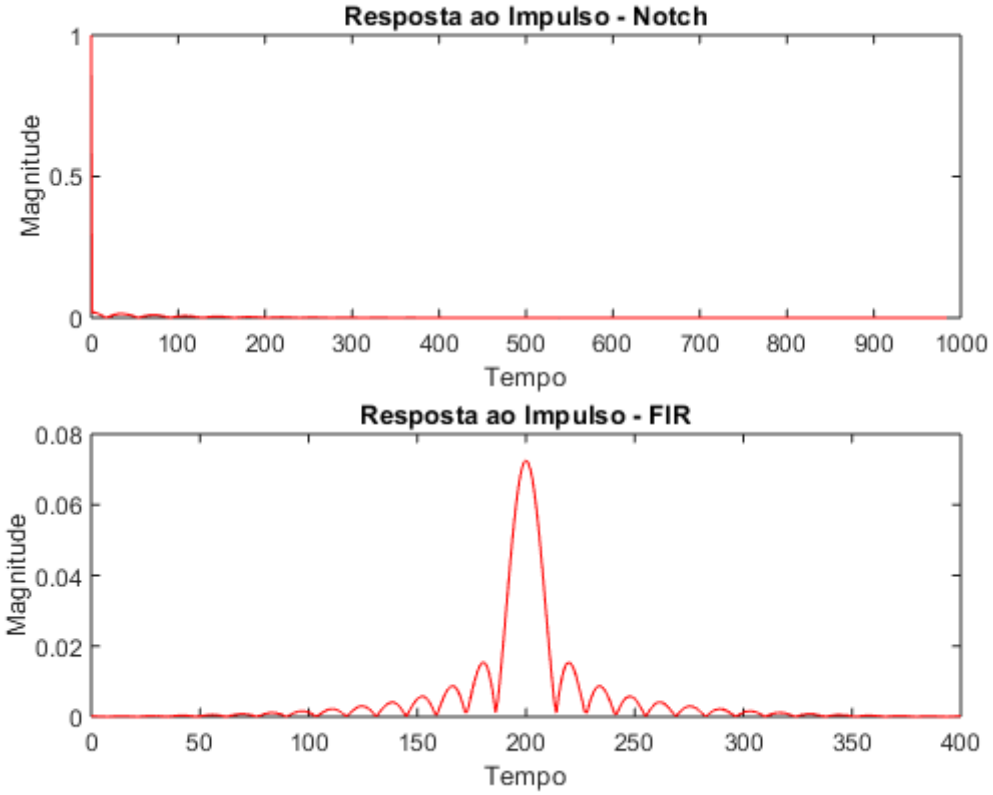
Fonte: Autoral.

A frequência de corte escolhida foi de 800 Hz, considerando que essa foi a frequência que melhor se adequou à filtragem na parte 1. Ao ouvir o áudio a voz humana masculina é distinguível e audível, no entanto, ela soa abafada, pois algumas de suas componentes não puderam ser separadas do ruído e foram eliminadas.

Já a resposta ao impulso dos filtros implementados foi gerada através do código mostrado abaixo e pode ser observada na Figura 19.

```
%Resposta ao impulso do Filtro Notch
figure;                                          %Novo grafico
[h1,w1] = impz(a,b);                  %coeficientes do filtro
subplot(2,1,1);         %define a escala e posiçao do gráfico
plot(w1,abs(h1), 'r');           %plota a resposta ao impulso
title('Resposta ao Impulso - Notch');     %Titulo do gráfico
xlabel('Tempo');                           %Titulo do eixo x
ylabel('Magnitude');                       %Titulo do eixo y

%Resposta ao impulso do Filtro FIR
[h2,w2] = impz(q,1);                  %coeficientes do filtro
subplot(2,1,2);         %define a escala e posiçao do gráfico
plot(w2,abs(h2),'r');            %plota a resposta ao impulso
title('Resposta ao Impulso - FIR');       %Titulo do gráfico
xlabel('Tempo');                           %Titulo do eixo x
ylabel('Magnitude');                       %Titulo do eixo y
```

Figura 19. Resposta ao impulso dos filtros utilizados



Fonte: Autoral.

Observa-se que a resposta ao impulso do filtro Notch possui pequenas oscilações em torno do zero, enquanto a do filtro FIR produz uma resposta ao impulso centralizada e que possui oscilações em sua cauda, sendo ela finita, enquanto a do Notch tende a infinito.

5. CONCLUSÃO

A decisão de implementar um filtro FIR ou IIR depende da sua aplicação, em que as características de cada tipo devem ser levadas em consideração. Como o filtro FIR apresenta memória finita, sempre será estável e qualquer transitório tem duração limitada. Entretanto, esse tipo de filtro necessita de vários coeficientes, exigindo mais memória, maior número de operações e, consequentemente, mais tempo de processamento por um processador. Por outro lado, os filtros IIR apresentam característica de corte de frequência acentuada com um filtro de ordem relativamente baixa, o que reduz a complexidade e o tempo de processamento.

O filtro que melhor atendeu a este trabalho foi o IIR, pois era necessário implementar um filtro que tivesse menor faixa de transitório possível, sendo sua descida o mais próxima da frequência de corte. Com a implementação do filtro FIR, mesmo com a utilização de uma ordem elevada, não foi possível obter a mesma inclinação em torno da frequência de corte obtida com a implementação de um filtro Chebyshev tipo I, sendo os resultados obtidos com este melhores que os obtidos com o FIR. Vale ressaltar que tanto o filtro FIR como o filtro IIR cumpriram seu objetivo de filtragem do sinal de áudio com interferências, e as diferenças percebidas foram mínimas.

6. REFERÊNCIAS

MATHWORKS. **Documentation.** MathWorks Corporation, 2017. Disponível em: Acesso em: 24 junho 2021.

OPPENHEIM, Alan V.; SCHAFER, Ronald W. **Digital signal processing.** [S.l.]: Prentice-Hall, 1998.

MARQUES, Miguel P. **Sistemas e Técnicas de Produção Áudio.** 1ª. ed. [S.l.]: FCA, 2014.

ROBERTS, M. J. **Fundamentos de Sinais e Sistemas.** Tradução de Carlos Henrique Nogueira de Resende Barbosa. Porto Alegre: AMGH, 2010.

LATHI, B. P. **Sinais e Sistemas Lineares.** 2ª. ed. Porto Alegre: Bookman, 2006.

INTRANET. **Filtro Passivo.** Disponível em: <https://intranet.ctism.ufsm.br/gsec/Apostilas/filtropassivo.pdf> Acesso em: 25 de junho de 2021.

CIN UFPE. **Tutorial Filtros.** Disponível em: <https://www.cin.ufpe.br/~es238/arquivos/laboratorios/tutorial_filtros> Acesso em: 25 de junho de 2021.

DECOM UNICAMP. **Slides - Projeto Filtros.** Disponível em: <https://www.decom.fee.unicamp.br/~rlopes/IE550/2S2013/Notas/Slides_ProjetoFiltros.pdf> Acesso em: 25 de junho de 2021.

ELE ITA. **Aula Filtros.** Disponível em: <http://www.ele.ita.br/~yukio/eea05/Aula3_EEA05_2018.pdf> Acesso em: 25 de junho de 2021.

NETO. **Estrutura de Filtros Digitais**. Disponível em: <https://slideplayer.com.br/slide/7796735/> Acesso em: 25 de junho de 2021.

FAMBRINI. **Conversor Analogico - Digital.** Disponível em: <https://pt.slideshare.net/ffambrini/slides-9-erm> Acesso em: 25 de junho de 2021.

SANTANA. **Projeto de controladores digitais no domínio da frequência.** Disponível em: <http://professor.ufop.br/sites/default/files/adrielle/files/projetofrequencia1.pdf> Acesso em: 25 de junho de 2021.

7. ANEXOS

Os códigos completos implementados na primeira e na segunda parte foram exportados do Matlab para um arquivo em pdf e seguem nas páginas finais do trabalho. Os códigos das funções para o filtro FIR e IIR foram feitas em arquivos .m separados, seguem:

```
%Beatriz Cristina Pereira de Assis
%Projeto PDS - FUNÇÃO FILTRO IIR

function y= filtro_iir(a,b,x)

  M = length(x);                        %tamanho do vetor do sinal de entrada
  N = length(a);                              %tamanho do filtro IIR (Ordem)
  T = length(b);                  %tamanho do vetor do sinal de realimentacao

  y = zeros(1,M);               %gera vetor para armazenar a saida do sistema
  aux = zeros(N,1);       %gera vetor com os coeficientes da eq. dif. entrada
  aux1 = zeros(T-1,1);      %gera vetor com os coeficientes da eq. dif. saída

for i = 1:M
    aux = circshift(aux,1);   % desloca circularmente os valores de aux em 1
    aux(1) = x(i);      %atualiza a 1 posição de aux com a amostra atual de x

    y(i) = ((a*aux)-(b(2:T)*aux1));                        %saida do sistema

    aux1 = circshift(aux1,1); %desloca circularmente os valores de aux1 em 1
    aux1(1) = y(i);   %atualiza a 1 posição de aux1 com a amostra atual de y
end
end
```

```
%Beatriz Cristina Pereira de Assis
%Projeto PDS - FUNÇÃO FILTRO FIR

function y= filtro_fir(b,x)

M= length(x);                         %tamanho do vetor do sinal de entrada
N= length(b);                               %tamanho do filtro FIR (Ordem)
y= zeros(1,M);           % gera o vetor para armazenar a saida do sistema
aux= zeros(1,N);        %gera vetor auxiliar para fazer a eq de diferenca
for i= 1:M
    aux= circshift(aux,1); %desloca circularmente os valores de aux em 1
    aux(1)=x(i);   %atualiza a 1 posição de aux com a amostra atual de x
    y(i)= b*aux';                                          %Saida do sistema
end
end
```

```
%Beatriz Cristina Pereira de Assis
%Projeto PDS - Filtragem de áudio com filtros IIR

%Analise no dominio do tempo
[x,fa] = audioread('fala_sirene_tm1.wav'); %leitura do audio
subplot(2,1,1);                  %define a escala do gráfico
plot(x(1:400));             %plota as 400 primeiras amostras
title('Sinal no dominio do Tempo');       %Titulo do gráfico
xlabel('Tempo');                           %Titulo do eixo x
ylabel('Magnitude');                       %Titulo do eixo y
figure;

%Análise espectral da frequencia
X = fft(x);                         %Faz a transformada de fourier
N = length(X);                        %Pega a quantidade de pontos
X = X/(N/2);                                        %Normalizacao
f = [0: N-1]*fa/(N-1); %Vetor que contem as frequencias de f a fa
subplot(2,1,1);                        %define a escala do gráfico
plot(f(1:N/2),abs(X(1:N/2)));           %Plota f sem espelhamento
title('Espectro de Frequencia do Sinal');      %Titulo do gráfico
xlabel('Frequencia');                           %Titulo do eixo x
ylabel('Magnitude');                            %Titulo do eixo y

%.........................PRIMEIRA PARTE.............................

%Primeira Filtragem - Notch
wc = 2*pi*300;                  %Frequência de corte do filtro Notch
r =0.99;                                        %Raio do filtro Notch
a =[1 -2*cos(wc/fa) 1];                   %Numerador do filtro Notch
b =[1 -2*r*cos(wc/fa) r^2];             %Denominador do filtro Notch
[H,W] = freqz(a,b,512,fa);   %Resposta em frequência do filtro Notch
hold on                                   %Mantém o gráfico anterior
plot(W,abs(H),'r');  %plota a resposta em frequencia do filtro Notch

y = filtro_iir(a,b,x); % Aplicando o filtro Notch ao sinal original
Y = fft(y);                 %Calcula a transformada de Fourier de y
subplot(2,1,2);               %define a escala e posiçao do gráfico
plot(f(1:N/2),abs(Y(1:N/2)))    %plota o espectro do áudio filtrado

%Segunda Filtragem - Chebyshev Tipo I
wn = 800/(fa/2);                            %Frequencia de corte
[t,u]= cheby1(10,1,wn);                        %Filtro Chebyshev
[h,w] = freqz(t,u,512,fa);    %Resposta em frequência do filtro
hold on;                              %Mantém o gráfico anterior
plot(w, 2000*abs(h)); %plota a resposta em frequencia do filtro
title('Espectro de Frequencia do Sinal');    %Titulo do gráfico
xlabel('Frequencia');                         %Titulo do eixo x
ylabel('Magnitude');                          %Titulo do eixo y

figure;                                                 %Novo grafico
p = filtro_iir(t,u,y);              %implementação da função desenvolvida
```

```
52 P = fft(p);                          %Calcula a transformada de Fourier de P
53 subplot(2,1,1);                        %define a escala e posiçao do gráfico
54 plot(f(1:N/2),abs(P(1:N/2)))             %plota o espectro do áudio filtrado
55 title('Espectro de Frequencia do Sinal');               %Titulo do gráfico
56 xlabel('Frequencia');                                    %Titulo do eixo x
57 ylabel('Magnitude');                                    %Titulo do eixo y
58 sound(p,fa)                                      %Reproduz o áudio filtrado
59 audiowrite('C:\Users\Windows\Downloads\PDS\audio_filtrado_1.wav',p,fa);
60 %Grava o áudio filtrado
61
62 %Resposta ao impulso do Filtro Notch
63 figure;                                          %Novo grafico
64 [h1,w1] = impz(a,b);                  %coeficientes do filtro
65 subplot(2,1,1);         %define a escala e posiçao do gráfico
66 plot(w1,abs(h1), 'r');           %plota a resposta ao impulso
67 title('Resposta ao Impulso - Notch');     %Titulo do gráfico
68 xlabel('Tempo');                            %Titulo do eixo x
69 ylabel('Magnitude');                        %Titulo do eixo y
70
71 %Resposta ao impulso do Filtro Chebyshev
72 [h2,w2] = impz(t,u);                  %coeficientes do filtro
73 subplot(2,1,2);         %define a escala e posiçao do gráfico
74 plot(w2,abs(h2),'r');           %plota a resposta ao impulso
75 title('Resposta ao Impulso - Chebyshev'); %Titulo do gráfico
76 xlabel('Tempo');                            %Titulo do eixo x
77 ylabel('Magnitude');                        %Titulo do eixo y
78
```

```
%Beatriz Cristina Pereira de Assis
%Projeto PDS - Filtragem de áudio com filtros FIR e IIR

%Analise no dominio do tempo
[x,fa] = audioread('fala_sirene_tm1.wav'); %leitura do audio
subplot(2,1,1);                  %define a escala do gráfico
plot(x(1:400));             %plota as 400 primeiras amostras
title('Sinal no dominio do Tempo');       %Titulo do gráfico
xlabel('Tempo');                           %Titulo do eixo x
ylabel('Magnitude');                       %Titulo do eixo y
figure;                                        %Novo grafico

%Análise espectral da frequencia
X = fft(x);                          %Faz a transformada de fourier
N = length(X);                         %Pega a quantidade de pontos
X = X/(N/2);                                        %Normalizacao
f = [0: N-1]*fa/(N-1); %Vetor que contem as frequencias de f a fa
subplot(2,1,1);                         %define a escala do gráfico
plot(f(1:N/2),abs(X(1:N/2)));           %Plota f sem espelhamento
title('Espectro de Frequencia do Sinal');      %Titulo do gráfico
xlabel('Frequencia');                           %Titulo do eixo x
ylabel('Magnitude');                            %Titulo do eixo y

%.............................SEGUNDA PARTE.............................
%Primeira Filtragem - Notch
wc = 2*pi*300;                  %Frequência de corte do filtro Notch
r =0.99;                                        %Raio do filtro Notch
a =[1 -2*cos(wc/fa) 1];                  %Numerador do filtro Notch
b =[1 -2*r*cos(wc/fa) r^2];              %Denominador do filtro Notch
[H,W] = freqz(a,b,512,fa);   %Resposta em frequência do filtro Notch
hold on                                  %Mantém o gráfico anterior
plot(W,abs(H),'r');  %plota a resposta em frequencia do filtro Notch

y = filtro_iir(a,b,x); % Aplicando o filtro Notch ao sinal original
Y = fft(y);                 %Calcula a transformada de Fourier de y
subplot(2,1,2);               %define a escala e posiçao do gráfico
plot(f(1:N/2),abs(Y(1:N/2)))    %plota o espectro do áudio filtrado

%Segunda Filtragem - FILTRO FIR
wn = 2*(800/fa);                            %Frequencia de corte
q =fir1(400, wn);                                   %Filtro FIR
[h,w]=freqz(q,1,512,fa);      %Resposta em frequencia do filtro
hold on;                                 %Mantém o gráfico anterior
plot(w,2000*abs(h));  %plota a resposta em frequencia do filtro
title('Espectro de Frequencia do Sinal');    %Titulo do gráfico
xlabel('Frequencia');                         %Titulo do eixo x
ylabel('Magnitude');                          %Titulo do eixo y

figure;                                                 %Novo grafico
p= filtro_fir(q,y);                  %implementação da função desenvolvida
P = fft(p);                         %Calcula a transformada de Fourier de P
```

```
52 subplot(2,1,1);                      %define a escala e posiçao do gráfico
53 plot(f(1:N/2),abs(P(1:N/2)))          %plota o espectro do áudio filtrado
54 title('Espectro de Frequencia do Sinal');              %Titulo do gráfico
55 xlabel('Frequencia');                                   %Titulo do eixo x
56 ylabel('Magnitude');                                    %Titulo do eixo y
57 sound(p,fa)                                        %Reproduz o áudio filtrado
58 audiowrite('C:\Users\Windows\Downloads\PDS\audio_filtrado_2.wav',p,fa);
59 %Grava o áudio filtrado
60
61 %Resposta ao impulso do Filtro Notch
62 figure;                                          %Novo grafico
63 [h1,w1] = impz(a,b);                 %coeficientes do filtro
64 subplot(2,1,1);        %define a escala e posiçao do gráfico
65 plot(w1,abs(h1), 'r');          %plota a resposta ao impulso
66 title('Resposta ao Impulso - Notch');     %Titulo do gráfico
67 xlabel('Tempo');                            %Titulo do eixo x
68 ylabel('Magnitude');                        %Titulo do eixo y
69
70 %Resposta ao impulso do Filtro FIR
71 [h2,w2] = impz(q,1);                 %coeficientes do filtro
72 subplot(2,1,2);        %define a escala e posiçao do gráfico
73 plot(w2,abs(h2),'r');           %plota a resposta ao impulso
74 title('Resposta ao Impulso - FIR');       %Titulo do gráfico
75 xlabel('Tempo');                            %Titulo do eixo x
76 ylabel('Magnitude');                        %Titulo do eixo y
```