

**INSTITUTO FEDERAL**  
**CEARÁ**  
**Campus Fortaleza**

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO CEARÁ  
Engenharia de Telecomunicações – Departamento de Telemática

## **Trabalho Implementação de Filtragem de Sinal de Áudio Utilizando os Tipos de Filtro FIR e IIR**

**Disciplina:** Filtros Digitais

**Autor:** Wellington Dantas de Almeida

**Professor:** Dr. Ricardo Rodrigues de Araújo

Campus Fortaleza  
Setembro de 2016

## SUMÁRIO

1. INTRODUÇÃO .....	2
2. FUNDAMENTAÇÃO TEÓRICA .....	2
2.1 Filtros FIR.....	2
2.1.1 Projeto de Filtros FIR .....	3
<i>Método das Janelas</i> .....	3
2.2 Filtros IIR.....	4
2.2.1 Projeto de Filtro IIR.....	5
<i>Filtro de Butterworth</i> .....	6
<i>Filtro de Chebyshev</i> .....	7
<i>Filtro Elíptico</i> .....	8
<i>Transformação Bilinear</i> .....	8
<i>Filtro Notch</i> .....	9
3. METODOLOGIA .....	10
<i>Filtragem com FIR</i> .....	10
<i>Filtragem com IIR</i> .....	10
4. RESULTADOS .....	11
<i>Resultados com o filtro FIR</i> .....	11
<i>Resultados com o filtro IIR</i> .....	14
<i>Resultados comparativos</i> .....	17
5. CONCLUSÃO .....	17
6. REFERÊNCIAS .....	17
7. ANEXOS.....	17

## 1. INTRODUÇÃO

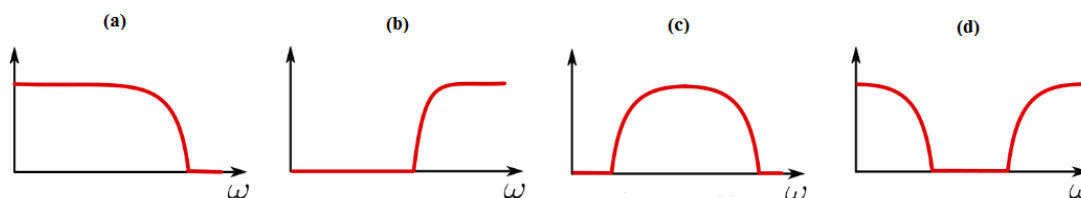
Um filtro digital é um sistema que retira alguns componentes de frequência de um sinal, mas considerando um contexto mais amplo, qualquer sistema que modifica certas frequências em relação ao seu estado inicial é também chamado de filtro.

Em sistemas de comunicação, mais notadamente em sinais de áudio é possível encontrar sons, ruídos e ou vibrações juntos ao sinal de voz, e consequentemente interferido nas comunicações de dados, e umas das soluções que pode ser encontrada para eliminar essas interferências no áudio é com a aplicação de filtros.

Com isso, este trabalho tem como objetivo implementar estruturas de filtros digitais do tipo FIR e IIR para retirar interferências sobre o som da voz humana, deixando-a o mais audível possível. Também tem como objetivo fazer a comparação dos resultados adquiridos quando se usa os filtros FIR e IIR. Sendo que muitos dos cálculos e resultados foram alcançados com a utilização de *softwares* de computação numérica como o *Matlab*, *Scilab* e ou *Octave*.

## 2. FUNDAMENTAÇÃO TEÓRICA

Os filtros digitais são sistemas discretos que atuam sobre sequências alterando suas características em função de seus componentes espectrais. Em outras palavras, no processamento digital de sinais, a função dos filtros digitais consiste em retirar as partes indesejadas do sinal, tais como o ruído aleatório, ou para extrair partes úteis do sinal. Na Figura 1 é apresentado quatro formas de retirar ruídos indesejados de sinais no domínio da frequência.



**Figura 1.** Filtros com resposta em frequência: (a) passa-baixa; (b) passa-alta; (c) passa-banda e (d) rejeita-banda.

Os filtros digitais são divididos em dois tipos: Filtros FIR (em inglês, *Finite Impulse Response*) e Filtros IIR (em inglês, *Infinite Impulse Response*), que serão enfatizados neste trabalho.

### 2.1 Filtros FIR

Um filtro de resposta ao impulso finito é uma estrutura de filtragem que pode ser utilizado para implementar quase que qualquer tipo de resposta de frequência digital. Esse filtro tem como características básicas:

- A resposta ao impulso finito;
- Natureza não recursiva;
- Estabilidade BIBO estáveis;

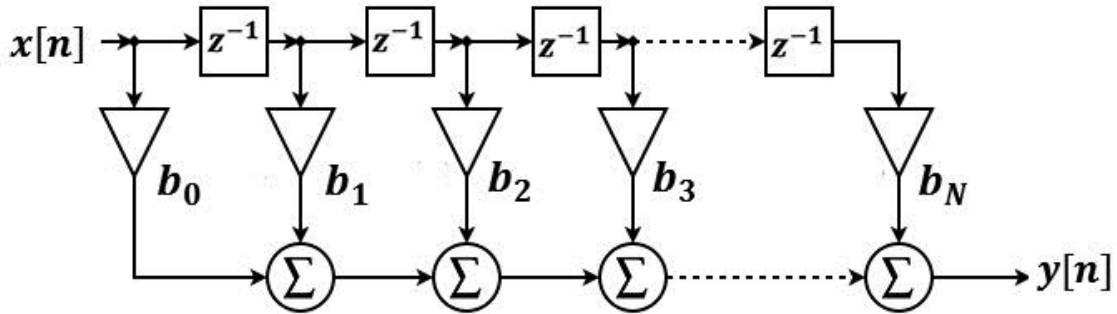
- Fase linear e memória finita.

Este tipo de filtro apresenta a função de transferência discreta, como apresentado na Equação 1, ou pode ser reescrito como uma função polinomial com potências negativas de Z na Equação 2.

$$\frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^M a_k z^{(M-k)}}{z^M} \quad (1)$$

$$H(z) = b_0 + b_1 z^{-1} + \dots + b_N z^{-N} \quad (2)$$

Um filtro FIR é normalmente implementado por meio de uma série de atraso, multiplicadores e adicionadores para criar uma saída do filtro, como o da Figura 2.



**Figura 2.** Estrutura de diagrama de blocos de um filtro FIR.

Em comparação com os filtros IIR, o filtro FIR tem a desvantagem de que eles requerem mais memória e são mais lentos para alcançar uma dada característica de resposta do filtro. Porém, as vantagens de filtros FIR superam as suas desvantagens, por isso eles são, em algumas ocasiões, mais utilizados do que o filtro IIR. Uma entre as suas vantagens, é destacado a simplicidade de implementá-lo.

### 2.1.1 Projeto de Filtros FIR

#### *Método das Janelas*

Seja a resposta impulsiva de um filtro ideal  $h_d[n]$  e  $h[n]$  a resposta ao impulso do filtro FIR. Deseja-se encontrar uma função Equação 3 de maneira a minimizar a função de erro quadrático.

$$E = \sum_{n=-\infty}^{\infty} |h_d[n] - h[n]|^2 \quad (3)$$

Os parâmetros ajustáveis são os coeficientes do filtro  $h[n]$ , apresentados nas equações abaixo.

$$h[n] = \begin{cases} h_d[h], & 0 \leq h \leq N \\ 0, & \text{caso contrário} \end{cases}$$

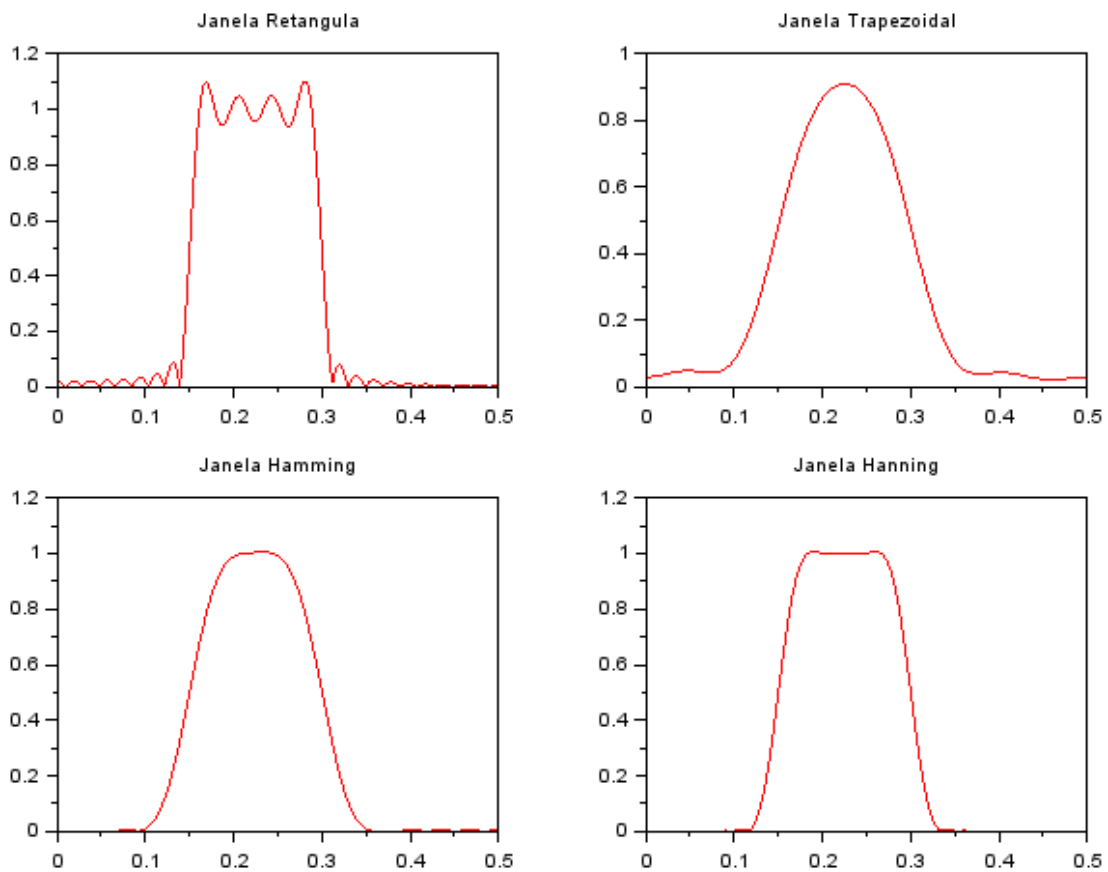
$$h[n] = h_d[n] \cdot \omega[n]$$

$$\text{onde } \omega[n] = \begin{cases} 1, & 0 \leq h \leq N \\ 0, & \text{caso contrário} \end{cases}$$

O erro quadrático de aproximação será igual ao apresentado na Equação 4.

$$E = \sum_{n=-\infty}^{-1} |h_d[n]|^2 + \sum_{n=N+1}^{\infty} |h_d[n]|^2 \quad (4)$$

Na Figura 3 é apresentado algumas janelas utilizando o filtro passa-banda, tais como: a janela retangular; trapezoidal; hamming e hanning.



**Figura 3.** Método das janelas do filtro FIR.

## 2.2 Filtros IIR

Um filtro de resposta ao impulso infinito tem como características a entrada e saída regidas por equações de diferença recursiva e também com coeficientes constantes de natureza recursiva: Algumas características básicas desse filtro são:

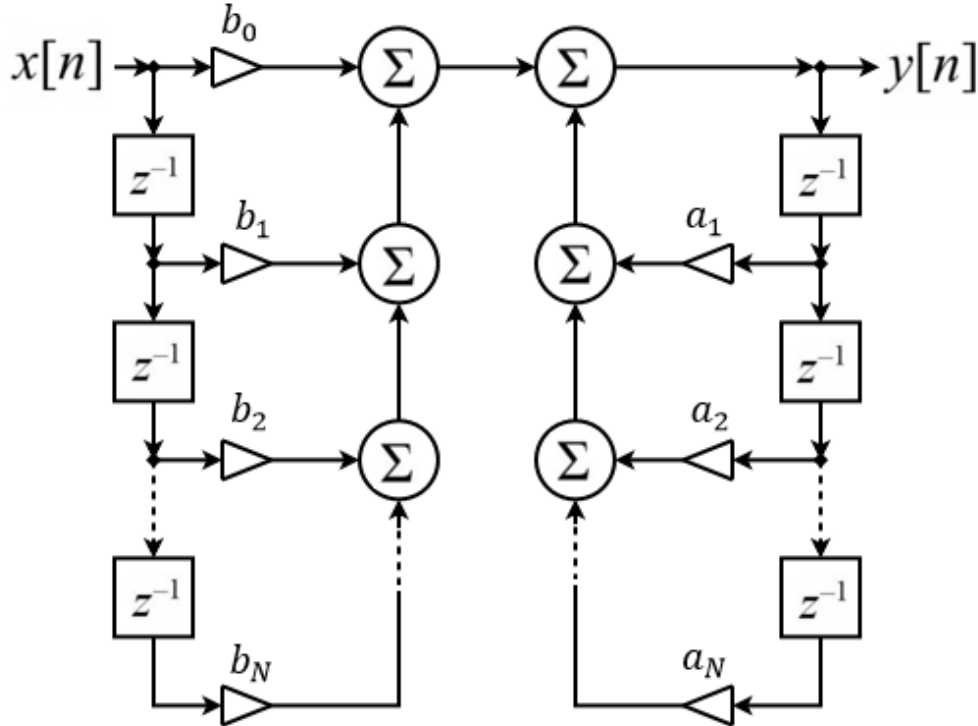
- Resposta ao impulso infinito;

- Estabilidade depende da posição dos polos;
- Resposta em fase não linear.

E apresentam uma dada resposta ao impulso usando uma ordem menor comparado com os filtros FIR. Este tipo de filtro apresenta a função de transferência discreta, como apresentado na Equação 5 ou pode ser reescrito como uma função polinomial com potências negativas de  $z$  na Equação 6. Na Figura 4 apresenta a estrutura de um Filtro IIR.

$$\frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^M a_k z^{(M-k)}}{\sum_{j=0}^N b_j z^{(N-k)}} \quad (5)$$

$$H(z) = \frac{b_0 + b_1 z^{-1} + \dots + b_N z^{-N}}{1 + a_1 z^{-1} + \dots + a_N z^{-N}} \quad (6)$$



**Figura 4.** Estrutura de diagrama de blocos de um filtro IIR.

### 2.2.1 Projeto de Filtro IIR

Nesse projeto, o filtro digital com função de transferência  $H(z)$  é criado a partir de uma mudança de variável, a função  $H(s)$  em Laplace é convertida em  $H(z)$ , funções que geram respostas em frequência que atendam as especificações de filtro. As funções de aproximação mais usadas são:

- Butterworth;
- Chebyshev;
- Elíptica.

As especificações de um filtro podem ser:

- Banda passante;
- Banda de rejeição;
- Banda de transição;
- Ripple na banda passante;
- Ripple na banda de rejeição.

### ***Filtro de Butterworth***

O filtro de *Butterworth* é um tipo de filtro de processamento de sinais concebido para ter uma resposta de frequência tão plana quanto possível na faixa de passagem. A função de *butterworth* de ordem  $N$  é apresentado na Equação 7.

$$|H(\omega)|^2 = \frac{1}{1 + \left(\frac{\omega}{\omega_c}\right)^{2N}} \quad \text{e ou} \quad \omega_p = \omega_p \left(\frac{\varepsilon}{1-\varepsilon}\right)^{\frac{1}{2N}} \quad \text{e ou} \quad \omega_s = \omega_c \left(\frac{1-\delta}{\delta}\right)^{\frac{1}{2N}} \quad (7)$$

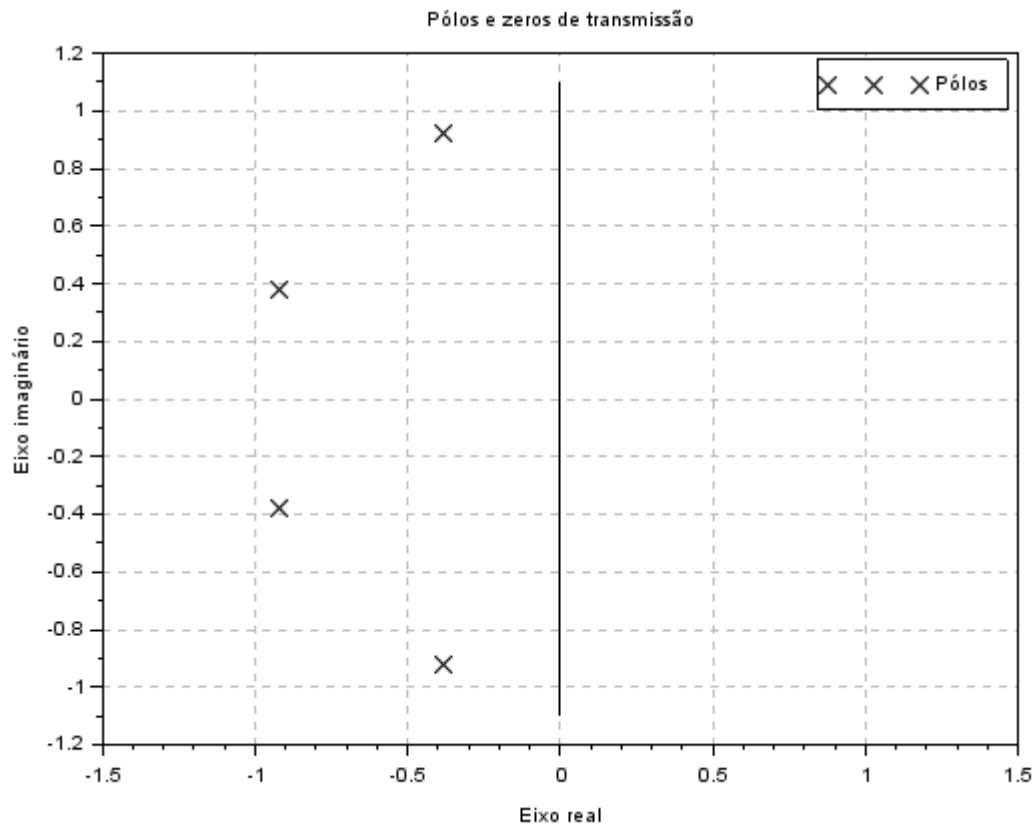
Sendo que  $\omega_c$  é a frequência de corte. Abaixo é demonstrado algumas equivalências de matemáticas.

$$|H(j\omega)|^2 = H(s) \cdot H(-s) \Big|_{s=j\omega}$$

$$H(s) \cdot H(-s) = \frac{1}{1 + \left(\frac{s}{\omega_c}\right)^{2N}}, \text{ cujo os polos são } s = \frac{\omega_c e^{j\pi(2k+N+1)}}{2N}, \text{ para } k = 0, 1, \dots, 2N - 1.$$

Para  $N=4$  teríamos o seguinte polinômio característico Equação 8. E na Figura 5 o diagrama de polos e zeros da respectiva Equação 8.

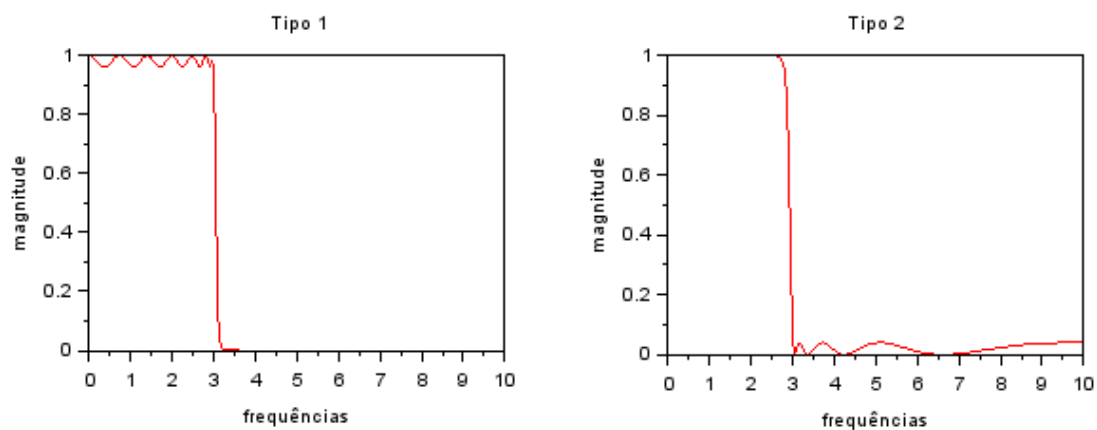
$$H(s) = \frac{1}{(s^2 + 0,765s + 1)} \cdot \frac{1}{(s^2 + 1,848s + 1)} \quad (8)$$



**Figura 5.** Diagrama de polos e zeros para  $N=4$ .

### *Filtro de Chebyshev*

Este tipo de filtro possui oscilações na banda passante (tipo 1) ou na banda de rejeição (tipo 2). Os filtros *Chebyshev* possuem uma menor faixa de transição se comparado ao filtro *Butterworth*, tornando-o nesse aspecto mais próximo a característica de um filtro ideal. Na figura 6 é apresentado os dois tipos de filtro *chebyshev*.

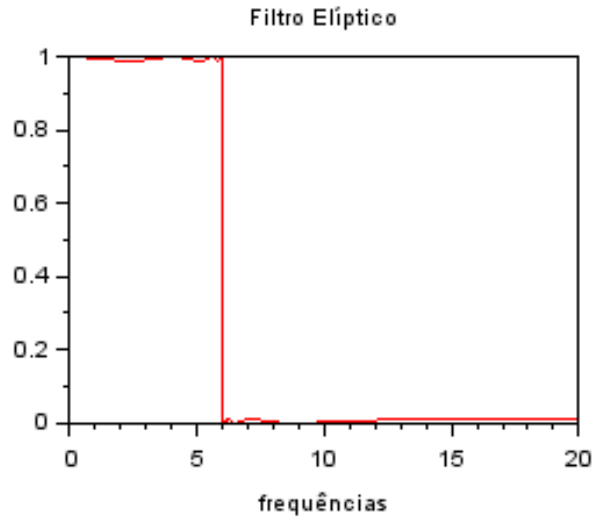


**Figura 6.** Plotagem dos tipos de filtro de chebyshev, tipo 1 e tipo 2.



### ***Filtro Elíptico***

Esse filtro é formado pela combinação de filtro de *chebyshev* tipo 1 e tipo 2. Possui oscilações na banda passante e rejeição. A principal vantagem desse filtro está em sua banda de transição, na qual é mais rápida dentre os filtros. Na Figura 7 é apresentado esse filtro.



**Figura 7.** Plotagem do filtro elíptico.

### ***Transformação Bilinear***

A transformação bilinear é uma relação matemática que pode ser utilizado para converter a função de transferência de um filtro específico no domínio de Laplace complexo para o domínio Z, e vice-versa. Por este método, a função de transferência do filtro digital IIR é obtido a partir da função de transferência de um filtro analógico (protótipo). Abaixo é demonstrado algumas equivalências de matemáticas.

Fazendo  $s = \frac{z-1}{z+1}$  obtém-se a função de transferência em Z.

$$H(z) = H(s) \Big|_{s=\frac{z-1}{z+1}} \quad s = \frac{z-1}{z+1} \rightarrow s(z+1) = z-1 \rightarrow sz - z = -s - 1 \rightarrow$$

$$z(s-1) = -s-1 \rightarrow z = \frac{s+1}{s-1}$$

- Forma triangula

$$s = \delta + j\omega$$

- Forma Polar

$$z = re^{j\omega}$$

$$z = \frac{\sigma + j\omega + 1}{1 - \sigma - j\omega} \quad r = |z| = \left( \frac{(1 + \sigma)^2 + \omega^2}{(1 - \sigma)^2 + \omega^2} \right)^{\frac{1}{2}}$$

$$\varphi = \arg(z) = \tan^{-1}\left(\frac{\omega}{1 + \sigma}\right) + \tan^{-1}\left(\frac{\omega}{1 - \sigma}\right)$$

### Pré-warping

É um procedimento realizado para compensar a distorção de frequência (*warping*), produzida pela transformação bilinear. Assim, as frequências a serem usadas no protótipo analógico devem ser obtidos como a apresentada na Figura 8.

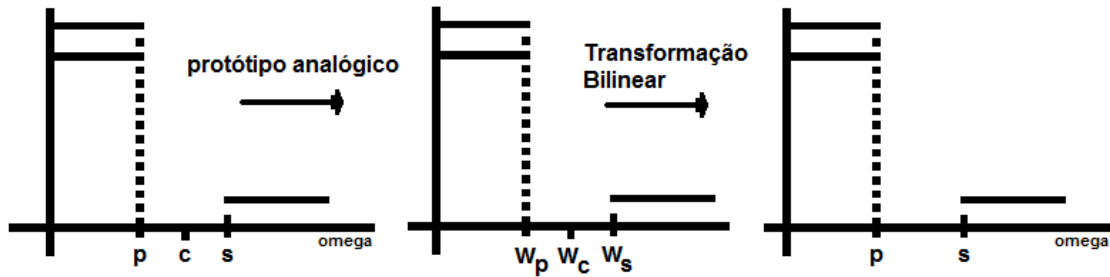


Figura 8. Representação da transformação bilinear.

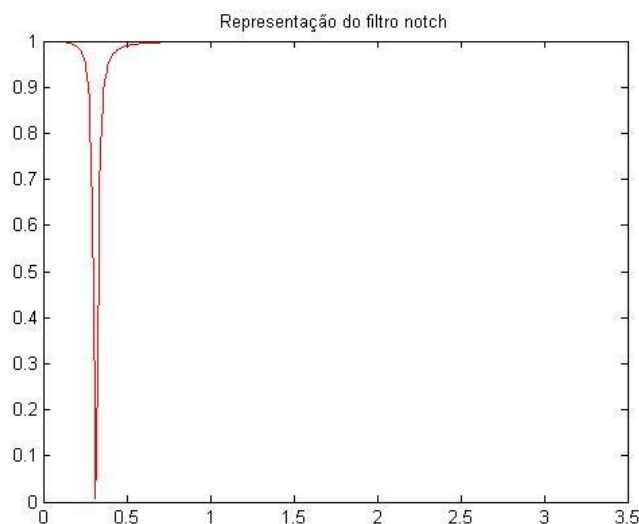
$$\omega = \tan\left(\frac{\omega_a}{2}\right) \quad \omega'_p = \tan\left(\frac{\omega_{a_p}}{2}\right) \quad \omega'_s = \tan\left(\frac{\omega_{a_s}}{2}\right)$$

$$\omega_{a'_p} = 2 \tan^{-1}(\omega'_p) = 2 \tan^{-1}\left(\tan\left(\frac{\omega_{a_p}}{2}\right)\right) = \omega_{a_p}$$

$$\omega_{a'_s} = 2 \tan^{-1}(\omega'_s) = 2 \tan^{-1}\left(\tan\left(\frac{\omega_{a_s}}{2}\right)\right) = \omega_{a_s}$$

### Filtro Notch

No processamento de sinais, o filtro *notch* é capaz de rejeitar uma faixa de frequência bastante estreita. Geralmente é utilizado para eliminar intervalos de frequência muito específico. Esse filtro tem como vantagem, o fato de atuar em intervalos pequenos de frequência, e interferir pouco na qualidade do sinal. Na Figura 9 é apresentado o comportamento do filtro rejeita-faixa de *notch*.



**Figura 9.** Representação do filtro Notch.

### 3. METODOLOGIA

Para o desenvolvimento desse trabalho, foi utilizado um sinal de voz de áudio, com ruídos e interferências, e o mesmo deveria passar por um tratamento de filtragem, a ponto de ser possível a verificação da informação que o sinal de voz transmite. Mais precisamente foi utilizado um sinal áudio onde a mesma continha o ruído de uma sirene, e vibrações sendo emitida ao mesmo tempo em que o tom da fala humana, deixando a voz humana sem som audível, por causa das interferências.

Para a análise do sinal, é utilizando o *software* de computação numérica Matlab®, que foi de grande importância no processamento do sinal utilizado. O mesmo captura informações do arquivo de áudio, como por exemplos, números de amostras, frequência, e número de bits. Com essas informações é possível plotar o espectro do sinal de áudio e verificar o seu comportamento.

Analisando o espectro do sinal é possível verificar quais ruídos devem ser filtrados. Isso porque a fala humana está no intervalo de 50Hz à 4000Hz, assim, frequências maiores que as do intervalo podem ser descartadas do sinal que realmente deseja-se ter. Para o procedimento de filtragem, foram utilizados os dois tipos de filtros, FIR e IIR.

#### ***Filtragem com FIR***

Para a filtragem com o filtro FIR, foi utilizado o método das janelas, de onde foi dividido em dois estágios. No primeiro estágio foi desenvolvido uma equação que gera o rejeita-faixa. No segundo estágio foi utilizado uma função própria do Matlab© que gera o filtro passa-baixa.

#### ***Filtragem com IIR***

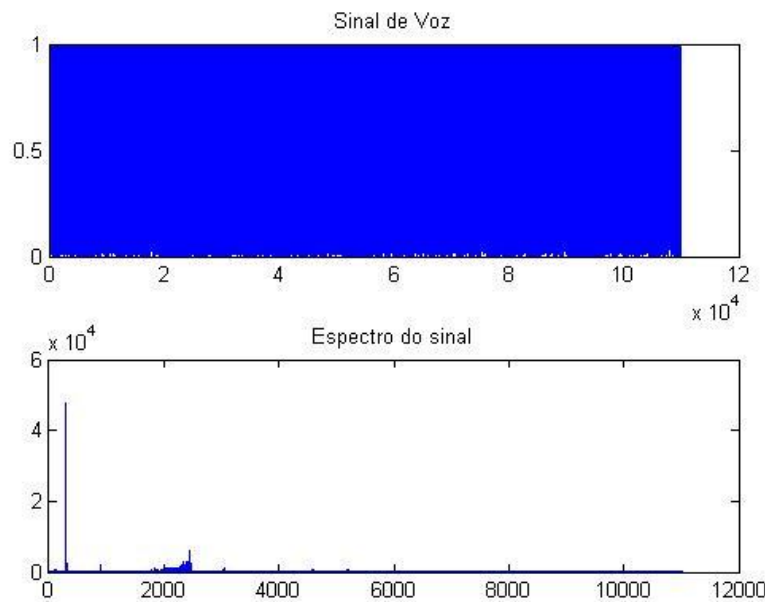
Para a filtragem do sinal com o filtro IIR, foi utilizado dois tipos de projetos de filtros, divididos em dois estágios, no primeiro estágio é aplicado duas vezes o filtro rejeita-faixa de *notch* para filtrar uma frequência indesejada, que se encontra no intervalo

de frequências da voz humana, e no segundo estágio é utilizado um filtro passa-baixa de *butterworth*.

O passa-baixa de *butterworth* foi escolhido por ser o mais adequado para retirar as frequências que estão acima da faixa de frequências da voz humana, além disso, esse filtro possui uma resposta mais próxima à ideal. O rejeita-faixa de *notch* foi escolhido por conseguir retirar uma pequena faixa de frequência que se encontrava dentro do intervalo de frequências da voz.

#### 4. RESULTADOS

Os dois tipos de filtros filtraram o mesmo arquivo de áudio disponibilizado, e os resultados se comportaram de maneiras semelhantes. Na figura 10 é apresentado o sinal do áudio e o seu espectro em função da frequência, comum aos dois tipos de filtros.



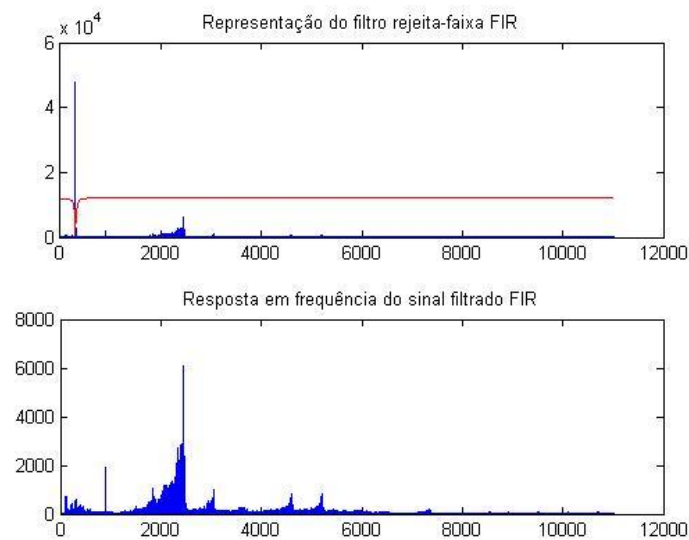
**Figura 10.** Plotagem do sinal de áudio e seu respectivo espectro.

##### ***Resultados com o filtro FIR***

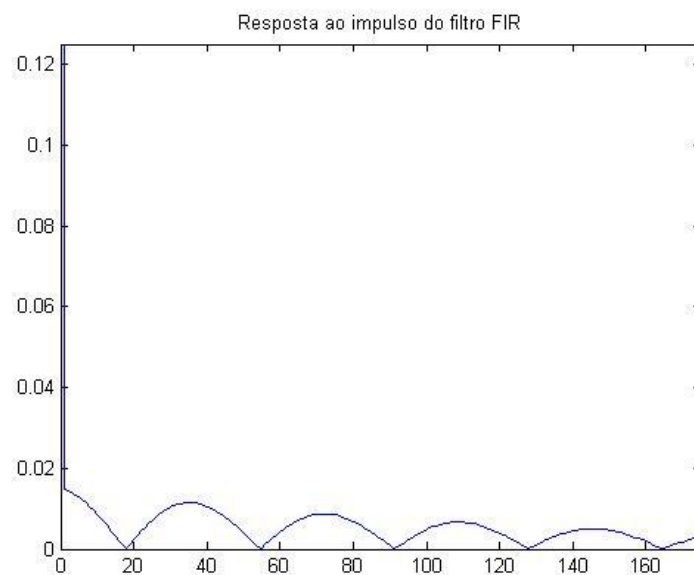
Na filtragem do filtro FIR, foi verificado que a frequência de 300Hz interferia no sinal de áudio, com isso, foi projetado um filtro rejeita-faixa com as seguintes características.

- $f_c=300\text{Hz}$ , de frequência de corte;
- $r=0.9926$ , coeficiente de regulação;
- $w_c=0.090702$ , de frequência de corte discreta.

Na Figura 11 é apresentado o filtro rejeita-faixa do tipo FIR, projetado com os valores acima. Percebe-se que o filtro eliminou somente a frequência de 300Hz do sinal, a Figura 12 apresenta a respectiva resposta ao impulso desse filtro.



**Figura 11.** Representação do filtro rejeita-faixa FIR (cor vermelho) filtrando o sinal, e a forma do sinal depois de filtrado.



**Figura 12.** Resposta ao impulso do filtro rejeita faixa FIR.

Na Tabela 1 são mostrados os respectivamente os coeficientes numerador e denominador do filtro rejeita-faixa FIR implementado neste projeto.

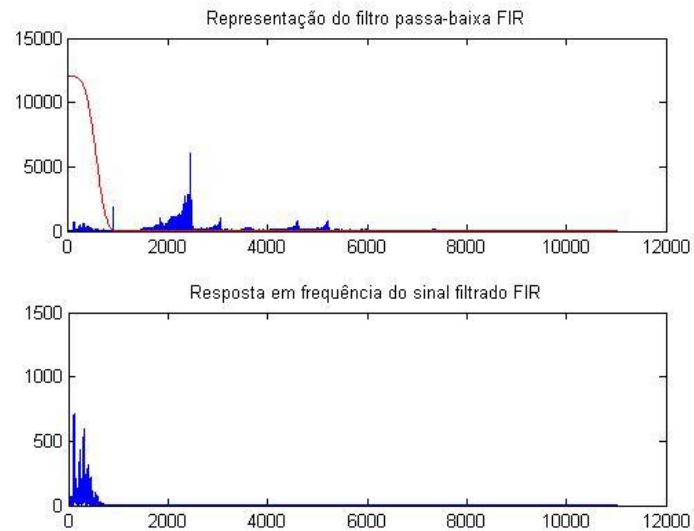
**Tabela 1.** Coeficientes dos filtros rejeita-faixa FIR.

Numerador	1.0000 -1.9927 1.0000
Denominador	1.0000 -1.9780 0.9853

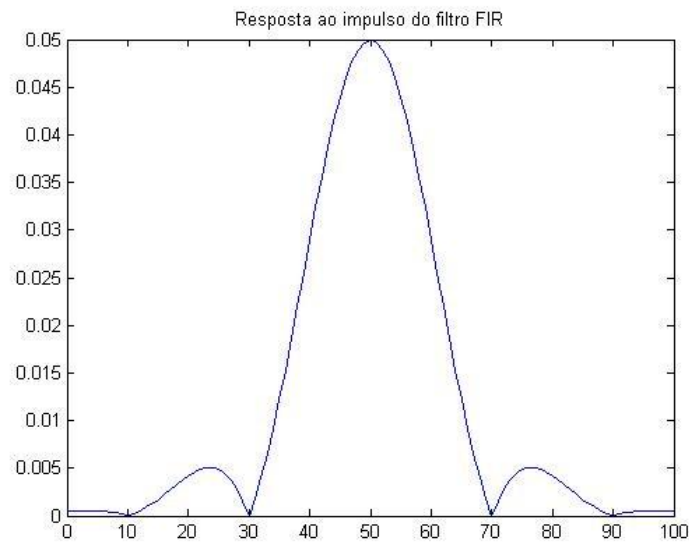
Foi verificado que frequências acima de 900Hz deveriam ser eliminadas do sinal para se alcançar resultados mais satisfatórios, com isso foi empregado um filtro passa-baixa FIR com as seguintes características:

- $N=100$ ; ordem do filtro;
- $W_n=0.05$ , frequência de corte.

Na Figura 13 é apresentado o filtro passa-baixa do tipo FIR, projetado com os valores acima. Percebe-se que o filtro eliminou as frequências acima de 900Hz do sinal, a Figura 14 apresenta a respectiva resposta ao impulso desse filtro.



**Figura 13.** Representação do filtro passa-baixa FIR (cor vermelho) filtrando o sinal, e a forma do sinal depois de filtrado.

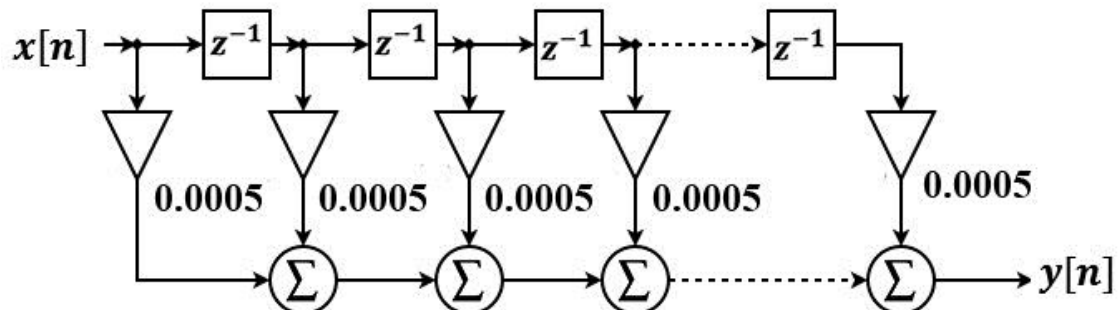


**Figura 14.** Resposta ao impulso do filtro passa-baixa FIR.

A ordem do filtro passa-baixa é  $N=100$ , com isso, são muitos os coeficientes desse filtro. Com isso, na Tabela 2 optou-se por apresentar somente alguns coeficientes, porém no console do Matlab®, são apresentados todos os coeficientes do numerador e denominador. E a Figura 15 mostra o diagrama de blocos.

**Tabela 2.** Coeficientes dos filtros passa-baixa FIR.

Numerador	0.0005	0.0005	0.0005	0.0005	.....	.....	0.0005	0.0005	0.0005	0.0005
Denominador	1									

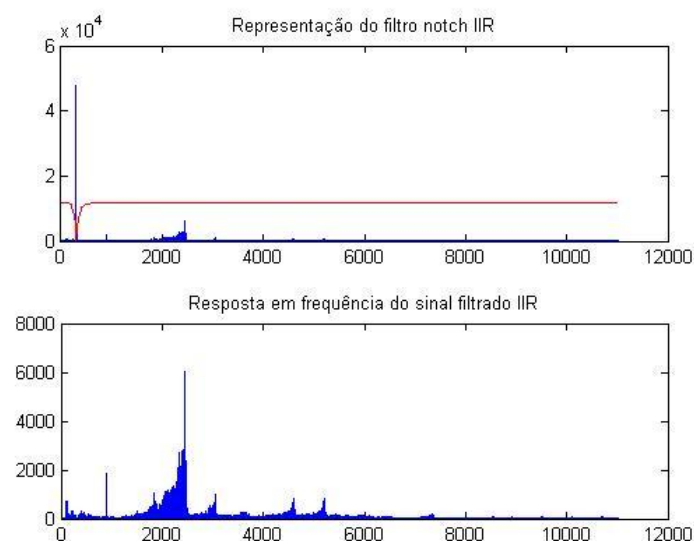


**Figura 15.** Diagrama de blocos passa-baixa FIR com ordem  $N=100$ .

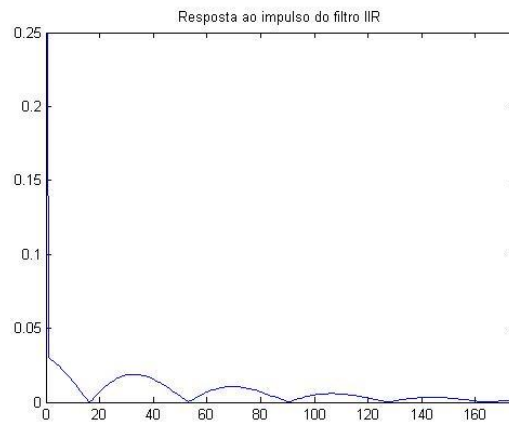
### Resultados com o filtro IIR

Semelhante como na filtragem do filtro FIR, foi verificado que a frequência de 300Hz interferia no sinal de áudio, com isso, também foram projetados dois filtros *notch* com as seguintes parâmetros  $W_o=0.0275$  e  $BW=0.01$  utilizando a função *iirnotch*. Foram projetados dois filtros *notch* para se alcançar mais consistência na filtragem.

Na Figura 16 é apresentado o filtro notch do tipo IIR, projetado com os parâmetros mencionados acima. Percebe-se que o filtro eliminou somente a frequência de 300Hz do sinal, a Figura 17 apresenta a respectiva resposta ao impulso desse filtro.



**Figura 16.** Representação do filtro rejeita-faixa IIR (cor vermelha) filtrando o sinal, e a forma do sinal depois de filtrado.



**Figura 17.** Resposta ao impulso do filtro rejeita faixa FIR.

Na Tabela 3 são apresentados os coeficientes do numerador e denominador do filtro rejeita-faixa IIR implementado neste projeto.

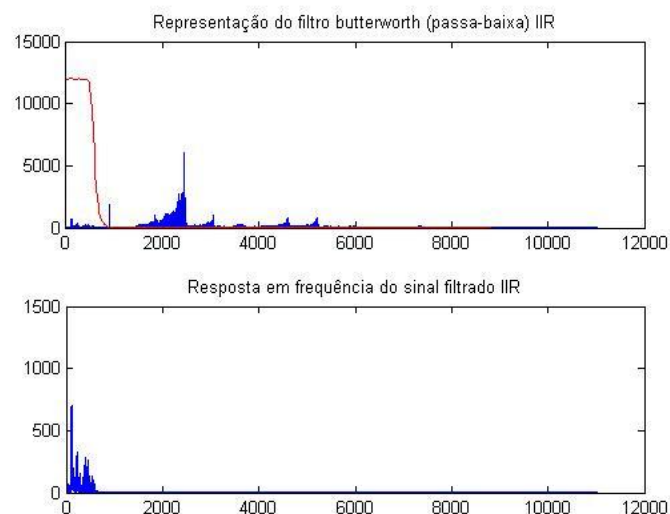
**Tabela 3.** Coeficientes dos filtros rejeita-faixa IIR.

Numerador	0.9845 -1.9617 0.9845
Denominador	1.0000 -1.9617 0.9691

Para retirar as frequências acima de 900Hz também foi utilizado um filtro passa-baixa, mas de *butterworth* do tipo IIR. Este filtro foi empregado com as seguintes características:

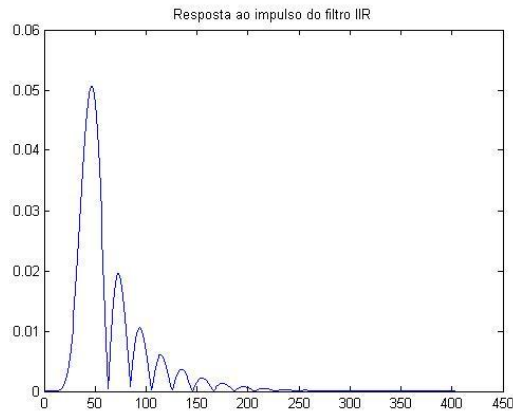
- $N=10$ ; ordem do filtro;
- $W_n=0.05$ , frequência de corte.

Na Figura 18 é apresentado o filtro passa-baixa do tipo FIR, projetado com os valores acima. Percebe-se que o filtro eliminou as frequências acima de 900Hz do sinal, a Figura 19 apresenta a respectiva resposta ao impulso desse filtro.



**Figura 18.** Representação do filtro passa-baixa IIR (cor vermelho) filtrando o sinal, e a forma do sinal depois de filtrado.



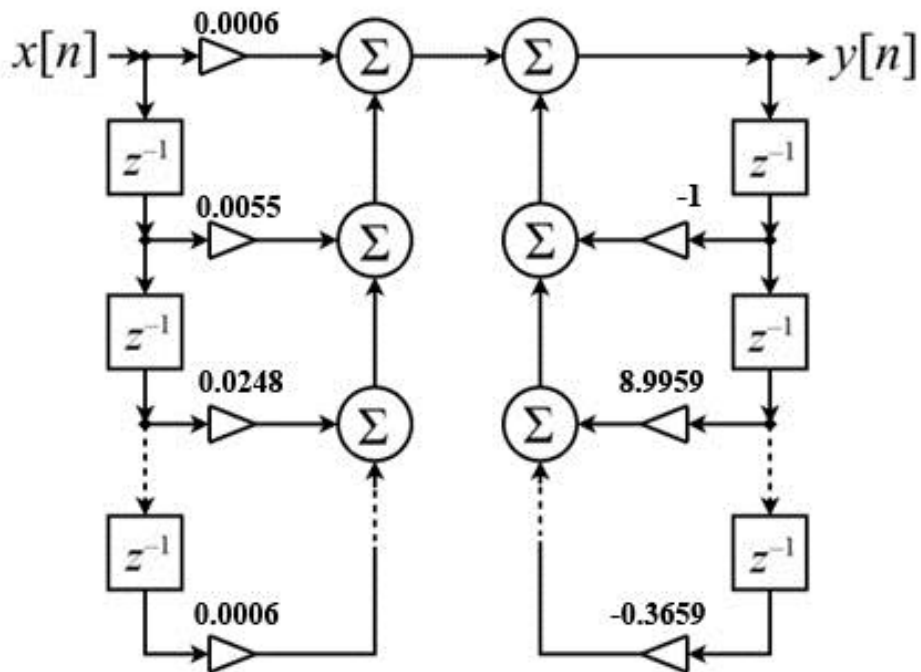


**Figura 19.** Resposta ao impulso do filtro passa-baixa IIR.

Na Tabela 4 são apresentados os coeficientes do numerador e denominador do filtro passa-baixa IIR de ordem  $N=10$  implementado neste projeto. Na Figura 20 mostrado o diagrama de blocos.

**Tabela 4.** Coeficientes dos filtros passa-baixa IIR.

Numerador	0.0006	0.0055	0.0248	0.0662	0.1158	0.1390	0.1158	0.0662	0.0248	0.0055	0.0006
Denominador	1.0000	-8.9959	36.4633	-87.6908	138.5599	-150.3021	113.3487	-58.6791	19.9563	-4.0260	0.3659



**Figura 20.** Diagrama de blocos passa-baixa IIR com ordem  $N=10$ .

### ***Resultados comparativos***

Os dois tipos filtros realização a filtragem do arquivo de áudio, de forma que o mesmo pudesse se tornar audível. Os filtros rejeita-faixa de FIR e IIR também funcionaram de maneiras semelhantes na filtragem da frequência, assim como também, os filtros passa-baixa FIR e IIR, no final o resultado foi semelhantes nos dois tipos de filtros. O filtro IIR obteve um comportamento melhor considerando a ordem do filtro, 10 vezes melhor que o filtro FIR. Considerando a opinião do autor deste trabalho, o som depois de filtrado utilizando o filtro IIR se mostrou mais adequado audivelmente.

## **5. CONCLUSÃO**

Conclui-se que implementação de filtros digitais para o tratamento de sinais de áudio é bastante interessante, principalmente no sentido de ser observado na prática a filtragem de sinais. Os resultados se comportaram de maneira fiel a fundamentação teórica, e foi possível utilizar vários projetos de filtros na implementação deste trabalho. Também a escolha dos projetos de filtros foram as mais adequadas para a realização deste trabalho.

Por fim, a utilização de *software* matemáticos foi de grande importância para calcular e simular os resultados obtidos.

## **6. REFERÊNCIAS**

DINIZ, Paulo SR; DA SILVA, Eduardo AB; NETTO, Sergio L. **Processamento Digital de Sinais:- Projeto e Análise de Sistemas**. Bookman Editora, 2014.

HAYKIN, SIMON S.; VAN VEEN, Barry. **Sinais e sistemas**. Bookman, 2001.

OPPENHEIM, Alan V. **Sinais e sistemas**. Prentice-Hall, 2010.

OPPENHEIM, A. V.; SCHAFER, R. W. **Processamento em tempo discreto de sinais**. 2012.

LATHI, Bhagwandas Pannalal. **Sinais e Sistemas Lineares-2**. Bookman, 2007.

## **7. ANEXOS**

### ***Filtragem utilizando o filtro FIR***

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Autor: WELLINGTON DANTAS DE ALMEIDA - ENG. DE TELECOMUNICAÇÕES - IFCE
%Cadeira: FILTROS DIGITAIS - Prof. DR. RICARDO RODRIGUES
%Ferramentas: MÉTODO DAS JANELAS
%Descrição: Neste código são utilizado dois filtro FIR para a
filtragem de
%          uma som. No som há uma pessoa falando, uma sirene e um
chiado.
```

```

%           O código retira o som da sirene e do chiado. Deixando o
som da
%           voz da pessoa audível.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

close all;
clc;
clear all;
% Carregamento do sinal de áudio
[x,fs,nbits]=wavread('C:\Users\usuario\Desktop\TRABALHOS\FILTRO-
DIGITAIS\Codigos\Sons\fala_sirene_tml.wav');

% Play na musica original
%wavplay(x,fs);

% Capturando informações
Ts = 1/fs;                % Período de amostragem
t = 0:Ts:1-Ts;            % Intervalo de amostragem
fa=[0:1:length(x)-1];    % fa percorre vetor de amostras dos sinal
f=fa.*fs/(length(x)-1);  % Vetor de frequências em Hz
X=abs(fft(x));            % Espectro de frequência do sinal de áudio

% Apresentação de plots
subplot(2,1,1);
plot(fa,abs(x));
title('Sinal de Voz');
subplot(2,1,2);
plot(f(1:round(length(x)/2)),X(1:round(length(x)/2)));
title('Espectro do sinal');

% ----- 1º Estágio de filtragem -----
% Filtro rejeita-faixa.
fc = 300;                % Frequência a ser eliminada
r = 0.9926;              % Coeficiente para regulação
wc = (2*pi*fc)/fs;       % Conversão da frequência de corte
b = [1 -2*cos(wc) 1];    % Coeficientes do numerador do filtro
a = [1 -2*r*cos(wc) r^2]; % Coeficientes do denominador do filtro
[H1,W1] = freqz(b, a, 512); % Resposta em Frequência do Filtro Notch
e = filter(b,a,x);       % Filtrando o sinal
E = abs(fft(e));         % Resposta em frequência do sinal
filtrado
[I,T] = impz(b, a);      % Resposta ao impulso do filtro FIR

% Coeficientes do segundo filtro notch
disp('Coeficientes do filtro rejeita-faixa FIR');
disp(b);
disp(a);

% Apresentação de plots
figure;
subplot(2,1,1);
plot(f(1:round(length(x)/2)),X(1:round(length(x)/2)));
hold on
plot(W1*(fs/(2*pi)), abs(H1*12000), 'r');
title('Representação do filtro rejeita-faixa FIR');
subplot(2,1,2);
plot (f(1:round(length(e)/2)), E(1:round(length(e)/2)));
title('Resposta em frequencia do sinal filtrado FIR');
figure;
plot(T,abs(I));

```

```

title('Resposta ao impulso do filtro FIR');

% ----- 2º Estágio de filtragem -----
% Filtro passa-baixa.
B=fir1(100,0.05, 'low');
A=1;
[H1,W1] = freqz(B, 1, 1024); % Resposta em Frequência
y = filter(B,A,e);           % Filtrando o sinal
Y = abs(fft(y));             % Resposta em frequência do sinal
filtrado
[I1,T1] = impz(B, A);        % Resposta ao impulso do filtro FIR

% Coeficientes do segundo filtro notch
disp('Coeficientes do filtro passa-baixa FIR');
disp(B);
disp(A);

% Apresentação de plots
figure;
subplot(2,1,1);
plot(f(1:round(length(e)/2)),E(1:round(length(e)/2)));
hold on
plot(W1*(fs/(2*pi)), abs(H1*12000), 'r');
title('Representação do filtro passa-baixa FIR');
subplot(2,1,2);
plot(f(1:round(length(y)/2)), Y(1:round(length(y)/2)));
title('Resposta em frequência do sinal filtrado FIR');
figure;
plot(T1,abs(I1));
title('Resposta ao impulso do filtro FIR');

% Salvando sinal filtrado
wavwrite(y,fs,nbits,'C:\Users\usuario\Desktop\TRABALHOS\FILTRO-
DIGITAIS\Codigos\Sons\fala_sirene_tmFIR.wav');

% Play no voz filtrado
[x1,fs1,nbits1]=wavread('C:\Users\usuario\Desktop\TRABALHOS\FILTRO-
DIGITAIS\Codigos\Sons\fala_sirene_tmFIR.wav');
am=(1)*x1;           % Amplificação do sinal.
wavplay(am,fs1);

disp('100%!');

```

### ***Filtragem utilizando o filtro IIR***

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Autor: WELLINGTON DANTAS DE ALMEIDA - ENG. DE TELECOMUNICAÇÕES - IFCE
%Cadeira: FILTROS DIGITAIS - Prof. DR. RICARDO RODRIGUES
%Ferramentas: FILTRO DE BUTTERWORTH E NOTCH.
%Descrição: Neste código são utilizado dois filtro IIR para a
filtragem de
%          uma som. No som há uma pessoa falando, uma sirene e um
chiado.
%          O código retira o som da sirene e do chiado. Deixando o
som da
%          voz da pessoa audível.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

close all;
clc;
clear all;
% Carregamento do sinal de áudio
[x,fs,nbits]=wavread('C:\Users\usuario\Desktop\TRABALHOS\FILTRO-
DIGITAIS\Codigos\Sons\fala_sirene_tml.wav');

% Play na musica original
%wavplay(x,fs);

% Capturando informações
Ts = 1/fs; % Período de amostragem
t = 0:Ts:1-Ts; % Intervalo de amostragem
fa=[0:1:length(x)-1]; % fa percorre vetor de amostras dos sinal
f=fa.*fs/(length(x)-1); % Vetor de frequências em Hz
X=abs(fft(x)); % Espectro de frequência do sinal de áudio

% Apresentação de plots
subplot(2,1,1);
plot(fa,abs(x));
title('Sinal de Voz');
subplot(2,1,2);
plot(f(1:(round(length(x)/2))),X(1:(round(length(x)/2))));
title('Espectro do sinal');

% ----- 1º Estágio de filtragem -----
% Filtro notch
Wo = 0.0275; BW=0.01; % Parâmetros de entrada
[b,a] = iirnotch(Wo,BW); % Filtro notch
[H1, W1]= freqz(b, a, 512); % Resposta em frequência do filtro
y = filter(b,a,x); % Filtrando o sinal
Y = abs(fft(y)); % Resposta em frequência do sinal
filtrado

% Coeficientes do primeiro filtro notch
disp('Coeficientes do primeiro filtro notch');
disp(b);
disp(a);

% Filtro notch
Wo = 0.0275; BW=0.01; % Parâmetros de entrada
[b1,a1] = iirnotch(Wo,BW); % Filtro notch
[H1, W1]= freqz(b1, a1, 512); % Resposta em frequência do filtro
e = filter(b1,a1,y); % Filtrando o sinal
E = abs(fft(e)); % Resposta em frequência do sinal
filtrado
[G,U] = impz(b1, a1); % Resposta ao impulso do filtro IIR

% Coeficientes do segundo filtro notch
disp('Coeficientes do segundo filtro notch');
disp(b1);
disp(a1);

% Apresentação de plots
figure;
subplot(2,1,1);
plot(f(1:(round(length(x)/2))),X(1:(round(length(x)/2))));

```

```

hold on;
plot(W1*(fs/(2*pi)), abs(H1*12000), 'r');
title('Representação do filtro notch IIR');
subplot(2,1,2);
plot(f(1:round(length(e)/2)), E(1:round(length(e)/2)));
title('Resposta em frequência do sinal filtrado IIR');
figure;
plot(U,abs(G));
title('Resposta ao impulso do filtro IIR');

% ----- 2° Estágio de filtragem -----
% Filtro de passa-baixa de butterworth
[b2, a2] = butter(10, 0.05, 'low'); % Parametros de entrada
[H1, W1]= freqz(b2, a2, 512);      % Resposta em frequência do filtro
s = filter(b2,a2,e);               % Filtrando o sinal
S = abs(fft(s));                   % Resposta em frequência do sinal
filtrado
[I,T] = impz(b2, a2);               % Resposta ao impulso do filtro
IIR

% Coeficientes do filtro de butterworth
disp('Coeficientes do filtro de butterworth');
disp(b2);
disp(a2);

% apresentação de plots
figure;
subplot(2,1,1);
plot(f(1:(round(length(e)/2))),E(1:(round(length(e)/2))));
hold on;
plot(W1*(fs/(2*pi)), abs(H1*12000), 'r');
title('Representação do filtro notch IIR');
subplot(2,1,2);
plot(f(1:round(length(s)/2)), S(1:round(length(s)/2)) );
title('Resposta em frequência do sinal filtrado IIR');
figure;
plot(T,abs(I));
title('Resposta ao impulso do filtro IIR');

% Salvando sinal filtrado
wavwrite(s,fs,nbits,'C:\Users\usuario\Desktop\TRABALHOS\FILTRO-
DIGITAIS\Codigos\Sons\fala_sirene_tmIIR.wav');

% Play na voz filtrada
[x1,fs1,nbits1]=wavread('C:\Users\usuario\Desktop\TRABALHOS\FILTRO-
DIGITAIS\Codigos\Sons\fala_sirene_tmIIR.wav');
am=(1)*x1;          % Amplificação do sinal.
wavplay(am,fs1);

disp('100%!');

```