



Universidade Federal do Ceará  
Centro de Tecnologia  
Departamento de Engenharia de Teleinformática  
Engenharia de Telecomunicações

## Projeto de Filtro Digital

Aplicação em Sinais de Áudio

Aluno	Lucas de Souza Abdalah - 385472
Professores	André Lima Ferrer de Almeida e João César Moura Mota
Disciplina	Processamento Digital de Sinais - TI0119

Fortaleza, 26 de junho de 2018

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Amostragem por Trem de Impulsos Periódicos . . . . .	1
1.2	Filtros Seletivos em Frequência . . . . .	2
1.2.1	Ideal . . . . .	2
1.2.2	Real . . . . .	3
1.3	Projeto de Filtro . . . . .	3
1.3.1	Visão Geral . . . . .	3
1.3.2	Especificações . . . . .	3
<b>2</b>	<b>Desenvolvimento</b>	<b>5</b>
2.1	Problema Proposto . . . . .	5
2.2	Análise do Problema . . . . .	6
2.2.1	Canal Esquerdo . . . . .	7
2.2.2	Canal Direito . . . . .	8
2.3	Projeto de Filtro . . . . .	9
2.3.1	Análise Manual . . . . .	9
2.3.2	Análise Computacional . . . . .	10
<b>3</b>	<b>Análise dos Resultados</b>	<b>16</b>
<b>4</b>	<b>Referências</b>	<b>17</b>

# 1 Introdução

## 1.1 Amostragem por Trem de Impulsos Periódicos

É utilizada a propriedade de amostragem da função,  $x(t)\delta(t) = x(0)\delta(t)$ . Logo, a técnica "captura" essas amostras ao efetuar o produto entre o sinal  $x_c(t)$  de tempo contínuo pelo Trem de Impulsos periódicos  $s(t)$

$$x_s(t) = x_c(t)s(t), \quad (1)$$

A equação geral adquire essa forma ao explicitar os termos

$$x_s(t) = x_c(t) \sum_{n=-\infty}^{\infty} \delta(t - nT) \sum_{n=-\infty}^{\infty} x_c(t)\delta(t - nT) \quad (2)$$

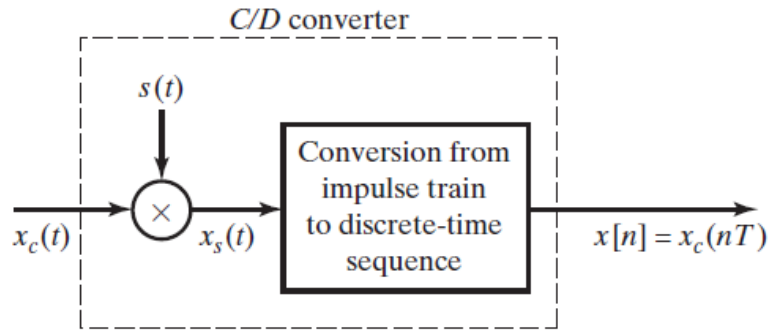


Figura 1: Diagrama de blocos de um conversor de tempo Contínuo-Discreto.

Fonte: Oppenheim, A. V., and Schafer, R. W., Discrete-Time Signal Processing - Prentice Hall (2009).

O diagrama da figura 1 descreve justamente a operação da amostragem. A multiplicação resulta no sinal de tempo contínuo que representa uma soma infinita das contribuições do sinal contínuo, espaçadas em  $T$ , devido a multiplicação pelo trem de impulsos periódico. Ao visualizar o grande bloco de conversão (*C/D Converter*), há de se atentar à discretização do sinal contínuo observada na equação (1).

## 1.2 Filtros Seletivos em Frequência

### 1.2.1 Ideal

Um filtro seletivo em frequência é um dispositivo que permite a passagem de frequências em certas faixas e atenua as frequências fora dessa faixa.

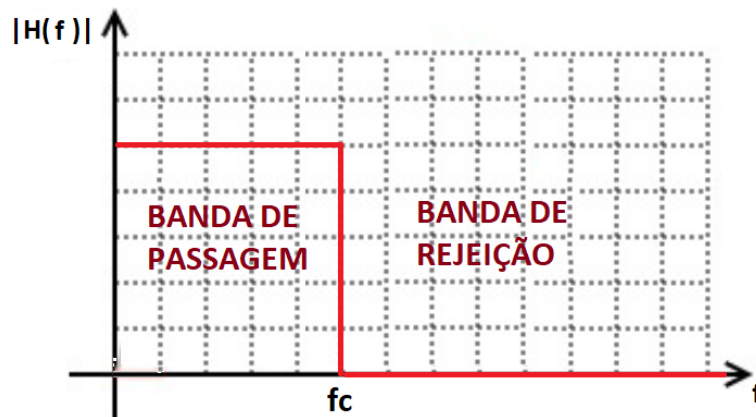


Figura 2: Tipo de filtro seletivo em frequência ideal.

Fonte: Elaborado pelo autor.

Um filtro ideal teria banda passante sem deformidades e iria atenuar perfeitamente todas as frequências fora da faixa de passagem, além de não ter uma transição entre as duas bandas mostradas na figura 2, o que na prática, infelizmente, nenhum filtro passa-faixa faz.

Não há como atenuar totalmente todas as frequências fora da faixa desejada, na região de descontinuidade há um período de acomodação do sinal e há variações nos limiares máximos e mínimos do filtro, efeito esse que é conhecido como fenômeno de Gibbs.

Essas nuances do filtro real podem ser analisadas melhor em um projeto real, onde é possível observar e avaliar como resolver cada característica do filtro realizável.

### 1.2.2 Real

Como é trabalhado computacionalmente, a filtragem aplicada envolve discretização de um sinal no tempo. Logo o sistema a ser projetado deve trabalhar com sinais discretos e somente após o sinal ser filtrado, reconstrução do sinal para tempo contínuo, como a figura 3 indica.

Também é importante fazer algumas considerações iniciais: o sinal de entrada tem banda limitada e, como há um processo de amostragem, é admitido que o teorema de Nyquist é respeitado em todas as etapas do processo.

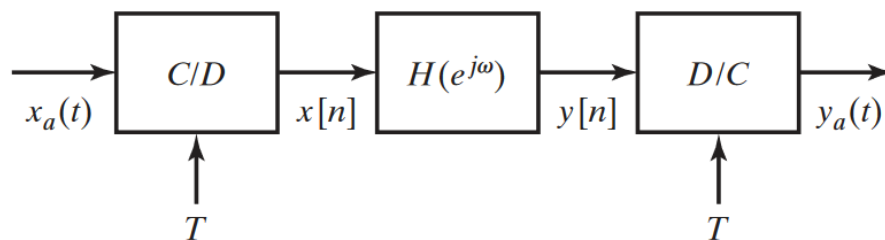


Figura 3: Diagrama de um sistema de filtragem discreta para sinais contínuos.

Fonte: Oppenheim, A. V., and Schaffer, R. W., Discrete-Time Signal Processing - Prentice Hall (2009).

## 1.3 Projeto de Filtro

### 1.3.1 Visão Geral

Ao iniciar esse projeto é interessante dividi-lo em etapas. (nesta introdução será descrito apenas a primeira dessas).

- **Especificação** dos parâmetros são descritos.
- **Aproximação** das especificações iniciais, respeitando causalidade e estabilidade.
- **Realização** do sistema.

### 1.3.2 Especificações

Felizmente, não é necessário ter total controle de seu comportamento, é possível trabalhar apenas alguns parâmetros (figura 4) desejados e verificar possíveis pontos de instabilidades.

Como a imagem abaixo mostra, parâmetros de efetividade devem ser definidos para estimar um filtro realizável.

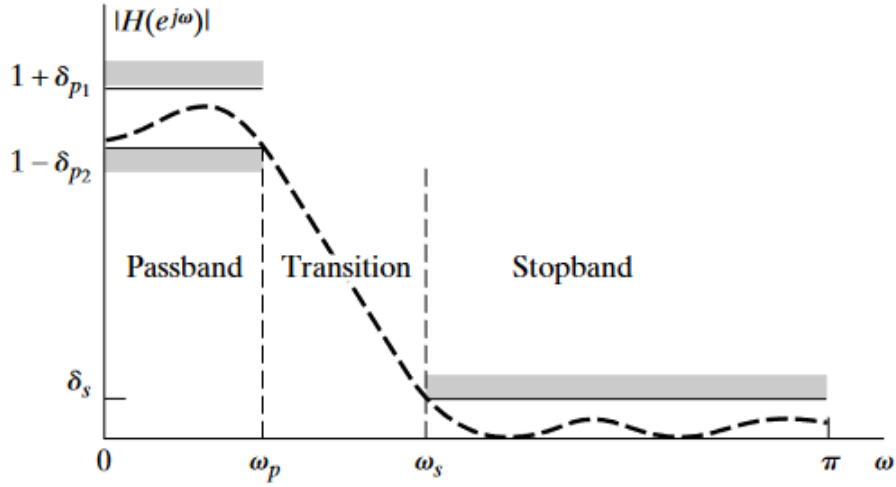


Figura 4: Esboço de um filtro real.

Fonte: Oppenheim, A. V., and Schafer, R. W., Discrete-Time Signal Processing - Prentice Hall (2009).

Dentro da faixa de passagem, a resposta em módulo deve se manter entre  $1 - \delta_1$  e  $1 + \delta_1$

$$1 - \delta_1 \leq |H(e^{j\omega})| \leq 1 + \delta_1, (0 \leq \omega \leq \omega_p) \quad (3)$$

em que  $\omega_p$  é a frequência de corte da faixa de passagem e  $\delta_1$  envolve a tolerância da variação do ganho.

Dentro da faixa de rejeição, a resposta do módulo não deve ultrapassar  $\delta_2$

$$|H(e^{j\omega})| \leq \delta_2, (\omega_s \leq |\omega|) \quad (4)$$

A largura da faixa de transição é finita e vale  $\omega_s - \omega_p$

Importante informar também que há classes de filtros IIR (Resposta Infinita ao Impulso) e FIR (Resposta Finita ao Impulso). Em cada uma dessas há diversos modelos, cada um com as devidas especificidades e de acordo com essas que é escolhido o modelo utilizado no projeto.

## **2 Desenvolvimento**

### **2.1 Problema Proposto**

Davis, um trompetista profissional, estava gravando algumas notas em seu pequeno estúdio caseiro. Ao verificar o áudio capturado pelo seu microfone, ele percebeu que havia um problema: um tom de alta frequência, provavelmente produzido por um defeito no seu microfone, estava contaminando a sua música. Como não havia outro microfone estúdio, ele decidiu pedir o suporte de uma empresa especializada em processamento de sinais para resolver este problema. Ele enviou ao suporte um arquivo de áudio e solicitou a remoção da interferência sem prejudicar a qualidade da sua música.

## 2.2 Análise do Problema

Inicialmente, tem-se um sinal de áudio é estéreo (como o próprio plot da imagem indica) e para trabalhar apenas com um vetor de cada vez, as colunas da matriz foram separadas e analisadas individualmente.

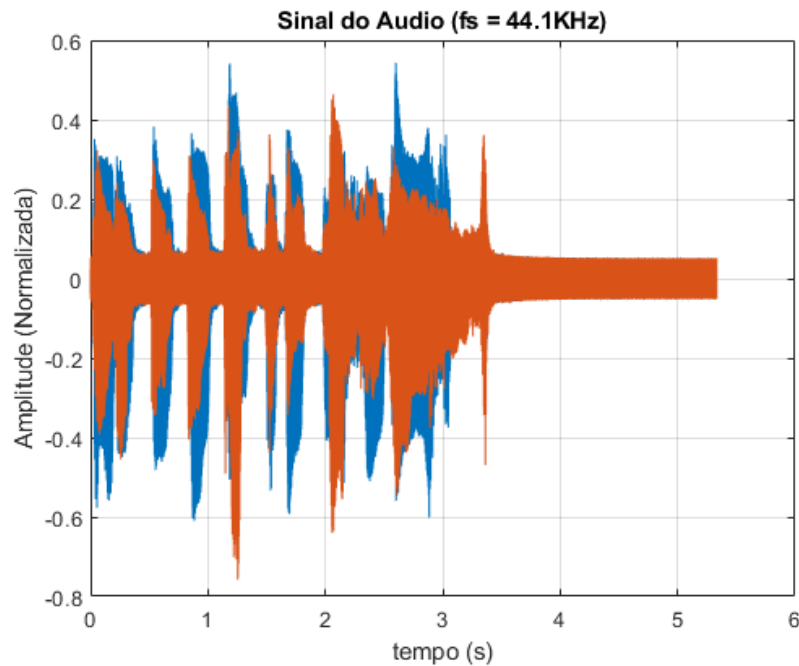


Figura 5: Sinal de Áudio Original.

Fonte: Elaborada pelo autor.

Ao ouvir o som, um ruído é muito perceptível e inclusive mais intenso que a melodia. Instintivamente, sabe-se que ele é ruído de alta frequência pois é bastante agudo, ou como é dito informalmente, ele é bem "fino", de modo que está em frequência bem acima da música, logo um filtro passa-baixas deve ser o ideal para solucionar o problema.

A suspeita é confirmada ao aplicar a transformada rápida de Fourier no sinal e observar a plotagem do sinal na frequência (figura 6). O sinal referente a melodia é concentrado de aproximadamente 0Hz-4kHz e bem 'distante' das componentes do sinal desejado, há um pico de frequência na componente de 11kHz. Ao utilizar a função **spectrogram** também é possível observar a componente que destoa das demais, entretanto com frequência normalizada).



### 2.2.1 Canal Esquerdo

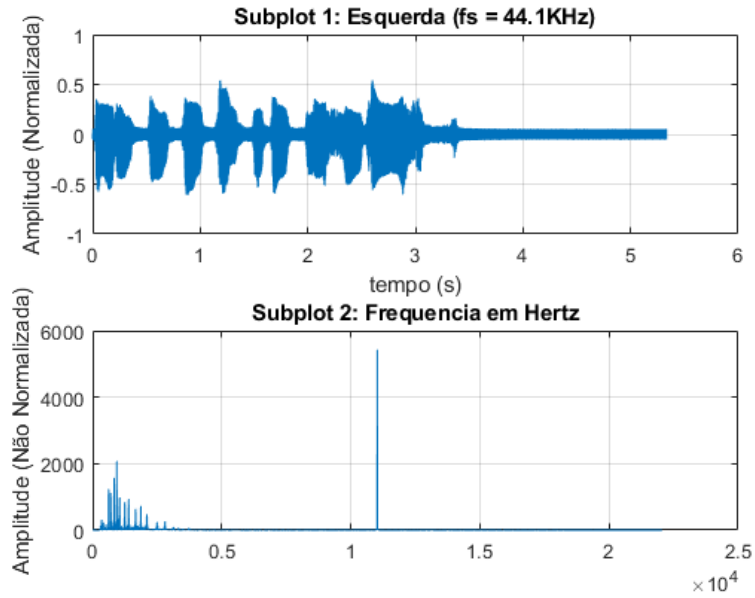


Figura 6: Sinal de áudio e sua transformada de Fourier.

Fonte: Elaborada pelo autor.

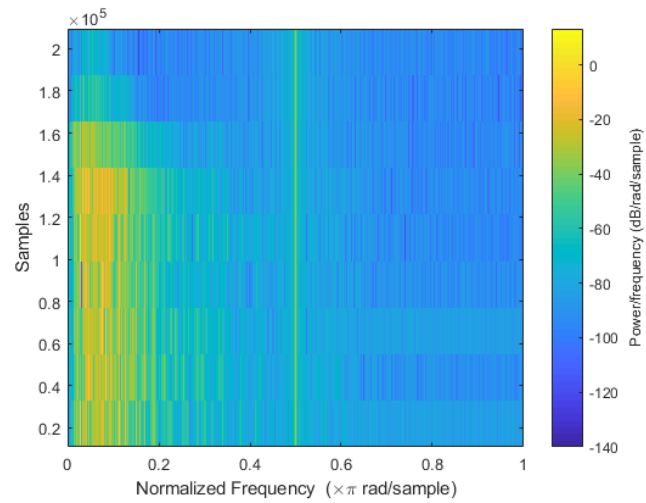


Figura 7: Espectro da frequência com **spectrogram**.

Fonte: Elaborada pelo autor.

### 2.2.2 Canal Direito

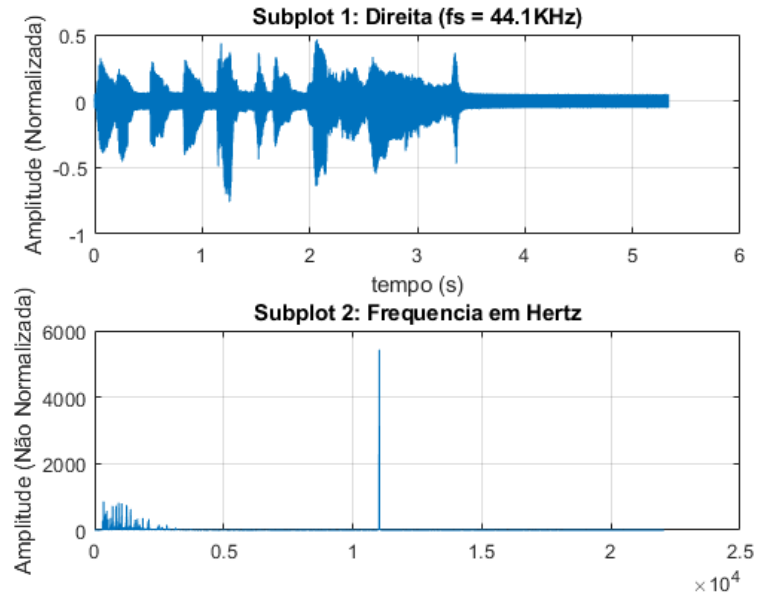


Figura 8: Sinal de áudio e sua transformada de Fourier.  
Fonte: Elaborada pelo autor.

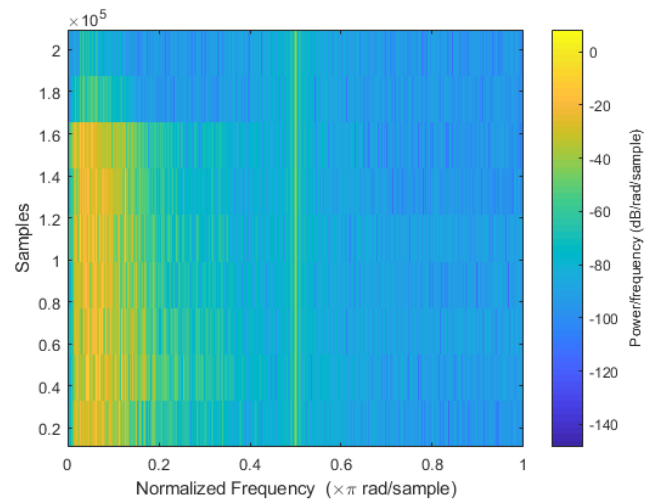


Figura 9: Espectro da frequência com **spectrogram**.  
Fonte: Elaborada pelo autor.

## 2.3 Projeto de Filtro

### 2.3.1 Análise Manual

Como é de conhecimento do projeto, isso é um sinal de áudio, logo para manter a fidelidade do que foi gravado originalmente, ao atenuar a interferência, é essencial que não haja muita oscilação na banda de passagem, pois caso contrário variações indesejadas do ganho em frequências que devem permanecer inalteradas o máximo possível serão notadas.

Para a situação descrita é interessante usar o filtro de **Butterworth**, pois este se aproxima do resultado esperado. A equação que envolve a sua magnitude é dada por:

$$|H(j\Omega)| = \frac{1}{1 + (\frac{\Omega}{\Omega_C})^{2N}} \quad (5)$$

E o projeto do filtro segue com as especificações, justificadas por tudo explanado até aqui.

$$d_{P1} = d_{P2} = 0,001 \quad (6)$$

$$d_S = 0,001 \quad (7)$$

$$f_C = 4,8KHz \quad (8)$$

$$f_S = 10kHz \quad (9)$$

$$\omega_P = \frac{2\pi f_C}{f_{samp}} = 0,218\pi \quad (10)$$

$$\omega_S = \frac{2\pi f_S}{f_{samp}} = 0,454\pi \quad (11)$$

- Na banda de passagem:  
 $1 - 0,001 \leq |H(j\Omega)| \leq 1 + 0,001$
- Na banda de rejeição:  
 $|H(j\Omega)| \leq 0,001$

Utilizando a expressão da magnitude e aplicados os valores das especificações, tem-se um sistema de duas equações e duas variáveis.

$$1 + \left(\frac{0,218\pi}{\Omega_C}\right)^{2N} = \left(\frac{1}{0.999}\right)^2$$

$$1 + \left(\frac{0,454\pi}{\Omega_C}\right)^{2N} = \left(\frac{1}{0.001}\right)^2$$

A solução da ordem do filtro é dada por:

$$N = 0,5 \log_{\frac{\omega_s}{\omega_p}} \left(\frac{1}{\delta_2}\right)^2 - \left(\frac{1}{1 - \delta_1}\right)^2$$

Ao obter essa expressão, é possível ver que o filtro é da ordem  $N = 9,42$ .

Como  $N$  deve ser inteiro é interessante utilizar o inteiro maior, para garantir que as especificações serão cumpridas. Logo,  $N = 10$  e por consequência  $\Omega_C = 1,53$ .

### 2.3.2 Análise Computacional

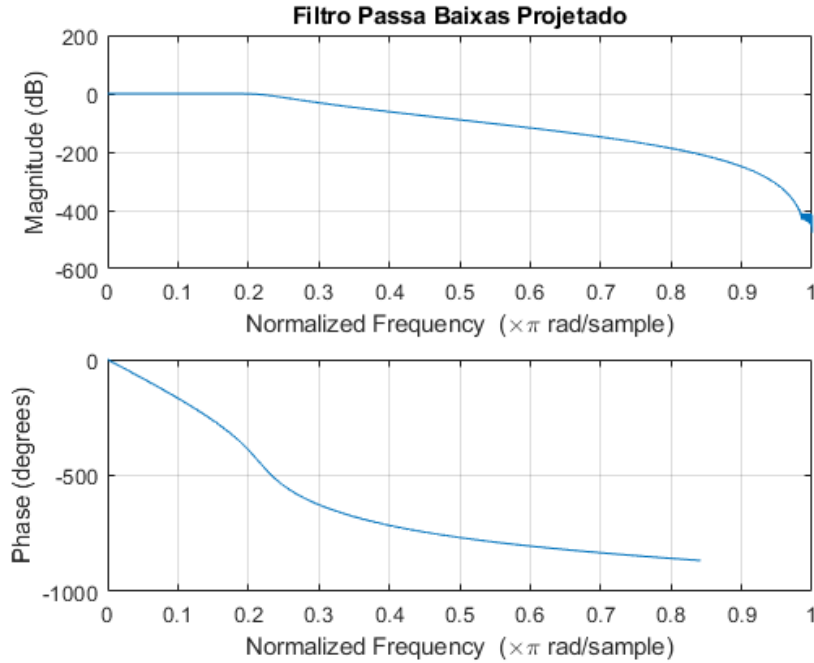


Figura 10: Diagrama de Bode do Filtro ordem 10.

Fonte: Elaborada pelo autor.

Observe o plot do Diagrama de Bode do referido filtro, ele atenuando em 100dB na frequência normalizada de 0,6.

Vendo esse gráfico e após todo o desenvolvimento do filtro computacional surgem perguntas do gênero: "Ele funciona? É estável? E se tivesse uma ordem menor? E se mudasse as especificações? Ele não é ideal, então como proceder?"

Para sanar essas dúvidas, nas próximas há uma sequência de esclarecimentos.

O mapa de polos e zeros no plano Z indica que o filtro tem todos os polos dentro do círculo unitário.

*O que isso implica?*

Estabilidade do sistema e que todos os S do espaço Laplaciano foram mapeados dentro do próprio círculo e todos ainda no semi plano esquerdo!

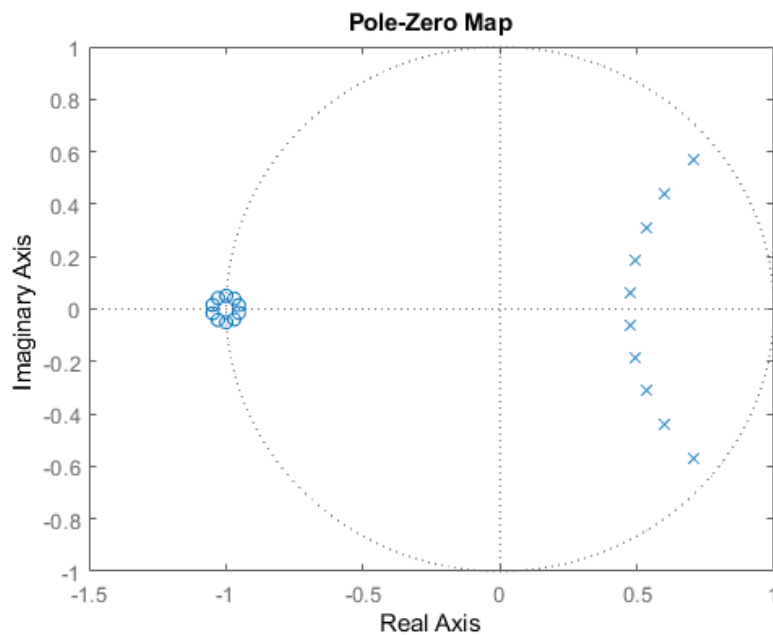


Figura 11: Mapa de Polos e Zeros do Plano Z.

Fonte: Elaborada pelo autor.

Observe também que há zeros fora da circunferência unitária do Plano Z, logo o sistema projetado não é de fase mínima.

O mapa de polos e zeros no plano S, o qual foi mapeado na figura anterior. Indicando os zeros do lado direito e nenhum polo fora da região de convergência.

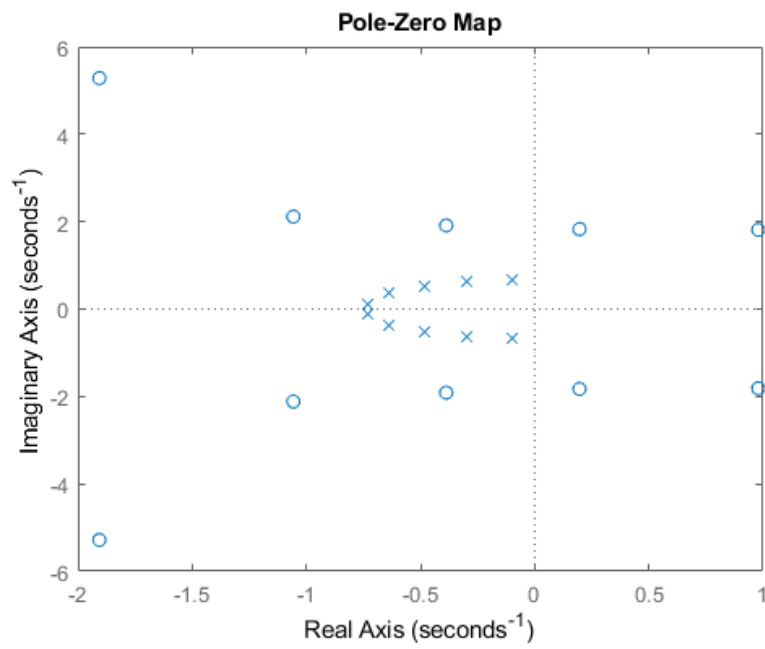


Figura 12: Mapa de Polos e Zeros do Plano Z.  
Fonte: Elaborada pelo autor.

Ao modificar algumas especificações, tem-se um sistema novo com todas as características alterada. O interesse nesse caso é só mostrar o quanto a ordem do filtro importou nesse caso.

Pra estimativas mais conservadoras o filtro funciona muito melhor e decai mais rápido (como pode ser percebido no áudio em anexo.) Já no filtro de ordem  $n = 5$  (figura 13), o ruído ainda que levemente atenuado permanece perceptível para um homem, ou seja, o filtro de menor ordem nesse caso foi insuficiente.

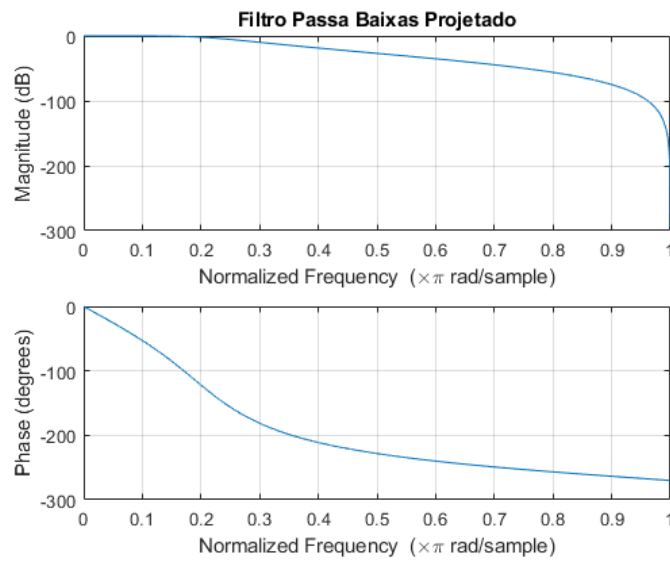


Figura 13: Filtro Passa Baixas de Ordem 5

Fonte: Elaborada pelo autor.

E observando justamente os dois filtros em uma plotagem Magnitude *versus* Frequencia (normalizada), é visível a diferença gritante de atenuação na faixa de interesse.

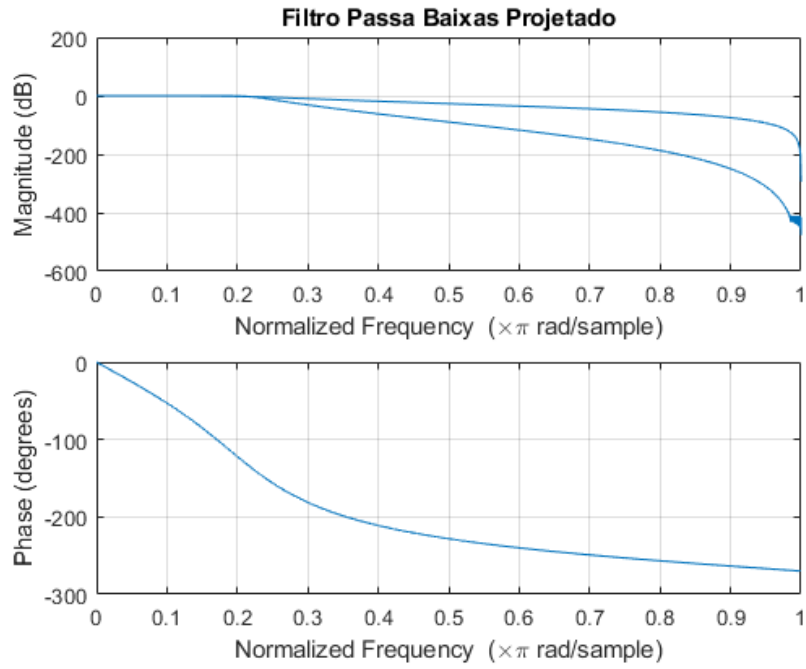


Figura 14: Filtros de Ordens 5 e 10.

Fonte: Elaborada pelo autor.



Por fim, cada canal já devidamente filtrado, sem aquela componente de frequência que prejudicava a performance do artista. Depois desse procedimento, os vetores de áudio são reunidos e exportados como um arquivo estéreo.

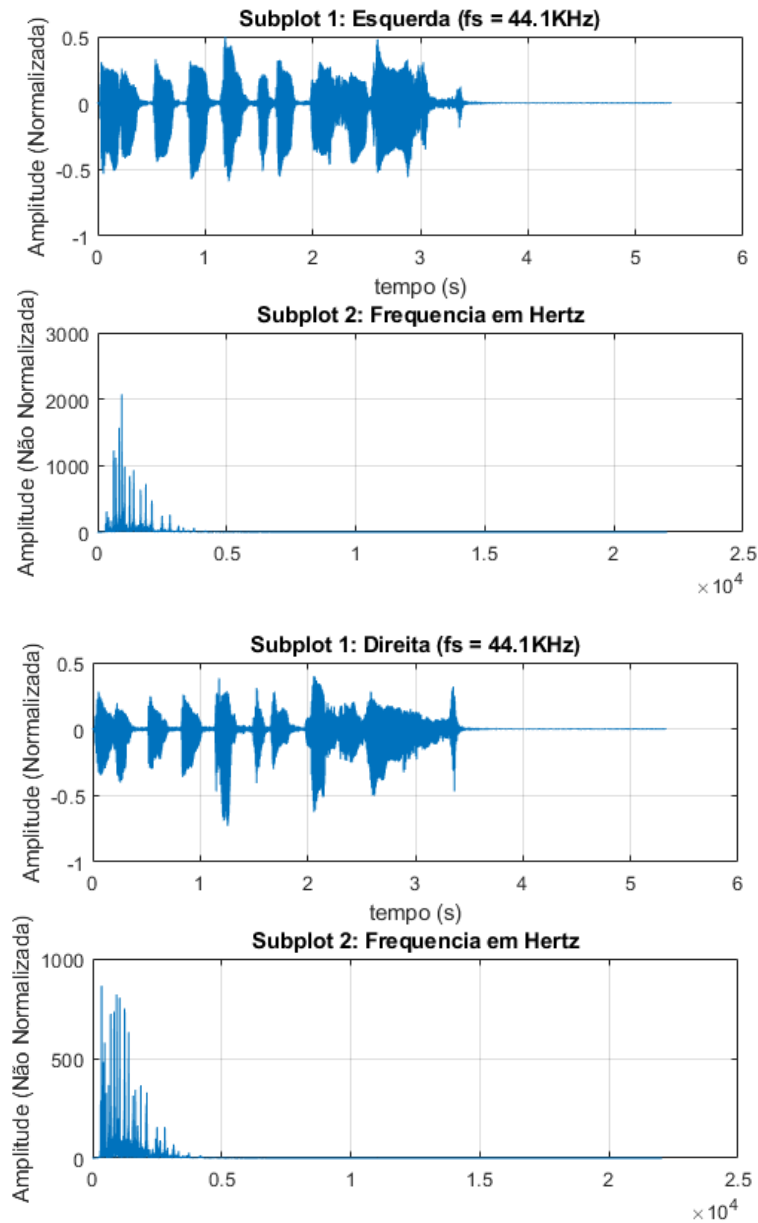


Figura 15: Canais do Áudio Filtrado

Fonte: Elaborada pelo autor.

### 3 Análise dos Resultados

É interessante citar que a técnica por invariância ao impulso para filtro IIR é bem apropriada para sinais limitados em bandas (como o passa baixas que foi utilizado no problema).

A eficiência do filtro projetado pode ser considerada boa do ponto de vista científico, pois todas as atenuações de frequência seguiram as especificações, logo não há distorção perceptível do som esperado pelo artista, o ganho na banda de passagem mantém-se em um nível também aceitável.

Do ponto de vista do músico, que fará essa análise qualitativa, pode-se considerar um sucesso, pois de forma alguma o ruído é ouvido e como a frequência de corte do projeto está após as frequências do trompete, tudo está nos conformes.

Por fim, uma análise da função custo x benefício. Nesse projeto, a ordem utilizada é inversamente proporcional à qualidade no filtro. Logo, se a ordem aumentar, o filtro é mais eficiente e há menos imprecisões. Contudo, onde fica o limiar do muito eficiente e extremamente custoso? Nesse caso, a percepção do ouvido humano que aponta, à partir do momento que o ruído não é percebido, o método já é extremamente eficiente.

## 4 Referências

- [1] Oppenheim, Alan V. – Signals and systems, 2nd Edition;
- [2] Oppenheim, A. V., and Schafer, R. W., Discrete-Time Signal Processing - Prentice Hall (2009);
- [3] **Butterworth filter design**. Disponível em:  
< [mathworks.com/help/signal/ref/butter.html](https://www.mathworks.com/help/signal/ref/butter.html) >. Acesso: 08 de junho 2018.
- [4] **Chebyshev Type I filter design**. Disponível em:  
< <https://www.mathworks.com/help/signal/ref/cheby1.html> >. Acesso: 08 de junho 2018.
- [5] **Window-based FIR filter design**. Disponível em:  
< <https://www.mathworks.com/help/signal/ref/fir1.html> >. Acesso: 25 de junho 2018.

## Apêndice A - fftedit (Matlab)

```
function [filefft , freq] = fftedit(file , fs)
    % Run the Fast Fourier Transform and do the abs of the
    % complex numbers

    % To make it faster and time axes
    n = length(file);
    t = n/fs; %File length and time range
    k = 0:n-1;
    freq=k/t; % Creates the frequency range

    % Help to cut filefft and frequency vectors
    periodo=ceil(n/2);
    freq=freq(1:periodo);

    %Do the abs of fftn data and cut
    filefft = abs(fft(file));
    filefft=filefft(1:periodo);
end
```

## Apêndice B - trompete (Matlab)

```
function [y,fs] = trompete(filename)
    clc; % Limpa a janela de comandos

    % Recebendo o audio com 2 canais
    [file , fs] = audioread(filename); % Le o audio contido no
        mesmo diretorio - y carrega os dados e fs a freq de
        amostragem
    t = linspace(0,length(file)/fs,length(file)); % Tempo do
        Audio

    % Grafico 1 - Amostras x Tempo audio stereo
    figure(1);
        ('GRAFICO_1_-AMOSTRAS_x_TEMPO');
    % Plot do Sinal contra o Tempo
        plot(t, file);
    title ("Sinal do Audio_(fs = 44.1KHz)"); %title of this
        graphic
        xlabel ("tempo_(s)");
    ylabel ("Amplitude_(Normalizada)");
    %x is the time index in seconds, y is the amplitude's
        samples
        grid on; %turn on the grid

    % ESQUERDA
    E = file(:,1); % Canal Esquerdo
    n = length(E);
    % FFT para analise
    [p, q] = fftedit(E, fs);

    % Subplot do AUDIO NO TEMPO E NA FREQUENCIA
    figure(2)
    subplot(2,1,1);
    plot(t, E); % Esquerda contra o Tempo
    title ("Subplot_1:-Esquerda_(fs = 44.1KHz)"); %title of this
        graphic
        xlabel ("tempo_(s)"); % x is the time index in seconds,
    ylabel ("Amplitude_(Normalizada)"); % y is the amplitude's
        samples
    grid on; %turn on the grid
    subplot(2,1,2);
    plot(q,p); % FFT em Hertz
    title ("Subplot_2:-Frequencia_em_Hertz");
    ylabel ("Amplitude_(No_Normalizada)");
    grid on; %turn on the grid

    % Esquerda
    figure(3);
    spectrogram(E, fs); % Analise na Frequencia
```

```

% ORDEM 10
delta1 = 1e-3; % Limite da Banda de Passagem
deltas = 1e-3; % Limite da Banda de Rejeicao
wp = 0.218*pi; % Corte 1
ws = 0.454*pi; % Corte 2

% Calculo da ordem N e do omegac
auxdelta = log((1/deltas)^2 - (1/1-delta1)^2); % mudanca de
base
ausw = log(ws/wp); % % mudanca de base
disp('Ordem_N:');
ORDEM10 = 0.5*(auxdelta/ausw); %
disp('Ordem_N_inteiro:'); disp(ORDEM10);%
ORDEM10 = ceil(ORDEM10); disp(ORDEM10);%

omegac = (wp/(((1/(1-delta1))^2 - 1)^0.5*ORDEM10));
disp('Omegac:'); disp(omegac);

% FILTRO
fc = 4800; % Frequencia de corte calculada
[a,b] = butter(ORDEM10,fc/(fs/2)); % Filtro ButterWorth
yE = filter(a,b,E); %audio filtrado
% sound(yE,fs);

% PLOT DO FILTRO
figure(4);
freqz(a,b,fs);
title('Filtro_Passa_Baixas_Projetado');

% POLOS NO PLANO Z
figure(5);
Hz = filt(a,b,1);
pzmap(Hz);

% POLOS NO PLANO S
figure(6);
Hs = d2c(Hz);
pzmap(Hs);

% Direita
D = file(:,2); % Canal Direito

% FFT para analise
[p, q] = fftedit(D, fs);

% Subplot do AUDIO NO TEMPO E NA FREQUENCIA
figure(7)
subplot(2,1,1);

```

```

plot(t, D); % Direita contra o Tempo
title ("Subplot_1:_Direita_(fs_=44.1KHz)"); %title of this
    graphic
    xlabel ("tempo_(s)"); % x is the time index in seconds,
ylabel ("Amplitude_(Normalizada)"); % y is the amplitude's
    samples
grid on; %turn on the grid
subplot(2,1,2);
plot(q,p); % FFT em Hertz
title ("Subplot_2:_Frequencia_em_Hertz");
ylabel("Amplitude_(No_Normalizada)")
grid on; %turn on the grid

figure(8);
spectrogram(D, fs); % Analise na Frequencia

yD = filter(a,b,D); %audio filtrado

y = [yE yD];

sound(y, fs);
audiowrite('q1fitro.wav',y, fs);

    % ORDEM 5
delta1 = 1e-1; % Limite da Banda de Passagem
deltas = 1e-1; % Limite da Banda de Rejeicao
wp = 0.218*pi; % Corte 1
ws = 0.480*pi; % Corte 2
% Calculo da ordem N e do omegac
auxdelta = log((1/deltas)^2 - (1/1-delta1)^2); % mudanca de
    base
ausw = log(ws/wp); % % mudanca de base
disp('Ordem_N:');
ORDEM5 = 0.5*(auxdelta/ausw); %
disp('Ordem_N_inteiro:'); disp(ORDEM5);%
ORDEM5 = ceil(ORDEM5); disp(ORDEM5);

% FILTRO
fc = 4800; % Frequencia de corte calculada
[c,d] = butter(ORDEM5,fc/(fs/2)); % Filtro ButterWorth
xE = filter(c,d,E); %audio filtrado
xD = filter(c,d,D); %audio filtrado

figure(9);
freqz(a,b,fs);
hold on;
freqz(c,d,fs);
title('Filtro_Passa_Baixas_Projetado');

```

```

x = [xE xD];

audiowrite('qlfitroruim.wav',x,fs);


[p, q] = fftedit(yE, fs);
% Subplot do AUDIO NO TEMPO E NA FREQUENCIA
figure(10)
subplot(2,1,1);
plot(t, yE); % Esquerda contra o Tempo
title ("Subplot_1: Esquerda (fs = 44.1KHz)"); %title of this
    graphic
    xlabel ("tempo(s)"); % x is the time index in seconds,
ylabel ("Amplitude (Normalizada)"); % y is the amplitude's
    samples
grid on; %turn on the grid
subplot(2,1,2);
plot(q,p); % FFT em Hertz
title ("Subplot_2: Frequencia em Hertz");
ylabel ("Amplitude (No Normalizada)")
grid on; %turn on the grid


[p, q] = fftedit(yD, fs);
% Subplot do AUDIO NO TEMPO E NA FREQUENCIA
figure(11)
subplot(2,1,1);
plot(t, yD); % Direita contra o Tempo
title ("Subplot_1: Direita (fs = 44.1KHz)"); %title of this
    graphic
    xlabel ("tempo(s)"); % x is the time index in seconds,
ylabel ("Amplitude (Normalizada)"); % y is the amplitude's
    samples
grid on; %turn on the grid
subplot(2,1,2);
plot(q,p); % FFT em Hertz
title ("Subplot_2: Frequencia em Hertz");
ylabel ("Amplitude (No Normalizada)")
grid on; %turn on the grid

```

end