



UNIVERSIDADE FEDERAL DE MINAS GERAIS – UFMG  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA - DEE  
Circuitos Eletrônicos Analógicos - Prof. Hilton Mota

## **Trabalho Prático - Filtros LIT de Tempo Discreto**

Nander Santos do Carmo – 2018019931

Victor Freitas Borges - 2018122155

Belo Horizonte, março de 2021

## SUMÁRIO

INTRODUÇÃO.....	5
OBJETIVOS.....	6
1ª ETAPA: OBTENÇÃO DO SINAL DE ÁUDIO.....	7
2ª ETAPA: ESPECTRO DE FREQUÊNCIAS DO SINAL.....	9
3ª ETAPA: BANCO DE FILTROS.....	11
Filtro de Subgraves [16~60]Hz.....	12
Filtro de Graves [60~250]Hz.....	15
Filtro de Médio-graves [250~2000]Hz.....	18
Filtro de Médio-agudos [2000~4000]Hz.....	21
Filtro de Agudos [4000~6000]Hz.....	24
Filtro de Brilho [6000~16000]Hz.....	27
Análise do Banco de Filtros e Metodologia de Projeto.....	30
4ª ETAPA: RECONSTRUÇÃO DO SINAL.....	32
5ª ETAPA: REPRODUÇÃO DOS SINAIS.....	36
CONCLUSÃO.....	37
BIBLIOGRAFIA.....	39

## LISTA DE IMAGENS

Figura 1: Script para obtenção do sinal de áudio.....	7
Figura 2: Sinal de Áudio Obtido.....	7
Figura 3: Frequência de Amostragem do Sinal de Áudio.....	8
Figura 4: Espectro de Frequências do Sinal de Áudio.....	9
Figura 5: Diagramas de Magnitude e Fase do Espectro de Frequência do Sinal de Áudio.....	9
Figura 6: Parâmetros Globais do Banco de Filtros.....	11
Figura 7: Script do Filtro de Subgraves.....	12
Figura 8: Informações do Filtro de Subgraves.....	12
Figura 9: Resposta de Magnitude e Fase do Filtro de Subgraves.....	13
Figura 10: Diagrama de Polos e Zeros do Filtro de Subgraves.....	13
Figura 11: Função de Transferência do Filtro de Subgraves.....	13
Figura 12: Sinal de Áudio Propagado pelo Filtro de Subgraves.....	14
Figura 13: Script do Filtro de Graves.....	15
Figura 14: Informações do Filtro de Graves.....	15
Figura 15: Resposta de Magnitude e Fase do Filtro de Graves.....	16
Figura 16: Diagrama de Polos e Zeros do Filtro de Graves.....	16
Figura 17: Função de Transferência do Filtro de Graves.....	16
Figura 18: Sinal de Áudio Propagado pelo Filtro de Graves.....	17
Figura 19: Script do Filtro de Médio-graves.....	18
Figura 20: Informações do Filtro de Médio-graves.....	18
Figura 21: Resposta de Magnitude e Fase do Filtro de Médio-graves.....	19
Figura 22: Diagrama de Polos e Zeros do Filtro de Médio-graves.....	19
Figura 23: Função de Transferência do Filtro de Médio-graves.....	19
Figura 24: Sinal de Áudio Propagado pelo Filtro de Médio-graves.....	20
Figura 25: Script do Filtro de Médio-agudos.....	21
Figura 26: Informações do Filtro de Médio-agudos.....	21
Figura 27: Resposta de Magnitude e Fase do Filtro de Médio-agudos.....	22
Figura 28: Diagrama de Polos e Zeros do Filtro de Médio-agudos.....	22
Figura 29: Função de Transferência do Filtro de Médio-agudos.....	22
Figura 30: Sinal de Áudio Propagado pelo Filtro de Médio-agudos.....	23

Figura 31: Script do Filtro de Agudos.....	24
Figura 32: Informações do Filtro de Agudos.....	24
Figura 33: Resposta de Magnitude e Fase do Filtro de Agudos.....	25
Figura 34: Diagrama de Polos e Zeros do Filtro de Agudos.....	25
Figura 35: Função de Transferência do Filtro de Agudos.....	25
Figura 36: Sinal de Áudio Propagado pelo Filtro de Agudos.....	26
Figura 37: Script do Filtro de Brilho.....	27
Figura 38: Informações do Filtro de Brilho.....	27
Figura 39: Resposta de Magnitude e Fase do Filtro de Agudos.....	28
Figura 40: Diagrama de Polos e Zeros do Filtro de Brilho.....	28
Figura 41: Função de Transferência do Filtro de Brilho.....	28
Figura 42: Sinal de Áudio Propagado pelo Filtro de Brilho.....	29
Figura 43: Script para Reconstrução do Sinal de Áudio Filtrado.....	32
Figura 44: Sinal de Áudio Filtrado e Reconstruído.....	33
Figura 45: Sinal Reconstruído com Ganho de -40dB.....	33
Figura 46: Sinal Reconstruído com Ganho de 6.02dB.....	34
Figura 47: Script para Geração dos Diagramas de Magnitude e Fase do Espectro de Frequência do Sinal de Áudio Filtrado e Reconstruído.....	35
Figura 48: Diagrama de Magnitude e Fase do Espectro de Frequência do Sinal de Áudio Filtrado e Reconstruído.....	35
Figura 49: Script para Reprodução dos Sinais de Áudio.....	36
Figura 50: Resposta de Magnitude Total do Bando de Filtros.....	37
Figura 51: Diagrama de Polos e Zeros Total do Banco de Filtros.....	37
Figura 52: Atraso de Grupo do Banco de Filtros.....	38

## INTRODUÇÃO

Os filtros são uma construção que tem como objetivo isolar o sinal de interesse dos demais sinais que não são desejados. Para isso eles podem ser construídos a partir de diversas topologias e variações, com aplicação tanto para sinais tempo contínuo quanto para sinais tempo discreto, sendo possível de serem implementados via circuito elétrico ou via software. Para compreender o funcionamento de um filtro é necessário compreender as ferramentas de análise no domínio da frequência, sendo uma delas a Transformada de Fourier. Uma vez aplicada, a transformada permite que a análise deixe de acontecer no domínio do tempo e passe para o domínio da frequência. Nesse novo domínio podemos visualizar o funcionamento do filtro como um seletor de frequências, de modo que o princípio de funcionamento se baseia em atenuar componentes em frequências não desejadas. Dessa forma, o sinal filtrado possui somente as componentes de interesse.

Para o contexto do trabalho o filtro a ser implementado é de tempo discreto e vai tratar um sinal de áudio que foi amostrado com frequência de 44100Hz. Sendo assim, o sinal de entrada é discreto, de forma que a ferramenta aplicada para a sua análise é a transformada rápida de Fourier (FFT). A topologia definida é a Chebyshev do tipo 1, em que é um filtro construído a partir de uma função de transferência com polos e zeros não nulos, de modo a permitir um ripple na banda de passagem. Ademais, essa topologia nos permite controlar a seletividade em termos de controlar a faixa de frequência que possui menor atenuação.

Para todas as etapas de cálculo, projeto e visualização dos resultados será utilizado o MATLAB como ferramenta de trabalho, logo todos os scripts utilizarão funções disponíveis nesse ambiente e, assim, qualquer dúvida com relação à implementação dos códigos foi sanada através da consulta à documentação oficial, cujos links estão referenciados no final deste documento.

## OBJETIVOS

Desenvolver um sistema para análise espectral e filtragem para um sinal de áudio fornecido. Logo, o projeto deve apresentar um banco de filtros capaz de dividir o espectro de frequências do sinal fornecido nas seguintes faixas:

- Subgraves: entre 16 Hz e 60 Hz;
- Graves: entre 60 Hz e 250 Hz;
- Médio-graves: entre 250 Hz e 2 kHz;
- Médio-agudos: entre 2 kHz e 4 kHz;
- Agudos: entre 4 kHz e 6 kHz;
- Brilho: entre 6 kHz e 16 kHz.

Além disso, os filtros do banco do sistema devem atender minimamente aos seguintes requisitos de projeto:

- Filtro IIR baseado em um Filtro Chebyshev Tipo I;
- *Ripple* na faixa de passagem de 0.5dB;
- Atenuação da faixa de rejeição de 40dB;
- Ganho variável entre -40dB e 6.02dB;

## 1ª ETAPA: OBTENÇÃO DO SINAL DE ÁUDIO

A etapa do projeto consistiu na obtenção do vetor de dados do sinal de entrada, assim como sua frequência de amostragem. Assim, para a análise do arquivo de áudio foi produzido o seguinte script mostrado na Figura 1. O gráfico do sinal pelo tempo está representado na Figura 2, enquanto a frequência de amostragem do sinal recuperada está mostrada na Figura 3 .

```
%% Sinal de Entrada no Tempo

filename = 'Blind_intro.wav';

[signal,Fs] = audioread(filename);
signalLength = length(signal);

fprintf('\nFrequência de amostragem do sinal (Fs): %i\n', Fs);

maxTime = length(signal)/Fs;
timeVector = 0: maxTime / signalLength: maxTime - maxTime / signalLength;

figure();
plot(timeVector,signal);
title('Sinal original');
xlabel('Tempo (s)');
ylabel('Amplitude');
axis([0, 8.5, -1.5, 1.5]);
print('-clipboard','-dbitmap');
```

Figura 1: Script para obtenção do sinal de áudio.

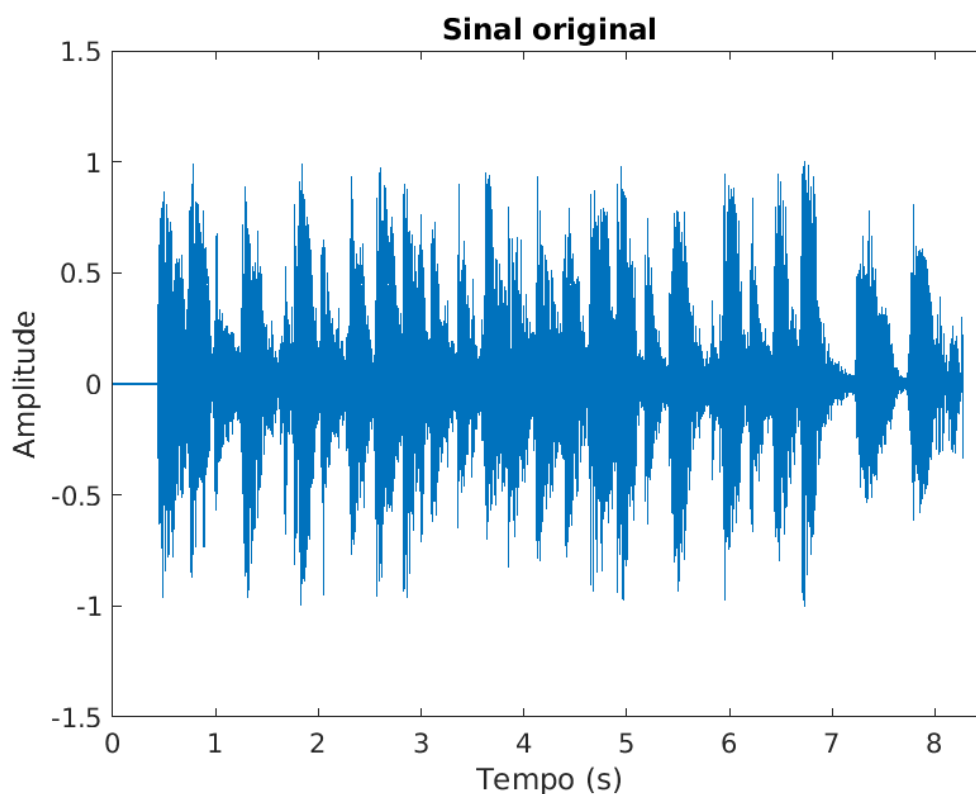
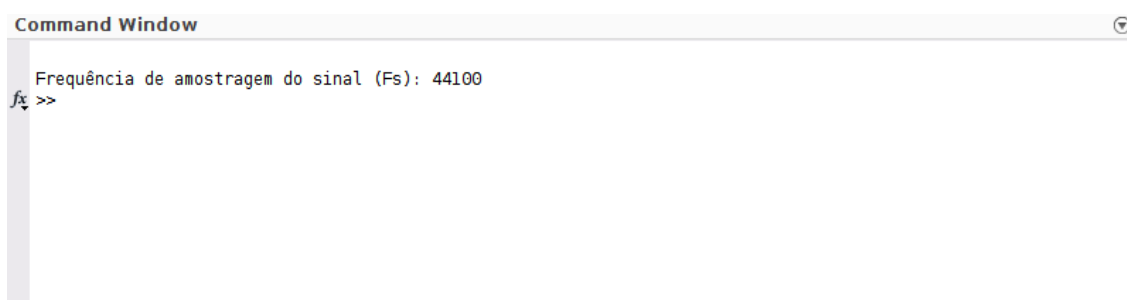


Figura 2: Sinal de Áudio Obtido.



*Figura 3: Frequência de Amostragem do Sinal de Áudio.*



## 2ª ETAPA: ESPECTRO DE FREQUÊNCIAS DO SINAL

A segunda etapa do projeto consistia no cálculo e apresentação do espectro de frequências do sinal fornecido, através da transformada rápida de Fourier, a FFT. Para isso o script mostrado na Figura 4 foi desenvolvido.

```
% Espectro de Frequências do Sinal de Áudio

Xw_signal = fftshift(fft(signal./signalLength));
freqVector = linspace(-Fs/2, Fs/2, signalLength);

figure();
subplot(2, 1, 1);
plot(freqVector, abs(Xw_signal(:,1)));
title('Espectro de frequência do sinal');
ylabel('Magnitude');
xlabel('Frequência [Hz]');
axis([-22.05E3, 22.05E3, 0, 15E-3]);
subplot(2, 1, 2);
plot(freqVector, unwrap(angle(Xw_signal(:,1))));
ylabel('Fase');
xlabel('Frequência [Hz]');
axis([-22.05E3, 22.05E3, -0.1E4, 3.1E4]);
print('-clipboard', '-dbitmap');
```

Figura 4: Espectro de Frequências do Sinal de Áudio.

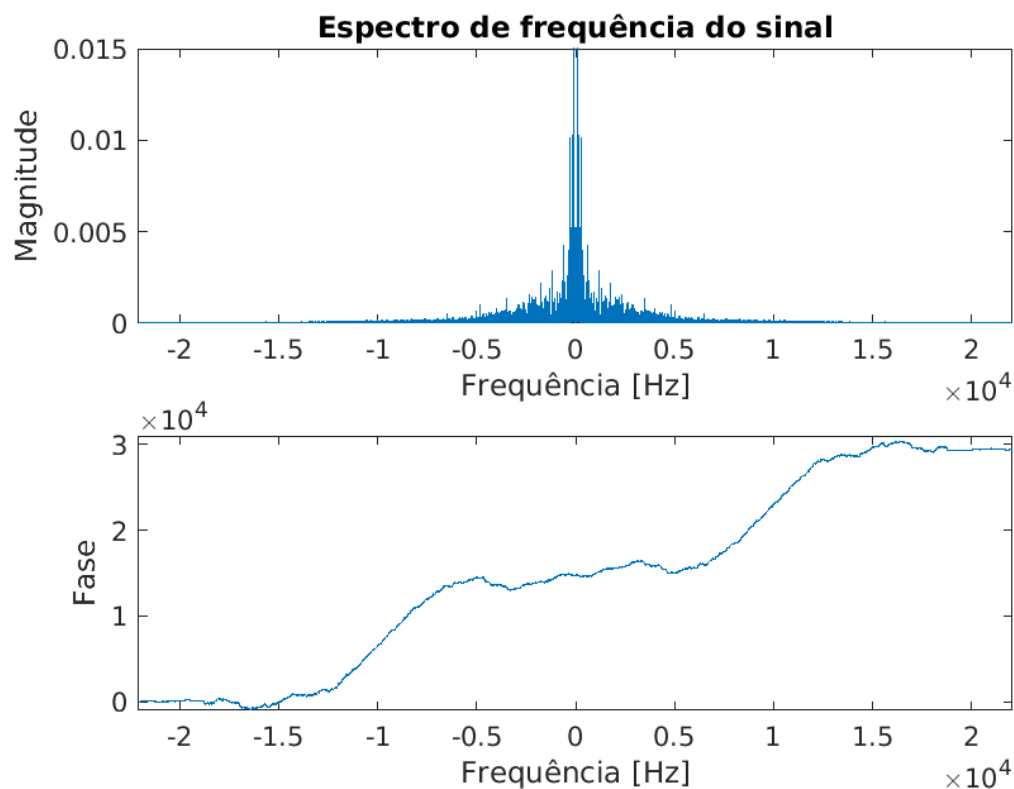


Figura 5: Diagramas de Magnitude e Fase do Espectro de Frequência do Sinal de Áudio.

No script é importante destacar o uso da função *fftshift* para representar o espectro de frequência centralizada em uma frequência de 0 Hz. Além disso, de forma a ajustar a amplitude do diagrama de magnitude do espectro foi feita uma normalização do sinal de entrada.

Como pode ser visto nos diagramas de magnitude e fase do espectro de frequências do sinal mostrado na Figura 5 acima, o sinal de entrada possui uma presença mais forte de sinais de baixa frequência do que de alta frequência.

### 3ª ETAPA: BANCO DE FILTROS

Essa sessão irá abordar o projeto de cada um dos 6 filtros que compõem o banco de filtragem do sistema. Inicialmente serão mostrados de forma individual para cada filtro o script utilizado para a sua criação e os resultados de simulação do filtro, envolvendo:

- Script do filtro;
- Informações completas do filtro;
- Resposta de amplitude e fase;
- Diagrama de polos e zeros;
- Função de transferência;
- Sinal de áudio após ser filtrado exclusivamente pelo filtro.

Assim, a análise acerca do método escolhido para o projeto dos filtros será realizada em uma seção posterior do relatório, enquanto a análise do impacto dos atrasos de fase e grupo será realizada no momento da recomposição do sinal filtrado após passar por todo o banco de filtros, com o auxílio da análise do sinal no domínio do tempo e de seu espectro de frequências.

Por fim, para a simplificação do projeto do banco de filtros foram definidos os parâmetros globais dos filtros, conforme mostrado na Figura 6.

---

```
%% Parâmetros do Banco de Filtros
```

```
Rp = 0.5;  
Rs = 40;  
delta_freq_low = 10;  
delta_freq_high = 100;
```

*Figura 6: Parâmetros Globais do Banco de Filtros.*

## Filtro de Subgraves [16~60]Hz

O primeiro filtro do banco desenvolvido é o filtro de subgraves, para frequências de 16Hz até frequências de 60Hz.

```
%% Filtro de Subgraves [16~60]Hz

fp_16_60 = [16, 60]; % Frequências da banda passante em Hz
fs_16_60 = [fp_16_60(1) - delta_freq_low, fp_16_60(2) + delta_freq_low]; % Frequências da faixa de corte em Hz
gain_16_60 = 0; % Ganho em dB do Filtro (-40~-6.02)dB;

bpFilt_16_60 = designfilt(...
    'bandpassiir', 'DesignMethod', 'cheby1',...
    'StopbandFrequency1', fs_16_60(1),...
    'StopbandFrequency2', fs_16_60(2),...
    'StopbandAttenuation1', Rs,...
    'StopbandAttenuation2', Rs,...
    'PassbandFrequency1', fp_16_60(1),...
    'PassbandFrequency2', fp_16_60(2),...
    'PassbandRipple', Rp,...
    'SampleRate', Fs);

[num_16_60, den_16_60] = tf(bpFilt_16_60);
Hz_16_60 = tf(num_16_60, den_16_60, 1 / Fs);
display(Hz_16_60);

fvtool(bpFilt_16_60, 'FrequencyScale', 'log');

filtered_16_60 = filter(bpFilt_16_60, signal);

figure();
subplot(2, 1, 1);
plot(timeVector, filtered_16_60);
axis([0, 8.5, -1, 1]);
title('Filtro de Subgraves [16~60]Hz');
xlabel('Tempo (s)');
ylabel('Amplitude');
subplot(2, 1, 2);
title('Sinal original');
plot(timeVector, signal);
axis([0, 8.5, -1, 1]);
xlabel('Tempo (s)');
ylabel('Amplitude');
print('-clipboard', '-dbitmap');

%sound(filtered_16_60, Fs);
```

Figura 7: Script do Filtro de Subgraves.

IIR Digital Filter (real)	
-----	
Number of Sections	: 9
Stable	: Yes
Linear Phase	: No
Design Method Information	
Design Algorithm	: Chebyshev type I
Design Options	
Match Exactly	: passband
Design Specifications	
Sample Rate	: 44.1 kHz
Response	: Bandpass
Specification	: Fst1,Fp1,Fp2,Fst2,Ast1,Ap,Ast2
Passband Ripple	: 0.5 dB
First Stopband Edge	: 6 Hz
First Passband Edge	: 16 Hz
Second Passband Edge	: 60 Hz
Second Stopband Edge	: 70 Hz
First Stopband Atten.	: 40 dB
Second Stopband Atten.	: 40 dB

Figura 8: Informações do Filtro de Subgraves.

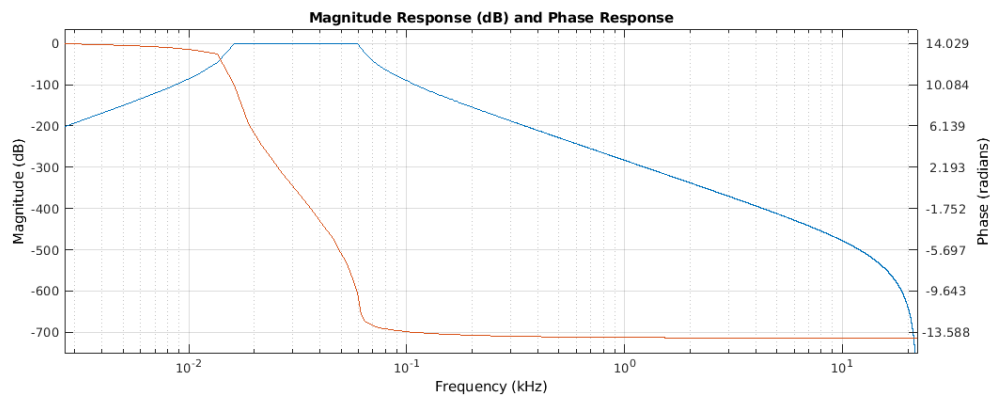


Figura 9: Resposta de Magnitude e Fase do Filtro de Subgraves.

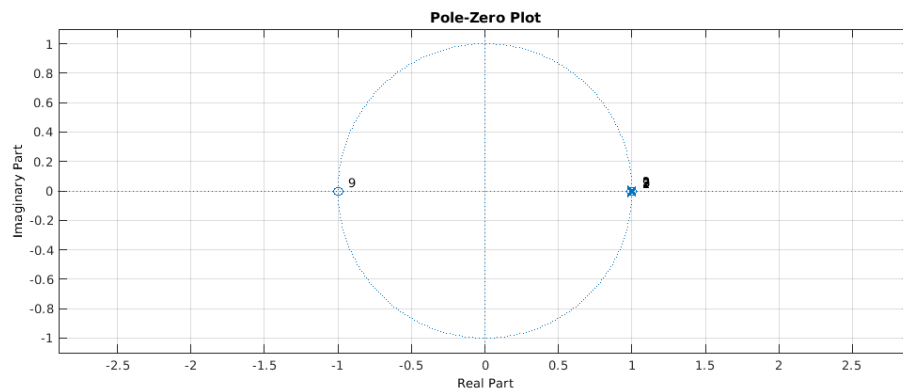


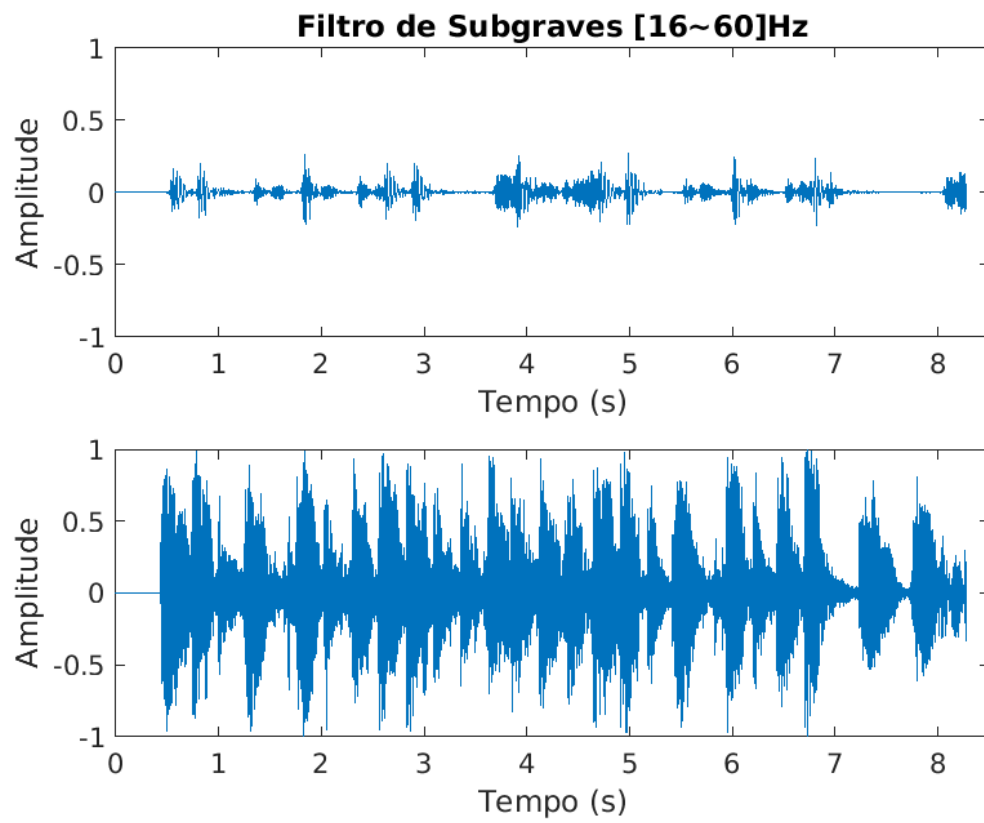
Figura 10: Diagrama de Polos e Zeros do Filtro de Subgraves.

Hz\_16\_60 =

$$\begin{aligned}
 & 3.254e-25 z^{18} - 2.929e-24 z^{16} + 1.172e-23 z^{14} - 2.734e-23 z^{12} + 4.101e-23 z^{10} - 4.101e-23 z^8 + 2.734e-23 z^6 \\
 & \quad - 1.172e-23 z^4 + 2.929e-24 z^2 - 3.254e-25 \\
 & \hline
 & z^{18} - 17.99 z^{17} + 152.9 z^{16} - 815 z^{15} + 3055 z^{14} - 8550 z^{13} + 1.852e04 z^{12} - 3.173e04 z^{11} + 4.362e04 z^{10} \\
 & \quad - 4.844e04 z^9 + 4.358e04 z^8 - 3.168e04 z^7 + 1.847e04 z^6 - 8523 z^5 + 3043 z^4 - 811.1 z^3 \\
 & \quad + 152 z^2 - 17.88 z + 0.9929
 \end{aligned}$$

Sample time: 2.2676e-05 seconds  
Discrete-time transfer function.

Figura 11: Função de Transferência do Filtro de Subgraves.



*Figura 12: Sinal de Áudio Propagado pelo Filtro de Subgraves.*

## Filtro de Graves [60~250]Hz

O segundo filtro do banco desenvolvido é o filtro de graves, para frequências de 60Hz até frequências de 250Hz.

```
%% Filtro de Graves [60~250]Hz

fp_60_250 = [60, 250]; % Frequências da banda passante em Hz
fs_60_250 = [fp_60_250(1) - delta_freq_low, fp_60_250(2) + delta_freq_high]; % Frequências da faixa de corte em Hz
gain_60_250 = 0; % Ganho em dB do Filtro (-40~-6.02)dB;

bpFilt_60_250 = designfilt(...
    'bandpassiir', 'DesignMethod', 'cheby1',...
    'StopbandFrequency1', fs_60_250(1),...
    'StopbandFrequency2', fs_60_250(2),...
    'StopbandAttenuation1', Rs,...
    'StopbandAttenuation2', Rs,...
    'PassbandFrequency1', fp_60_250(1),...
    'PassbandFrequency2', fp_60_250(2),...
    'PassbandRipple', Rp,...
    'SampleRate', Fs);

[num_60_250, den_60_250] = tf(bpFilt_60_250);
Hz_60_250 = tf(num_60_250, den_60_250, 1 / Fs);
display(Hz_60_250);

fvtool(bpFilt_60_250, 'FrequencyScale', 'log');

filtered_60_250 = filter(bpFilt_60_250, signal);

figure();
subplot(2, 1, 1);
plot(timeVector, filtered_60_250);
axis([0.5, 8.5, -1, 1]);
title('Filtro de Graves [60~250]Hz');
xlabel('Tempo (s)');
ylabel('Amplitude');
subplot(2, 1, 2);
title('Sinal original');
plot(timeVector, signal);
axis([0.5, 8.5, -1, 1]);
xlabel('Tempo (s)');
ylabel('Amplitude');
print('-clipboard', '-dbitmap');

%sound(filtered_60_250, Fs);
```

Figura 13: Script do Filtro de Graves.

```
IIR Digital Filter (real)
-----
Number of Sections : 9
Stable             : Yes
Linear Phase       : No

Design Method Information
Design Algorithm : Chebyshev type I

Design Options
Match Exactly : passband

Design Specifications
Sample Rate      : 44.1 kHz
Response         : Bandpass
Specification    : Fst1,Fp1,Fp2,Fst2,Ast1,Ap,Ast2
Passband Ripple  : 0.5 dB
First Stopband Edge : 50 Hz
First Passband Edge : 60 Hz
Second Passband Edge : 250 Hz
Second Stopband Edge : 350 Hz
First Stopband Atten. : 40 dB
Second Stopband Atten. : 40 dB
```

Figura 14: Informações do Filtro de Graves.

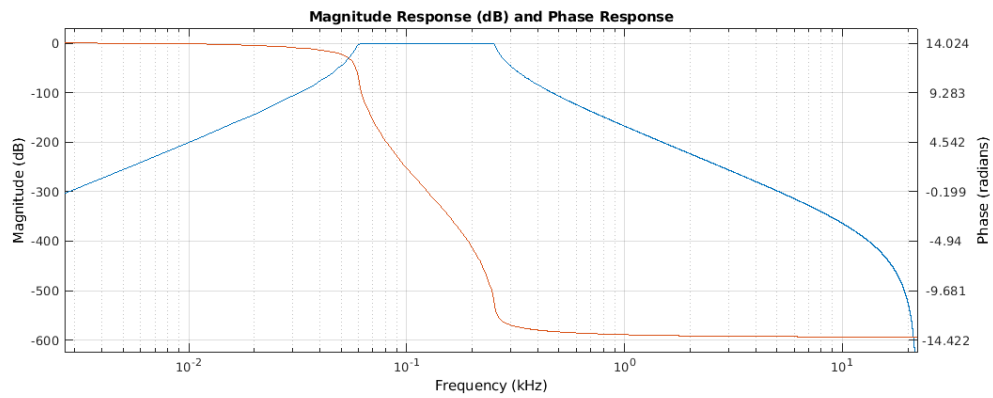


Figura 15: Resposta de Magnitude e Fase do Filtro de Graves.

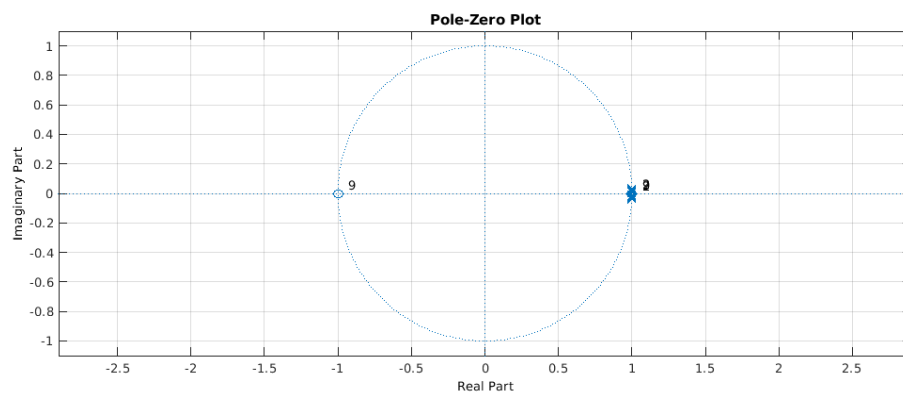


Figura 16: Diagrama de Polos e Zeros do Filtro de Graves.

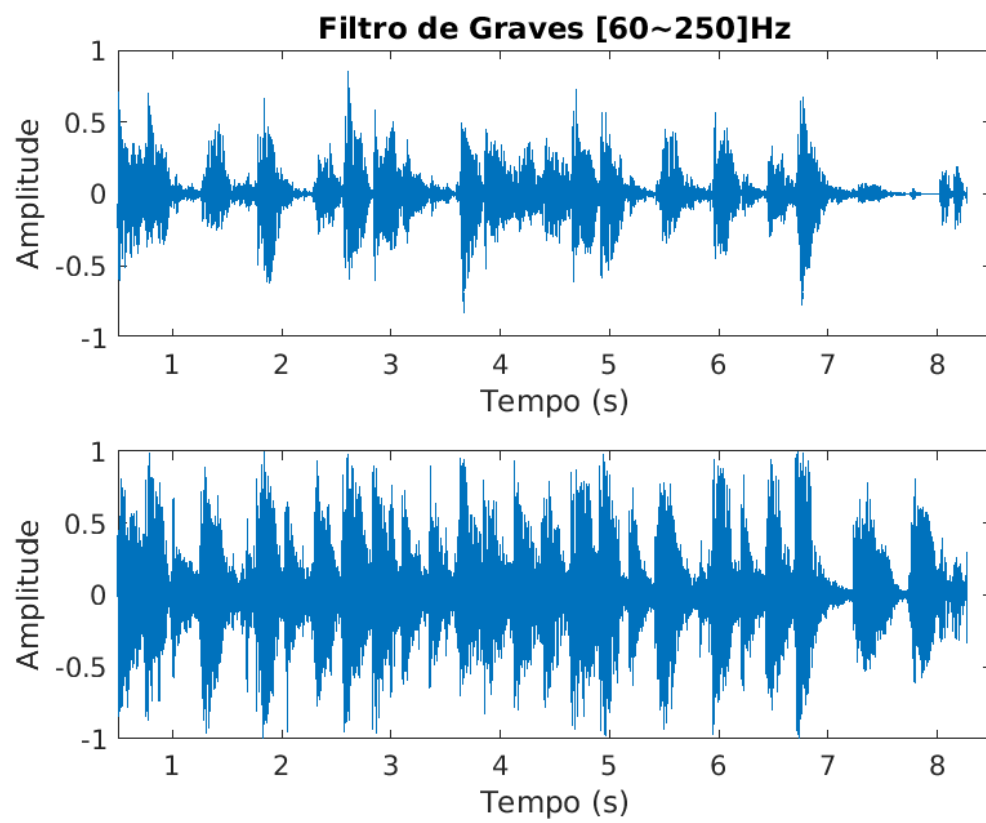
Hz\_60\_250 =

$$\begin{aligned}
 &1.679\text{e-}19 z^{18} - 1.511\text{e-}18 z^{16} + 6.045\text{e-}18 z^{14} - 1.41\text{e-}17 z^{12} + 2.116\text{e-}17 z^{10} - 2.116\text{e-}17 z^8 + 1.41\text{e-}17 z^6 \\
 &\quad - 6.045\text{e-}18 z^4 + 1.511\text{e-}18 z^2 - 1.679\text{e-}19 \\
 &\hline
 &z^{18} - 17.96 z^{17} + 152.4 z^{16} - 811.3 z^{15} + 3037 z^{14} - 8487 z^{13} + 1.835\text{e}04 z^{12} - 3.141\text{e}04 z^{11} \\
 &\quad + 4.311\text{e}04 z^{10} - 4.782\text{e}04 z^9 + 4.296\text{e}04 z^8 - 3.119\text{e}04 z^7 + 1.817\text{e}04 z^6 - 8371 z^5 + 2985 z^4 \\
 &\quad - 794.7 z^3 + 148.8 z^2 - 17.48 z + 0.9695
 \end{aligned}$$

Sample time: 2.2676e-05 seconds  
Discrete-time transfer function.

Figura 17: Função de Transferência do Filtro de Graves.





*Figura 18: Sinal de Áudio Propagado pelo Filtro de Graves.*

## Filtro de Médio-graves [250~2000]Hz

O primeiro filtro do banco desenvolvido é o filtro de médio-graves, para frequências de 250Hz até frequências de 2000Hz.

```
%% Filtro de Médio-graves [250~2000Hz]

fp_250_2k = [250, 2000]; % Frequências da banda passante em Hz
fs_250_2k = [fp_250_2k(1) - delta_freq_high, fp_250_2k(2) + delta_freq_high]; % Frequências da faixa de corte em Hz
gain_250_2k = 0; % Ganho em dB do Filtro (-40~-6.02)dB;

bpFilt_250_2k = designfilt(...
    'bandpassfir', 'DesignMethod', 'cheby1',...
    'StopbandFrequency1', fs_250_2k(1),...
    'StopbandFrequency2', fs_250_2k(2),...
    'StopbandAttenuation1', Rs,...
    'StopbandAttenuation2', Rs,...
    'PassbandFrequency1', fp_250_2k(1),...
    'PassbandFrequency2', fp_250_2k(2),...
    'PassbandRipple', Rp,...
    'SampleRate', Fs);

[num_250_2k, den_250_2k] = tf(bpFilt_250_2k);
Hz_250_2k = tf(num_250_2k, den_250_2k, 1 / Fs);
display(Hz_250_2k);

fvtool(bpFilt_250_2k, 'FrequencyScale', 'log');

filtered_250_2k = filter(bpFilt_250_2k, signal);

figure();
subplot(2, 1, 1);
plot(timeVector, filtered_250_2k);
axis([0, 8.5, -1, 1]);
title('Filtro de Médio-graves [250~2000Hz]');
xlabel('Tempo (s)');
ylabel('Amplitude');
subplot(2, 1, 2);
title('Sinal original');
plot(timeVector, signal);
axis([0, 8.5, -1, 1]);
xlabel('Tempo (s)');
ylabel('Amplitude');
print('-clipboard', '-dbitmap');

%sound(filtered_250_2k, Fs);
```

Figura 19: Script do Filtro de Médio-graves.

```
IIR Digital Filter (real)
-----
Number of Sections : 18
Stable             : Yes
Linear Phase       : No

Design Method Information
Design Algorithm : Chebyshev type I

Design Options
Match Exactly : passband

Design Specifications
Sample Rate      : 44.1 kHz
Response         : Bandpass
Specification    : Fst1,Fp1,Fp2,Fst2,Ast1,Ap,Ast2
Passband Ripple  : 0.5 dB
First Stopband Edge : 150 Hz
First Passband Edge : 250 Hz
Second Passband Edge : 2 kHz
Second Stopband Edge : 2.1 kHz
First Stopband Atten. : 40 dB
Second Stopband Atten. : 40 dB
```

Figura 20: Informações do Filtro de Médio-graves.

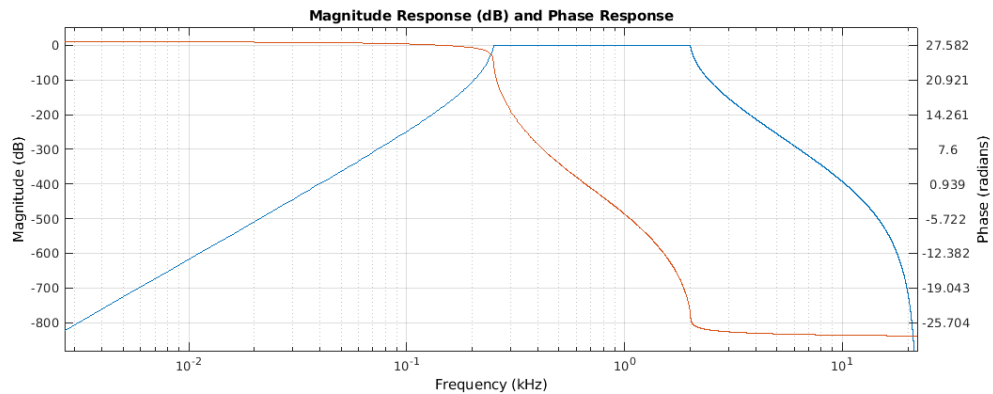


Figura 21: Resposta de Magnitude e Fase do Filtro de Médio-graves.

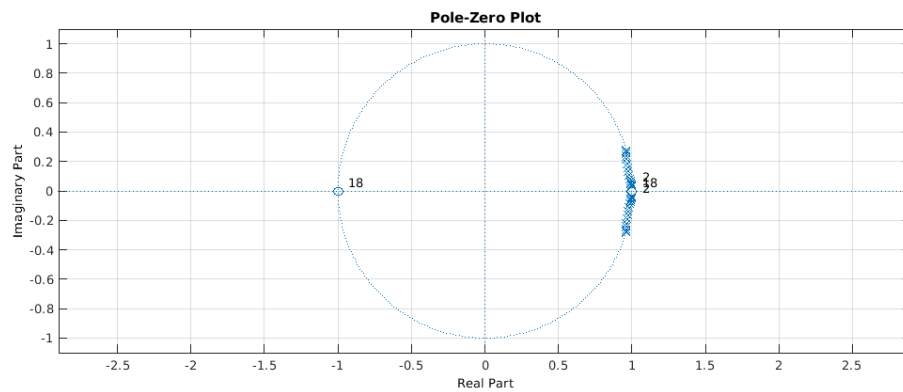


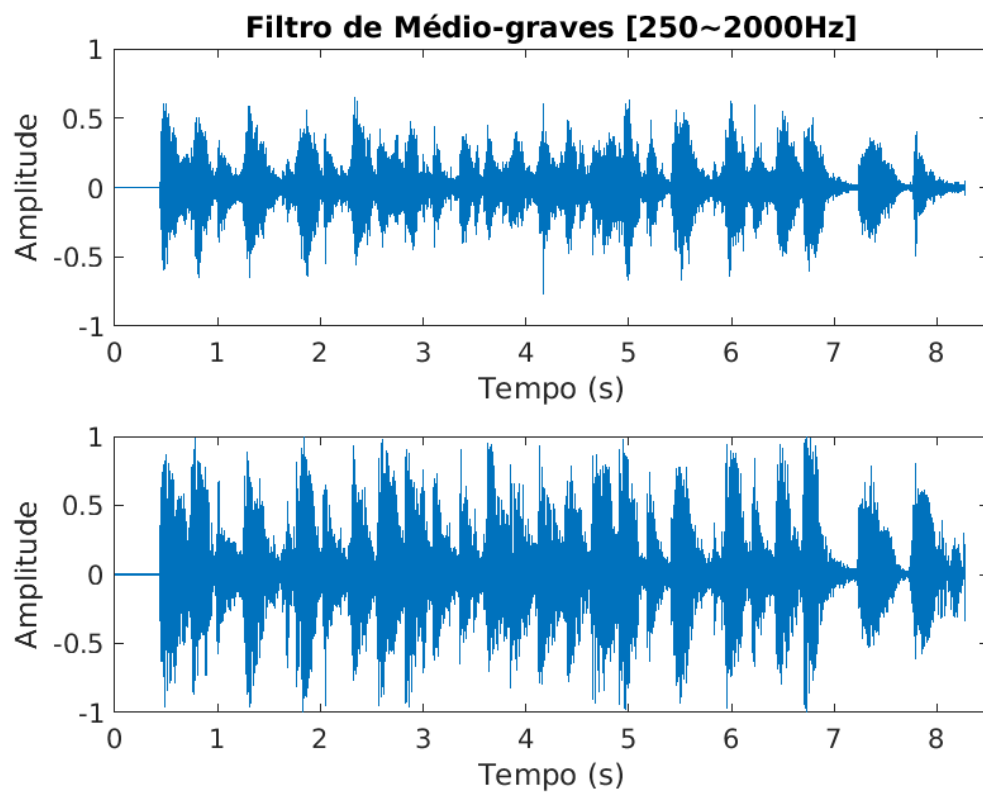
Figura 22: Diagrama de Polos e Zeros do Filtro de Médio-graves.

Hz\_250\_2k =

$$\begin{aligned}
 &1.027\text{e-}21 \, z^{36} - 1.849\text{e-}20 \, z^{34} + 1.571\text{e-}19 \, z^{32} - 8.38\text{e-}19 \, z^{30} + 3.143\text{e-}18 \, z^{28} - 8.799\text{e-}18 \, z^{26} + 1.907\text{e-}17 \, z^{24} \\
 &- 3.268\text{e-}17 \, z^{22} + 4.494\text{e-}17 \, z^{20} - 4.993\text{e-}17 \, z^{18} + 4.494\text{e-}17 \, z^{16} - 3.268\text{e-}17 \, z^{14} + 1.907\text{e-}17 \, z^{12} \\
 &- 8.799\text{e-}18 \, z^{10} + 3.143\text{e-}18 \, z^8 - 8.38\text{e-}19 \, z^6 + 1.571\text{e-}19 \, z^4 - 1.849\text{e-}20 \, z^2 + 1.027\text{e-}21 \\
 &----- \\
 &z^{36} - 35.26 \, z^{35} + 604.8 \, z^{34} - 6724 \, z^{33} + 5.445\text{e}04 \, z^{32} - 3.423\text{e}05 \, z^{31} + 1.739\text{e}06 \, z^{30} - 7.331\text{e}06 \, z^{29} \\
 &+ 2.616\text{e}07 \, z^{28} - 8.019\text{e}07 \, z^{27} + 2.135\text{e}08 \, z^{26} - 4.979\text{e}08 \, z^{25} + 1.024\text{e}09 \, z^{24} - 1.868\text{e}09 \, z^{23} \\
 &+ 3.035\text{e}09 \, z^{22} - 4.406\text{e}09 \, z^{21} + 5.727\text{e}09 \, z^{20} - 6.677\text{e}09 \, z^{19} + 6.99\text{e}09 \, z^{18} - 6.573\text{e}09 \, z^{17} \\
 &+ 5.549\text{e}09 \, z^{16} - 4.203\text{e}09 \, z^{15} + 2.851\text{e}09 \, z^{14} - 1.727\text{e}09 \, z^{13} + 9.321\text{e}08 \, z^{12} - 4.46\text{e}08 \, z^{11} \\
 &+ 1.883\text{e}08 \, z^{10} - 6.962\text{e}07 \, z^9 + 2.236\text{e}07 \, z^8 - 6.168\text{e}06 \, z^7 + 1.44\text{e}06 \, z^6 - 2.791\text{e}05 \, z^5 + 4.371\text{e}04 \, z^4 \\
 &- 5313 \, z^3 + 470.5 \, z^2 - 27 \, z + 0.754
 \end{aligned}$$

Sample time: 2.2676e-05 seconds  
Discrete-time transfer function.

Figura 23: Função de Transferência do Filtro de Médio-graves.



*Figura 24: Sinal de Áudio Propagado pelo Filtro de Médio-graves.*

## Filtro de Médio-agudos [2000~4000]Hz

O primeiro filtro do banco desenvolvido é o filtro de médio-agudos, para frequências de 2000Hz até frequências de 4000Hz.

```
%% Filtro de Médio-agudos [2000-4000Hz]

fp_2k_4k = [2000, 4000]; % Frequências da banda passante em Hz
fs_2k_4k = [fp_2k_4k(1) - delta_freq_high, fp_2k_4k(2) + delta_freq_high]; % Frequências da faixa de corte em Hz
gain_2k_4k = 0; % Ganho em dB do Filtro (-40~-6.02)dB;

bpFilt_2k_4k = designfilt(...
    'bandpassiir', 'DesignMethod', 'cheby1',...
    'StopbandFrequency1', fs_2k_4k(1),...
    'StopbandFrequency2', fs_2k_4k(2),...
    'StopbandAttenuation1', Rs,...
    'StopbandAttenuation2', Rs,...
    'PassbandFrequency1', fp_2k_4k(1),...
    'PassbandFrequency2', fp_2k_4k(2),...
    'PassbandRipple', Rp,...
    'SampleRate', Fs);

[num_2k_4k, den_2k_4k] = tf(bpFilt_2k_4k);
Hz_2k_4k = tf(num_2k_4k, den_2k_4k, 1 / Fs);
display(Hz_2k_4k);

fvtool(bpFilt_2k_4k, 'FrequencyScale', 'log');

filtered_2k_4k = filter(bpFilt_2k_4k, signal);

figure();
subplot(2, 1, 1);
plot(timeVector, filtered_2k_4k);
axis([0, 8.5, -1, 1]);
title('Filtro de Médio-agudos [2000~4000Hz]');
xlabel('Tempo (s)');
ylabel('Amplitude');
subplot(2, 1, 2);
title('Sinal original');
plot(timeVector, signal);
axis([0, 8.5, -1, 1]);
xlabel('Tempo (s)');
ylabel('Amplitude');
print('-clipboard', '-dbitmap');

%sound(filtered_2k_4k, Fs);
```

Figura 25: Script do Filtro de Médio-agudos.

IIR Digital Filter (real)

---

Number of Sections : 17  
 Stable : Yes  
 Linear Phase : No

Design Method Information  
 Design Algorithm : Chebyshev type I

Design Options  
 Match Exactly : passband

Design Specifications  
 Sample Rate : 44.1 kHz  
 Response : Bandpass  
 Specification : Fst1,Fp1,Fp2,Fst2,Ast1,Ap,Ast2  
 Passband Ripple : 0.5 dB  
 First Stopband Edge : 1.9 kHz  
 First Passband Edge : 2 kHz  
 Second Passband Edge : 4 kHz  
 Second Stopband Edge : 4.1 kHz  
 First Stopband Atten. : 40 dB  
 Second Stopband Atten. : 40 dB

Figura 26: Informações do Filtro de Médio-agudos.

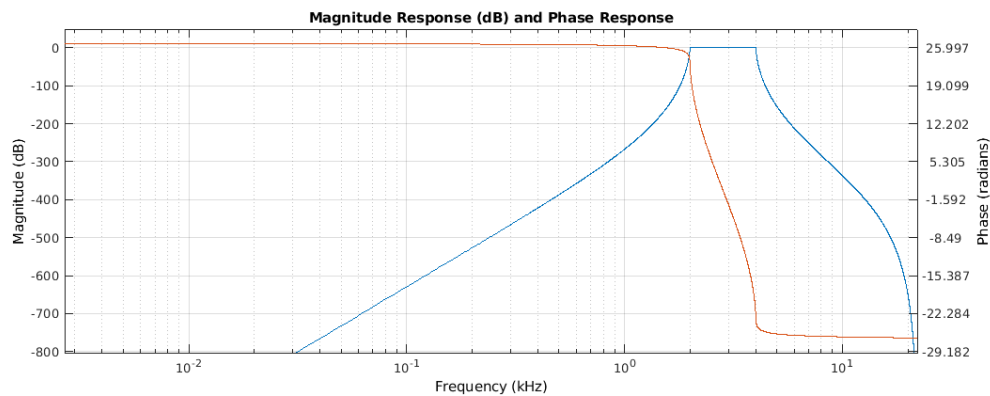


Figura 27: Resposta de Magnitude e Fase do Filtro de Médio-agudos.

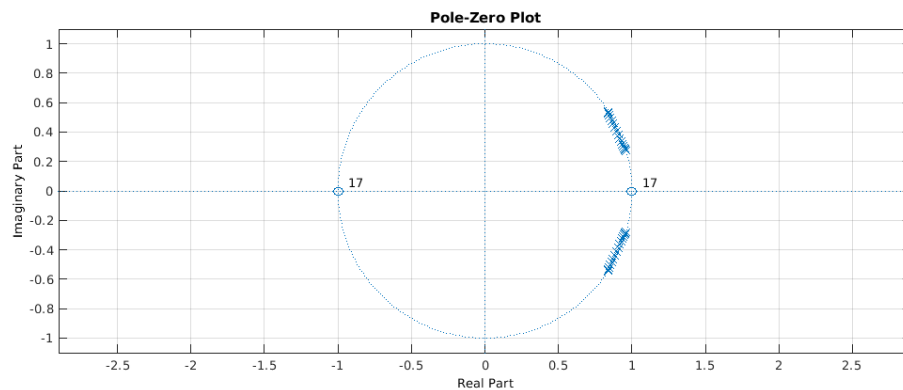


Figura 28: Diagrama de Polos e Zeros do Filtro de Médio-agudos.

Hz\_2k\_4k =

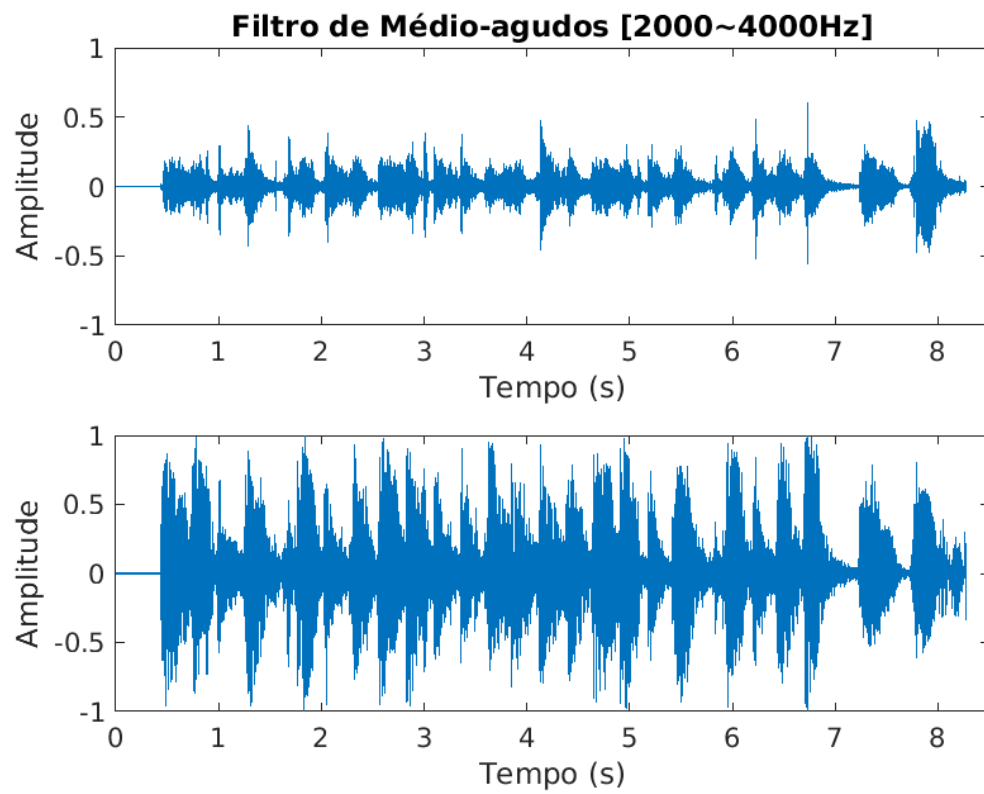
$$\begin{aligned}
 &1.571\text{e-}19 \, z^{34} - 2.671\text{e-}18 \, z^{32} + 2.137\text{e-}17 \, z^{30} - 1.069\text{e-}16 \, z^{28} + 3.74\text{e-}16 \, z^{26} - 9.724\text{e-}16 \, z^{24} + 1.945\text{e-}15 \, z^{22} \\
 &- 3.056\text{e-}15 \, z^{20} + 3.82\text{e-}15 \, z^{18} - 3.82\text{e-}15 \, z^{16} + 3.056\text{e-}15 \, z^{14} - 1.945\text{e-}15 \, z^{12} + 9.724\text{e-}16 \, z^{10} \\
 &- 3.74\text{e-}16 \, z^8 + 1.069\text{e-}16 \, z^6 - 2.137\text{e-}17 \, z^4 + 2.671\text{e-}18 \, z^2 - 1.571\text{e-}19
 \end{aligned}$$


---


$$\begin{aligned}
 &z^{34} - 30.65 \, z^{33} + 458.7 \, z^{32} - 4464 \, z^{31} + 3.176\text{e}04 \, z^{30} - 1.759\text{e}05 \, z^{29} + 7.898\text{e}05 \, z^{28} - 2.951\text{e}06 \, z^{27} \\
 &+ 9.358\text{e}06 \, z^{26} - 2.554\text{e}07 \, z^{25} + 6.068\text{e}07 \, z^{24} - 1.265\text{e}08 \, z^{23} + 2.33\text{e}08 \, z^{22} - 3.81\text{e}08 \, z^{21} \\
 &+ 5.553\text{e}08 \, z^{20} - 7.233\text{e}08 \, z^{19} + 8.438\text{e}08 \, z^{18} - 8.826\text{e}08 \, z^{17} + 8.279\text{e}08 \, z^{16} - 6.963\text{e}08 \, z^{15} \\
 &+ 5.245\text{e}08 \, z^{14} - 3.531\text{e}08 \, z^{13} + 2.119\text{e}08 \, z^{12} - 1.129\text{e}08 \, z^{11} + 5.312\text{e}07 \, z^{10} - 2.194\text{e}07 \, z^9 \\
 &+ 7.886\text{e}06 \, z^8 - 2.44\text{e}06 \, z^7 + 6.408\text{e}05 \, z^6 - 1.401\text{e}05 \, z^5 + 2.481\text{e}04 \, z^4 - 3421 \, z^3 + 344.9 \, z^2 \\
 &- 22.62 \, z + 0.7241
 \end{aligned}$$

Sample time: 2.2676e-05 seconds  
Discrete-time transfer function.

Figura 29: Função de Transferência do Filtro de Médio-agudos.



*Figura 30: Sinal de Áudio Propagado pelo Filtro de Médio-agudos.*

## Filtro de Agudos [4000~6000]Hz

O primeiro filtro do banco desenvolvido é o filtro de agudos, para frequências de 4000Hz até frequências de 6000Hz.

```
%% Filtro de Agudos [4000-6000Hz]

fp_4k_6k = [4000, 6000]; % Frequências da banda passante em Hz
fs_4k_6k = [fp_4k_6k(1) - delta_freq_high, fp_4k_6k(2) + delta_freq_high]; % Frequências da faixa de corte em Hz
gain_4k_6k = 0; % Ganho em dB do Filtro (-40~-6.02)dB;

bpFilt_4k_6k = designfilt(...
    'bandpassiir', 'DesignMethod', 'cheby1',...
    'StopbandFrequency1', fs_4k_6k(1),...
    'StopbandFrequency2', fs_4k_6k(2),...
    'StopbandAttenuation1', Rs,...
    'StopbandAttenuation2', Rs,...
    'PassbandFrequency1', fp_4k_6k(1),...
    'PassbandFrequency2', fp_4k_6k(2),...
    'PassbandRipple', Rp,...
    'SampleRate', Fs);

[num_4k_6k, den_4k_6k] = tf(bpFilt_4k_6k);
Hz_4k_6k = tf(num_4k_6k, den_4k_6k, 1 / Fs);
display(Hz_4k_6k);

fvtool(bpFilt_4k_6k, 'FrequencyScale', 'log');

filtered_4k_6k = filter(bpFilt_4k_6k, signal);

figure();
subplot(2, 1, 1);
plot(timeVector, filtered_4k_6k);
axis([0, 8.5, -1, 1]);
title('Filtro de Agudos [4000-6000Hz]');
xlabel('Tempo (s)');
ylabel('Amplitude');
subplot(2, 1, 2);
title('Sinal original');
plot(timeVector, signal);
axis([0, 8.5, -1, 1]);
xlabel('Tempo (s)');
ylabel('Amplitude');
print('-clipboard', '-dbitmap');

%sound(filtered_4k_6k, Fs);
```

Figura 31: Script do Filtro de Agudos.

IIR Digital Filter (real)	
-----	
Number of Sections	: 16
Stable	: Yes
Linear Phase	: No
Design Method Information	
Design Algorithm : Chebyshev type I	
Design Options	
Match Exactly : passband	
Design Specifications	
Sample Rate	: 44.1 kHz
Response	: Bandpass
Specification	: Fst1,Fp1,Fp2,Fst2,Ast1,Ap,Ast2
Passband Ripple	: 0.5 dB
First Stopband Edge	: 3.9 kHz
First Passband Edge	: 4 kHz
Second Passband Edge	: 6 kHz
Second Stopband Edge	: 6.1 kHz
First Stopband Atten.	: 40 dB
Second Stopband Atten.	: 40 dB

Figura 32: Informações do Filtro de Agudos.



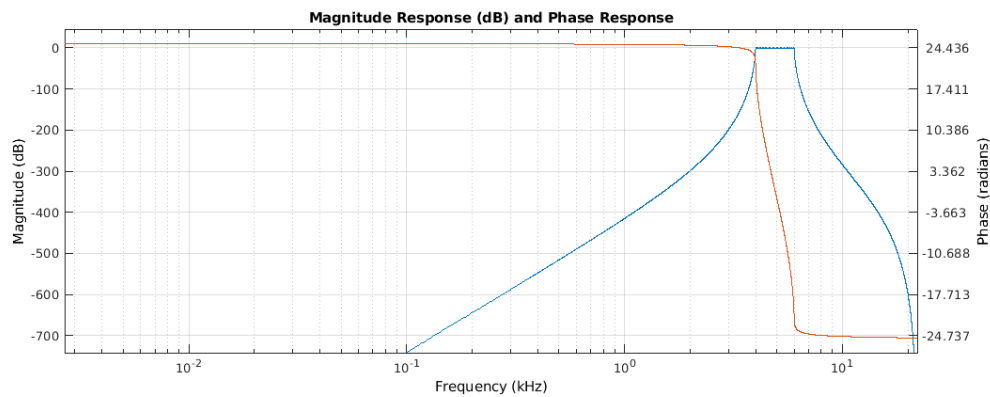


Figura 33: Resposta de Magnitude e Fase do Filtro de Agudos.

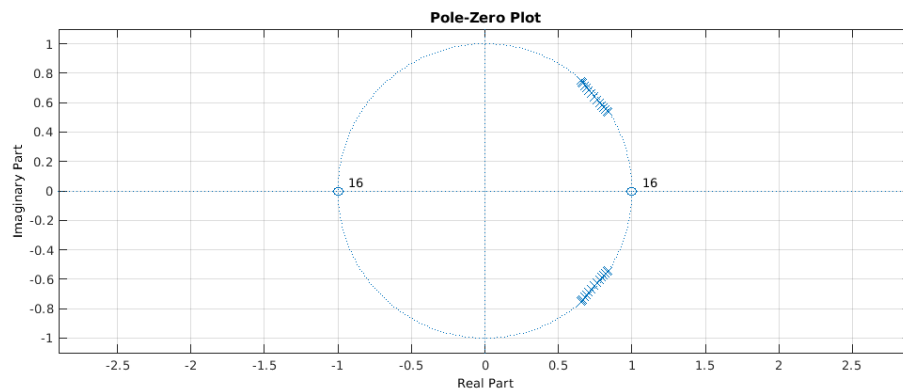


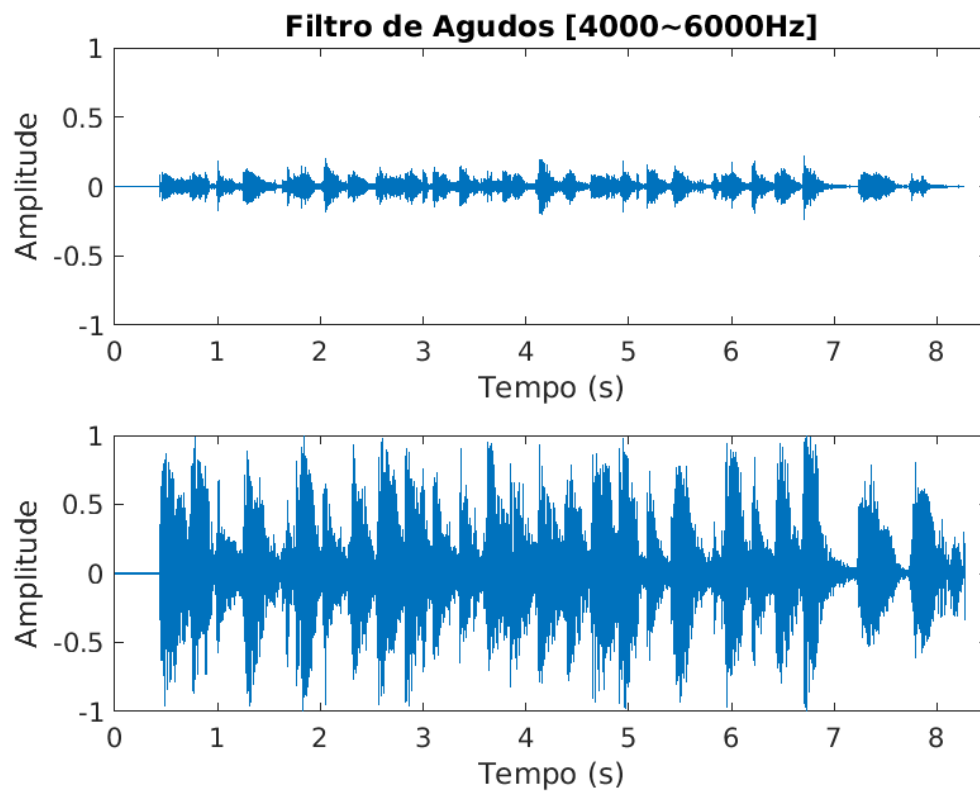
Figura 34: Diagrama de Polos e Zeros do Filtro de Agudos.

Hz\_4k\_6k =

$$\begin{aligned}
 &2.202\text{e-}18 \, z^{32} - 3.523\text{e-}17 \, z^{30} + 2.642\text{e-}16 \, z^{28} - 1.233\text{e-}15 \, z^{26} + 4.008\text{e-}15 \, z^{24} - 9.619\text{e-}15 \, z^{22} \\
 &+ 1.763\text{e-}14 \, z^{20} - 2.519\text{e-}14 \, z^{18} + 2.834\text{e-}14 \, z^{16} - 2.519\text{e-}14 \, z^{14} + 1.763\text{e-}14 \, z^{12} - 9.619\text{e-}15 \, z^{10} \\
 &+ 4.008\text{e-}15 \, z^8 - 1.233\text{e-}15 \, z^6 + 2.642\text{e-}16 \, z^4 - 3.523\text{e-}17 \, z^2 + 2.202\text{e-}18 \\
 &\hline
 &z^{32} - 23.97 \, z^{31} + 285 \, z^{30} - 2233 \, z^{29} + 1.296\text{e}04 \, z^{28} - 5.92\text{e}04 \, z^{27} + 2.215\text{e}05 \, z^{26} - 6.969\text{e}05 \, z^{25} \\
 &+ 1.877\text{e}06 \, z^{24} - 4.39\text{e}06 \, z^{23} + 9.006\text{e}06 \, z^{22} - 1.633\text{e}07 \, z^{21} + 2.634\text{e}07 \, z^{20} - 3.796\text{e}07 \, z^{19} \\
 &+ 4.902\text{e}07 \, z^{18} - 5.688\text{e}07 \, z^{17} + 5.936\text{e}07 \, z^{16} - 5.574\text{e}07 \, z^{15} + 4.708\text{e}07 \, z^{14} - 3.573\text{e}07 \, z^{13} \\
 &+ 2.43\text{e}07 \, z^{12} - 1.476\text{e}07 \, z^{11} + 7.978\text{e}06 \, z^{10} - 3.811\text{e}06 \, z^9 + 1.597\text{e}06 \, z^8 - 5.81\text{e}05 \, z^7 + 1.81\text{e}05 \, z^6 \\
 &- 4.741\text{e}04 \, z^5 + 1.017\text{e}04 \, z^4 - 1718 \, z^3 + 214.8 \, z^2 - 17.71 \, z + 0.724
 \end{aligned}$$

Sample time: 2.2676e-05 seconds  
Discrete-time transfer function.

Figura 35: Função de Transferência do Filtro de Agudos.



*Figura 36: Sinal de Áudio Propagado pelo Filtro de Agudos.*

## Filtro de Brilho [6000~16000]Hz

O primeiro filtro do banco desenvolvido é o filtro de brilho, para frequências de 6000Hz até frequências de 16000Hz.

```
%% Filtro de Brilho [6000~16000Hz]

fp_6k_16k = [6000, 16000]; % Frequências da banda passante em Hz
fs_6k_16k = [fp_6k_16k(1) - delta_freq_high, fp_6k_16k(2) + delta_freq_high]; % Frequências da faixa de corte em Hz
gain_6k_16k = 0; % Ganho em dB do Filtro (-40~6.02)dB;

bpFilt_6k_16k = designfilt(...
    'bandpassiir', 'DesignMethod', 'cheby1',...
    'StopbandFrequency1', fs_6k_16k(1),...
    'StopbandFrequency2', fs_6k_16k(2),...
    'StopbandAttenuation1', Rs,...
    'StopbandAttenuation2', Rs,...
    'PassbandFrequency1', fp_6k_16k(1),...
    'PassbandFrequency2', fp_6k_16k(2),...
    'PassbandRipple', Rp,...
    'SampleRate', Fs);

[num_6k_16k, den_6k_16k] = tf(bpFilt_6k_16k);
Hz_6k_16k = tf(num_6k_16k, den_6k_16k, 1 / Fs);
display(Hz_6k_16k);

fvtool(bpFilt_6k_16k, 'FrequencyScale', 'log');

filtered_6k_16k = filter(bpFilt_6k_16k, signal);

figure();
subplot(2, 1, 1);
plot(timeVector, filtered_6k_16k);
axis([0, 8.5, -1, 1]);
title('Filtro de Brilho [6000~16000Hz]');
xlabel('Tempo (s)');
ylabel('Amplitude');
subplot(2, 1, 2);
title('Sinal original');
plot(timeVector, signal);
axis([0, 8.5, -1, 1]);
xlabel('Tempo (s)');
ylabel('Amplitude');
print('-clipboard', '-dbitmap');

%sound(filtered_6k_16k, Fs);
```

Figura 37: Script do Filtro de Brilho.

```
IIR Digital Filter (real)
-----
Number of Sections : 27
Stable             : Yes
Linear Phase       : No

Design Method Information
Design Algorithm : Chebyshev type I

Design Options
Match Exactly : passband

Design Specifications
Sample Rate      : 44.1 kHz
Response         : Bandpass
Specification    : Fst1,Fp1,Fp2,Fst2,Ast1,Ap,Ast2
Passband Ripple  : 0.5 dB
First Stopband Edge : 5.9 kHz
First Passband Edge : 6 kHz
Second Passband Edge : 16 kHz
Second Stopband Edge : 16.1 kHz
First Stopband Atten. : 40 dB
Second Stopband Atten. : 40 dB
```

Figura 38: Informações do Filtro de Brilho.

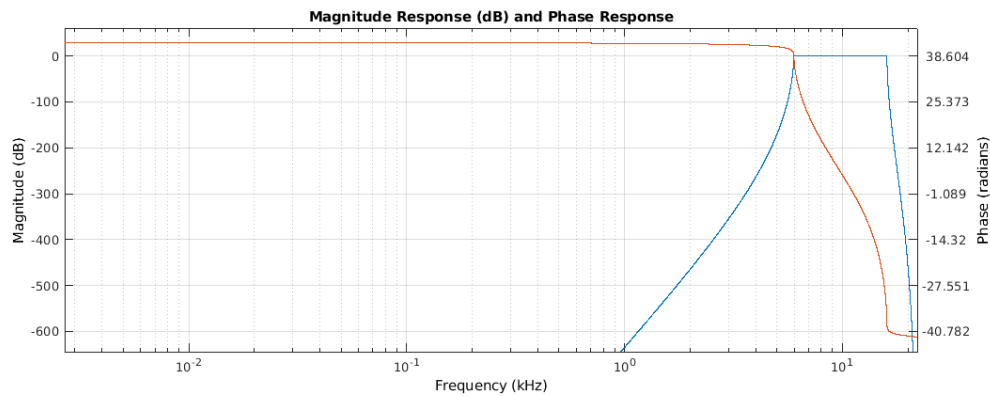


Figura 39: Resposta de Magnitude e Fase do Filtro de Agudos.

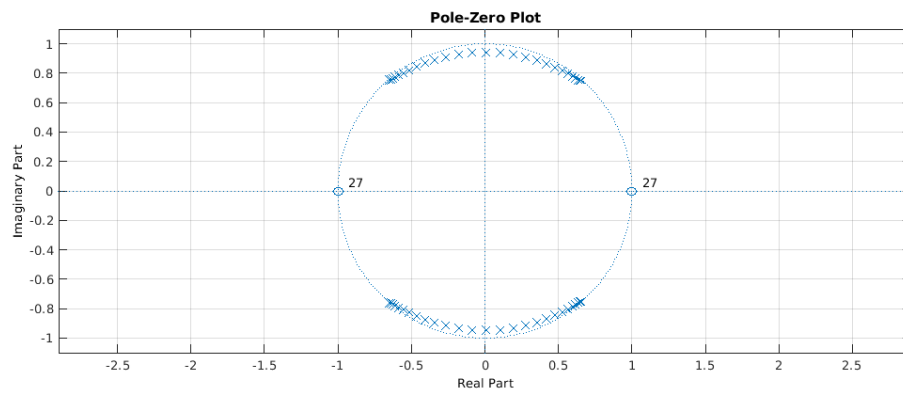


Figura 40: Diagrama de Polos e Zeros do Filtro de Brilho.

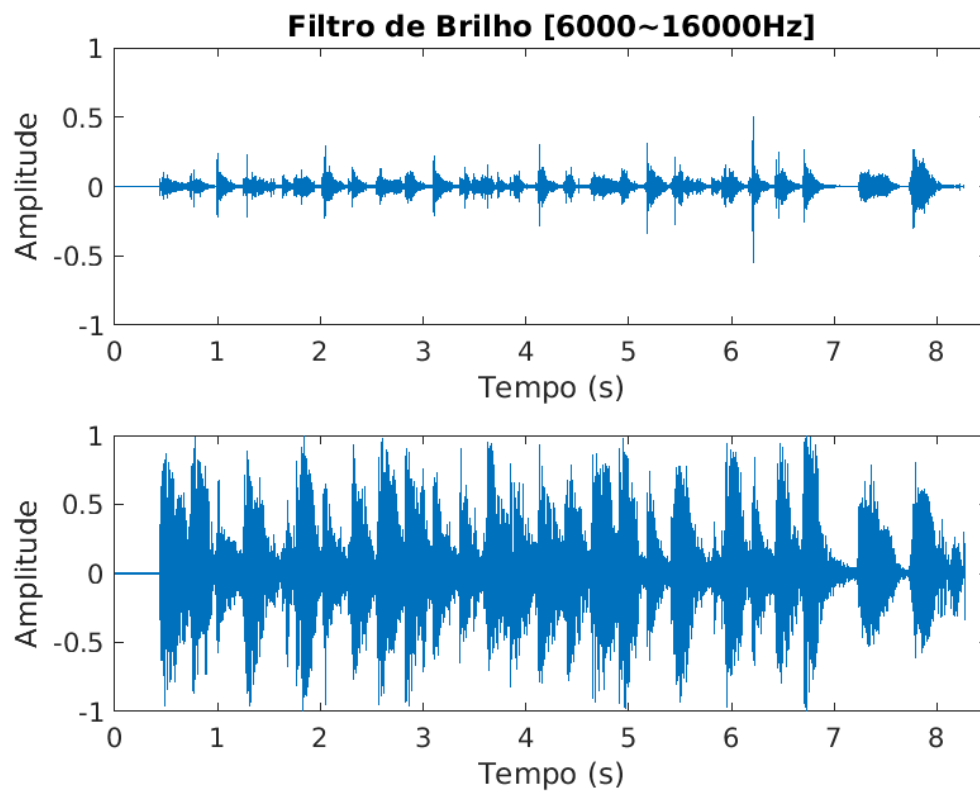
Hz\_6k\_16k =

$$\begin{aligned}
 &6.525e-12 z^{54} - 1.762e-10 z^{52} + 2.29e-09 z^{50} - 1.909e-08 z^{48} + 1.145e-07 z^{46} - 5.268e-07 z^{44} + 1.932e-06 z^{42} \\
 &- 5.795e-06 z^{40} + 1.449e-05 z^{38} - 3.058e-05 z^{36} + 5.505e-05 z^{34} - 8.507e-05 z^{32} + 0.0001134 z^{30} \\
 &- 0.0001309 z^{28} + 0.0001309 z^{26} - 0.0001134 z^{24} + 8.507e-05 z^{22} - 5.505e-05 z^{20} + 3.058e-05 z^{18} \\
 &- 1.449e-05 z^{16} + 5.795e-06 z^{14} - 1.932e-06 z^{12} + 5.268e-07 z^{10} - 1.145e-07 z^8 + 1.909e \\
 &-08 z^6 - 2.29e-09 z^4 + 1.762e-10 z^2 - 6.525e-12
 \end{aligned}$$

$$\begin{aligned}
 &z^{54} - 0.1871 z^{53} + 12.77 z^{52} - 2.365 z^{51} + 85.62 z^{50} - 15.63 z^{49} + 396 z^{48} - 71.02 z^{47} + 1407 z^{46} \\
 &- 247 z^{45} + 4062 z^{44} - 696.7 z^{43} + 9872 z^{42} - 1649 z^{41} + 2.065e04 z^{40} - 3351 z^{39} + 3.779e04 z^{38} \\
 &- 5940 z^{37} + 6.114e04 z^{36} - 9283 z^{35} + 8.821e04 z^{34} - 1.29e04 z^{33} + 1.141e05 z^{32} - 1.601e04 z^{31} \\
 &+ 1.329e05 z^{30} - 1.783e04 z^{29} + 1.397e05 z^{28} - 1.784e04 z^{27} + 1.326e05 z^{26} - 1.605e04 z^{25} \\
 &+ 1.137e05 z^{24} - 1.298e04 z^{23} + 8.797e04 z^{22} - 9401 z^{21} + 6.121e04 z^{20} - 6080 z^{19} + 3.814e04 z^{18} \\
 &- 3490 z^{17} + 2.115e04 z^{16} - 1763 z^{15} + 1.035e04 z^{14} - 774.5 z^{13} + 4418 z^{12} - 291.2 z^{11} \\
 &+ 1618 z^{10} - 91.48 z^9 + 496.2 z^8 - 23.17 z^7 + 123 z^6 - 4.463 z^5 + 23.24 z^4 - 0.5848 z^3 \\
 &+ 2.997 z^2 - 0.03947 z + 0.1998
 \end{aligned}$$

Sample time: 2.2676e-05 seconds  
Discrete-time transfer function.

Figura 41: Função de Transferência do Filtro de Brilho.



*Figura 42: Sinal de Áudio Propagado pelo Filtro de Brilho.*

## Análise do Banco de Filtros e Metodologia de Projeto

Conforme visto nos tópicos anteriores todos os filtros foram implementados utilizando a função do MATLAB *designfilt*, sendo representado através dos objetos `bpFilt_x_y` nos scripts, onde *x* representa a frequência esquerda da banda passante e *y* representa a frequência direita da banda passante do filtro. Essa função ajuda no desenvolvimento de filtros do tipo IIR, desde que configurada corretamente. Assim, respeitados os valores individuais dos filtros, como as frequências da banda passante e da faixa de rejeição, os parâmetros da função foram definidos da seguinte forma:

- `bandpassiir`: esse parâmetro define o filtro como um passa-baixas do tipo IIR;
- `DesignMethod`, `cheby1`: Essa dupla de parâmetros define a topologia do filtro a ser desenvolvido. No caso foi escolhido um filtro Chebyshev do tipo I;
- `StopbandFrequency1`, `fs_x_y(1)`: Essa dupla de parâmetros define a frequência de corte da faixa de rejeição de baixas frequências do filtro. No caso, `fs_x_y(1)` representa a frequência esquerda de corte da faixa de rejeição do filtro;
- `StopbandFrequency2`, `fs_x_y(2)`: Mesmo que o tópico anterior, porém para a faixa de rejeição a altas frequências do filtro, assim, `fs_x_y(2)` representa a frequência esquerda de corte da faixa de rejeição do filtro;
- `StopbandAttenuation1` e `StopbandAttenuation2`, `Rs`: Esses parâmetros que recebem o valor `Rs` representam o valor da atenuação da faixa de rejeição do filtro;
- `PassbandFrequency1`, `fp_x_y(1)`: Representa a frequência esquerda de corte da faixa de passagem do filtro dada pelo primeiro elemento do vetor de frequências de passagem `fp_x_y`;

- PassbandFrequency2,  $fp\_x\_y(2)$ : Representa a frequência direita de corte da faixa de passagem dada pelo segundo elemento do vetor de frequências de passagem  $fp\_x\_y$ ;
- PassbandRipple,  $Rp$ : Representa o valor de oscilação (*ripple*) máximo da faixa de passagem, dado por  $Rp$ ;
- SampleRate,  $F_s$ : Representa a frequência de amostragem do filtro, dada por  $F_s$ .

Assim, definidos esses parâmetros do filtro, é preciso destacar como foram definidos os valores das frequências da faixa de rejeição. Inicialmente foi escolhido uma diferença de -10Hz para a frequência esquerda da faixa de rejeição e +10Hz para a frequência direita da faixa de rejeição para todos os filtros, porém para os filtros de alta frequência (acima de 250Hz, por exemplo) esse valor exigia uma parede muito íngreme na resposta do filtro, o que provocava uma alta distorção na fase do filtro, comprometendo a filtragem do sinal. Assim, para frequências acima de 250Hz o delta de frequência assumido para determinar a banda de rejeição do filtro passou a ser de 100Hz.

Assim, o Como pode ser observado nas imagens presentes na seção de cada um dos filtros, utilizando a função *tf* aplicada ao objeto do filtro gerado, é possível obter a função de transferência. Além disso, através da função *fvtool* é possível obter as imagens relacionadas a resposta de magnitude e fase, diagrama de polos e zeros do filtro, além de informação acerca do atraso de grupo e respostas ao impulso e degrau.

Por fim, de forma a aplicar o filtro produzido no sinal de áudio carregado para o script é utilizada a função *filter*. Essa função retorna então o sinal de áudio já filtrado.

## 4ª ETAPA: RECONSTRUÇÃO DO SINAL

Essa etapa visa realizar a reconstrução do sinal através da soma do sinal de saída resultante de cada um dos filtros pertencentes ao banco. Além disso, é preciso aplicar o ganho variável de cada estágio de filtro, visto que a saída de cada filtro gerado possui ganho unitário. Esse ganho é representado pela variável  $k_{x,y}$  e pode ajustado individualmente, conforme necessário. O script utilizado para realizar a recomposição do sinal, representado pela variável `filtered_signal` pode ser observado na Figura 43.

---

```
%% Reconstrução do Sinal Filtrado e Ajuste de Ganho

k_16_60 = 1; % Pode variar entre 0.1 (-40dB) até 2 (6.02dB). 1 representa um ganho de 0dB
k_60_250 = 1; % Pode variar entre 0.1 (-40dB) até 2 (6.02dB). 1 representa um ganho de 0dB
k_250_2k = 1; % Pode variar entre 0.1 (-40dB) até 2 (6.02dB). 1 representa um ganho de 0dB
k_2k_4k = 1; % Pode variar entre 0.1 (-40dB) até 2 (6.02dB). 1 representa um ganho de 0dB
k_4k_6k = 1; % Pode variar entre 0.1 (-40dB) até 2 (6.02dB). 1 representa um ganho de 0dB
k_6k_16k = 1; % Pode variar entre 0.1 (-40dB) até 2 (6.02dB). 1 representa um ganho de 0dB

filtered_signal = 0;
filtered_signal = filtered_signal + filtered_16_60 * k_16_60;
filtered_signal = filtered_signal + filtered_60_250 * k_60_250;
filtered_signal = filtered_signal + filtered_250_2k * k_250_2k;
filtered_signal = filtered_signal + filtered_2k_4k * k_2k_4k;
filtered_signal = filtered_signal + filtered_4k_6k * k_4k_6k;
filtered_signal = filtered_signal + filtered_6k_16k * k_6k_16k;

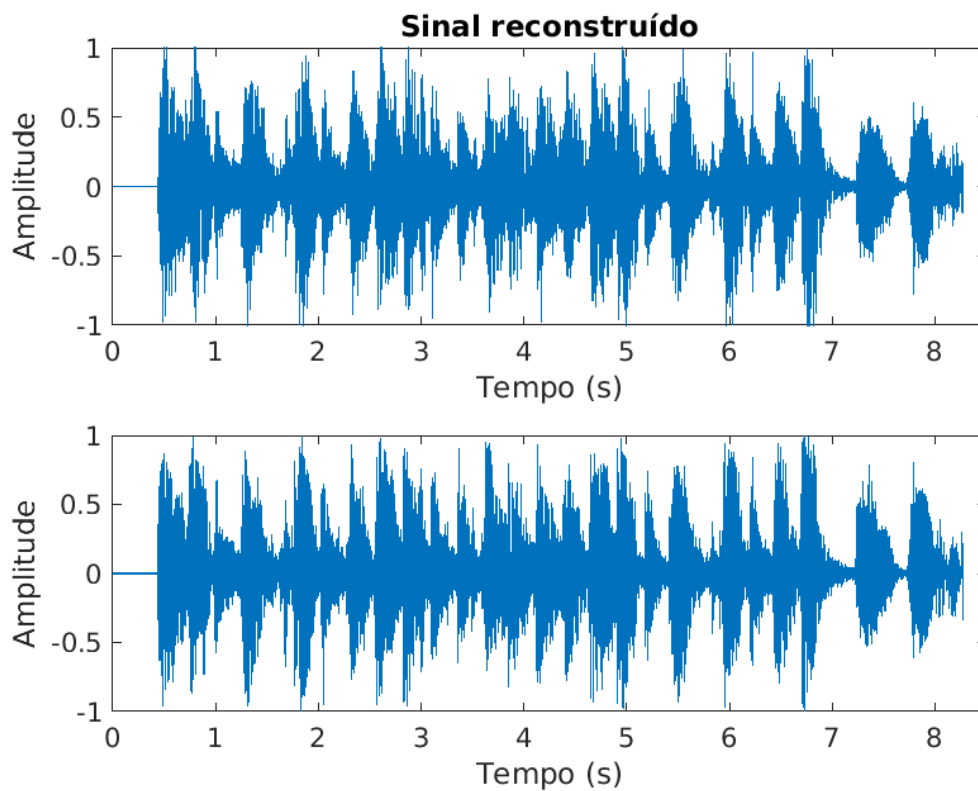
fvtool(bpFilt_16_60, bpFilt_60_250, bpFilt_250_2k, bpFilt_2k_4k, bpFilt_4k_6k, bpFilt_6k_16k, 'FrequencyScale', 'log');

figure();
subplot(2, 1, 1);
plot(timeVector, filtered_signal);
axis([0, 8.5, -1, 1]);
title('Sinal reconstruído');
xlabel('Tempo (s)');
ylabel('Amplitude');
subplot(2, 1, 2);
title('Sinal original');
plot(timeVector, signal);
axis([0, 8.5, -1, 1]);
xlabel('Tempo (s)');
ylabel('Amplitude');
print('-clipboard','-dbitmap');
```

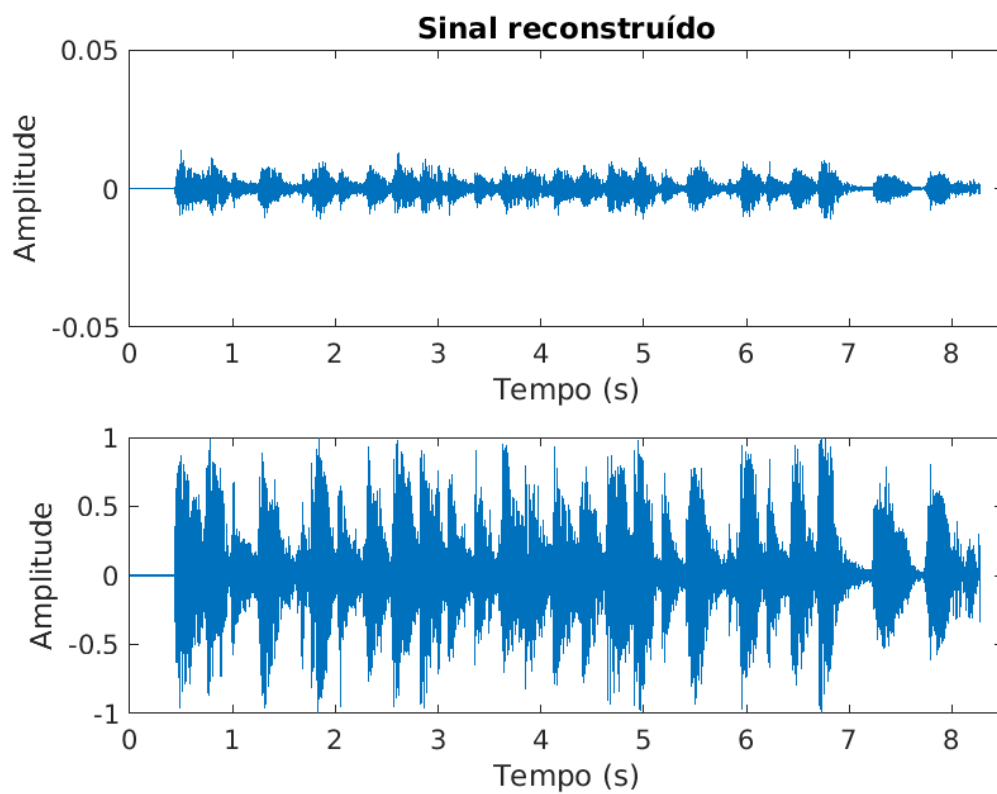
Figura 43: Script para Reconstrução do Sinal de Áudio Filtrado.

A Figura 44 representa o sinal de áudio reconstruído após passar pelo bando de filtros em comparação ao sinal original de áudio representado no gráfico inferior da imagem. É possível perceber que apesar de pequenas distorções no sinal reconstruído, ele possui forma muito parecida ao sinal original, o que valida o comportamento do sistema.

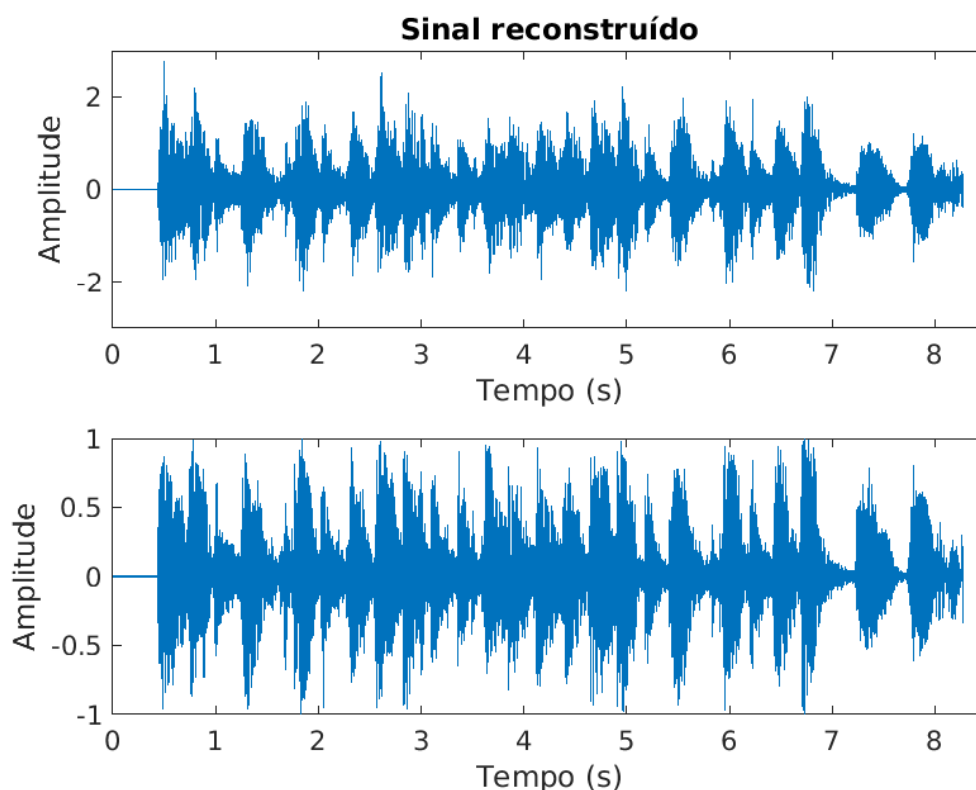




*Figura 44: Sinal de Áudio Filtrado e Reconstruído.*



*Figura 45: Sinal Reconstruído com Ganho de -40dB.*



*Figura 46: Sinal Reconstruído com Ganho de 6.02dB.*

Com relação ao ajuste de ganho a Figura 45 representa o sinal reconstruído em comparação ao sinal original quando aplicado um ganho de -40dB ao banco de filtros por inteiro. Enquanto isso, a Figura 46 representa o sinal após receber um ganho de 6.02dB em todos os espectros do filtro, conforme pode ser verificado observando a escala dos gráficos.

Por fim, foi possível plotar também o espectro de frequência do sinal filtrado (considerando novamente um ganho unitário). A Figura 47 representa o script utilizado para plotar os diagramas de magnitude e fase do espectro de frequência do sinal de áudio após ser filtrado e reconstruído. Conforme pode ser comparado com a Figura 5 é possível perceber que o espectro de frequência permanece praticamente inalterado. Isso é devido ao fato de que a maior parte do espectro de frequência do sinal estar dentro da faixa de passagem do banco de filtros: como as frequências de de cada filtro adjacente são iguais o banco como um todo funciona como um filtro passa faixa de 16Hz

a 16kHz. Assim é possível validar o comportamento do sistema com relação à resposta em frequência.

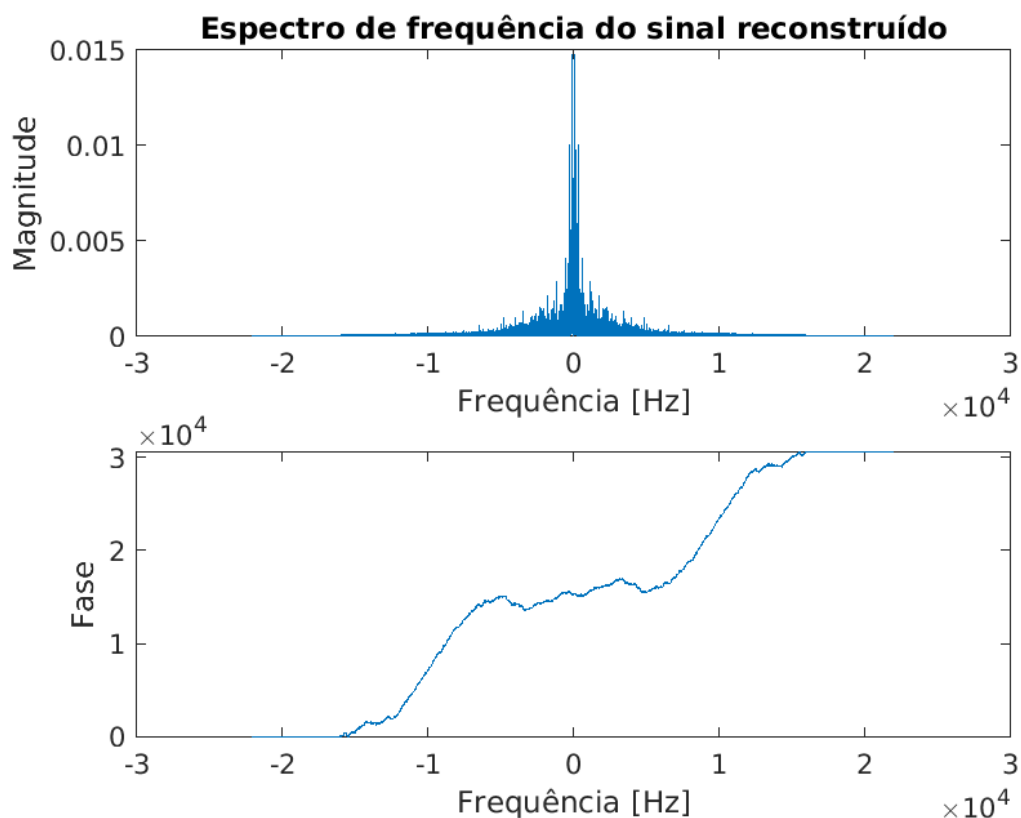
---

```
% Espectro de Frequências do Sinal de Áudio Filtrado

Xw_filtered = fftshift(fft(filtered_signal./signalLength));

figure();
subplot(2, 1, 1);
plot(freqVector, abs(Xw_filtered(:,1)));
title('Espectro de frequência do sinal reconstruído');
ylabel('Magnitude');
xlabel('Frequência [Hz]');
subplot(2, 1, 2);
plot(freqVector, unwrap(angle(Xw_filtered(:,1))));
ylabel('Fase');
xlabel('Frequência [Hz]');
print('-clipboard','-dbitmap');
```

*Figura 47: Script para Geração dos Diagramas de Magnitude e Fase do Espectro de Frequência do Sinal de Áudio Filtrado e Reconstruído..*



*Figura 48: Diagrama de Magnitude e Fase do Espectro de Frequência do Sinal de Áudio Filtrado e Reconstruído.*

## 5ª ETAPA: REPRODUÇÃO DOS SINAIS

Essa etapa é muito simples e, infelizmente, não será possível de comprovar sem que seja rodado o programa, porém basicamente, utilizando a função *sound* foi possível reproduzir tanto o sinal original de áudio original quanto o sinal filtrado e reconstruído. Assim, foi possível verificar que o resultado do sinal após o filtro é extremamente similar ao áudio original, apresentando uma distorção perceptível (apesar de leve) apenas no instante em que o som começa e no instante em que se encerra. O script usado para essa reprodução, apesar de muito simples, está representado na Figura 49.

---

### **%% Reprodução do Sinal Filtrado**

```
%sound(signal, Fs); % Reprodução do sinal original no sistema de áudio  
sound(filtered_signal, Fs); % Reprodução do sinal no sistema de áudio do computador
```

*Figura 49: Script para Reprodução dos Sinais de Áudio.*

## CONCLUSÃO

Conforme falado nos tópicos anteriores desse documento o sistema no final pode ser interpretado como um filtro passa-faixas de 16Hz até 16kHz, quando analisado o banco de filtros com um todo, devido ao caráter complementar das faixas de frequência de cada filtro individual. Assim esse sistema basicamente permite a passagem de frequências correspondente a 3 décadas de frequência a partir de 16Hz.

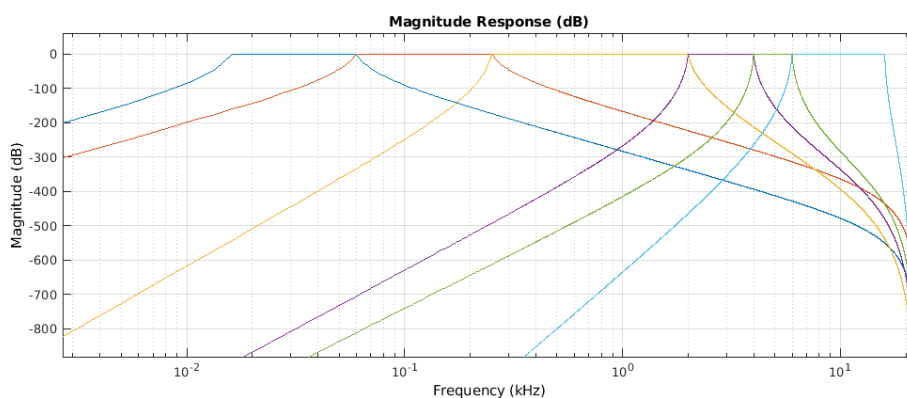


Figura 50: Resposta de Magnitude Total do Bando de Filtros.

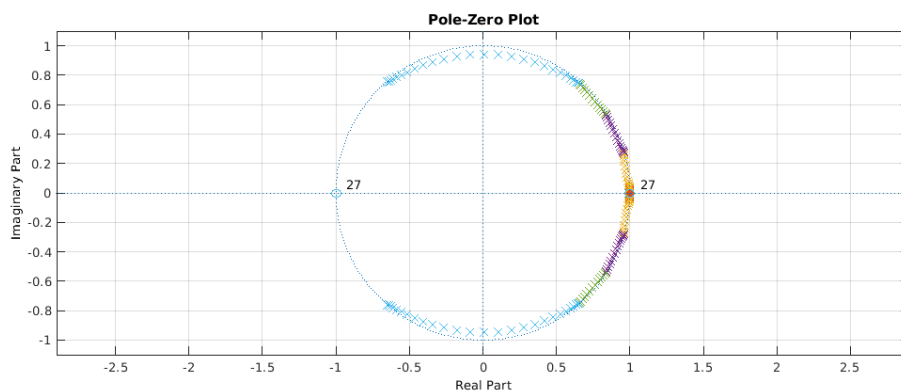
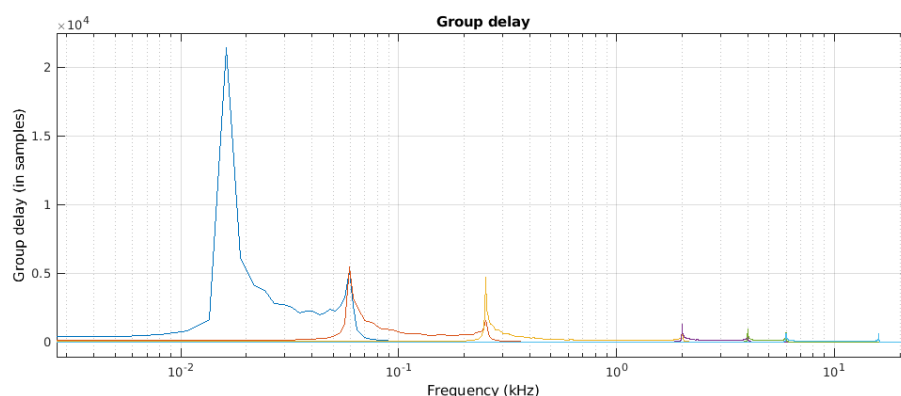


Figura 51: Diagrama de Polos e Zeros Total do Banco de Filtros.

Esse comportamento pode ser visto tanto ao analisar a reposta de magnitude total do banco de filtros representado na Figura 50. É possível perceber nessa imagem que o filtro permite a passagem de frequências de

forma contínua dentro da faixa de 16Hz e 16kHz. Da mesma forma, ao analisar o diagrama de polos e zeros total do filtro é possível ver que a resposta de magnitude do sistema seria exatamente esse. É possível perceber uma concentração de zeros em  $z=1$  (correspondente a baixas frequências) e em  $z=-1$  (correspondente a altas frequências), além de uma distribuição de polos a partir de valores próximos de  $z=1$  caminhando em direção a  $z=-1$  muito próximos do círculo de raio unitário. Essa distribuição de polos e zeros é um comportamento esperado de um filtro passa-faixas. Assim, através dessa análise já seria possível perceber o comportamento total do sistema visto que a frequência de amostragem utilizada de 44100Hz, delimita a frequência máxima dos componentes harmônicos do sinal de áudio a uma frequência de 22.05kHz, pelo critério de Nyquist. Logo como o banco passa frequências de 16Hz até 16kHz apenas frequências abaixo do subgrave e acima do espectro de filtro seriam atenuadas.

Por fim, com relação ao atraso de grupos apesar da Figura 52 apresentar atrasos consideráveis, sobretudo para o filtro de subgraves, não possível observar distorções do sinal decorrentes desse atraso nos sinais observados e durante a reprodução do sinal de áudio. Assim o comportamento do circuito foi considerado satisfatório nesse aspecto.



*Figura 52: Atraso de Grupo do Banco de Filtros.*

## BIBLIOGRAFIA

MATHWORKS. **Documentation:** **audioread**. Disponível em:  
<<https://www.mathworks.com/help/releases/R2020b/matlab/ref/audioread.html>>  
. Acesso em 23 de março de 2021.

MATHWORKS. **Documentation:** **sound**. Disponível em:  
<<https://www.mathworks.com/help/releases/R2020b/matlab/ref/sound.html>>.  
Acesso em 23 de março de 2021.

MATHWORKS. **Documentation:** **fft**. Disponível em:  
<<https://www.mathworks.com/help/releases/R2020b/matlab/ref/fft.html>>.  
Acesso em 23 de março de 2021.

MATHWORKS. **Documentation:** **fftshift**. Disponível em:  
<<https://www.mathworks.com/help/releases/R2020b/matlab/ref/fftshift.html>>.  
Acesso em 23 de março de 2021.

MATHWORKS. **Documentation:** **designfilt**. Disponível em:  
<<https://www.mathworks.com/help/releases/R2020b/signal/ref/designfilt.html>>.  
Acesso em 23 de março de 2021.

MATHWORKS. **Documentation:** **fvtool**. Disponível em:  
<[https://www.mathworks.com/help/releases/R2020b/dsp/ref/dsp.digitaldowncon  
verter.fvtool.html](https://www.mathworks.com/help/releases/R2020b/dsp/ref/dsp.digitaldownconverter.fvtool.html)>. Acesso em 24 de março de 2021.

MATHWORKS. **Documentation:** **filter**. Disponível em:  
<<https://www.mathworks.com/help/releases/R2020b/matlab/ref/filter.html>>.  
Acesso em 24 de março de 2021.

MATHWORKS. **Documentation:** **digitalFilter**. Disponível em:  
<<https://www.mathworks.com/help/releases/R2020b/signal/ref/digitalfilter.html>>.  
Acesso em 24 de março de 2021.

MATHWORKS. **Documentation:** **filtfilt**. Disponível em:  
<<https://www.mathworks.com/help/releases/R2020b/signal/ref/fftfilt.html>>.  
Acesso em 25 de março de 2021.