

---

# TRABALHO PRÁTICO - PROCESSAMENTO DE SINAIS

## Table of Contents

|                  |    |
|------------------|----|
| QUESTÃO 1: ..... | 1  |
| QUESTÃO 2: ..... | 2  |
| QUESTÃO 3: ..... | 4  |
| QUESTÃO 4: ..... | 17 |
| QUESTÃO 5: ..... | 17 |

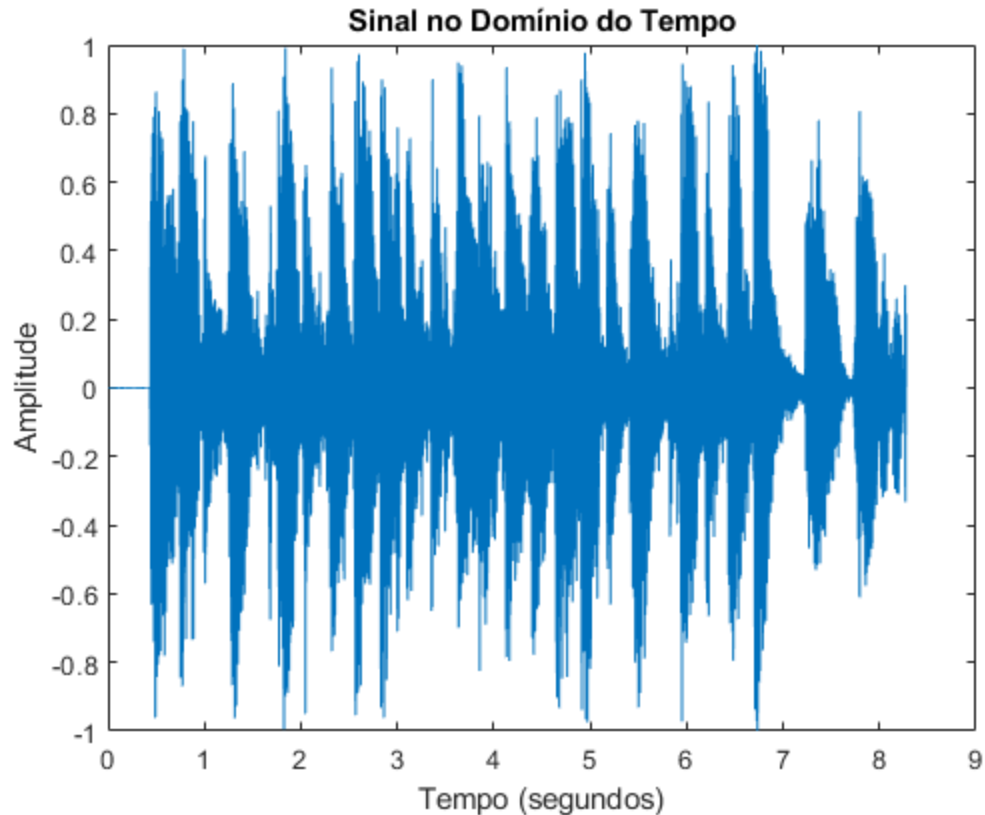
ELE 042 Vítor Gabriel Reis Caitité - 2016111849 \\ Willian Braga da Silva - 201602762

## QUESTÃO 1:

Abrir o arquivo Blind\_intro.wav e gerar um gráfico da forma de onda em função do tempo, similar ao mostrado na figura do tp.

```
%  
  
clear all;  
clc;  
audioArchive = 'Blind_intro.wav';  
% [Y, FS]=audioread(FILENAME) reads an audio file specified by the  
% string FILE, returning the sampled data in Y and the sample rate  
% FS, in Hertz.  
[Y, FS]=audioread(audioArchive);  
  
%Caso deseje ouvir o audio basta descomentar a linha abaixo  
%sound(Y, FS)  
  
%Geração da escala de tempo a partir de FS:  
period = 1/FS; % Período  
L = length(Y); % Largura de Faixa  
timeScale = linspace(0, L-1, L)*period; % Escala de Frequência  
%Plot:  
figure(1)  
plot(timeScale, Y); % Gráfico do sinal no  
tempo  
xlabel('Tempo (segundos)'); % Eixo x  
ylabel('Amplitude'); % Eixo y  
title('Sinal no Domínio do Tempo'); % Título  
  
% Descrição da Atividade: Nesse exercício o primeiro passo foi ler o  
% arquivo de áudio com a função audioread, a qual retorna os dados  
% amostrados e também a frequência de amostragem.  
  
%
```

---



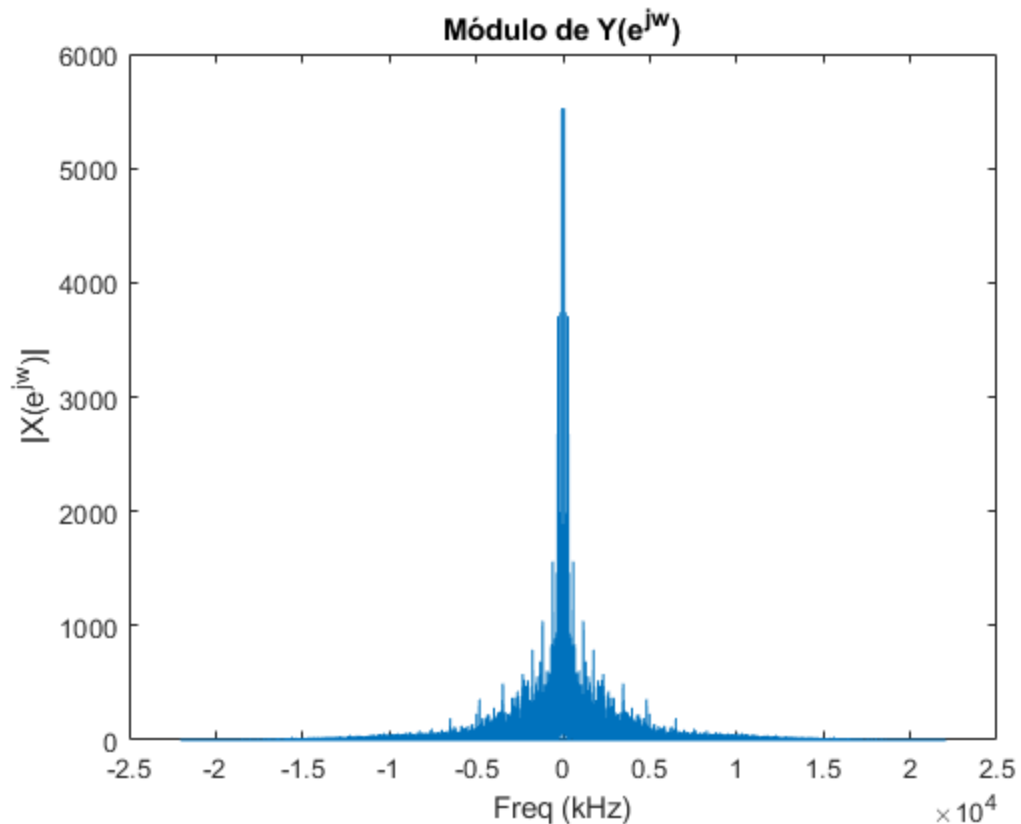
## QUESTÃO 2:

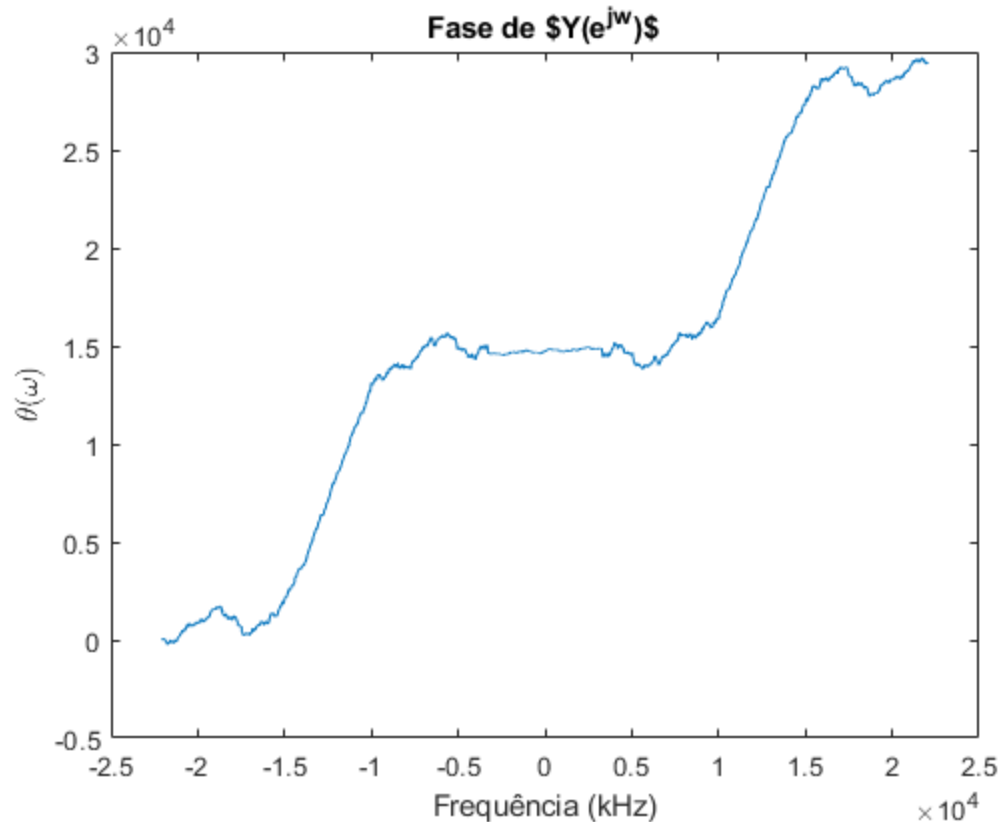
Calcular e mostrar a composição espectral do sinal utilizando a transformada rápida de Fourier (FFT). Gere os gráficos de amplitude e fase do espectro de frequências.

```
% Transformada Rápida de Fourier Y:
fftY = fft(Y);
% Módulo da fft:
moduloY = abs(fftY);
% Fase da fft:
faseY = unwrap(angle(fftY));
%Geração de escala de frequência positiva:
frequencyScale = (linspace(0, L/2, L/2+1))*FS/L;
%Geração de escala de frequência completa:
aux = length(frequencyScale)+1;
for i=length(frequencyScale):-1:1
    aux = aux-1;
    completeScale(i) = -frequencyScale(aux);
end
aux=0;
for i=(2*length(frequencyScale)-1):-1:length(frequencyScale)
    aux=aux+1;
    completeScale(i) = frequencyScale(aux);
end
f = linspace(-FS/2,FS/2, L);
%Plots:
```

```
completeScale=completeScale';  
figure(2)  
plot(completeScale, moduloY); % Plot módulo  
ylabel('|X(e^{j\omega})|'); % Eixo y  
xlabel('Freq (kHz)'); % Eixo x  
title('Módulo de Y(e^{j\omega})'); % Título  
figure(3)  
plot(f, faseY); % Plot fase  
ylabel('$\theta(\omega)$', 'interpreter', 'latex'); % Eixo y  
xlabel('Frequência (kHz)'); % Eixo x  
title('Fase de $Y(e^{j\omega})$'); % Título  
  
% Descrição da Atividade: Primeiramente gerou-se a transformada rápida  
% de  
% Fourier usando a função fft(), depois disso pôde-se obter o módulo e  
% fase  
% com as funções abs() e angle() respectivamente. Então bastou gerar  
% as  
% escalas de frequência e plotar.  
  
%
```

---





## QUESTÃO 3:

Faça o projeto do banco de filtros. . Subgraves: entre 16 Hz to 60 Hz; . Graves: entre 60 Hz to 250 Hz; . Médio-graves: entre 250 Hz to 2 kHz; . Médio-agudos: entre 2 kHz to 4 kHz; . Agudos: entre 4 kHz to 6 kHz; . Brilho: entre 6 kHz to 16 kHz;

```
% Vetor de Frequências Inferiores:
freqInferior = [16 60 250 2000 4000 6000];
% Vetor de Frequências Superiores:
freqSuperior = [60 250 2000 4000 6000 16000];
% Vetor de títulos dos filtros:
filterType = {'Filtro Subgrave'; 'Filtro Grave'; 'Filtro Medio Grave'
              'Filtro Medio Agudo'; 'Filtro Agudo'; 'Filtro Brilho'};

for count = 1:6
    % Definindo parâmetros para a função cheblord:
    % [N, Wp] = cheblord(Wp, Ws, Rp, Rs) returns the order N of the
    lowest
    % order digital Chebyshev Type I filter which has a passband
    ripple of
    % no more than Rp dB and a stopband attenuation of at least Rs dB.
    Wp
    % and Ws are the passband and stopband edge frequencies,
    normalized
    % from 0 to 1 (where 1 corresponds to pi radians/sample). For
    example,
```

```
% . Lowpass:      Wp = .1,      Ws = .2
% . Highpass:     Wp = .2,      Ws = .1
% . Bandpass:     Wp = [.2 .7], Ws = [.1 .8]
% . Bandstop:     Wp = [.1 .8], Ws = [.2 .7]

fpI = freqInferior(count)*2/FS;
fpS = freqSuperior(count)*2/FS;
RsInferior = fpI - 0.08;
RsSuperior = fpS + 0.08;
if RsInferior < 0
    RsInferior = 0.0000001;
end
if RsSuperior > 1
    RsSuperior = 0.999999;
end

% Gerando Ordem do Filtro Chebyshev tipo I:
[order ~] = cheblord([fpI fpS], [RsInferior RsSuperior], 0.5,
40);

% Designs a bandpass filter: cheby1(N,R,Wp,'bandpass')
% Cheby1 returns the filter coefficients in length N+1 vectors B
%(numerator) and A (denominator).
[B A] = cheby1(order, 0.5, [fpI fpS], 'bandpass');

[NUMd,DENd] = bilinear(B,A,FS)
% Gerando sinal passado pelo filtro:
y{count} = filter(B,A, Y);
%h = fvtool(B,A);

figure(count+3)
% Gráfico de Módulo e Fase do Filtro:
freqz(B, A);
title(filterType{count});

figure(10)
subplot(2, 3, count)
plot(timeScale, y{count}) % Sinal após o filtro
xlabel('Tempo(s)')
ylabel('Amplitude')
title(filterType{count})

figure(11)
% Diagrama de polos e zeros do filtro:
subplot(3, 2, count)
%Função de Transferência:
H(count) = tf(B, A, period)
pzmap(H(count))
title(filterType{count});

end

% Descrição da Atividade: Primeiramente gerou-se os filtros de
Chebyshev
```

TRABALHO PRÁTICO - PRO-  
CESSAMENTO DE SINAIS

---

```
% tipo I, utilizando as funções cheblord(), cheby1(), cada uma já
% explicada. Logo bastou passar o sinal pelos filtros especificados,
% utilizando a função filter(). Assim podê-se gerar todos os gráficos,
a
% função de transferência (usando a função tf()), e o diagrama de polos
e
% zeros (usando a função pzmap()).
```

```
%
```

---

*H* =

$$\frac{1.4e-05 z^4 - 2.8e-05 z^2 + 1.4e-05}{z^4 - 3.991 z^3 + 5.973 z^2 - 3.973 z + 0.9911}$$

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

*H* =

*From input 1 to output:*

$$\frac{1.4e-05 z^4 - 2.8e-05 z^2 + 1.4e-05}{z^4 - 3.991 z^3 + 5.973 z^2 - 3.973 z + 0.9911}$$

*From input 2 to output:*

$$1.745e-06 z^6 - 5.235e-06 z^4 + 5.235e-06 z^2 - 1.745e-06$$

---

$$z^6 - 5.965 z^5 + 14.83 z^4 - 19.65 z^3 + 14.66 z^2 - 5.831 z + 0.9667$$

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

*H* =

*From input 1 to output:*

$$\frac{1.4e-05 z^4 - 2.8e-05 z^2 + 1.4e-05}{z^4 - 3.991 z^3 + 5.973 z^2 - 3.973 z + 0.9911}$$

*From input 2 to output:*

$$1.745e-06 z^6 - 5.235e-06 z^4 + 5.235e-06 z^2 - 1.745e-06$$

---

$$z^6 - 5.965 z^5 + 14.83 z^4 - 19.65 z^3 + 14.66 z^2 - 5.831 z + 0.9667$$

*From input 3 to output:*

$$4.686e-06 z^{10} - 2.343e-05 z^8 + 4.686e-05 z^6 - 4.686e-05 z^4$$

$$+ 2.343e-05 z^2 - 4.686e-06$$

-----

$$z^{10} - 9.584 z^9 + 41.46 z^8 - 106.6 z^7 + 180.3 z^6 - 209.8 z^5$$

$$+ 170.1 z^4 - 94.79 z^3 + 34.78 z^2 - 7.585 z + 0.7466$$

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

$H =$

From input 1 to output:

$$1.4e-05 z^4 - 2.8e-05 z^2 + 1.4e-05$$

$$-----$$

$$z^4 - 3.991 z^3 + 5.973 z^2 - 3.973 z + 0.9911$$

From input 2 to output:

$$1.745e-06 z^6 - 5.235e-06 z^4 + 5.235e-06 z^2 - 1.745e-06$$

$$-----$$

$$z^6 - 5.965 z^5 + 14.83 z^4 - 19.65 z^3 + 14.66 z^2 - 5.831 z + 0.9667$$

From input 3 to output:

$$4.686e-06 z^{10} - 2.343e-05 z^8 + 4.686e-05 z^6 - 4.686e-05 z^4$$

$$+ 2.343e-05 z^2 - 4.686e-06$$

-----

$$z^{10} - 9.584 z^9 + 41.46 z^8 - 106.6 z^7 + 180.3 z^6 - 209.8 z^5 \\ + 170.1 z^4 - 94.79 z^3 + 34.78 z^2 - 7.585 z + 0.7466$$

From input 4 to output:

$$8.965e-06 z^{10} - 4.482e-05 z^8 + 8.965e-05 z^6 - 8.965e-05 z^4 \\ + 4.482e-05 z^2 - 8.965e-06$$

---


$$z^{10} - 8.798 z^9 + 35.62 z^8 - 87.28 z^7 + 143.3 z^6 - 164.6 z^5 \\ + 134 z^4 - 76.35 z^3 + 29.14 z^2 - 6.734 z + 0.7161$$

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

$H =$

From input 1 to output:

$$1.4e-05 z^4 - 2.8e-05 z^2 + 1.4e-05 \\ z^4 - 3.991 z^3 + 5.973 z^2 - 3.973 z + 0.9911$$

From input 2 to output:

$$1.745e-06 z^6 - 5.235e-06 z^4 + 5.235e-06 z^2 - 1.745e-06 \\ z^6 - 5.965 z^5 + 14.83 z^4 - 19.65 z^3 + 14.66 z^2 - 5.831 z + 0.9667$$

From input 3 to output:



$$4.686e-06 z^{10} - 2.343e-05 z^8 + 4.686e-05 z^6 - 4.686e-05 z^4$$

$$+ 2.343e-05 z^2 - 4.686e-06$$

---


$$z^{10} - 9.584 z^9 + 41.46 z^8 - 106.6 z^7 + 180.3 z^6 - 209.8 z^5$$

$$+ 170.1 z^4 - 94.79 z^3 + 34.78 z^2 - 7.585 z + 0.7466$$

*From input 4 to output:*

$$8.965e-06 z^{10} - 4.482e-05 z^8 + 8.965e-05 z^6 - 8.965e-05 z^4$$

$$+ 4.482e-05 z^2 - 8.965e-06$$

---


$$z^{10} - 8.798 z^9 + 35.62 z^8 - 87.28 z^7 + 143.3 z^6 - 164.6 z^5$$

$$+ 134 z^4 - 76.35 z^3 + 29.14 z^2 - 6.734 z + 0.7161$$

*From input 5 to output:*

$$8.965e-06 z^{10} - 4.482e-05 z^8 + 8.965e-05 z^6 - 8.965e-05 z^4$$

$$8.965e-06 \quad + 4.482e-05 z^2 -$$

-----

$$z^{10} - 7.317 z^9 + 26.04 z^8 - 58.5 z^7 + 91.32 z^6 - 103.2 z^5$$

$$+ 85.41 z^4 - 51.18 z^3 + 21.31 z^2 - 5.6 z + 0.7161$$

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

$H =$

From input 1 to output:

$$1.4e-05 z^4 - 2.8e-05 z^2 + 1.4e-05$$

-----

$$z^4 - 3.991 z^3 + 5.973 z^2 - 3.973 z + 0.9911$$

From input 2 to output:

$$1.745e-06 z^6 - 5.235e-06 z^4 + 5.235e-06 z^2 - 1.745e-06$$

-----

$$z^6 - 5.965 z^5 + 14.83 z^4 - 19.65 z^3 + 14.66 z^2 - 5.831 z + 0.9667$$

From input 3 to output:

$$4.686e-06 z^{10} - 2.343e-05 z^8 + 4.686e-05 z^6 - 4.686e-05 z^4$$

$$+ 2.343e-05 z^2 - 4.686e-06$$

-----

$$z^{10} - 9.584 z^9 + 41.46 z^8 - 106.6 z^7 + 180.3 z^6 - 209.8 z^5$$

$$0.7466 \quad + 170.1 z^4 - 94.79 z^3 + 34.78 z^2 - 7.585 z +$$

From input 4 to output:

$$8.965e-06 z^{10} - 4.482e-05 z^8 + 8.965e-05 z^6 - 8.965e-05 z^4$$

$$8.965e-06 \quad + 4.482e-05 z^2 -$$

-----

$$z^{10} - 8.798 z^9 + 35.62 z^8 - 87.28 z^7 + 143.3 z^6 - 164.6 z^5$$

$$0.7161 \quad + 134 z^4 - 76.35 z^3 + 29.14 z^2 - 6.734 z +$$

From input 5 to output:

$$8.965e-06 z^{10} - 4.482e-05 z^8 + 8.965e-05 z^6 - 8.965e-05 z^4$$

$$8.965e-06 \quad + 4.482e-05 z^2 -$$

-----

$$z^{10} - 7.317 z^9 + 26.04 z^8 - 58.5 z^7 + 91.32 z^6 - 103.2 z^5$$

$$0.7161 \quad + 85.41 z^4 - 51.18 z^3 + 21.31 z^2 - 5.6 z +$$

From input 6 to output:

$$0.006697 z^{12} - 0.04018 z^{10} + 0.1005 z^8 - 0.1339 z^6 + 0.1005 z^4$$

$$- 0.04018 z^2 +$$

$$0.006697$$

-----

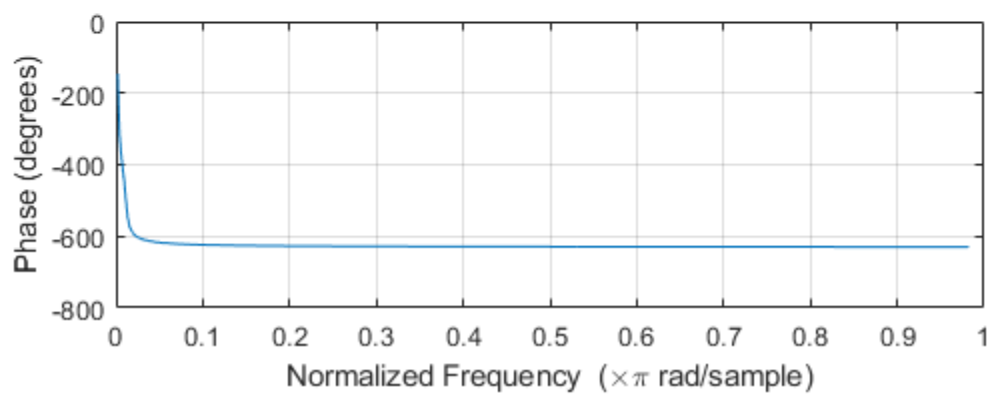
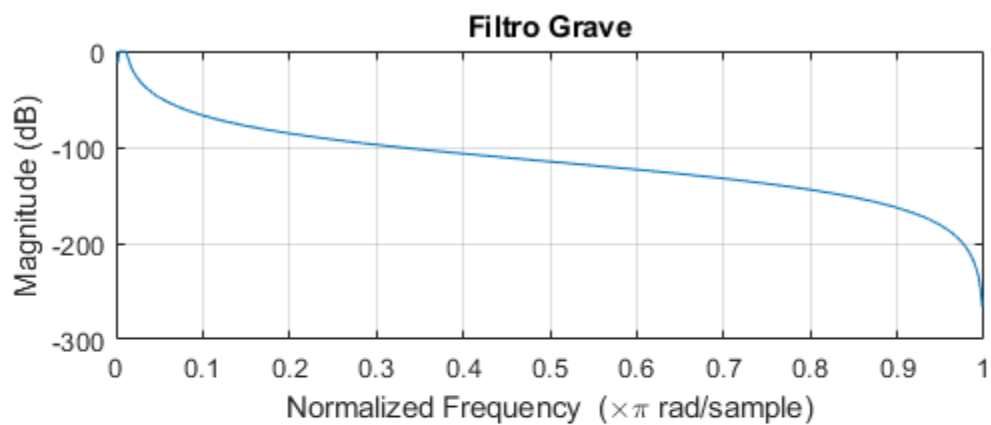
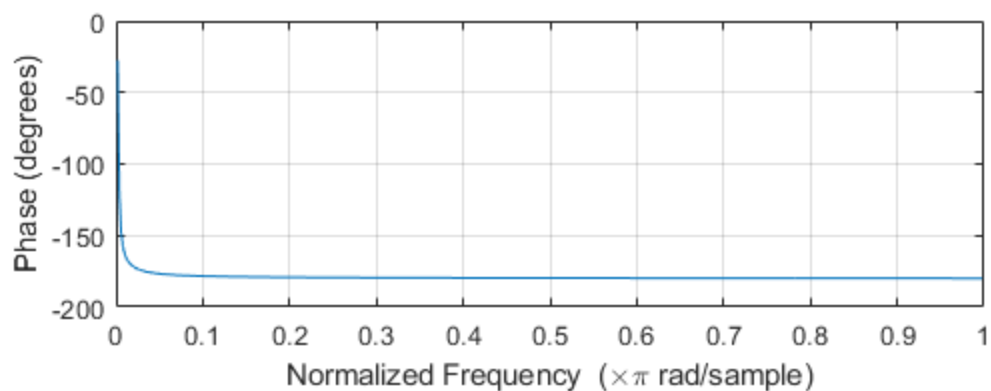
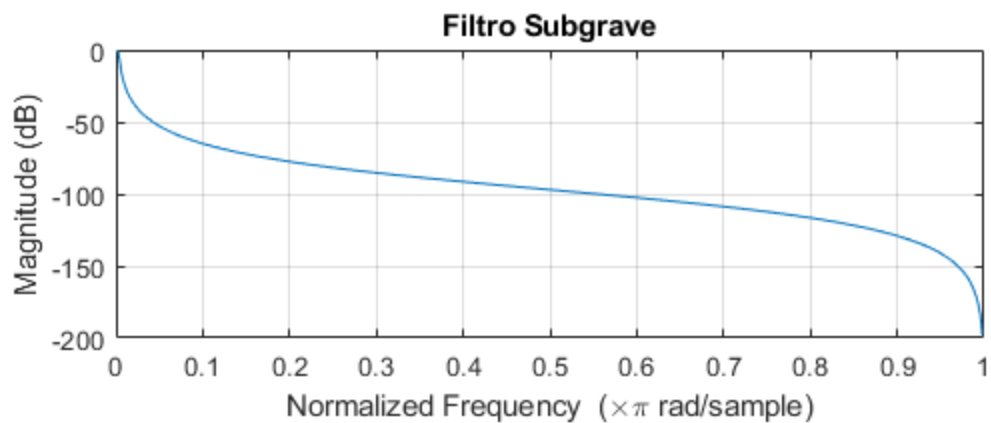
$$z^{12} - 0.03747 z^{11} + 1.962 z^{10} - 0.07381 z^9 + 2.939 z^8 - 0.09439 z^7$$

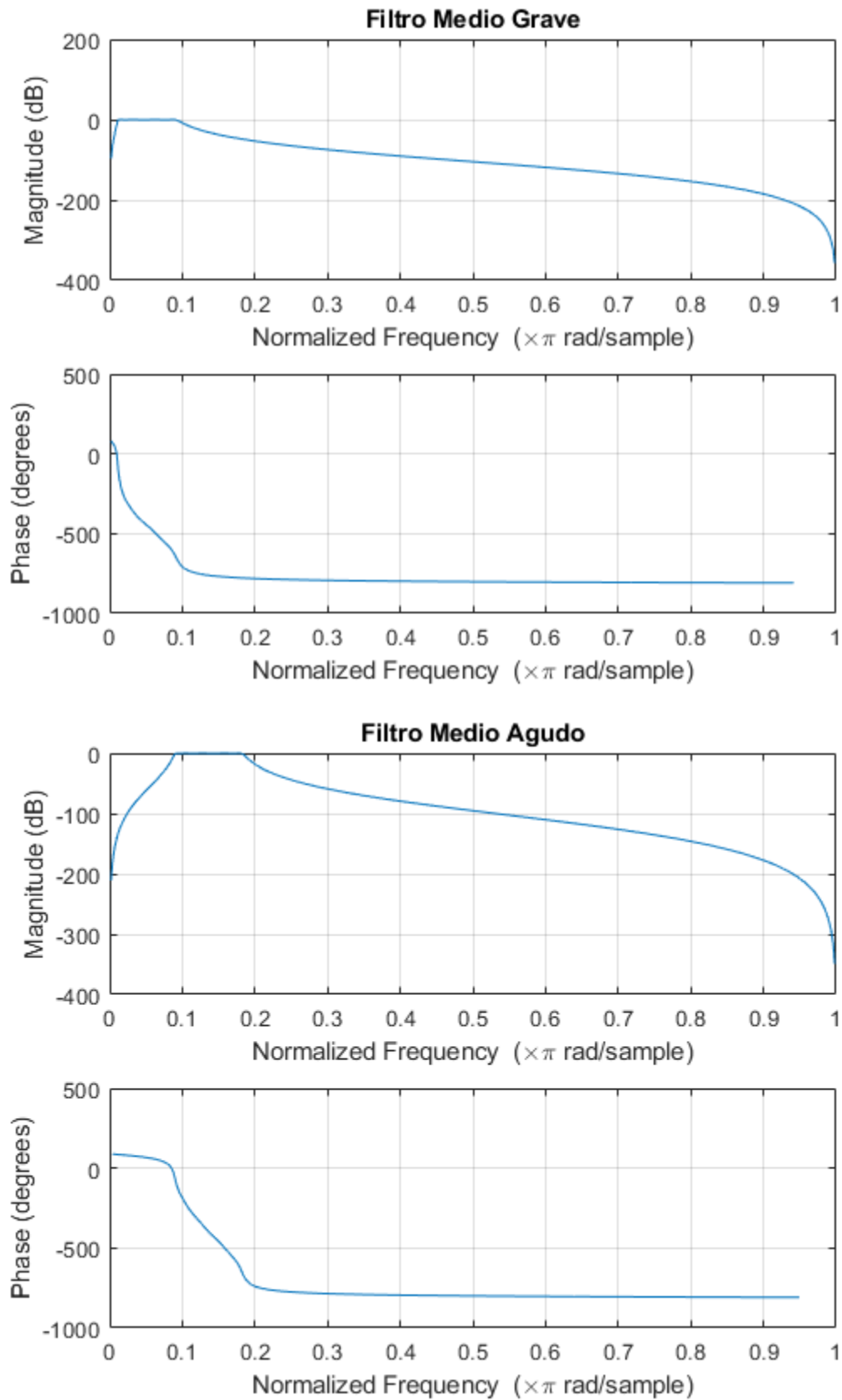
$$+ 2.768 z^6 - 0.0742 z^5 + 1.867 z^4 - 0.03678 z^3 + 0.8166 z^2$$

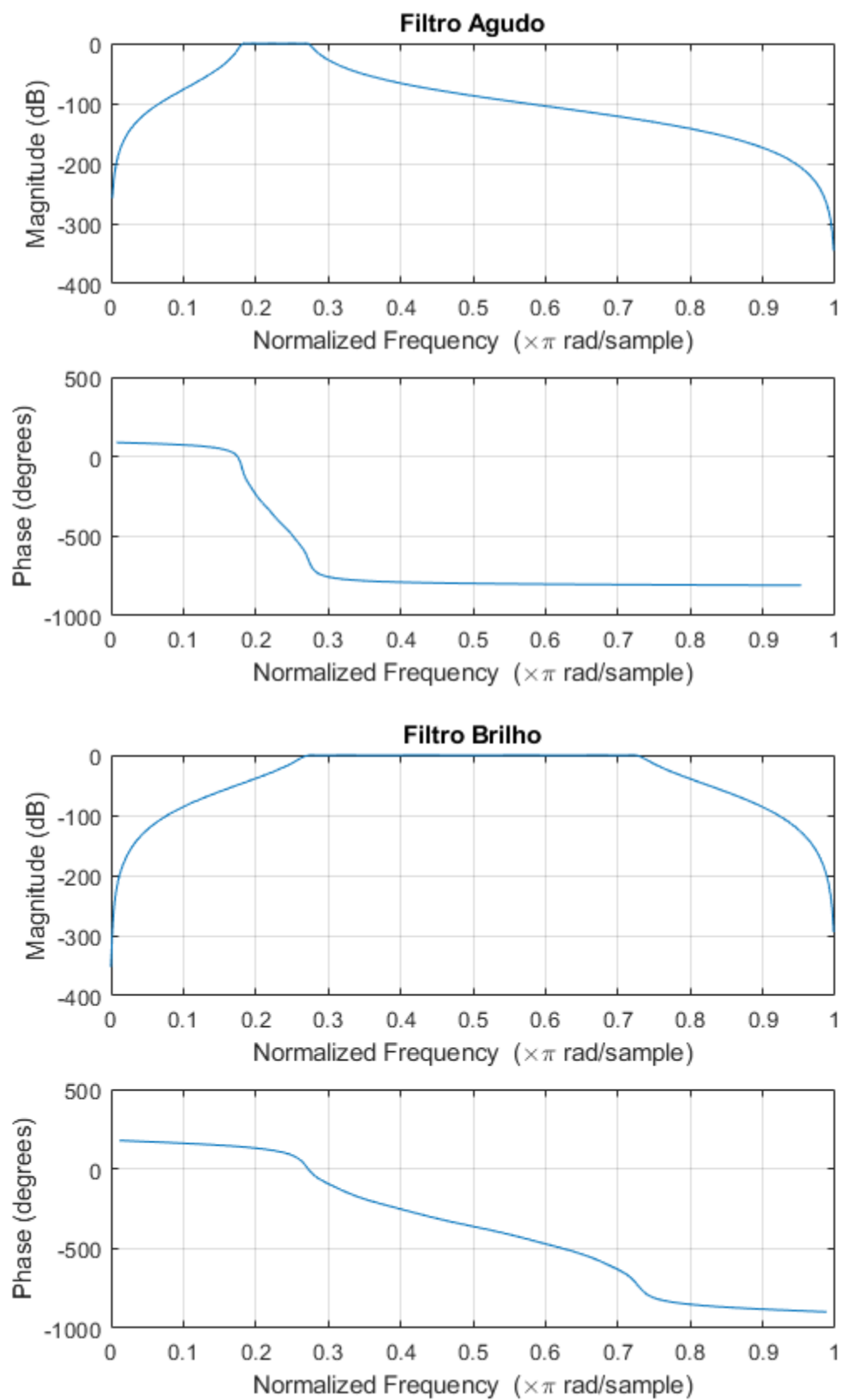
$$- 0.009334 z +$$

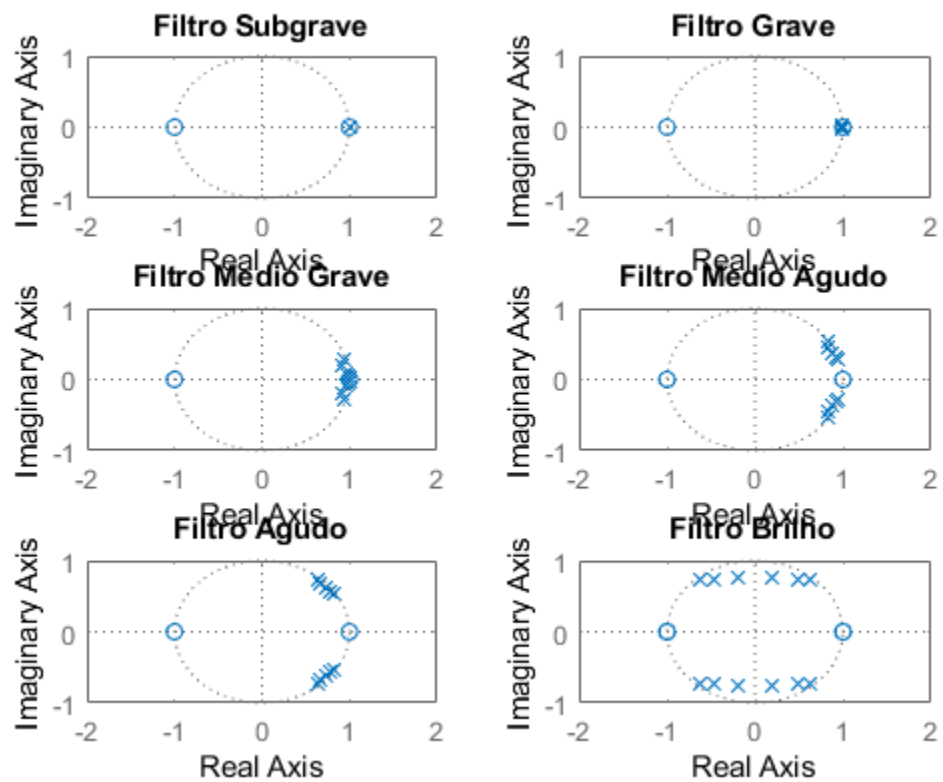
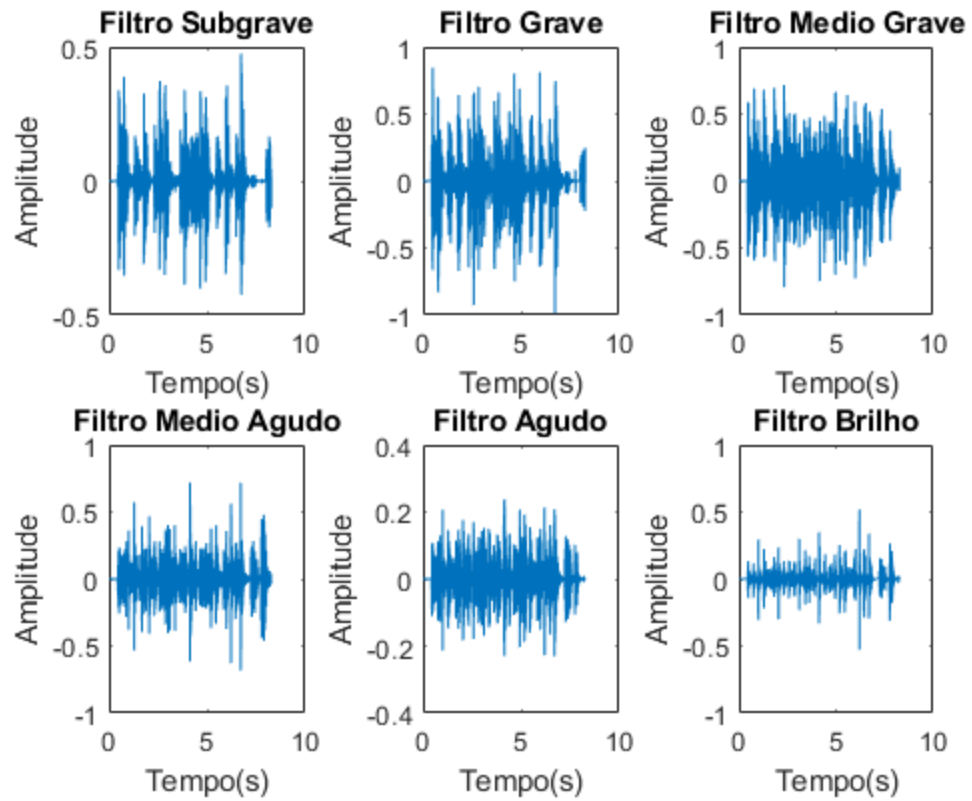
$$0.1945$$

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







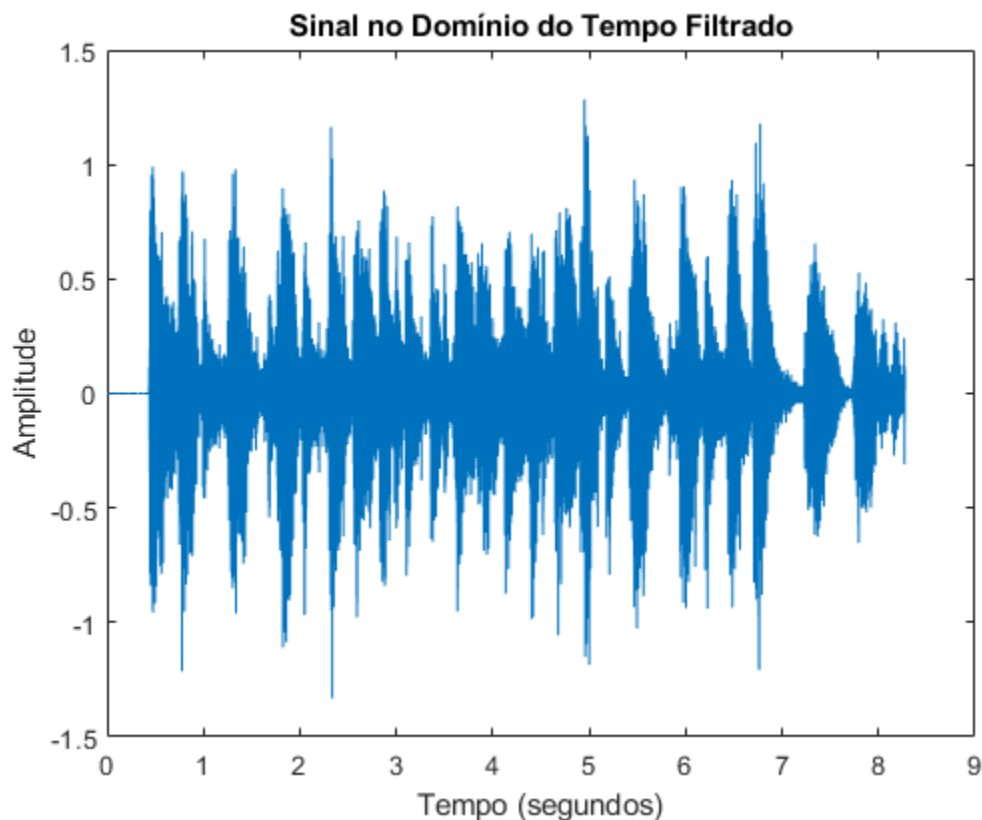




## QUESTÃO 4:

Após o ajuste dos ganhos, o sistema deverá realizar a filtragem e reconstrução do sinal por meio da soma das saídas dos filtros individuais.

```
finalY = Y{1}+Y{2}+Y{3}+Y{4}+Y{5}+Y{6};  
figure(12)  
plot(timeScale, finalY); % Gráfico do sinal no  
    tempo  
xlabel('Tempo (segundos)'); % Eixo x  
ylabel('Amplitude'); % Eixo y  
title('Sinal no Domínio do Tempo Filtrado'); % Título  
  
% Descrição da Atividade: Bastou somar todos os sinais, cada um  
    passado por  
% um tipo de filtro e plotar.
```



## QUESTÃO 5:

```
%Reproduza o sinal filtrado no sistema de áudio do computador.  
sound(finalY, FS);  
  
% Descrição da Atividade: Simplesmente utilizar a função sound para  
% reproduzir o audio.
```

*Published with MATLAB® R2017b*