UNIVERSIDADE ESTADUAL DE CAMPINAS

Faculdade de Engenharia Mecânica

ES879 - Sistemas de Aquisição de Dados

Trabalho Final

Bruno Monteiro Bonetti - 232488

Bruno Cardoso Holanda - 167542

Campinas - SP

2023

1. Introdução

Neste trabalho, propomos a aplicação de filtros digitais em uma parte do áudio da música "Viva la Vida", realizando uma análise em frequência através da Transformada de Fourier Discreta (DFT). Utilizaremos técnicas de decimação e interpolação para realizar um ajuste digital preciso na taxa de amostragem do sinal, e o uso de filtro passa-baixa. Além disso, vamos implementar um tratamento de eco no áudio.

2. Itens do trabalho

2.1. Item a

Escolhemos a música Viva la vida, da banda Coldplay, para ser analisada neste trabalho. Então, plotamos seu sinal no domínio do tempo, observando que o sinal de áudio disponibilizado possuía apenas 30 segundos. Para obter o espectro de frequências, realizamos a Transformada Discreta de Fourier (DFT) do sinal, usando o método da Transformada Rápida de Fourier (FFT).

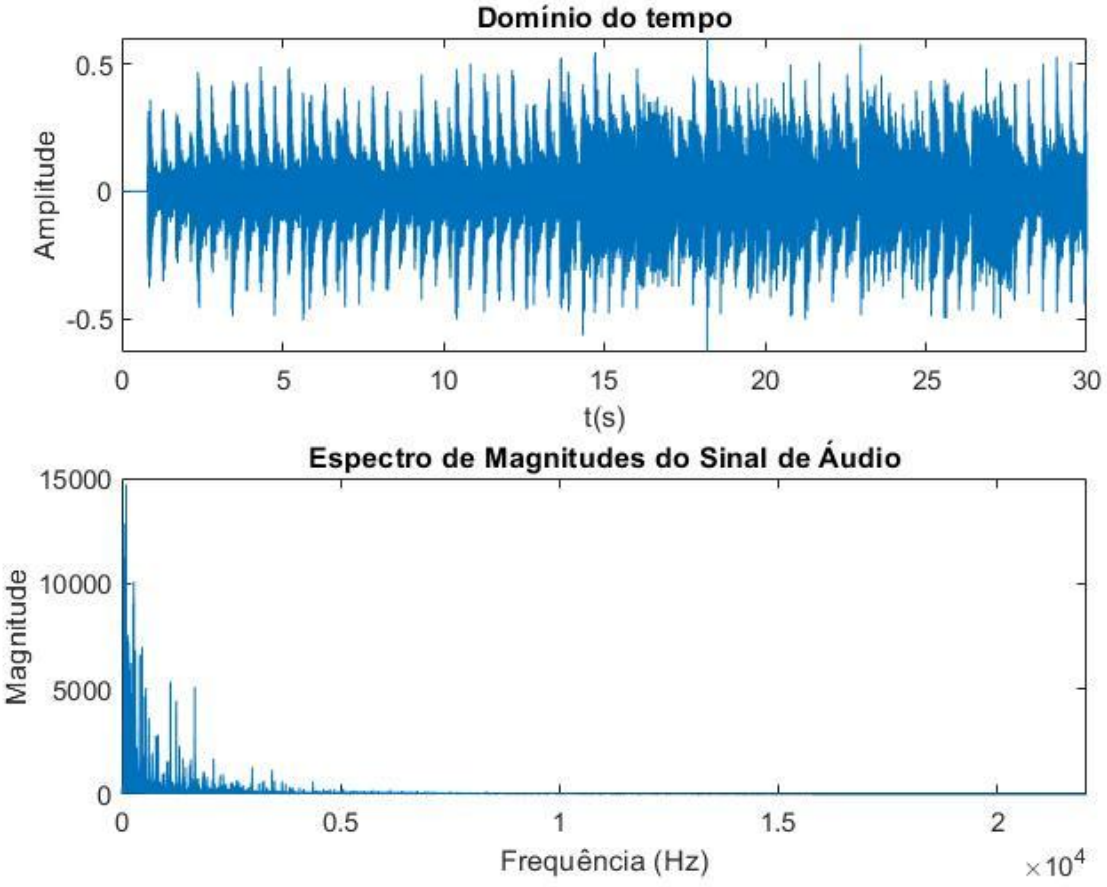

Figura 1: Representação do sinal de áudio disponibilizado, tanto no domínio do tempo, quanto no domínio da frequência (0 a Fs/2).

A taxa de amostragem (Fs) representa o número de amostras de um sinal que são registradas por segundo. No caso, a taxa de amostragem é de 44100 Hz, o que é uma taxa de amostragem padrão para áudio de qualidade de CD.

A metade da taxa de amostragem é conhecida como frequência de Nyquist e representa a frequência máxima que pode ser representada de forma única em um sinal amostrado. Neste caso, a frequência de Nyquist é 22050 Hz, o que significa que qualquer componente de frequência no sinal original acima dessa frequência pode causar aliasing.

A FFT é simplesmente uma forma mais rápida de calcular a DFT de um sinal. Ela utiliza alguns algoritmos que permitem reduzir o número de operações de N^2 para $N.log_2(N)$, sendo N o comprimento do sinal. Esta funcionalidade é implementada na função fft do MATLAB.

É importante salientar que, se a média do sinal usado na fft não for zero, um componente na frequência 0, denominado de ganho estático, irá aparecer. Então, de modo a eliminar esse possível ganho estático, subtraímos de nosso sinal a sua média.

Também, uma prática importante ao realizarmos a transformada de Fourier, é multiplicar seu resultado por 2. Essa multiplicação é essencial devido à natureza bilateral da transformada, que é simétrica em torno de 0. Quando realizamos a fft, as componentes negativas da transformada encontram-se espelhadas em torno de $Fs/2$. A energia em cada frequência é igualmente distribuída nas metades antes e após $Fs/2$ do espectro. Portanto, ao considerar apenas a primeira parte do espectro, multiplicamos por 2 para representar com precisão a amplitude total do sinal.

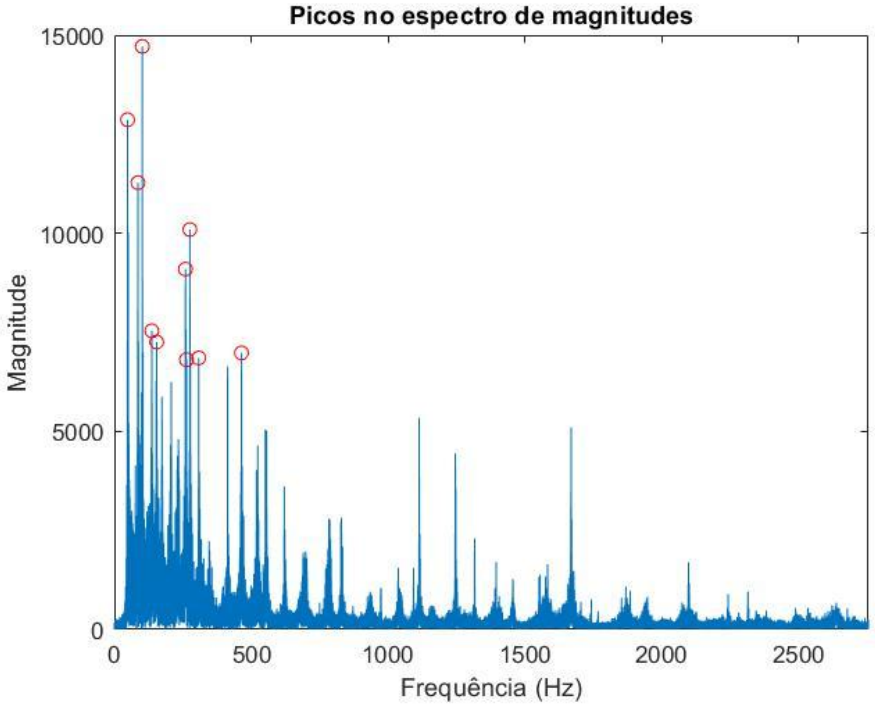

Figura 2: Espectro de frequências de 0 a Fs/16, com os 10 maiores picos de magnitude circulados em vermelho.

Frequência (Hz)	Magnitude
48.3000	12865.1368
86.4001	11276.5683
102.2334	14715.4545
136.9001	7538.1845
154.6335	7255.6021
259.8669	9095.2528
263.6669	6810.2268
276.0002	10094.7638
308.3669	6851.4922
464.8004	6979.3511

Tabela 1: Relação entre os 10 picos mais significativos de magnitude e sua respectiva frequência.

Percebe-se, por meio da figura 1, que as frequências mais significativas são as mais baixas. Então, através da figura 2, observamos quais as frequências que representam o maior conteúdo espectral do sinal analisado. Estas frequências estão localizadas no gráfico por meio de círculos vermelhos, e a relação entre frequência e magnitude pode ser vista na tabela 1. Pela tabela 1, percebemos que a frequência mais significativa é a de 102,2334 Hz. Utilizou-se a função *findpeaks* do *matlab,* para achar esses picos.

2.2. Item b

O segundo item do trabalho pede que seja feita a decimação, por um fator 8, do sinal, sem realizar nenhum tipo de pré-filtragem. Usamos a função *decimate* para realizar essa operação. Ficamos, portanto, com uma nova frequência de amostragem de 5512,5 Hz e uma nova frequência de Nyquist de 2756,25 Hz.

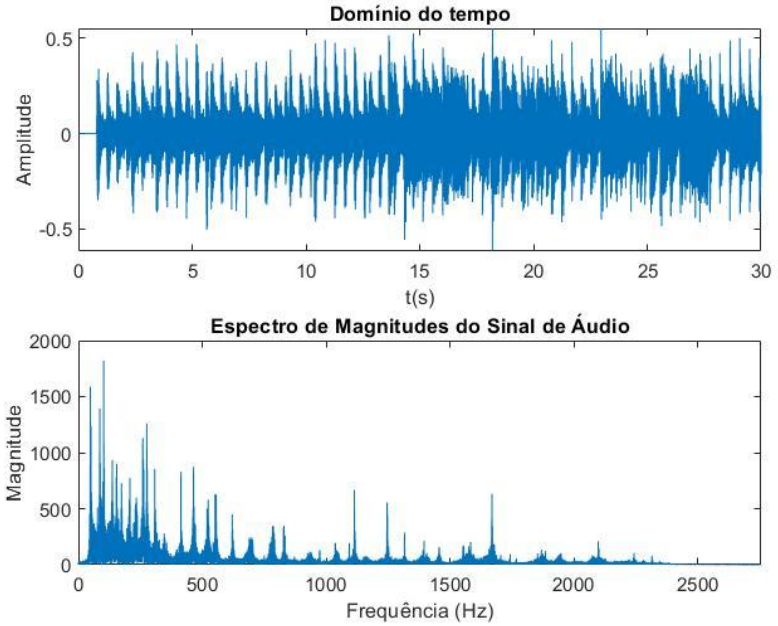

Figura 3: Representação do sinal de áudio decimado por um fator 8, tanto no domínio do tempo, quanto no domínio da frequência.

Ao fazer a decimação do sinal percebemos, através da figura 3, que a frequência de amostragem e a frequência de Nyquist também foram reduzidas pelo mesmo fator. Agora, será traçado o sinal original juntamente com o sinal decimado para observar as diferenças.

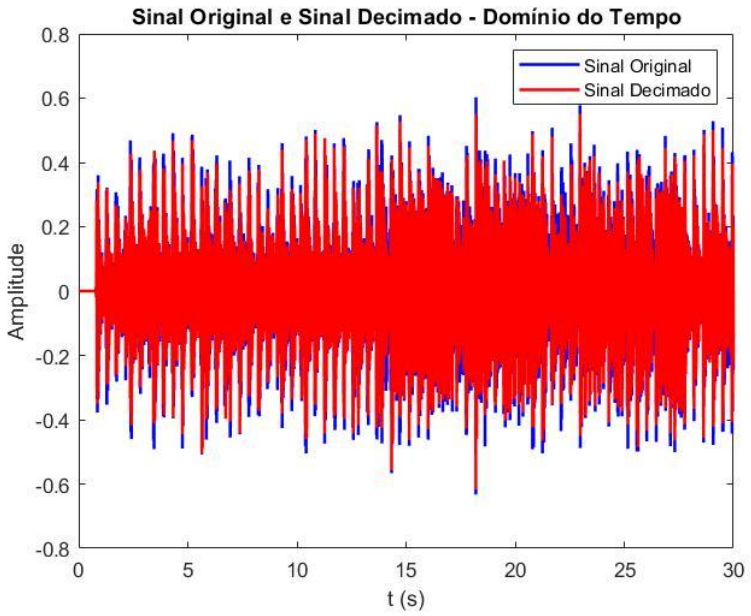

Figura 4: Sinal de áudio original e decimado sobrepostos no domínio do tempo.

É possível perceber, pela figura 4, que o sinal decimado segue o mesmo padrão do sinal original no domínio do tempo, o que era esperado.

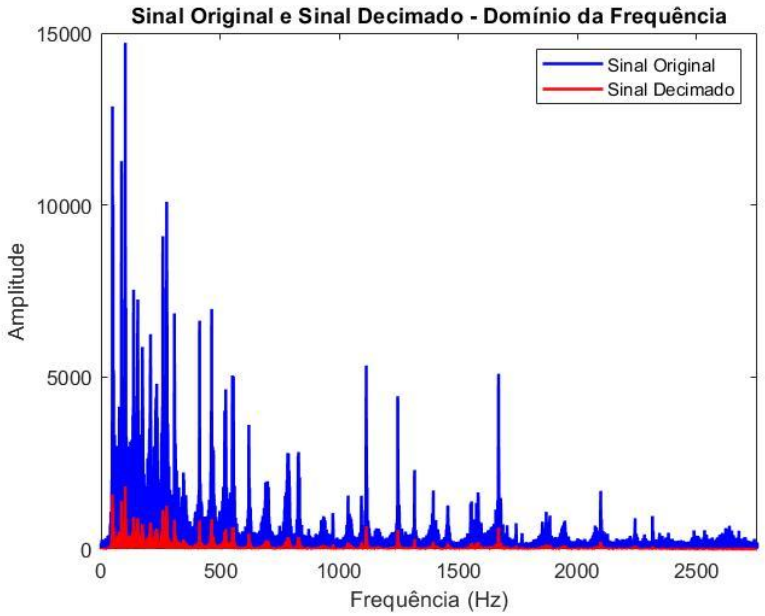

Figura 5: Espectro de magnitudes do sinal de áudio original e do decimado sobrepostos.

Já pela figura 5, é possível notar a redução da frequência de amostragem, mas também a drástica redução na magnitude do sinal. Esta redução na magnitude gera uma redução na energia total do sinal, e para compensar isso basta multiplicarmos a magnitude do sinal pelo fator de decimação. O resultado obtido pode ser visualizado na figura 6.

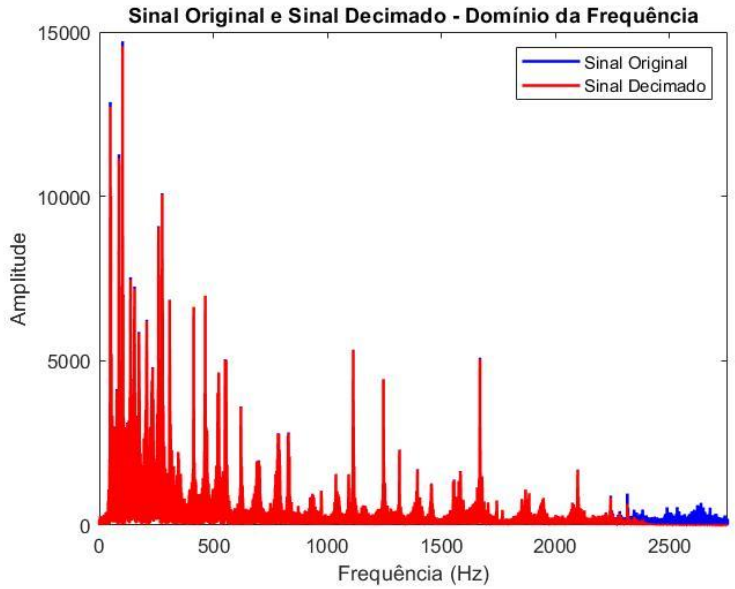

Figura 6: Espectro de magnitudes do sinal de áudio original e do decimado, com a correção, sobrepostos e visualizados de 0 a Fs/16 (frequência de Nyquist do sinal decimado).

2.3. Item c

Ao avaliarmos os dois áudios (original e após decimação), identificamos as seguintes diferenças perceptíveis: sensação de áudio abafado, redução na altura da voz do cantor e redução de agudos (resultante do "corte" das frequências altas).

Essas diferenças podem ser atribuídas ao fato de que a decimação pode influenciar nas características do sinal, especialmente se houver componentes de alta frequência no sinal original que estão próximos à nova frequência de Nyquist após a decimação. A perda de detalhes de alta frequência pode resultar em uma reprodução menos nítida de partes do sinal, como letras ou componentes musicais de alta frequência.

2.4. Item d

Pediu-se que fosse filtrado o sinal de áudio com um filtro passa-baixa. O MATLAB disponibiliza a função *fir1*, que cria um filtro do tipo FIR (resposta ao impulso finita), e que utiliza dois parâmetros: frequência de corte e ordem. Para entender a importância da ordem de um filtro, plotamos a resposta em frequência de filtros com a mesma frequência de corte e diferentes ordens.

Mantendo a frequência de corte constante igual a 500 Hz e variando a ordem do filtro, obtivemos os gráficos das figuras 7, 8 e 9:

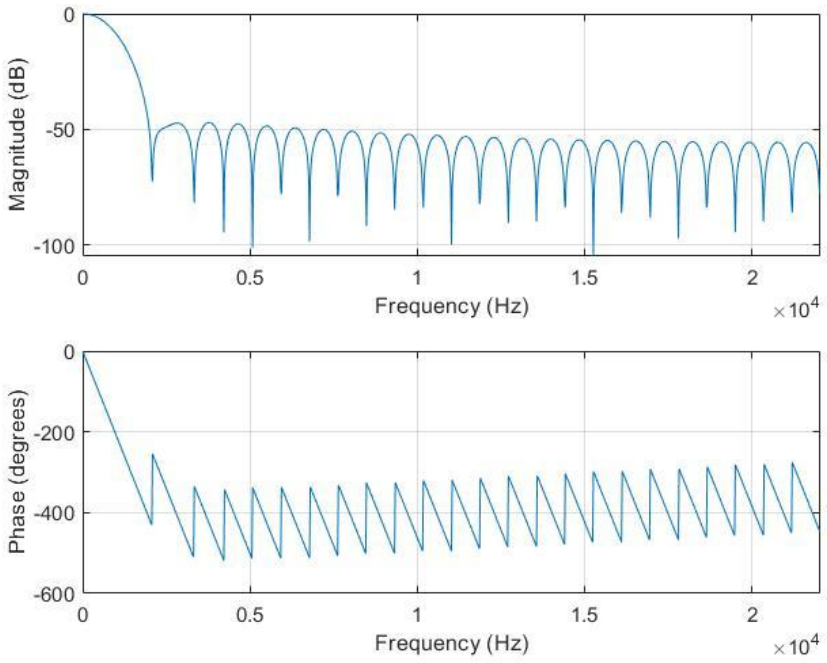

Figura 7: Resposta em frequência de filtro FIR com frequência de corte de 500 Hz e ordem do filtro igual a 50.

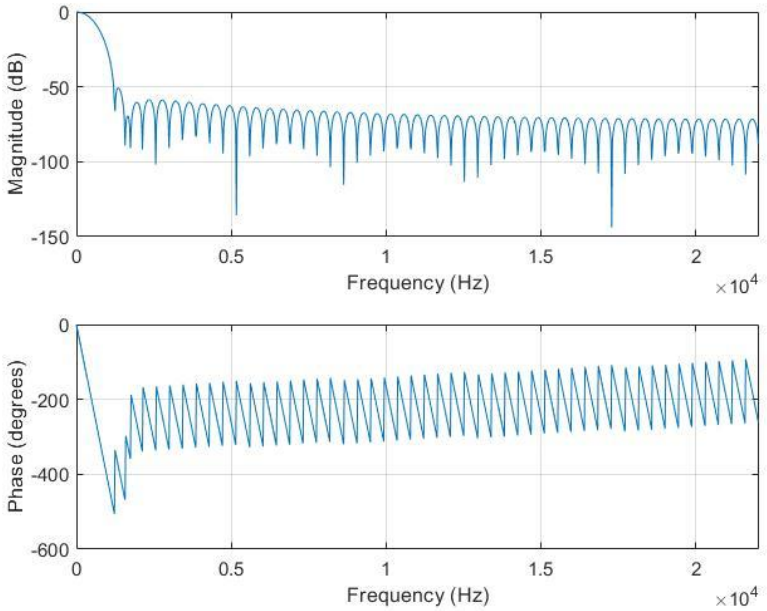

Figura 8: Resposta em frequência de filtro FIR com frequência de corte de 500 Hz e ordem do filtro igual a 100.

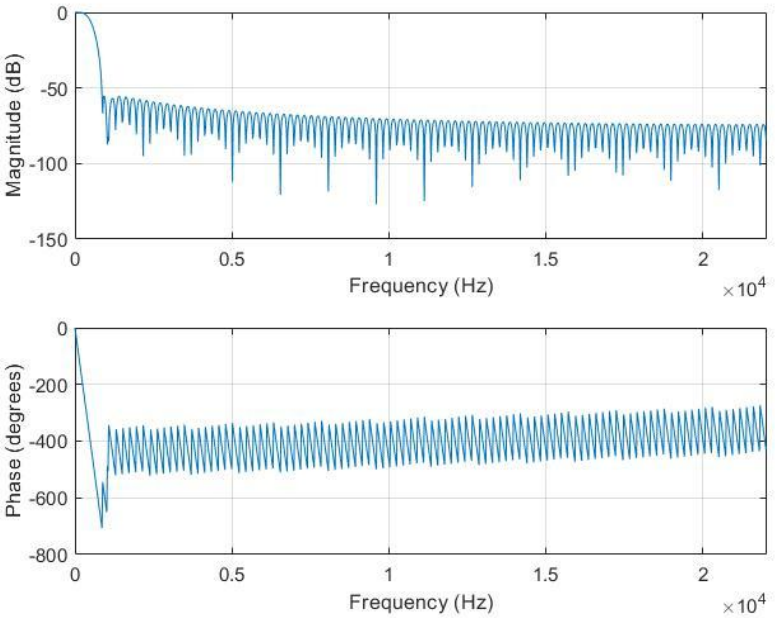

Figura 9: Resposta em frequência de filtro FIR com frequência de corte de 500 Hz e ordem do filtro igual a 200.

Observa-se, comparando os gráficos desses três valores distintos de ordem, que esse parâmetro define o quão abrupta é a atenuação de frequências a partir da frequência de corte desejada. Logo, quanto maior a ordem, menor é a faixa de transição do filtro. O contraponto de se usar ordens elevadas é um maior custo computacional.

Então, definimos a ordem igual a 200, e variamos a frequência de corte de modo a notar diferenças significativas ao ouvir o sinal filtrado. Optamos por frequências de corte de 50 Hz, 500 Hz, e 5000 Hz para fazer as análises.

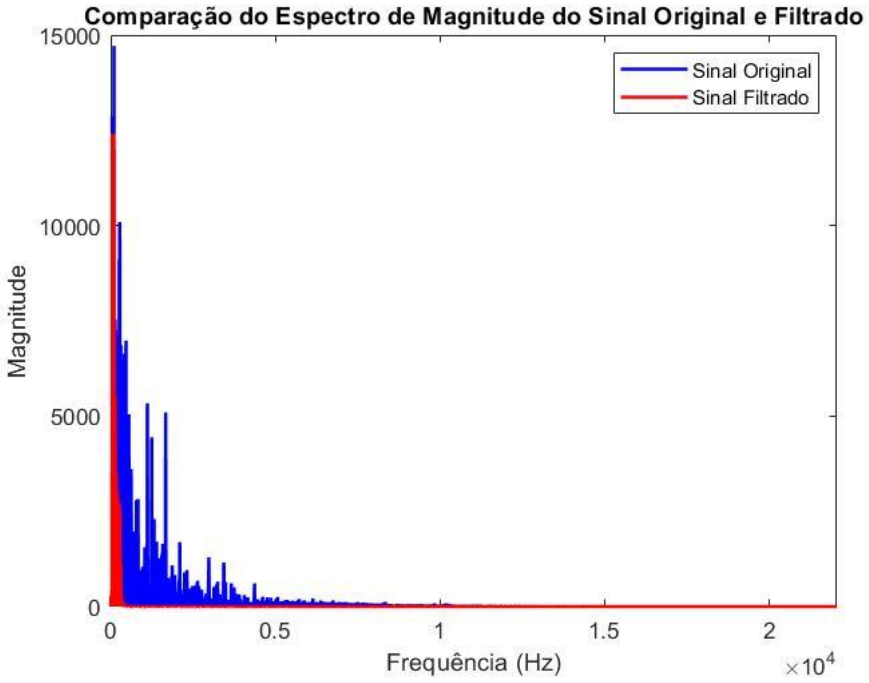

Figura 10: Comparação do espectro de frequências do sinal original e filtrado com ordem 200 e frequência de corte de 50 Hz.

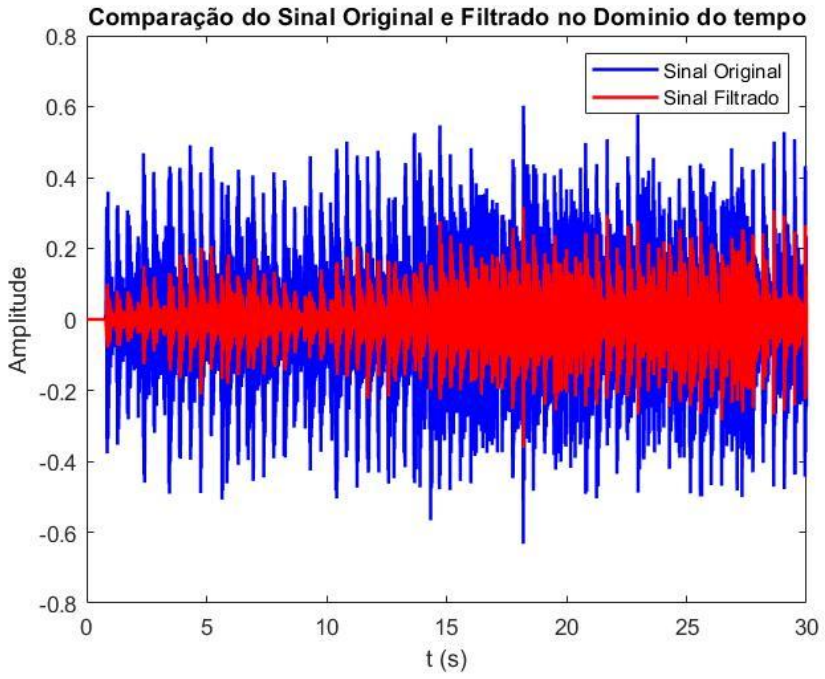

Figura 11: Comparação do sinal original e filtrado, no domínio do tempo, com ordem 200 e frequência de corte de 50 Hz.

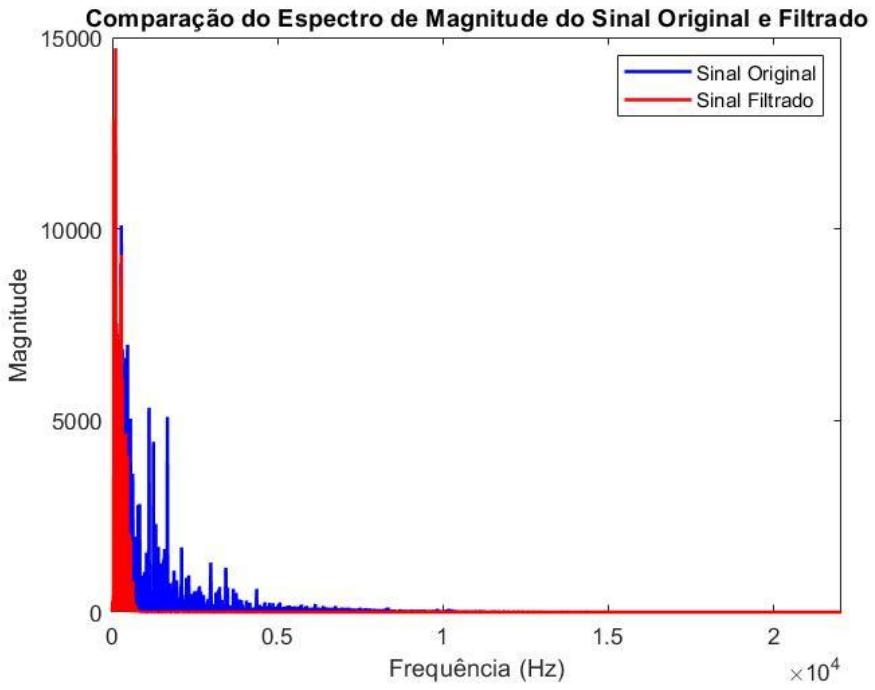

Figura 12: Comparação do espectro de frequências do sinal original e filtrado com ordem 200 e frequência de corte de 500 Hz.

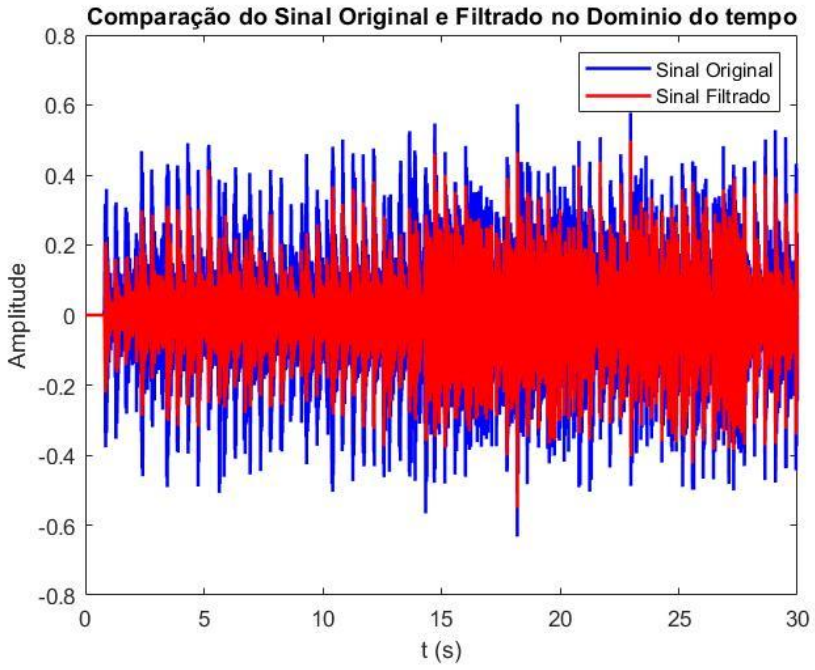

Figura 13: Comparação do sinal original e filtrado, no domínio do tempo, com ordem 200 e frequência de corte de 500 Hz.

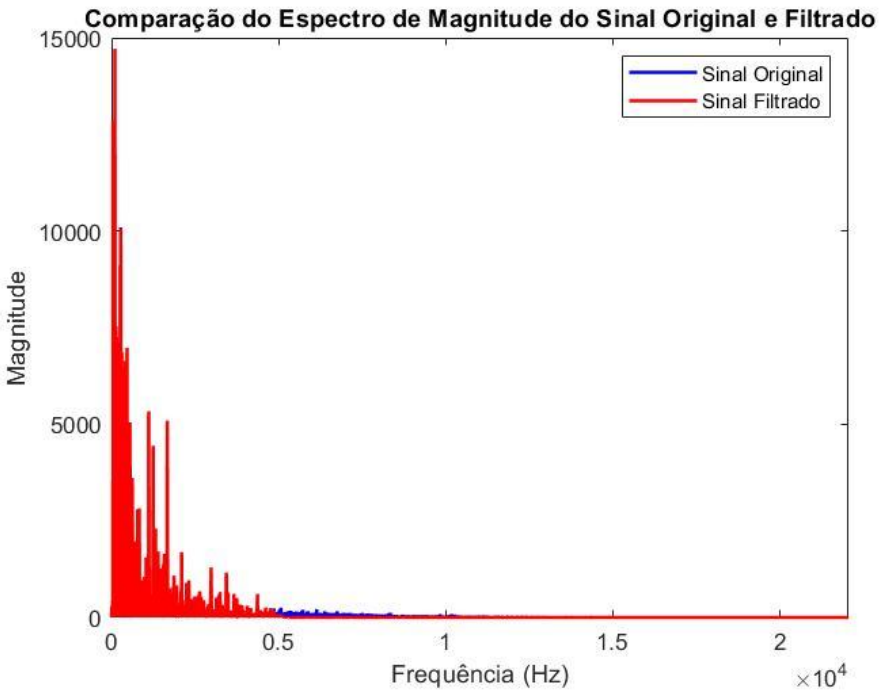

Figura 14: Comparação do espectro de frequências do sinal original e filtrado com ordem 200 e frequência de corte de 5000 Hz.

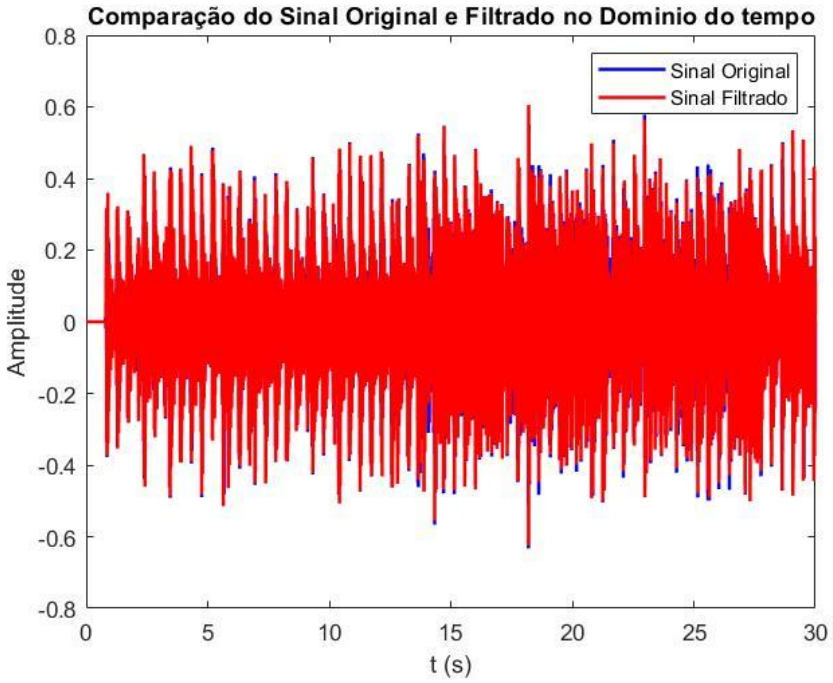

Figura 15: Comparação do sinal original e filtrado, no domínio do tempo, com ordem 200 e frequência de corte de 5000 Hz.

Analisando os gráficos de domínio do tempo obtidos pela filtragem do sinal, percebemos que para as frequências de corte de 50 Hz e 500 Hz, há uma grande diferença na amplitude do sinal original e do sinal filtrado. Essa diferença não é percebida na frequência de corte de 5000 Hz.

Esse comportamento no domínio do tempo pode ser explicado pelo espectro em frequência do sinal, após passar pelo filtro. Percebemos que muitas frequências significativas são cortadas usando as duas menores frequências de corte. Olhando para o espectro do sinal original, nota-se que a grande maioria das frequências mais importantes encontram-se antes dos 5000 Hz. Portanto, ao utilizarmos 5000 Hz como corte, era de se esperar que não houvesse tanto diferença no domínio do tempo ao retirar essas frequências que já não impactam tanto no sinal.

Ouvindo o sinal filtrado em 50 Hz, notamos que a altura da voz do cantor está bem baixa e fica marcada a presença do grave da música. Isso era esperado, visto que os graves são obtidos em frequências baixas.

Já para o sinal filtrado em 500 Hz, notamos a voz do cantor abafada, e, novamente, o grave da música é bem marcante.

Em 5000 Hz, ouve-se a música quase que perfeita, salvo uma sensação de abafamento da voz do cantor.

2.5. Item e

É pedido que façamos a decimação do sinal, por um fator M = 8, após a filtragem do sinal. Escolhemos para este item uma ordem de 200, e frequência de corte de 500 Hz. Os resultados são mostrados nas figuras abaixo.

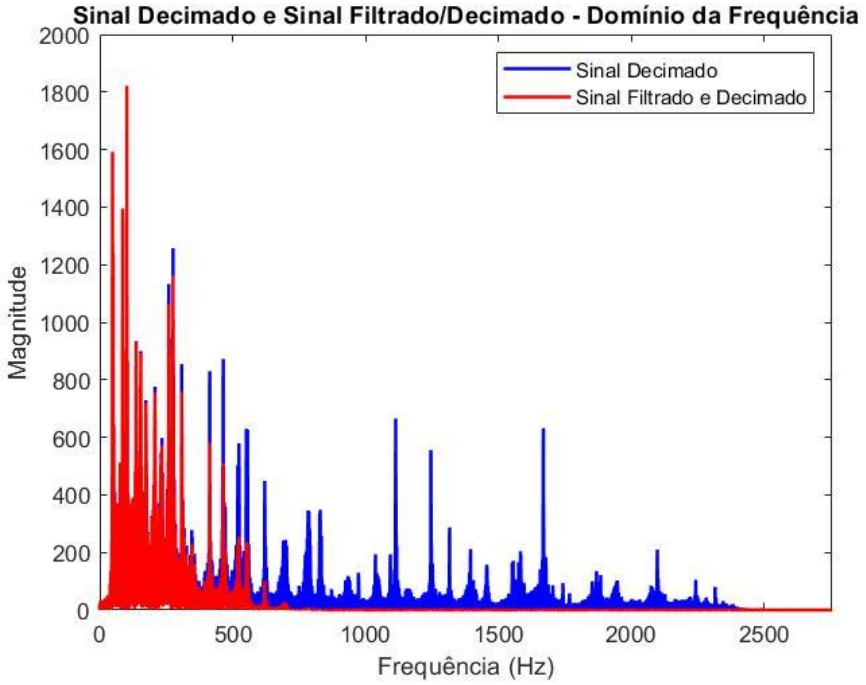

Figura 16: Comparação do espectro de frequências do sinal decimado com fator 8 e do filtrado, com ordem 200 e frequência de corte de 500 Hz, e decimado, com mesmo fator.

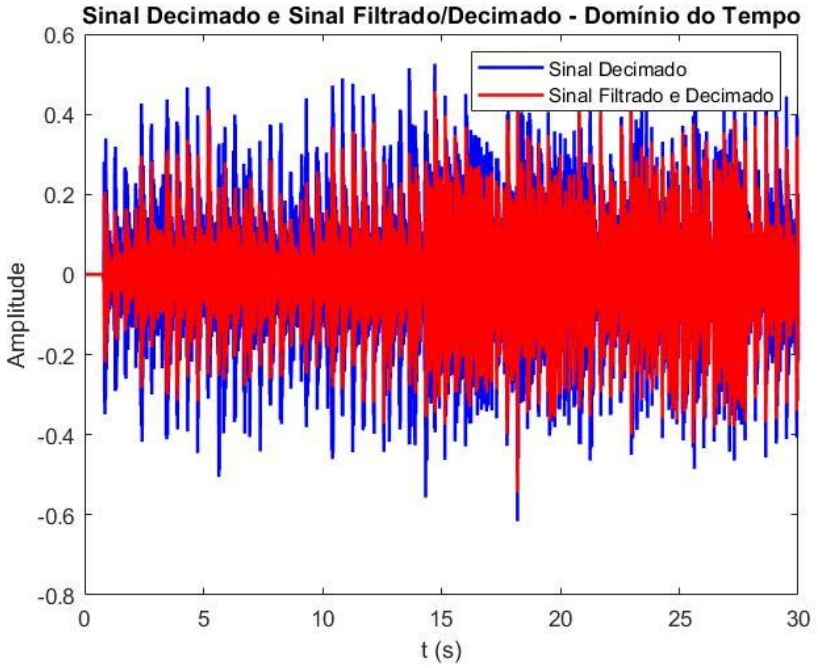

Figura 17: Comparação no domínio do tempo do sinal decimado com fator 8 e do filtrado, com ordem 200 e frequência de corte de 500 Hz, e decimado, com mesmo fator.

Ao ouvir ambos os áudios, percebemos que novamente o filtro realiza com sucesso aquilo que se propõe. Tanto áudio decimado, como áudio filtrado e decimado são abafados, porém o filtrado e decimado, ficou muito parecido com o sinal original filtrado, marcado pelos graves da música e diminuição da voz do cantor, o que era esperado.

2.6. Item f

Neste item, pede-se que seja feita interpolação do sinal de áudio, por um fator L = 4. Isso pode ser feito através da função *interp* do MATLAB. Percebe-se pela figura 18, que ao realizar a interpolação do sinal, sua magnitude em frequência é multiplicada pelo mesmo fator de interpolação. Uma correção, ou seja dividir a magnitude por tal fator, é necessário, caso queira manter a energia original do sinal, e pode ser observada na figura 19. No domínio do tempo, não são encontradas diferenças entre sinal original e sinal interpolado, como pode ser visto pela figura 20. Já, ao ouvir ambos os áudios, notou-se ligeiro aumento no volume do sinal interpolado, quando comparado ao original, porém em relação à qualidade, não notou-se diferenças.

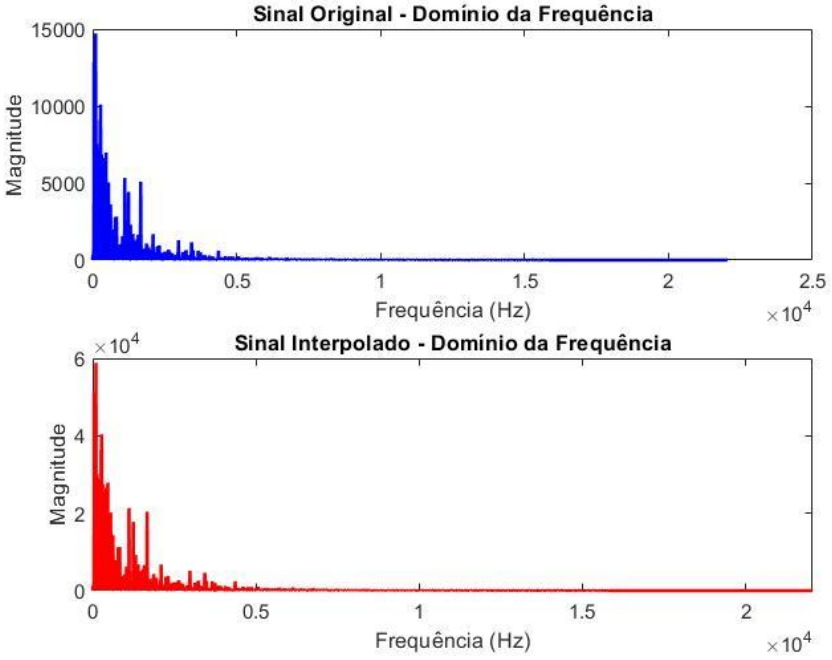

Figura 18: Comparação do espectro de frequências do sinal original e do superamostrado, por fator 4.

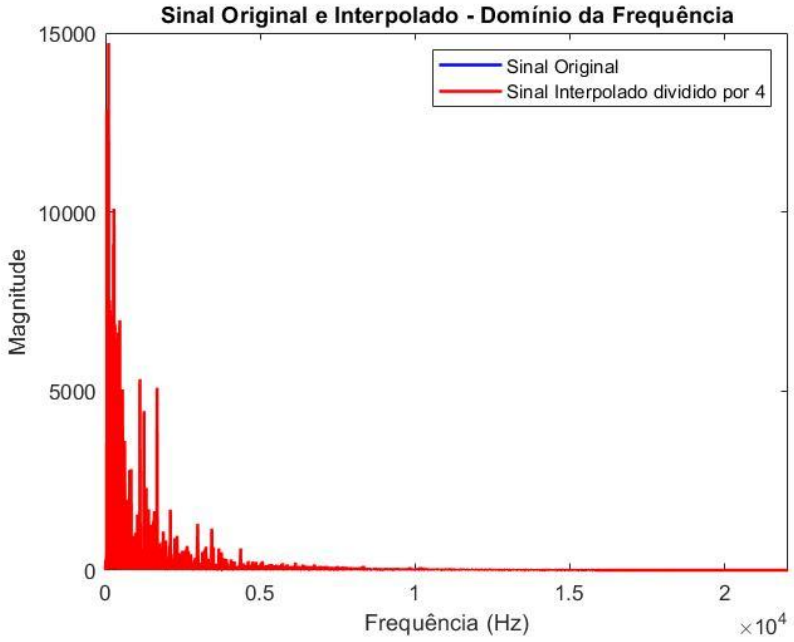

Figura 19: Espectro de frequências do sinal original e do superamostrado, com sua magnitude dividida pelo fator de interpolação, 4.

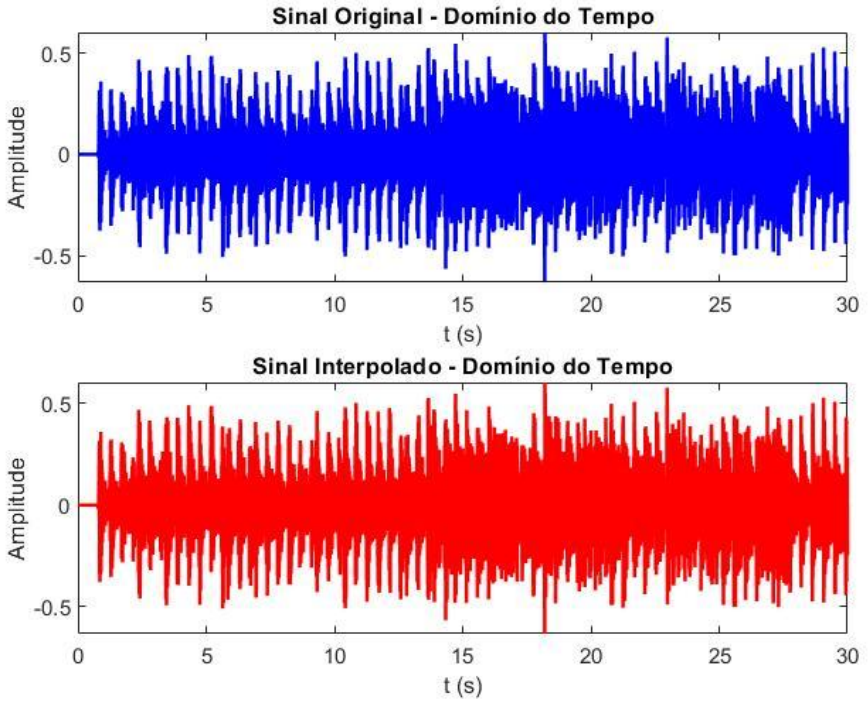

Figura 20: Gráficos do sinal original e do sinal interpolado, no domínio do tempo.

2.7. Item g

Um filtro de eco possui sua equação de diferenças dada no seguinte formato:

$$y[n] = x[n] + a\,x[n - D]$$

Aplicando a transformada Z nesta equação, ficamos com:

$$Y(z) = X(z) + a.z^{-D}.X(z)$$

Sabemos que a função de transferência é dada por $H(z) = Y(z)/X(z)$. Então, temos:

$$Y(z) = X(z).(1 + a.z^{-D})$$

$$\frac{Y(z)}{X(z)} = H(z) = 1 + a.z^{-D}$$

Fazendo a transformada Z inversa da função de transferência $H(z)$, obtemos a resposta ao impulso $h[n]$:

$$h[n] = Z^{-1}(H(z))$$

$$h[n] = \delta[n] + a.\delta[n - D]$$

Percebe-se que a resposta ao impulso será um um impulso unitário em $n = 0$ e um impulso atenuado (para $|a| < 1$), em $n = D$. Essa expressão condiz com o comportamento esperado de um eco.

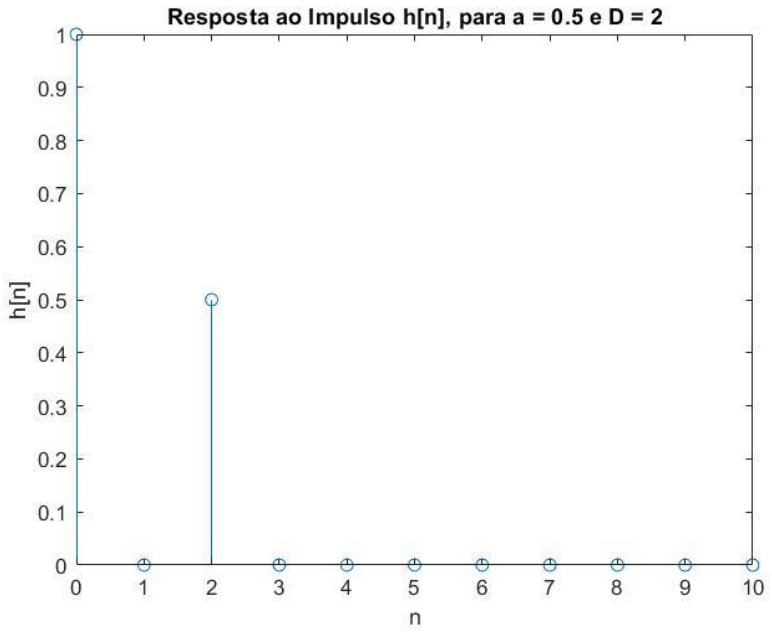

Figura 21: Resposta ao impulso do filtro de eco, no domínio discreto.

2.8. Item h

Para determinarmos a quantidade de amostras atrasadas D, em função do tempo de eco desejado t, em segundos, e da frequência de amostragem Fs , em Hz, temos:

$$D = t.Fs$$

Para $Fs = 44100\,Hz$ e tempo de atraso $t = 0,5s$, temos o seguinte valor:

$$D = 22050\,amostras$$

Usando esses parâmetros, obtivemos a resposta em frequência do filtro apresentada na figura 22.

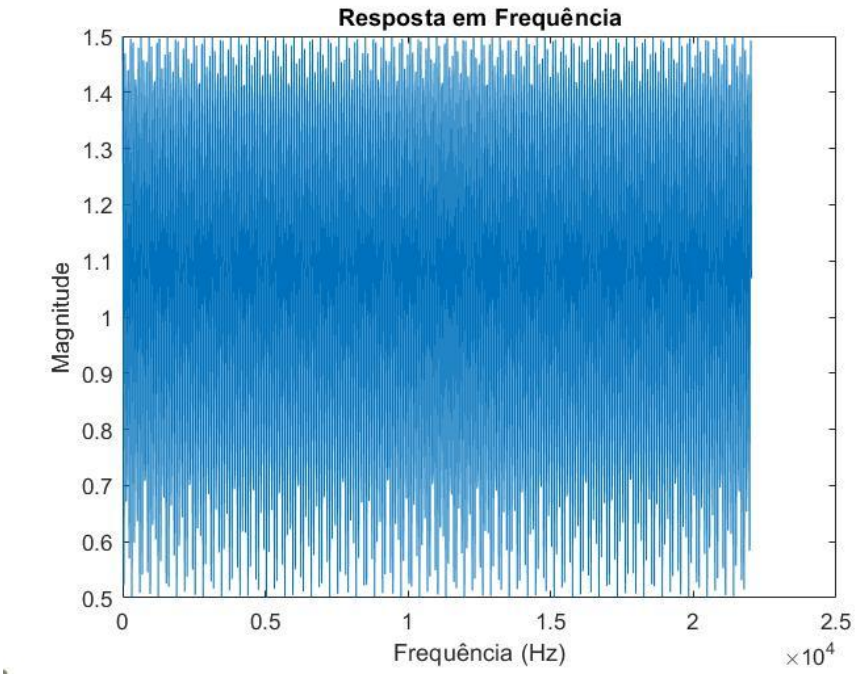

Figura 22: Resposta em frequência do filtro de eco.

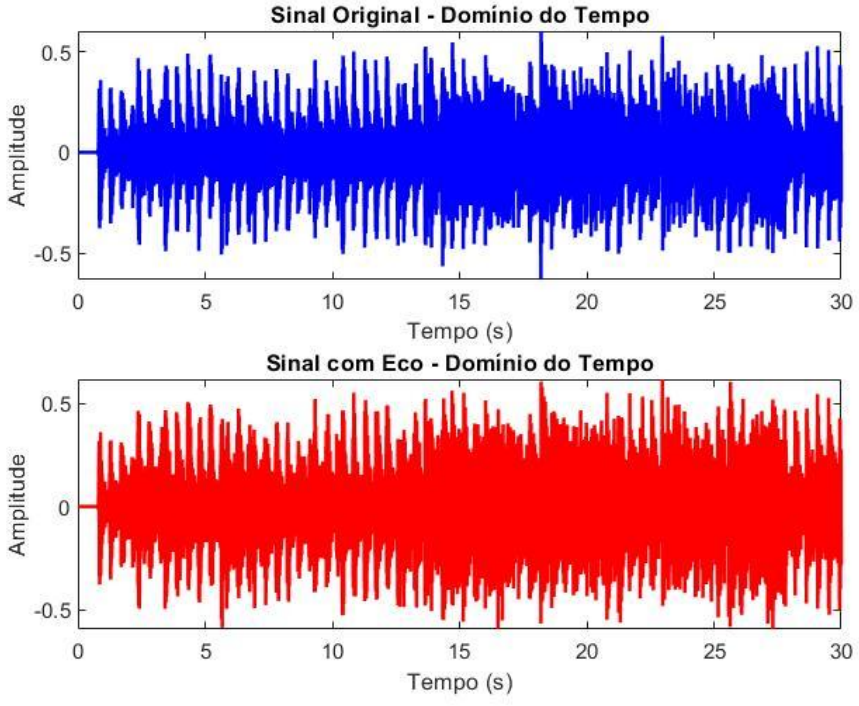

Figura 23: Comparação do sinal original e do sinal com eco, no domínio do tempo.

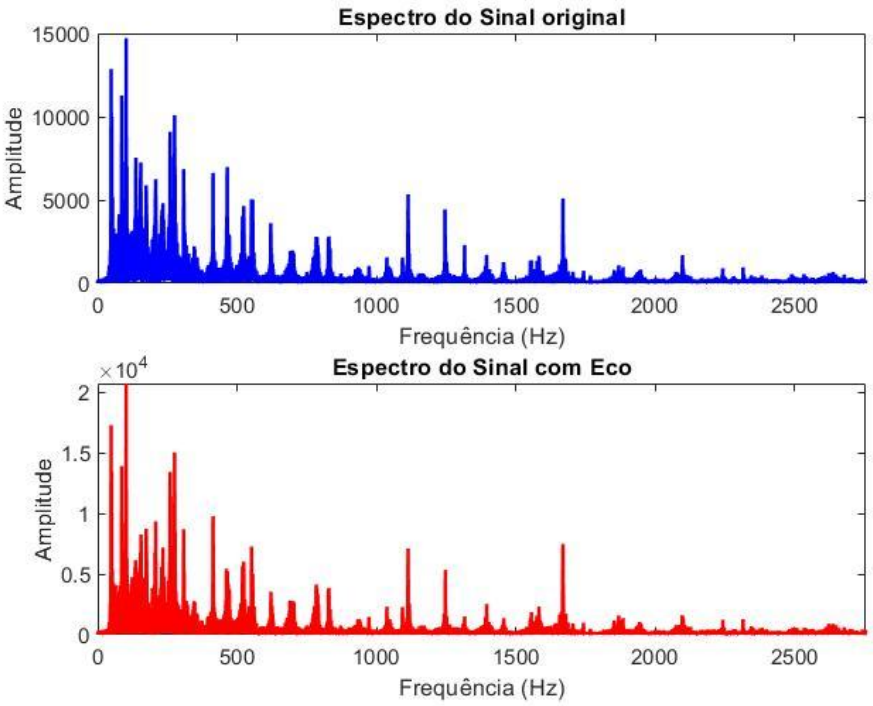

Figura 24: Comparação do espectro original e do espectro com eco.

Nas figuras 23 e 24, percebemos o efeito do eco nos domínios de tempo e frequência, respectivamente. Percebemos em ambos os gráficos diferenças nas magnitudes dos picos, sendo que, aqueles com eco aumentaram.

2.9. Item i

Usando tempo de atraso de $0.1\ s$ e $0.05\ s$, era possível perceber um efeito de “coral” na música. Com atraso de $0.025\ s$ já não era mais possível ouvir o efeito do eco. Logo, usando a fórmula do item anterior, obtemos a seguinte quantidade de amostras:

$$D = 0.025 * 44100 = 1102.5\ amostras$$

3. Conclusão

Por meio deste trabalho, pôde-se ter um contato mais prático com o objetivo desta matéria, a aquisição de dados e as análises realizadas após a obtenção do sinal, que neste caso era proveniente de uma música.

Foram realizadas técnicas como Transformada Discreta de Fourier, e sua versão otimizada, Transformada Rápida de Fourier, para gerar os espectros de frequência do sinal. Também, métodos para alterar a taxa de amostragem do sinal, como a decimação, que reduz, e a interpolação, que aumenta essa taxa. Também, utilizou-se de filtros para mudar o comportamento do sinal. Ao usar filtro passa-baixa, pudemos remover as frequências a partir de certo ponto, para obter resultados diferentes, baseados naquilo que se queria ouvir após a filtragem.

Por fim, realizou-se um projeto de filtro de eco, utilizando-se de sua equação a diferenças. O filtro de eco é determinado por dois fatores, sendo eles: a, que seria a atenuação do sinal, e D, que é a quantidade de amostras anteriores à amostra atual que será acrescentada à atual.

Conclui-se então que o propósito de estudar o processamento digital de um sinal de áudio real foi obtido com sucesso.

4. Anexos

Todos os códigos utilizados e imagens obtidas estão organizados e podem ser encontrados no seguinte link: https://github.com/CH-bruno/ES879-TrabalhoFinal

4.1. Item a

```
Unset
clc;
clear all;
close all;

[y, Fs] = audioread("Viva la vida - coldplay.wav");
y = y(:, 1) + y(:, 2);

ymedio = mean(y);

% Subtraindo a média de cada elemento do sinal
y = y - ymedio;
```

```
% Dominio do tempo
t = linspace(0, length(y)/Fs, length(y));
figure;
subplot(2,1,1);
plot(t, y);
title("Domínio do tempo");
xlabel("t(s)");
ylabel("Amplitude");

% Dominio da Frequencia
Y = 2 * fft(y);

frequencias = linspace(0, Fs, length(Y));

subplot(2,1,2);
plot(frequencias, abs(Y));
xlabel('Frequência (Hz)');
ylabel('Magnitude');
title('Espectro de Magnitudes do Sinal de Áudio');
xlim([0, Fs/2]);
```

4.2. Item b

Unset

```
clc;
clear all;
close all;

[y, Fs] = audioread("Viva la vida - coldplay.wav");
y = y(:, 1) + y(:, 2);

ymedio = mean(y);

% Subtraindo a média de cada elemento do sinal
y = y - ymedio;
```

```
% Função usada para decimar
y_decimado = decimate(y, 8);

t = linspace(0, length(y_decimado)/(Fs/8),
length(y_decimado));

%Plotando
figure;
subplot(2,1,1);
plot(t, y_decimado);
title("Domínio do tempo");
xlabel("t(s)");
ylabel("Amplitude");

%Dominio da Frequencia
Y = 2 * fft(y_decimado);

frequencias = linspace(0, Fs/8, length(Y));

% Plote o espectro de magnitudes
subplot(2,1,2);
plot(frequencias, abs(Y));
xlabel('Frequência (Hz)');
ylabel('Magnitude');
title('Espectro de Magnitudes do Sinal de Áudio');
xlim([0, Fs/16]);

%audiowrite("Audio Decimado por 8.wav", y_decimado,
round(Fs/8));

%soundsc(y, Fs)
%soundsc(y_decimado, Fs/8)
```

4.1. Item d

```
Unset
clc;
clear all;
close all;

[y, Fs] = audioread("Viva la vida - coldplay.wav");
y = y(:, 1) + y(:, 2);

ymedio = mean(y);

% Subtraindo a média de cada elemento do sinal
y = y - ymedio;

% Projeto do filtro FIR
Fc = 5000; % Frequência de corte(teste 50, 500, 5000)
N = 200; % Ordem do filtro (teste 50, 100, 200)
h = fir1(N + 1, Fc/(Fs/2));

% Resposta em Frequência do Filtro
freqz(h, 1, 1024, Fs);

% Aplicando o filtro FIR ao sinal original
y_filtrado = filter(h, 1, y);

%Dominio da Frequencia
Y_original = 2 * fft(y);
Y_filtrado = 2 * fft(y_filtrado);

%Espectros original e filtrado
figure;
plot(linspace(0, Fs, length(Y_original)), abs(Y_original),
'b', 'LineWidth', 1.5);
hold on;
plot(linspace(0, Fs, length(Y_filtrado)), abs(Y_filtrado),
'r', 'LineWidth', 1.5);
xlabel('Frequência (Hz)');
ylabel('Magnitude');
```

```
title('Comparação do Espectro de Magnitude do Sinal Original e
Filtrado');
legend('Sinal Original', 'Sinal Filtrado');
xlim([0, Fs/2]);

% Vetor de tempo para o sinal filtrado
t_filtrado = linspace(0, length(y_filtrado)/Fs,
length(y_filtrado));

%Plotando no domínio do tempo
figure;
plot(linspace(0, length(y)/Fs, length(y)), y, 'b',
'LineWidth', 1.5);
hold on;
plot(t_filtrado, y_filtrado, 'r', 'LineWidth', 1.5);
title("Comparação do Sinal Original e Filtrado no Dominio do
tempo");
xlabel("t (s)");
ylabel("Amplitude");
legend('Sinal Original', 'Sinal Filtrado');


%audiowrite("Audio Filtrado em 5000 Hz.wav", y_filtrado, Fs);

%soundsc(y, Fs)
%soundsc(y_filtrado, Fs)
```

4.1. Item e

Unset

```
clc;
clear all;
close all;

[y, Fs] = audioread("Viva la vida - coldplay.wav");
y = y(:, 1) + y(:, 2);
```

```
ymedio = mean(y);

y = y - ymedio;

% Projeto do filtro FIR
Fc = 500;
N = 200;
h = fir1(N + 1, Fc/(Fs/2));

y_filtrado = filter(h, 1, y);

y_filtrado_decimado = decimate(y_filtrado, 8);

y_decimado = decimate(y, 8);

% Vetor de tempo para o sinal decimado
t_decimado = linspace(0, length(y_decimado)/(Fs/8),
length(y_decimado));

% Vetor de tempo para o sinal filtrado/decimado
t_filtrado_decimado = linspace(0,
length(y_filtrado_decimado)/(Fs/8),
length(y_filtrado_decimado));

% Dominio da Frequencia do Sinal Original Decimado
Y_decimado = 2 * fft(y_decimado);
frequencies_decimado = linspace(0, Fs/8, length(Y_decimado));

% Dominio da Frequencia do Sinal Filtrado Decimado
Y_filtrado_decimado = 2 * fft(y_filtrado_decimado);
frequencies_filtrado_decimado = linspace(0, Fs/8,
length(Y_filtrado_decimado));

% Plot no domínio do tempo para o sinal decimado e o sinal
filtrado/decimado
figure;
plot(t_decimado, y_decimado, 'b', 'LineWidth', 1.5);
hold on;
```

```
plot(t_filtrado_decimado, y_filtrado_decimado, 'r',
'LineWidth', 1.5);
title("Sinal Decimado e Sinal Filtrado/Decimado - Domínio do
Tempo");
xlabel("t (s)");
ylabel("Amplitude");
legend('Sinal Decimado', 'Sinal Filtrado e Decimado');
xlim([0, length(y_decimado)/(Fs/8)]);

% Plot no domínio da frequência para o sinal decimado e o
sinal filtrado/decimado
figure;
plot(frequencies_decimado(1:length(frequencies_decimado)/2),
abs(Y_decimado(1:length(Y_decimado)/2)), 'b', 'LineWidth',
1.5);
hold on;
plot(frequencies_filtrado_decimado(1:length(frequencies_filtra
do_decimado)/2),
abs(Y_filtrado_decimado(1:length(Y_filtrado_decimado)/2)),
'r', 'LineWidth', 1.5);
xlabel("Frequência (Hz)");
ylabel("Magnitude");
title("Sinal Decimado e Sinal Filtrado/Decimado - Domínio da
Frequência");
legend('Sinal Decimado', 'Sinal Filtrado e Decimado');
xlim([0, Fs/16]);

%soundsc(y_decimado, Fs/8)
%soundsc(y_filtrado_decimado, Fs/8)

%audiowrite("Áudio filtrado em 500 Hz e Decimado em 8.wav",
y_filtrado_decimado, round(Fs/8));
```

4.1. Item f

```
Unset
clc;
```

```
clear all;
close all;

[y, Fs] = audioread("Viva la vida - coldplay.wav");
y = y(:, 1) + y(:, 2);

% Calculando a média do sinal
ymedio = mean(y);

y = y - ymedio;

%Interpolando com L = 4
L = 4;
y_superamostrado = interp(y, L);

t_superamostrado = linspace(0,
length(y_superamostrado)/(Fs*L), length(y_superamostrado));

%Plotando no domínio do tempo para o sinal original e
superamostrado
figure;
subplot(2,1,1);
plot(linspace(0, length(y)/Fs, length(y)), y, 'b',
'LineWidth', 1.5);
title("Sinal Original - Domínio do Tempo");
xlabel("t (s)");
ylabel("Amplitude");
subplot(2,1,2);
plot(t_superamostrado, y_superamostrado, 'r', 'LineWidth',
1.5);
title("Sinal Interpolado - Domínio do Tempo");
xlabel("t (s)");
ylabel("Amplitude");


%Dominio da Frequencia para o Sinal Original
Y_original = 2 * fft(y);
frequencies_original = linspace(0, Fs, length(Y_original));
```

```
% Domínio da Frequência para o Sinal Superamostrado
Y_superamostrado = 2 * fft(y_superamostrado);
frequencies_superamostrado = linspace(0, Fs*L,
length(Y_superamostrado));

% Plotando os espectros original e superamostrado
figure;
subplot(2,1,1);
plot(frequencies_original(1:length(frequencies_original)/2),
abs(Y_original(1:length(Y_original)/2)), 'b', 'LineWidth',
1.5);
xlabel("Frequência (Hz)");
ylabel("Magnitude");
title("Sinal Original - Domínio da Frequência");
subplot(2,1,2);
plot(frequencies_superamostrado(1:length(frequencies_superamos
trado)/2),
abs(Y_superamostrado(1:length(Y_superamostrado)/2)), 'r',
'LineWidth', 1.5);
xlabel("Frequência (Hz)");
ylabel("Magnitude");
title("Sinal Interpolado - Domínio da Frequência");
xlim([0, Fs/2]);

figure;

plot(frequencies_original(1:length(frequencies_original)/2),
abs(Y_original(1:length(Y_original)/2)), 'b', 'LineWidth',
1.5);
hold on;
plot(frequencies_superamostrado(1:length(frequencies_superamos
trado)/2),
abs(Y_superamostrado(1:length(Y_superamostrado)/2))/4, 'r',
'LineWidth', 1.5);
xlabel("Frequência (Hz)");
ylabel("Magnitude");
title("Sinal Original e Interpolado - Domínio da Frequência");
legend("Sinal Original", "Sinal Interpolado dividido por 4");
xlim([0, Fs/2]);
```

```
%soundsc(y, Fs)
%soundsc(y_superamostrado, Fs*L);

%audiowrite("Audio interpolado em 4.wav", y_superamostrado,
Fs*L);
```

4.1. Item g

```
Unset
%Parâmetros da função de transferência H(z)
a = 0.5;
D = 2;

syms z n;
n_values = 0:10;
hz = (z^D + a) / z^D;
hn = iztrans(hz);  % Transformada inversa de Z

hn_values = subs(hn, n, n_values);

h_func = matlabFunction(hn_values);

hn_numeric = h_func();

%Plotando h[n]
stem(n_values, hn_numeric, 'o');
xlim([0 10]);
title('Resposta ao Impulso h[n], para a = 0.5 e D = 2');
xlabel('n');
ylabel('h[n]');
```

4.1. Item h

Unset

```
clc;
clear all;
close all;

[y, Fs] = audioread("Viva la vida - coldplay.wav");
y = y(:, 1) + y(:, 2);

ymedio = mean(y);
y = y - ymedio;

%Especificações do filtro de eco desejado
a = 0.5;
t_atraso = 0.025;

% Calculando o número de amostras para o atraso desejado
D = round(t_atraso * Fs);

% Funçao de transferência do eco
num = [1, zeros(1, D), a];
den = 1;

% Resposta em frequência do filtro
[H, Freq] = freqz(num, den, 1024, Fs);

y_filtrado = filter(num, den, y);

figure;
plot(Freq, abs(H));
title('Resposta em Frequência');
xlabel('Frequência (Hz)');
ylabel('Magnitude');

figure;
subplot(2,1,1);
plot((0:length(y)-1)/Fs, y, 'b', 'LineWidth', 1.5);
title('Sinal Original - Domínio do Tempo');
xlabel('Tempo (s)');
ylabel('Amplitude');
```

```
subplot(2,1,2);
plot((0:length(y_filtrado)-1)/Fs, y_filtrado, 'r',
'LineWidth', 1.5);
title('Sinal com Eco - Domínio do Tempo');
xlabel('Tempo (s)');
ylabel('Amplitude');


Y_original = 2 * fft(y);
Y_filtrado = 2 * fft(y_filtrado);

frequencias = linspace(0, Fs, length(Y_original));

%Plotando espectro no domínio da frequência
figure;
subplot(2,1,1);
plot(frequencias, abs(Y_original), 'b', 'LineWidth', 1.5);
title('Espectro do Sinal original');
xlabel('Frequência (Hz)');
ylabel('Amplitude');
xlim([0 Fs/16]);

subplot(2,1,2);
plot(frequencias, abs(Y_filtrado), 'r', 'LineWidth', 1.5);
xlim([0 Fs/16]);
title('Espectro do Sinal com Eco');
xlabel('Frequência (Hz)');
ylabel('Amplitude');

soundsc(y_filtrado, Fs);
audiowrite("Audio com eco de 0,025s.wav", y_filtrado, Fs);
```