



Universidade Federal do Ceará

Campos Quixáda

Departamento de Engenharia Computacional

Processamento Digital de Sinais

Autor:

Pedro Emanuel Ferreira Paiva das Neves

Marcos Antonio Moreira Vieira

Professor Orientador:

Carlos Igor Ramos Bandeira

Quixáda, Ceará

Setembro de 2024

Sumário

1	Introdução	2
2	O Sinal Escolhido	2
3	Frequência do Sinal	4
4	Filtro Escolhido	7
5	Processamento do Sinal	8
6	Conclusão do Projeto	14

1 Introdução

modificação e interpretação de sinais, como áudio, vídeo, imagens e dados sensoriais, utilizando algoritmos computacionais. Sinais são informações que variam no tempo e no espaço e podem ser caracterizados como analógicos ou digitais. O processamento digital permite que sinais analógicos sejam analisados por meio da amostragem, que consiste em digitalizar esse sinal, tornando possível manipulá-los de maneira mais eficiente e flexível, oferecendo vantagens em termos de precisão e confiabilidade quando comparado ao processamento analógico.

Para a análise desta prática, foi baixado um sinal com ruído e, através do Octave, foram realizadas avaliações qualitativas e quantitativas dos sinais subsequentes. Este relatório tem como objetivo explorar o processo de tratamento de um sinal específico, desde a sua escolha até o seu processamento por meio de filtros digitais adequados.

2 O Sinal Escolhido

Sinais de áudio, por terem apenas uma dimensão no tempo, são mais fáceis de analisar também devido à avaliação qualitativa, que consiste em ouvir o áudio. Por isso, o sinal escolhido foi um sinal de áudio, mais precisamente uma música. A música escolhida se chama Laputa, da banda Panchiko, presente no álbum \$D E A T H M E T A L. Nesse álbum, foi incluída a música original junto com a mesma música, porém ruidosa, fornecendo assim um áudio com ruído e um original para comparação.

Na primeira análise do sinal, foram recortados os 20 primeiros segundos tanto da música normal quanto da música com ruído, para se obter uma amostra menor dos dois sinais e reduzir o processamento necessário.

```
1 pkg load signal
2 % carregar o arquivo da musica base
3 [x, fs] = audioread('C:\Users\Pedro\Desktop\pds - trabalho\music wav\
    Laputa\Laputa.wav');
4
```

```

5 % definir o corte da musica para os 20 primeiros segundos
6 max_time = 20;
7 n_samples = min(length(x), fs * max_time);
8 x = x(1 : n_samples);
9
10 % definir o tempo do grafico
11 n = length(x);
12 t = (0 : n - 1) / fs;
13 t = t * 1000;
14
15 % pegar a magnitude e a frequencia do audio
16 X = abs(fft(x));
17 f = (0 : n - 1) * (fs / n);
18
19 % salvar a musica recortada
20 audiowrite('C:\Users\Pedro\Desktop\pds - trabalho\music wav\Laputa\
    Laputa_mod.wav', x, fs);
21 disp('0 audio foi salvo como "Laputa_mod.wav".');
22
23 % agora vamos repertir o mesmo processo para o arquivo de musica ruidoso
24 [y, fs] = audioread('C:\Users\Pedro\Desktop\pds - trabalho\music wav\
    Laputa\Laputa_ROT.wav');
25 n_samples = min(length(y), fs * max_time);
26 y = y(1 : n_samples);
27 n2 = length(y);
28 t2 = (0 : n2 - 1) / fs;
29 t2 = t2 * 1000;
30 Y = abs(fft(y));
31 f2 = (0 : n2 - 1) * (fs / n2);
32 audiowrite('C:\Users\Pedro\Desktop\pds - trabalho\music wav\Laputa\
    Laputa_ROT_mod.wav', y, fs);
33 disp('0 audio foi salvo como "Laputa_ROT_mod.wav".');

```

Pode ouvir o recorte do áudio em: *Laputa_Mod* e *Laputa_ROT_Mod*.

Após o recorte do áudio, foi feita uma análise gráfica do áudio em função do tempo, tanto da música original quanto da música com ruído, comparando os dois gráficos. Esses gráficos foram gerados utilizando o código a seguir.

```
1 % plotar o graficos audio x tempo e Magnitude x frequencia
2 figure(1);
3 subplot(2,1,1);
4 plot(t, x);
5 title(' audio x tempo - Laputa');
6 ylabel('Amplitude');
7 xlabel('tempo (ms)');
8 subplot(2,1,2);
9 plot(t2, min(max(y*1.3, -1), 1));
10 title(' audio x tempo - Laputa ROT');
11 ylabel('Amplitude');
12 xlabel('tempo (ms)');
13 grid on;
14 %print('C:\Users\Pedro\Desktop\pds - trabalho\music wav\Laputa\
    graf_audioXtempo.png', '-dpng');
15 disp('A imagem foi salva como "graf_AudioXTempo_Laputa.png".');
```

observa-se que o áudio original tem uma amplitude maior que o ruidoso, embora o áudio com ruído apresente picos maiores, representando a interferência do ruído. Também é perceptível que, a partir dos 14 segundos, há um aumento na amplitude, devido aos agudos da voz do vocalista.

3 Frequência do Sinal

Após a análise visual do sinal escolhido, foi feita uma análise quantitativa da amostragem, começando com a observação dos histogramas tanto do áudio original quanto do ruidoso. O código utilizado é o seguinte.

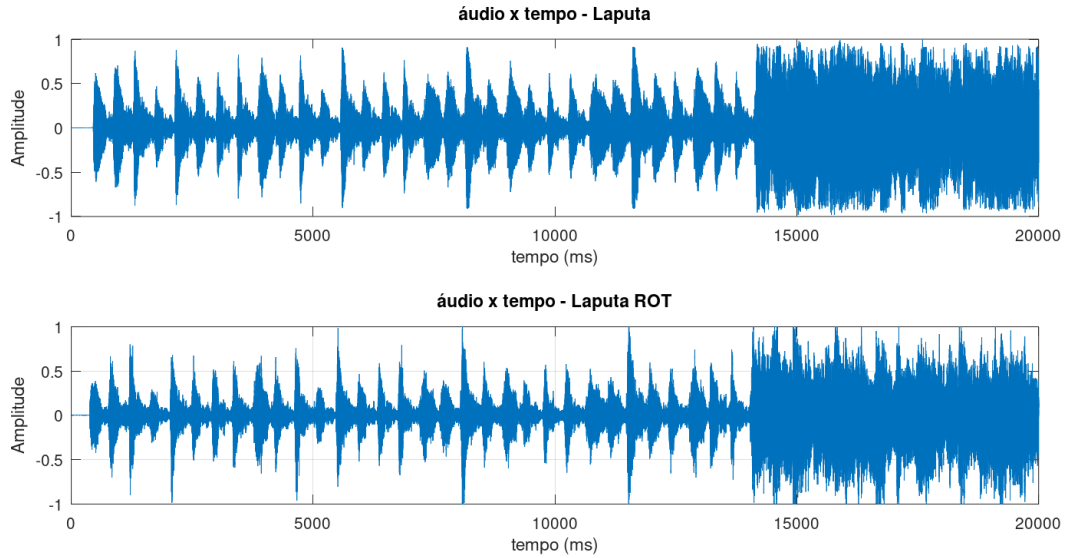


Figura 1: gráfico áudio x tempo

```

1 figure(2);
2 subplot(2,1,1);
3 plot(f(1:n/2), X(1:n/2));
4 title('Magnitude x Frequencia - Laputa');
5 ylabel('Magnitude');
6 xlabel('Frequencia(Hz)');
7 subplot(2,1,2);
8 plot(f2(1:n2/2), Y(1:n2/2));
9 title('Magnitude x Frequencia - Laputa ROT');
10 ylabel('Magnitude');
11 xlabel('Frequencia(Hz)');
12 grid on;
13 %print('C:\Users\Pedro\Desktop\pds - trabalho\music wav\Laputa\
    graf_magnitudeXfrequencia.png', '-dpng');
14 disp('A imagem foi salva como "graf_AudioxTempo_Laputa_ROT.png".');

```

Na imagem 2, observa-se que o áudio com ruído apresenta uma magnitude inferior ao áudio original, fato explicado pela amplitude maior do sinal original. Contudo, nos gráficos

3, verifica-se que o segundo gráfico tem mais picos, especialmente na faixa de 0 Hz a 1700 Hz, enquanto as frequências acima de 1700 Hz são menos acentuadas em comparação ao primeiro gráfico.

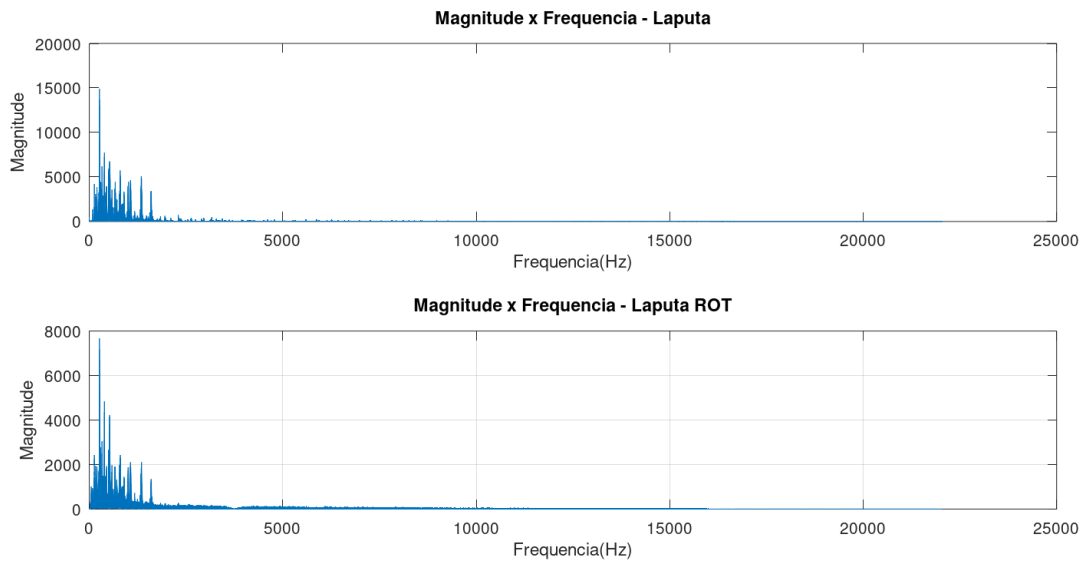


Figura 2: Histogramas obtidos

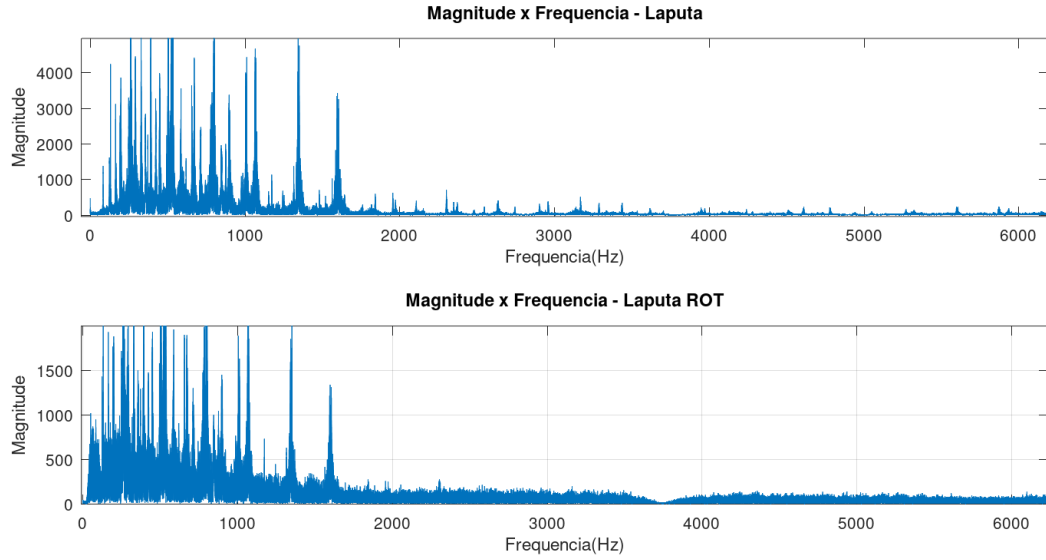


Figura 3: Histogramas obtidos análise

4 Filtro Escolhido

Devido a análise feita, o filtro escolhido para o processamento foi o filtro chebyshev tipo 1. O Chebyshev do tipo 1 é projetado para ter uma resposta em frequência com uma atenuação rápida e eficiente na banda de rejeição. Esse filtro é conhecido por apresentar ondulações na banda passante e uma transição mais rápida entre a banda passante e a banda de rejeição quando comparado a filtros como o Butterworth. O filtro Chebyshev Tipo 1 apresenta uma característica distintiva de ondulações na banda passante, mas uma banda de rejeição monotônica, sem ondulações. Essas ondulações são causadas pela minimização do erro máximo na resposta do filtro, resultando em uma resposta mais "agressiva" na faixa de corte. Esse filtro tem como função de transferencia a equação (1)

$$|H(\omega)| = \frac{1}{\sqrt{1 - \epsilon^2 T_n^2(\omega/\omega_c)}} \quad (1)$$

Onde ϵ é o fator de ondulação na banda passante, T_n é o polinômio de Chebyshev de

ordem n e ω_n é a frequência de corte angular.

O Chebyshev tipo 1 no tipo banda-faixa foi escolhido, justamente por deixar a transição mais acentuada entre as bandas passante e de rejeição, retirando o ruído das bandas de corte sem deixar o som muito abafado, acentuando a transição em agudo e grave. Na parte sem vocal, foi utilizada uma ordem sexta, e com vocal uma ordem 4 para não eliminar muito do agudo do vocal.

5 Processamento do Sinal

Para o processamento do sinal, o sinal ruidoso foi dividido em duas partes para melhorar o processamento do sinal, para isso foi plotado o histograma dessas duas partes

```
1 % carregar o audio ruidoso
2 [x, fs] = audioread('C:\Users\Pedro\Desktop\pds - trabalho\music wav\
    Laputa\Laputa_ROT_mod.wav');
3
4 % dividir o audio no instrumental e quando come a a cantar (14s)
5 n_samples = min(length(x), fs * 14);
6 x_1 = x(1 : n_samples);
7 x_2 = x(n_samples + 1 : end);
8 audiowrite('C:\Users\Pedro\Desktop\pds - trabalho\music wav\Laputa\
    audio_p1.wav', x_1, fs);
9 disp('O audio foi salvo como "audio_p1.wav".');
10 audiowrite('C:\Users\Pedro\Desktop\pds - trabalho\music wav\Laputa\
    audio_p2.wav', x_2, fs);
11 disp('O audio foi salvo como "audio_p2.wav".');
12
13 % magnitudes do espectro de frequencia
14 X_1 = abs(fft(x_1));
15 X_2 = abs(fft(x_2));
16
17 % frequencias do audios
```

```

18 n_1 = length(x_1);
19 f_1 = (0 : n_1 - 1) * (fs / n_1);
20 n_2 = length(x_2);
21 f_2 = (0 : n_2 - 1) * (fs / n_2);
22
23 %plotar os graficos pr processamento
24 figure(3);
25 subplot(2,1,1);
26 plot(f_1(1:n_1/2), X_1(1:n_1/2));
27 title('Espectro - Magnitude x Frequencia');
28 ylabel('Magnitude');
29 xlabel('Frequencia(Hz)');
30 subplot(2,1,2);
31 plot(f_2(1:n_2/2), X_2(1:n_2/2));
32 title('Espectro - Magnitude x Frequencia');
33 ylabel('Magnitude');
34 xlabel('Frequencia(Hz)');
35 grid on;
36 %print('C:\Users\Pedro\Desktop\pds - trabalho\music wav\Laputa\
    graf_espectros_pre.png', '-dpng');
37 disp('A imagem foi salva como "graf_espectros_pre.png".');

```

Pode ouvir os segmentos pré-processamento em: *Segmento1* e *Segmento2*.

A imagem 4 e 5 são histogramas dos dois segmentos de áudio com ruído pré-processamento.

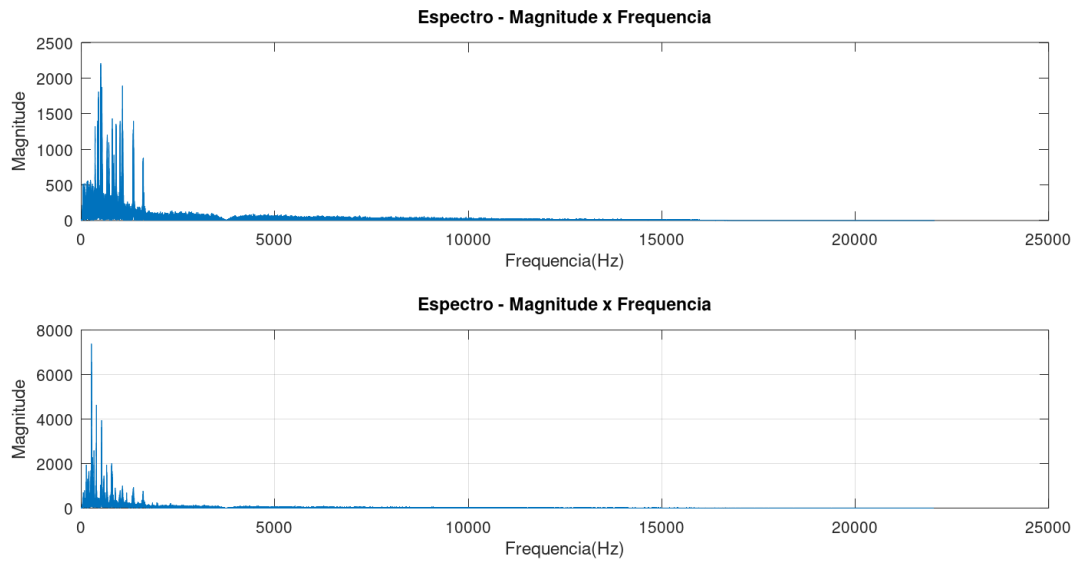


Figura 4: Histogramas pre processamento

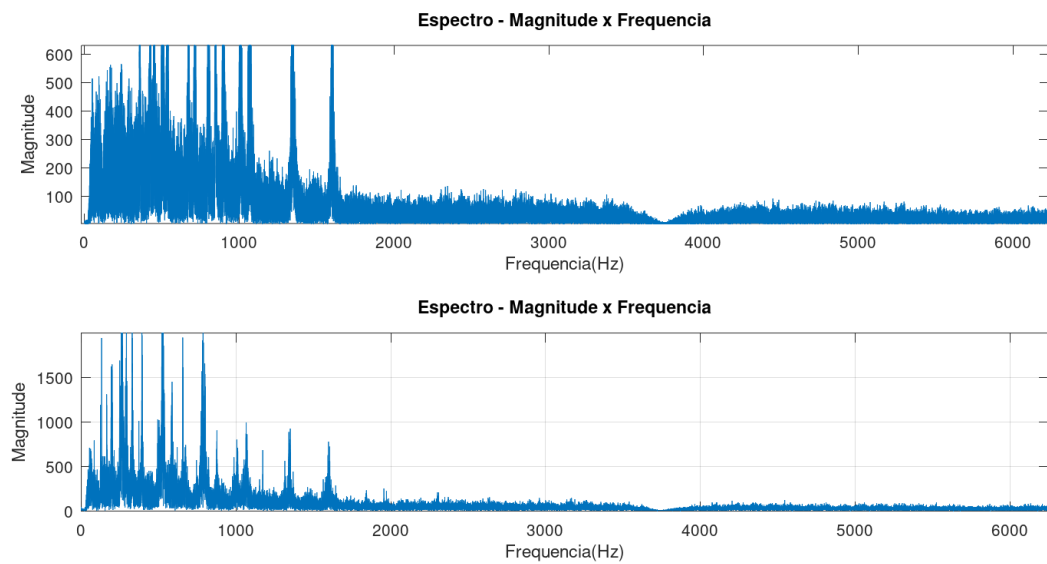


Figura 5: Histogramas pre processamento análise

Após a análise do gráfico, foi feito o processamento usando o filtro Chebyshev tipo 1, usando o filtro bandpass com a frequência de corte de 300 a 1700 Hz e de sexta ordem, já

no segundo segmento foi usado o filtro bandpass com a frequência de corte de 25 a 1700 Hz e de quarta ordem

```
1 % filtro Chebyshev tipo I no primeiro segmento
2 % o filtro vai ser usado no modo bandpass com frequências de corte de 300
   a 1700 Hz e de 6 ordem
3 fc = [300, 1700];
4 [num, den] = cheby1(6, 0.5, fc/(fs/2));
5 p = filter(num, den, x_1);
6 audiowrite('C:\Users\Pedro\Desktop\pds - trabalho\music wav\Laputa\
   audio_p1_filtrado.wav', p, fs);
7
8 % filtro Chebyshev tipo I no segundo segmento
9 % o filtro vai ser usado no modo bandpass com frequências de corte de 25
   a 1700 Hz e de 4 ordem
10 fc = [25, 1700];
11 [num, den] = cheby1(4, 0.5, fc/(fs/2));
12 q = filter(num, den, x_2);
13 audiowrite('C:\Users\Pedro\Desktop\pds - trabalho\music wav\Laputa\
   audio_p2_filtrado.wav', q, fs);
```

Pode ouvir os segmentos pós-processamento em: *Segmento1 – filtrado* e *Segmento2 – filtrado*.

Após o processamento dos segmentos, foi concatenado os áudios e plotando os histogramas pós-processamento.

```
1 % concatenar os dois segmentos filtrados
2 x_filtrado = [p; q];
3 audiowrite('C:\Users\Pedro\Desktop\pds - trabalho\music wav\Laputa\
   Laputa_ROT_filtrado.wav', min(max(x_filtrado*1.3, -1), 1), fs);
4 disp('O audio foi salvo como "Laputa_ROT_filtrado.wav".');
5
6 % magnitudes do espectro de frequência
7 P = abs(fft(p));
```

```

8  Q = abs(fft(q));
9
10 % frequ ncias do udios
11 n_1 = length(p);
12 f_1 = (0 : n_1 - 1) * (fs / n_1);
13 n_2 = length(q);
14 f_2 = (0 : n_2 - 1) * (fs / n_2);
15
16 %plotar os gr ficos p s processamento
17 figure(4);
18 subplot(2,1,1);
19 plot(f_1(1:n_1/2), P(1:n_1/2));
20 title('Espectro - Magnitude x Frequencia');
21 ylabel('Magnitude');
22 xlabel('Frequencia(Hz)');
23 subplot(2,1,2);
24 plot(f_2(1:n_2/2), Q(1:n_2/2));
25 title('Espectro - Magnitude x Frequencia');
26 ylabel('Magnitude');
27 xlabel('Frequencia(Hz)');
28 grid on;
29 %print('C:\Users\Pedro\Desktop\pds - trabalho\music wav\Laputa\
    graf_espectros_pos.png', '-dpng');
30 disp('A imagem foi salva como "graf_espectros_pos.png".');

```

Pode ouvir o áudio filtrado em: *Laputa_filtrado*

A imagem 4 e 5 são histogramas dos dois segmentos de áudio com ruído pós-processamento.

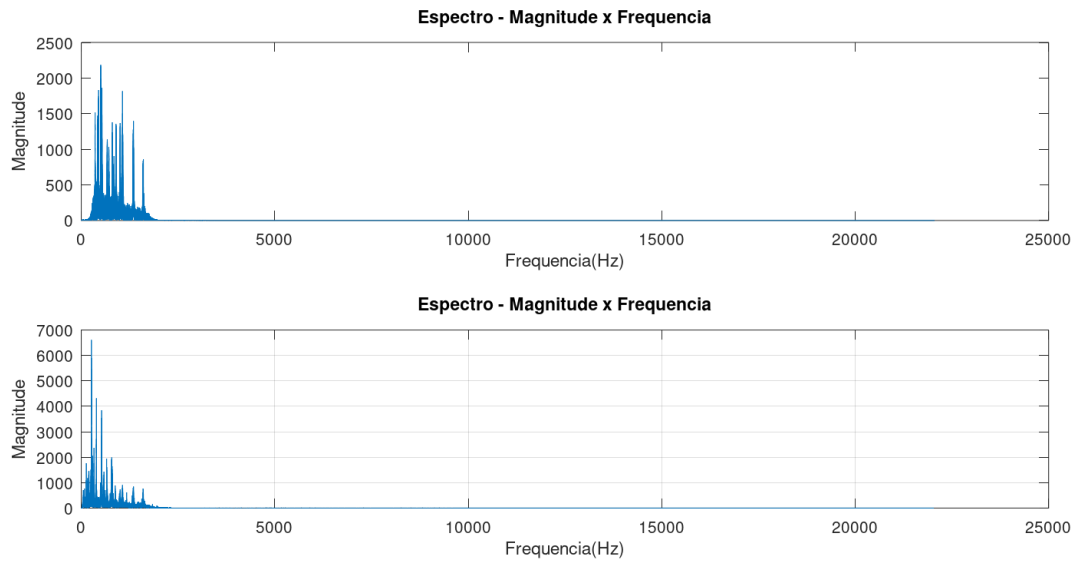


Figura 6: Histogramas pos processamento

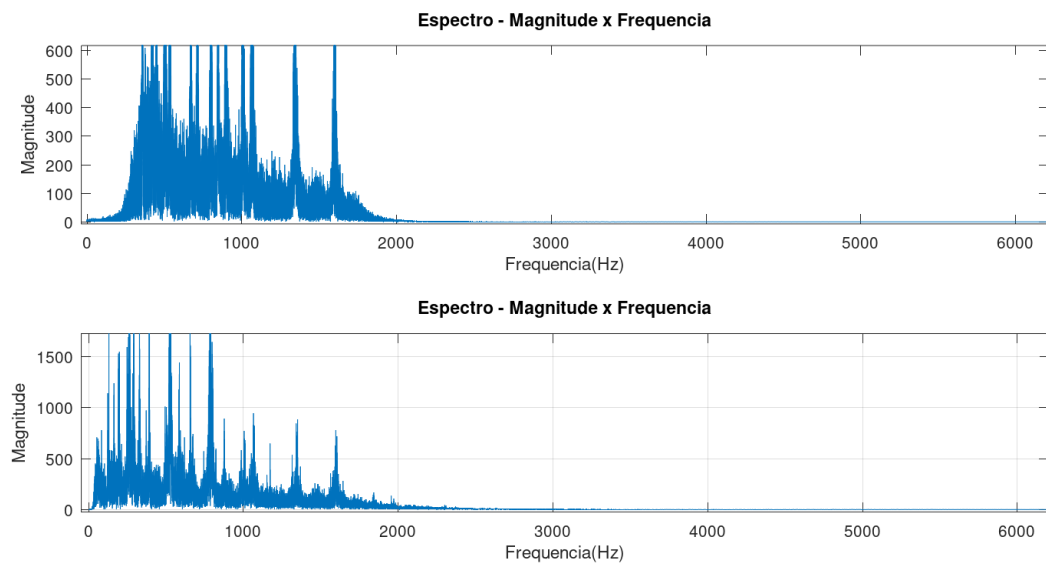


Figura 7: Histogramas pos processamento análise

6 Conclusão do Projeto

Para comparação do recortes dos áudios, foi usado o seguinte código, assim obtendo o gráfico 8.

```
1 % comparando os tres segmentos
2 [x, fs] = audioread('C:\Users\Pedro\Desktop\pds - trabalho\music wav\
    Laputa\Laputa_mod.wav');
3 n = length(x);
4 t = (0 : n - 1) / fs;
5 t = t * 1000;
6 [y, fs] = audioread('C:\Users\Pedro\Desktop\pds - trabalho\music wav\
    Laputa\Laputa_ROT_mod.wav');
7 n2 = length(y);
8 t2 = (0 : n2 - 1) / fs;
9 t2 = t2 * 1000;
10 [x_filtrado, fs] = audioread('C:\Users\Pedro\Desktop\pds - trabalho\music
    wav\Laputa\Laputa_ROT_filtrado.wav');
11 n3 = length(x_filtrado);
12 t3 = (0 : n3 - 1) / fs;
13 t3 = t3 * 1000;
14
15 % comparando os dois graficos do audio x tempo
16 figure(5);
17 subplot(3,1,1);
18 plot(t, x);
19 title ('audio x tempo - Laputa');
20 ylabel('Amplitude');
21 xlabel('tempo (ms)');
22 subplot(3,1,2);
23 plot(t2, min(max(y*1.3, -1), 1));
24 title ('audio x tempo - Laputa ROT');
25 ylabel('Amplitude');
26 xlabel('tempo (ms)');
```

```

27 subplot(3,1,3);
28 plot(t3, min(max(x_filtrado*1.3, -1), 1));
29 title('audio x tempo - Laputa filtrado');
30 ylabel('Amplitude');
31 xlabel('tempo (ms)');
32 grid on;
33 %print('C:\Users\Pedro\Desktop\pds - trabalho\music wav\Laputa\
    graf_comparacao.png', '-dpng');
34 disp('A imagem foi salva como "graf_comparacao_Laputa.png".');

```

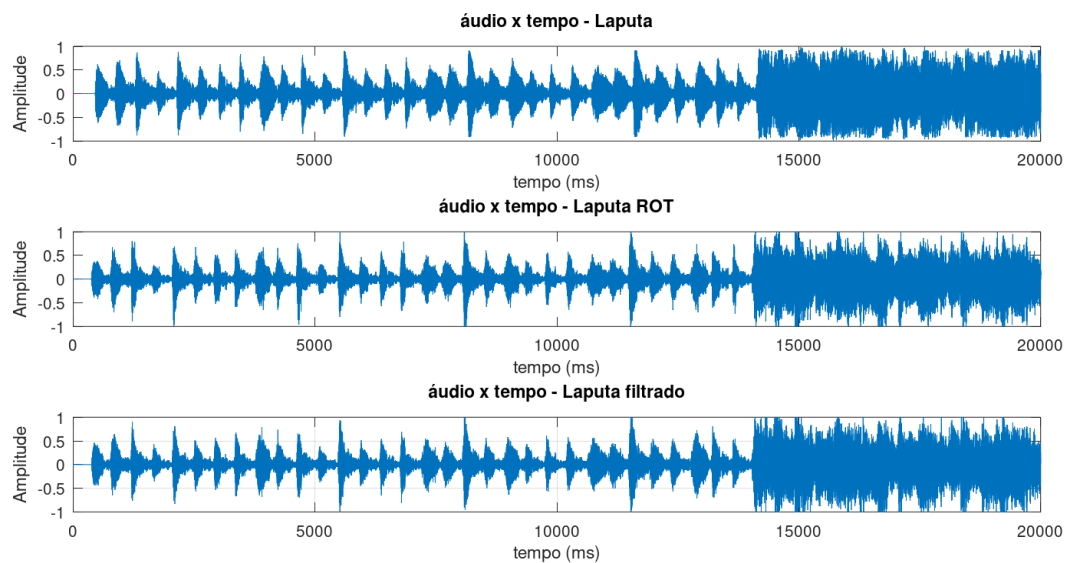


Figura 8: Comparação audio x tempo

Os áudios para comparação: *Laputa_Mod*, *Laputa_ROT_Mod* e *Laputa_filtrado*

Após o processamento do áudio utilizando o filtro Chebyshev Tipo I, foi possível observar uma redução significativa do ruído presente na faixa de frequência original. Comparando os histogramas pré e pós-processamento (figuras 5 e 7), percebe-se que as frequências indesejadas foram eficientemente atenuadas, especialmente nas bandas de corte estabelecidas.

Além disso, a análise qualitativa confirma que o som resultante apresenta uma clareza

superior, com o ruído sendo praticamente imperceptível, sem comprometer a qualidade das faixas vocais ou instrumentais. O uso de diferentes ordens de filtros para os segmentos com e sem vocal foi adequado, evitando a eliminação de frequências importantes nos agudos vocais e preservando a integridade das partes instrumentais.

Os gráficos de comparação entre o áudio original, o áudio ruído e o áudio filtrado (figura 8) mostram que o áudio filtrado apresenta menos picos abruptos, o que visualmente também indica uma maior suavidade na forma de onda resultante.

Portanto, o projeto alcançou seu objetivo de remover ruídos indesejados do áudio, preservando a integridade das frequências essenciais para a música, o que confirma a eficácia do filtro Chebyshev Tipo I no processamento de sinais de áudio para este caso específico.