

Bruno Lourenço da Cunha - 10372978
João Gabriel de Carvalho Ribeiro - 10783027
Kévin Gabriel Gonçalves de Oliveira - 10783156
Renan Ernesto Silva Pinto - 10875930
Vinicius Alves Matias - 10783052

Laboratório de Processamento Espectral

ACH2167 - Computação Sônica

Docente: Regis Rossi A. Faria

Julho de 2021
São Paulo / SP

1. Introdução

O propósito deste trabalho é realizar a análise espectral de um som escolhido pelo grupo (“notaDo.wav”) utilizando um código escrito na linguagem de programação Python em conjunto com algumas bibliotecas auxiliares.

Nosso objetivo neste trabalho é analisar a fonte sonora, identificar as frequências mais destacadas, aplicar uma filtragem com parâmetros baseados nessas frequências para extrair as de maior destaque e então gerar o arquivo de som produzido pela filtragem.

Para a análise da fonte sonora optamos por gerar gráficos visuais de forma que seja possível identificar as frequências destacadas. Primeiro aplicamos a transformada discreta de Fourier utilizando o algoritmo FFT (Transformada Rápida de Fourier) para transformar as amostras de áudio do domínio do tempo e amplitude para o domínio da frequência e magnitude. Isso possibilitará a manipulação das faixas de frequência e a análise das frequências presentes via espectrograma. A FFT pode ser definida a partir do tamanho N da amostra, índice k de frequência discreta e o índice n de tempo discreto:

$$F(k) = \sum_{n=0}^{N-1} f(n)e^{-i\frac{2\pi}{N}nk}$$

E a sua inversa como:

$$f(n) = \frac{1}{N} \sum_{k=0}^{N-1} F(k)e^{i\frac{2\pi}{N}nk}$$

Então para conseguirmos extrair as frequências destacadas da fonte sonora, utilizamos o filtro passa-faixa, onde são rejeitadas frequências fora do intervalo mínimo e máximo definido. Por fim, calculamos a inversa do conjunto filtrado, para trazer o vetor de magnitudes e frequências para o domínio da amplitude e tempo, possibilitando a construção de um arquivo de áudio a partir das amostras geradas dessas frequências filtradas.

2. Desenvolvimento

O formato digital do áudio consiste de 84490 amostras, capturadas a 44.1 kHz (taxa de amostragem), totalizando quase 2 segundos de áudio. A representação do áudio em questão da variação da amplitude no tempo (formato da onda) pode ser visualizada na figura 1. Ainda nesta figura é exibido o resultado da transformada rápida de Fourier para identificar as frequências presentes no arquivo de áudio, permitindo uma detecção visual de quatro picos que se destacam das outras frequências mediante sua alta magnitude. A tabela 1 exhibe as frequências mencionadas e a magnitude relacionada.

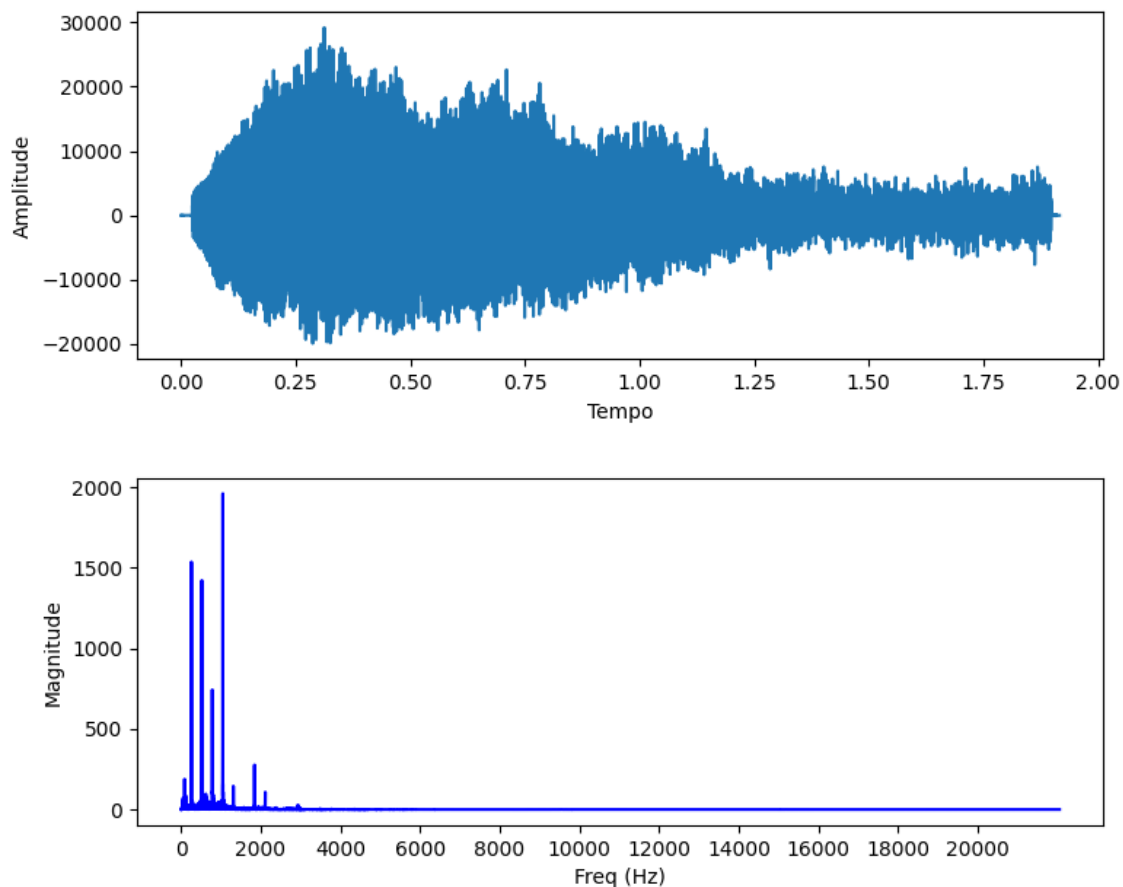


Figura 1 - Características da faixa sonora original e a FFT

Frequência	Magnitude
262 Hz	1536
524 Hz	1422
785 Hz	743
1049 Hz	1960

Tabela 2 - Frequências mais destacadas e suas magnitudes

Frequência	Magnitude
262 Hz	31.9 dB
524 Hz	31.5 dB
785 Hz	28.7 dB
1049 Hz	32.9 dB

Tabela 2 - Frequências mais destacadas e suas magnitudes em decibéis pela fórmula: $10 * \log_{10}(I)$

Frequência	Amplitude relativa
262 Hz	0 dB
524 Hz	-0.4 dB
785 Hz	-2.8 dB
1049 Hz	+4.2 dB

Tabela 3 - Amplitude relativas entre si das frequências destacadas

Identificadas as quatro frequências mais presentes no trecho de áudio, realizamos uma filtragem passa-faixa para estes parâmetros. Para tal, consideramos um intervalo de 4 Hz de desvio das frequências obtidas, isto é, passaremos as frequências entre 260 e 264 Hz, 522 e 526 Hz, 783 e 787 Hz, e 1047 e 1051 Hz.

Para gerar um novo som apenas com as frequências do áudio original que estejam nesse intervalo utilizamos a transformada inversa de Fourier e geramos a onda sonora da figura 2. Também nessa figura exibimos as frequências que se destacam no áudio. O som final removeu alguns ruídos de ambientação do áudio original, destacando o objeto de interesse (nota dó) no áudio final. A amplitude final reduziu significativamente, influenciando em um som de volume mais baixo.

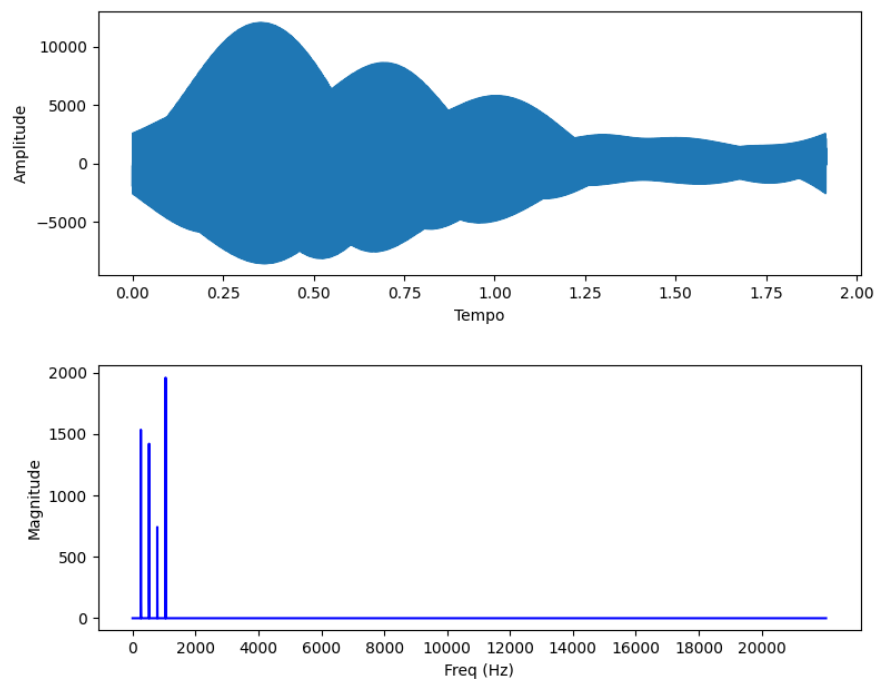


Figura 2 - Onda sonora modificada pelo filtro passa-faixa

Após a filtragem foi possível verificar uma mudança no formato de onda como pode ser visto em mais detalhes a seguir:

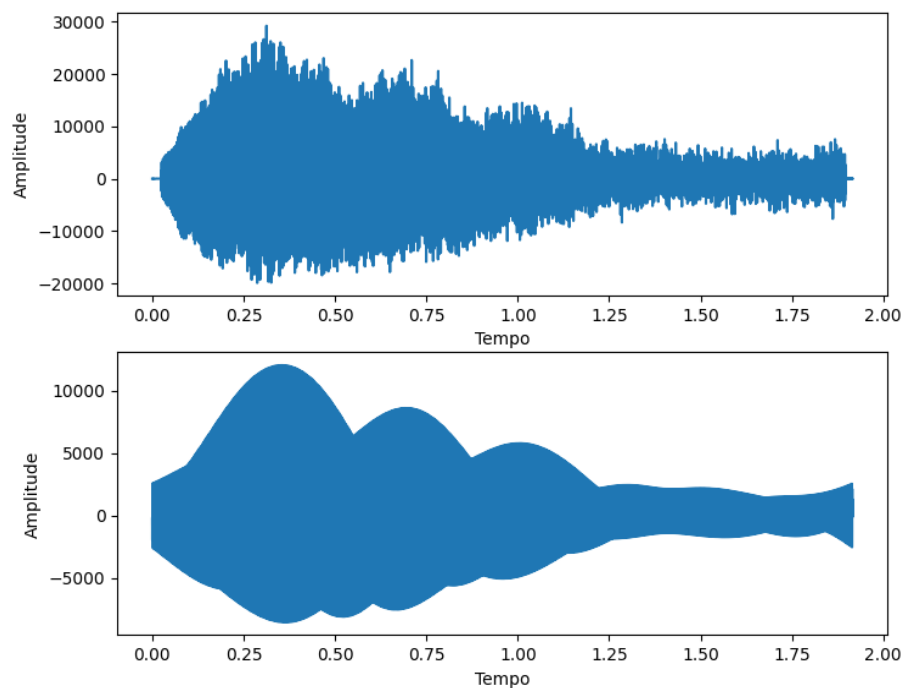


Figura 3 - Frequências da faixa sonora original e filtrada.

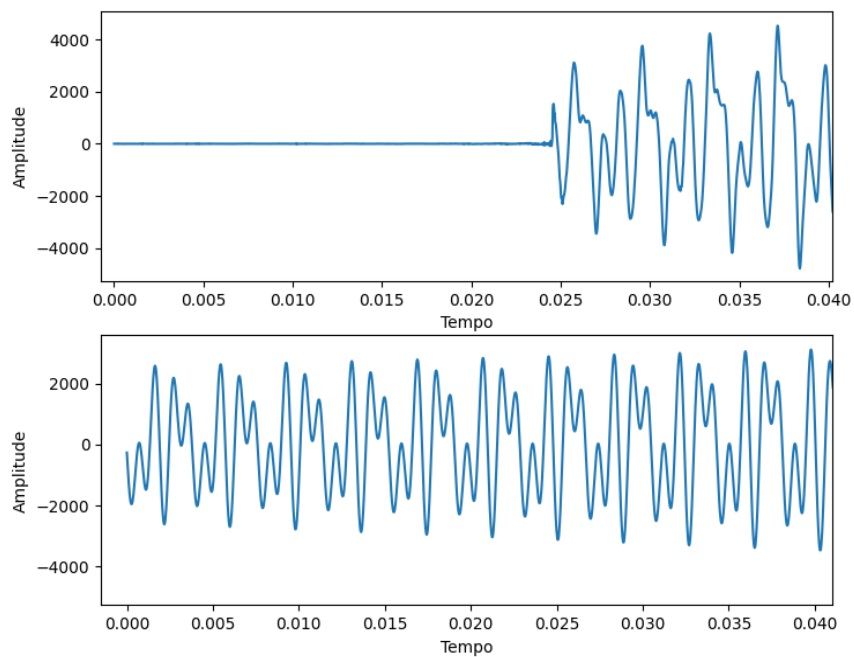


Figura 4 - Frequências da faixa sonora original e filtrada entre 0 e 0.040 segundos.

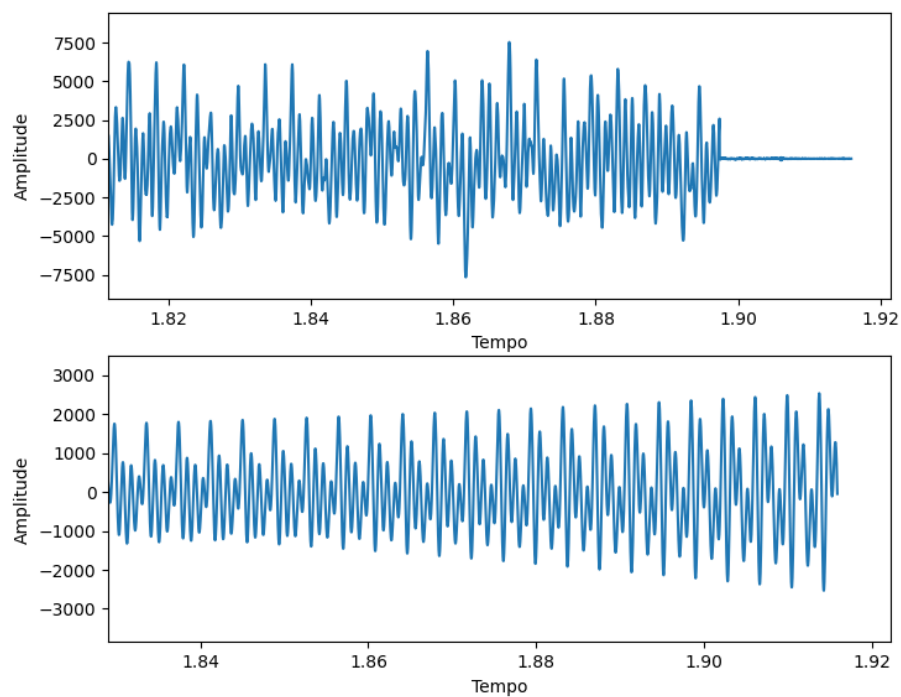


Figura 5 - Frequência da faixa sonora original e filtrada entre 1.82 e 1.92 segundos.

Além disso, o som resultante contém drasticamente menos ruído e se tornou um pouco mais grave devido a perda de algumas frequências altas que faziam parte da sua harmonia. Também é importante ressaltar que existiram dificuldades para a recuperação da parte imaginária da transformada, o que pode explicar a diferença nas amplitudes máximas.

3. Conclusão

Após tratar a fonte sonora com os procedimentos descritos ao longo do relatório, o grupo identificou as frequências que mais se destacam juntamente com suas respectivas magnitudes. Então, foi realizada a análise dos espectros do áudio e o grupo percebeu que, após a filtragem passa banda na faixa de frequências mais destacadas e a recuperação do onda sonora pela transformação inversa de Fourier, deixou o som do arquivo ficou “limpo” no sentido de sem ruídos e muito próximo de um tom complexo simples. Pode-se dizer ainda que o arquivo sonoro se trata de uma série harmônica visto que suas frequências são múltiplas de uma fundamental.

Todos os membros do grupo realizaram os procedimentos citados neste relatório em conjunto durante a reunião: idéias, testes, pesquisas, parametrizações e a escrita do relatório foi feita por todos os membros.

Fontes

- Performing a Fast Fourier Transform (FFT) on a Sound File, por John Paul Mueller, Luca Massaron. Disponível em:
<https://www.dummies.com/programming/python/performing-a-fast-fourier-transform-fft-on-a-sound-file/>
- Filtering a wav file using python. Disponível em:
<https://stackoverflow.com/questions/24920346/filtering-a-wav-file-using-python>
- How to implement band-pass Butterworth filter with Scipy.signal.butter. Disponível em:
<https://stackoverflow.com/questions/12093594/how-to-implement-band-pass-butterworth-filter-with-sciPy-signal-butter/12233959#12233959>
- Band-pass filter. Disponível em: https://en.wikipedia.org/wiki/Band-pass_filter
- Manipulação da biblioteca Matplotlib:
<https://stackoverflow.com/questions/12608788/changing-the-tick-frequency-on-x-or-y-axis-in-matplotlib>
- Fourier troubleshooting: <https://stackoverflow.com/questions/14574208/fftfft-audio-is-just-noise>
- BENITEZ, VICTOR & FUENTES, VALENTINO & Chavez-Burbano, Patricia. (2021). Separación De Flauta Dulce De Un Archivo De Audio Compuesto Por Piano Y Flauta Dulce. Disponível em:
https://www.researchgate.net/publication/28795056_Separacion_De_Flauta_Dulce_De_Un_Archivo_De_Audio_Compuesto_Por_Piano_Y_Flauta_Dulce