

ACH2167 - Computação Sônica  
Prof. Dr. Régis Rossi A. Faria

## **Laboratório de Processamento Espectral**

Gabriel B. Abacherly

10284420

31 de Julho de 2021

## 1. Introdução

Por meio deste trabalho, buscou-se realizar uma análise espectral em um dos sons presentes na pasta compartilhada do ambiente acadêmico, através da linguagem de programação Python.

Para isso, o arquivo “**Microondas.wav**” foi objeto de estudo, sendo processado e submetido à Transformada de Fourier, análises gráficas nos domínios do tempo e da frequência e, por fim, a uma filtragem passa-baixa.

## 2. Análise Espectral

Com o intuito de realizar a análise do arquivo escolhido, foi necessário aplicar a Transformada de Fourier. Por meio desta função, é possível decompor um sinal sonoro, por exemplo, nas frequências que o compõem.

Em outras palavras, a partir de um som, pode-se visualizar a intensidade de cada frequência presente.

Assim, após sua aplicação, será permitida a visualização e remoção de componentes de alta frequência, utilizando-se de um filtro passa-baixa.

### 2.1. Laboratório

Para a execução destas tarefas, optou-se pela utilização da linguagem de programação **Python**, através dos conhecidos **Jupyter Notebooks** e diversos pacotes relacionados a análise e visualização de dados, como:

- **Pandas:** Manipulação e análise de dados;
- **Matplotlib:** Criação de gráficos e visualizações de dados, em geral;
- **SciPy:** Biblioteca Open Source que fornece recursos matemáticos;
- **NumPy:** Suporte à arrays e matrizes multidimensionais e recursos matemáticos

#### 2.1.1. Projeto

O projeto é composto por sete arquivos:

- **Microondas.wav:** Arquivo original que será submetido a análise;
- **microondas\_passa\_baixa.wav:** Arquivo resultante após filtro passa-baixa;
- **analise\_espectral.ipynb:** Arquivo contendo as análises desenvolvidas;
- **analise\_espectral.html:** Arquivo estático com a análise, em caso de problemas com o Notebook;

- **requirements.txt**: Arquivo contendo as dependências do projeto;
- **tabela\_frequencias.csv**: Tabela de frequências e amplitudes relativas em CSV;
- **tabela\_frequencias.xlsx**: Tabela de frequências e amplitudes relativas em Excel.

### 2.1.2. Instalação e Execução

Uma vez que a análise foi realizada em **Notebooks** (arquivos **.ipynb**), é possível visualizar a saída de cada etapa do desenvolvimento sem a necessidade de instalar dependências, com exceção a linguagem Python e a biblioteca Jupyter.

Para instalar e executar todas as dependências do projeto, deve-se:

1. Abra o prompt de comando no diretório contendo os arquivos do projeto e execute “**pip install -r requirements.txt**”:

```
(venv) C:\Users\Gabriel\Desktop\Analise Espectral>pip install -r requirements.txt
Collecting anyio==3.3.0
  Using cached anyio-3.3.0-py3-none-any.whl (77 kB)
Collecting appdirs==1.4.4
  Using cached appdirs-1.4.4-py2.py3-none-any.whl (9.6 kB)
Collecting argon2-cffi==20.1.0
  Using cached argon2-cffi-20.1.0-cp39-cp39-win_amd64.whl (42 kB)
Collecting async-generator==1.10
  Using cached async_generator-1.10-py3-none-any.whl (18 kB)
Collecting attrs==21.2.0
  Using cached attrs-21.2.0-py2.py3-none-any.whl (53 kB)
Collecting audioread==2.1.9
  Using cached audioread-2.1.9.tar.gz (377 kB)
```

2. Após o fim da instalação, basta executar o comando “**jupyter notebook**” e, em um servidor local, será possível visualizar os arquivos no diretório em questão.

```
(venv) C:\Users\Gabriel\Desktop\Analise Espectral>jupyter notebook
[I 2021-07-31 15:59:49.587 LabApp] JupyterLab extension loaded from c:\users\gabriel\desktop\analise espectral\venv\lib\site-packages\jupyterlab
[I 2021-07-31 15:59:49.587 LabApp] JupyterLab application directory is c:\users\gabriel\desktop\analise espectral\venv\share\jupyter\lab
[I 15:59:49.600 NotebookApp] The port 8888 is already in use, trying another port.
[I 15:59:49.601 NotebookApp] Serving notebooks from local directory: C:\Users\Gabriel\Desktop\Analise Espectral
[I 15:59:49.601 NotebookApp] Jupyter Notebook 6.4.0 is running at:
[I 15:59:49.601 NotebookApp] http://localhost:8889/?token=fd5daaa07e214edd6c351cf5f2c67acf73eac80e905ba14e
[I 15:59:49.601 NotebookApp] or http://127.0.0.1:8889/?token=fd5daaa07e214edd6c351cf5f2c67acf73eac80e905ba14e
[I 15:59:49.601 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).

C:\Users\Gabriel\Desktop\Analise Espectral>

To access the notebook, open this file in a browser:
file:///C:/Users/Gabriel/AppData/Roaming/jupyter/runtime/nbsrvr-5048-open.html
Or copy and paste one of these URLs:
http://localhost:8889/?token=fd5daaa07e214edd6c351cf5f2c67acf73eac80e905ba14e
or http://127.0.0.1:8889/?token=fd5daaa07e214edd6c351cf5f2c67acf73eac80e905ba14e
```

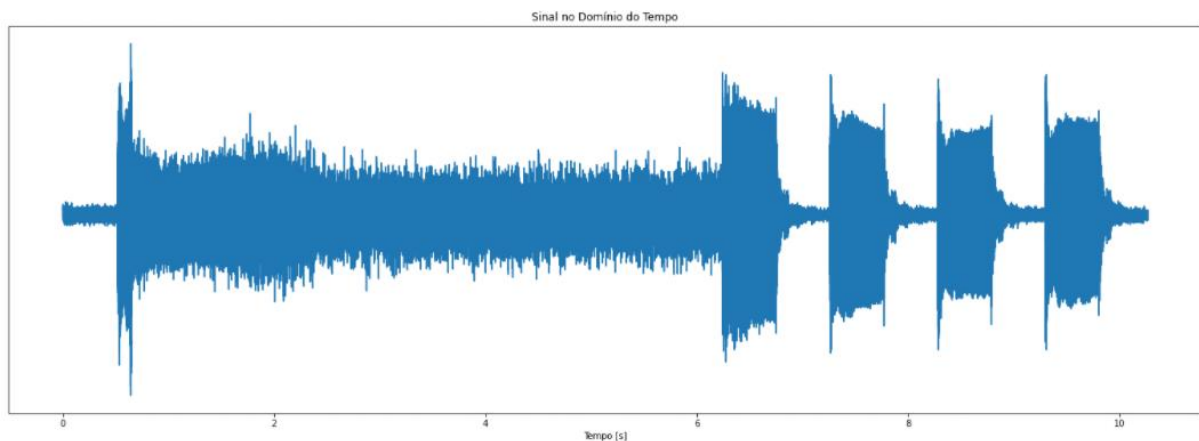
3. No servidor, é possível visualizar todos os arquivos e abrir/manipular o “**analise\_espectral.ipynb**”.



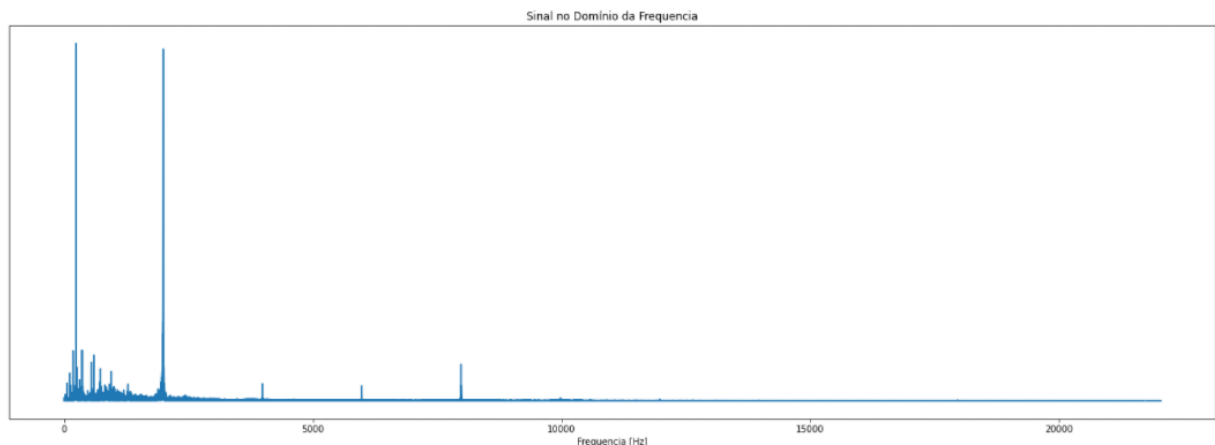
### 2.1.3. Operações

Inicialmente, validou-se a taxa de amostragem do arquivo e sua duração, sendo elas 44100 Hz e 10 segundos, respectivamente.

Em seguida, criou-se a visualização do sinal no domínio do tempo, com o intuito de analisar a variação da amplitude nesta dimensão, através da **Matplotlib**.



Após esta primeira visão, buscou-se representar este som no domínio da frequência. Para tal, foi utilizado o algoritmo “**Fast Fourier Transform**” (**FFT**), implementado na biblioteca **SciPy**, recorrendo, novamente, à Matplotlib para visualizar o resultado.



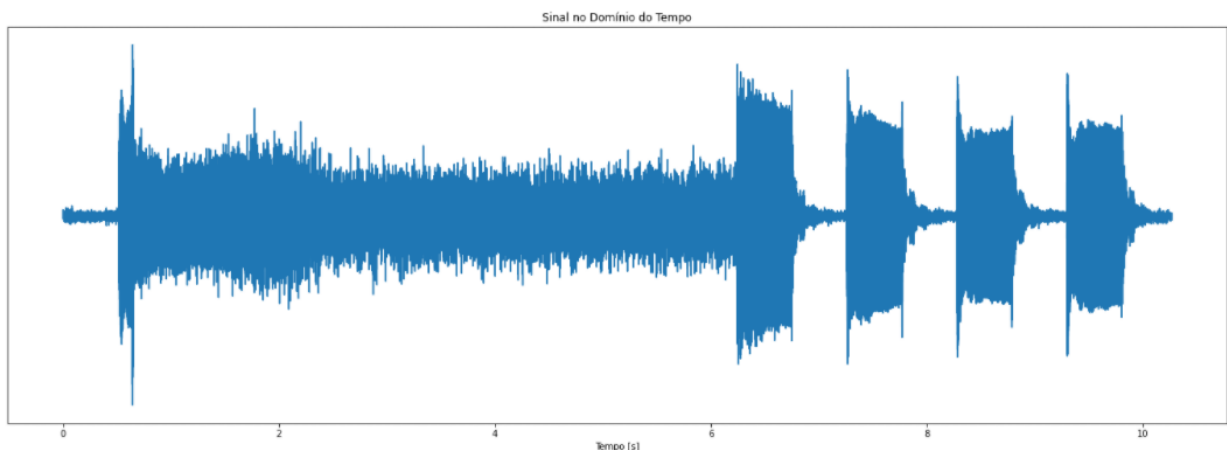
Para gerar uma tabela contendo as frequências (Hz) e as amplitudes (dB) presentes no gráfico acima, utilizou-se a biblioteca **Pandas**, exibindo amostras do resultado e, posteriormente, exportando-o nos arquivos com nome “**tabela\_frequencias**”, disponíveis em CSV e Excel.

Após a criação desta tabela, foi possível determinar os parâmetros utilizados no filtro **passa-baixa**. São eles:

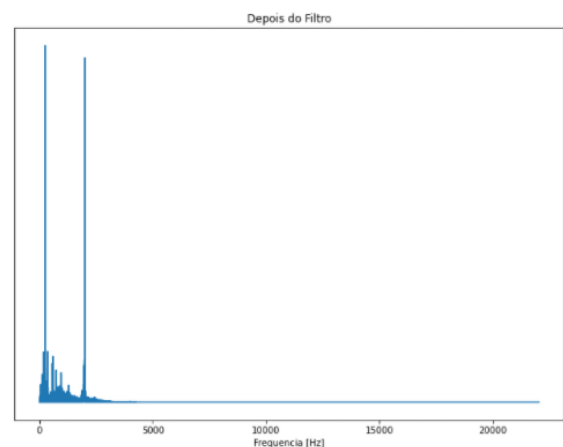
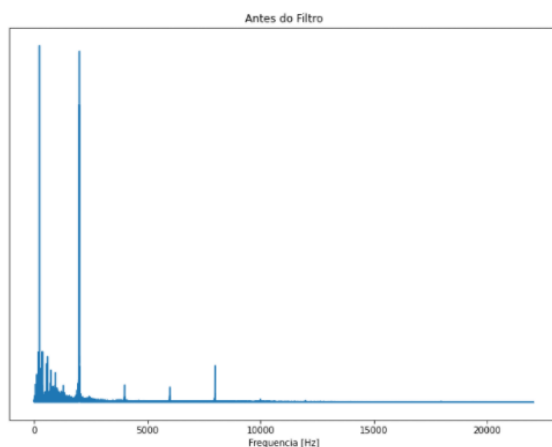
- Período da Amostra = 5.0
- Taxa de Amostragem = 30.0
- Corte = 2
- Frequência de Nyquist =  $0.5 * \text{Taxa de Amostragem}$
- Ordem = 5
- Quantidade de Amostras = Período \* Taxa de Amostragem

A frequência de corte é, então, normalizada e aplicada ao sinal sonoro em análise, respeitando o Teorema de Nyquist e utilizando uma implementação da biblioteca **SciPy**.

Em sequência, valida-se o sinal no domínio do tempo, novamente. Aqui, não são perceptíveis grandes alterações.



Por fim, há a conversão do sinal filtrado para o domínio da frequência e, abaixo, pode-se observar o resultado:



### 3. Considerações Finais

Após a realização das análises e observando o resultado final (arquivo **microondas\_passa\_baixa.wav**), percebe-se que, de fato, as frequências mais baixas foram mantidas. Desta forma, o som apresenta um aspecto “abafado”.

Porém, vale destacar que houve certa dificuldade em determinar e definir os parâmetros para a filtragem, na qual, idealmente, buscava-se um corte mais preciso.