UNIVERSIDADE DO VALE DO ITAJAÍ
ENGENHARIA DE COMPUTAÇÃO
NICOLE MIGLIORINI MAGAGNIN

FILTROS PASSA-BAIXA E PASSA-ALTA

Relatório apresentado como requisito parcial para a obtenção da M1 da disciplina de Processamento Digital de Sinais do curso de Engenharia de Computação pela Universidade do Vale do Itajaí da Escola do Mar, Ciência e Tecnologia.

Prof. Walter Antonio Gontijo

Itajaí
2021

1. OBJETIVO

O presente relatório tem por objetivo a descrição das atividades realizadas em sala de aula no dia 14 de setembro de 2021, sendo essas a implementação em linguagem Python de filtros passa-alta e passa-baixa utilizando sinais e coeficientes fornecidos pelo professor.

2. INTRODUÇÃO

Os filtros passa-baixa são circuitos que permitem a passagem de sinais de baixa frequência e reduzem a intensidade de sinais de alta frequência, enquanto filtros passa-alta permitem a passagem de sinais de alta frequência e reduzem a intensidade de filtros de baixa frequência.

3. DESENVOLVIMENTO

Para a implementação dos filtros solicitados foi usada a linguagem *Python*, juntamente com a biblioteca *numpy* e a IDE Visual Studio Code.

3.1 – A IMPLEMENTAÇÃO

```
vet_zeros = list(numpy.zeros(80)) #Cria lista de 0
vet_coef = [] #Cria vetor para receber os coeficientes
filepath = "Coefs_PA_1k.dat"  #Abre o arquivo
vet_entrada = list(numpy.memmap("two_sins.pcm", dtype='h', mode='r')) #lê o sinal sonoro de entrada
vet_saida = [] #Cria vetor para a saída

with open(filepath,'r') as f: #Abre o arquivo e lê
    for line in f:
        vet_coef.append(line) #Armazena no vetor
```

Figura 1 - Declaração de variáveis e importação do arquivo

Para iniciar a implementação dos filtros passa-baixa e passa alta, são declarados vetores, um para amostras (*vet_zeros*), outro para os coeficientes fornecidos (*vet_coef*), um para o sinal de entrada fornecido (*vet_entrada*) e um último para a saída (*vet_saida*). Assim, é lido o arquivo de coeficientes e passado ao vetor destinado, há um arquivo para passa-alta e outro para passa-baixa.

```
for i in range(len(vet_entrada)): #Loop pelo vetor de entrada (sinal sonoro)
    x=0 # x = 0
    for j in range(len(vet_coef)): #For pelo vetor de coeficiente
        x = x + (float(vet_coef[j]) * float(vet_zeros[j])) #Multiplica e soma com o anterior

    vet_saida.append(x) #Armazena todo somatório na primeira posição

    vet_zeros.pop(len(vet_zeros)-1) #Tira a última posição do vetor de zeros (amostras)
    vet_zeros.insert(0,vet_entrada[0]) #Insere a primeira posição do vetor de entrada
    vet_entrada.pop(0) #Tira a primeira posição do vetor de entrada



mp.plot(vet_saida) #Plota saída
mp.show()

with open('passa_alta.pcm', 'wb') as file: #Salva em PCM
    numpy.array(vet_saida, dtype=numpy.int16).tofile(file)
file.close()
```

Figura 2 – Implementação

O restante da implementação se dá em dois loops utilizando o laço *for*, onde o primeiro deles será responsável por zerar uma variável de armazenamento de soma e o segundo realiza uma convolução, multiplicando as posições do vetor de coeficientes e somando com a multiplicação anterior, depois, é adicionada a saída ao vetor destinado a isso e ainda no laço externo, é retirada a última posição do vetor de amostras e adicionada a primeira posição do vetor entrada no vetor de amostras (também na primeira posição), por fim, retirando essa posição do vetor de entrada.

A saída é plotada utilizando *matplotlib.pylab* e o filtro é salvo em arquivo .pcm. A única diferença entre a implementação passa-alta e passa-baixa é seu coeficiente de entrada.

3.2 – PASSA-ALTA RESULTADOS

3.2.1 – SINAL TWO_SINS.PCM

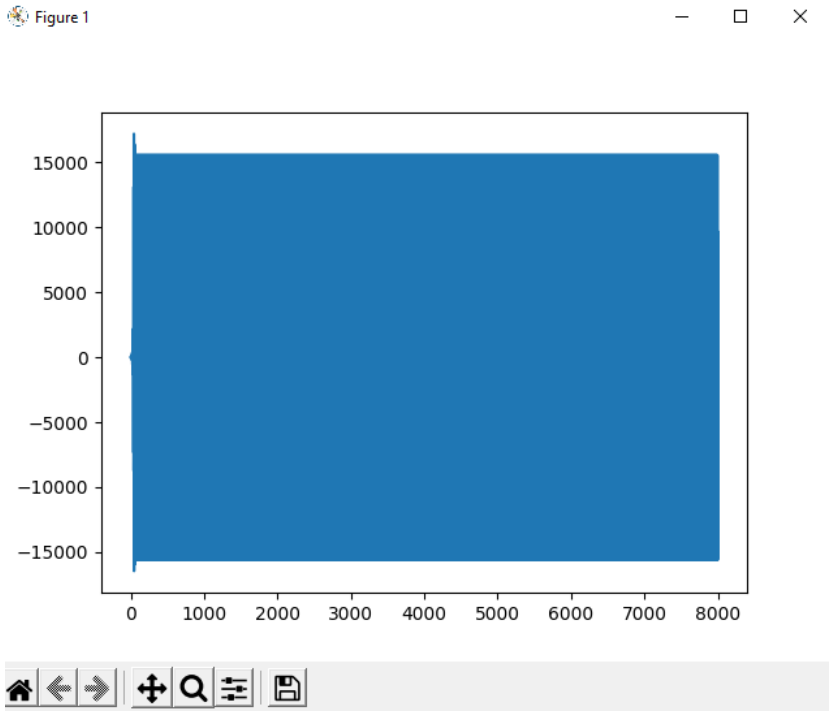

Figura 3 - Filtro passa-alta sinal two_sins plotado no VS code

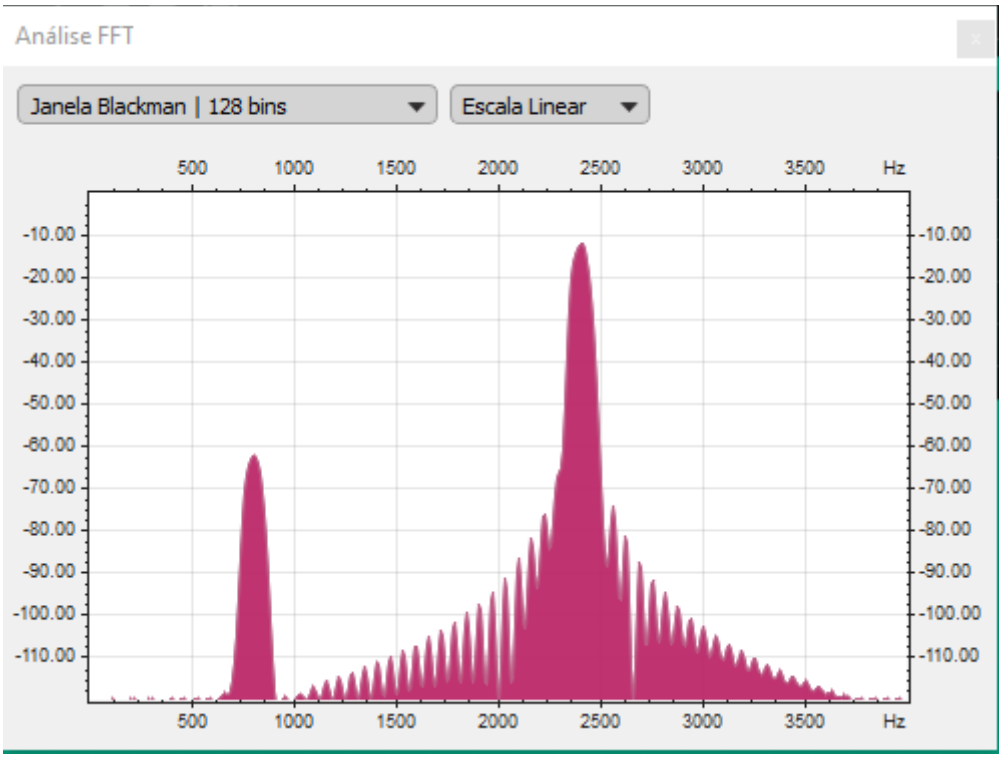

Figura 4 - Filtro passa-alta com o sinal two_sins plotado no OceanAudio

3.2.2 – SINAL SWEEP_100_3k4.PCM

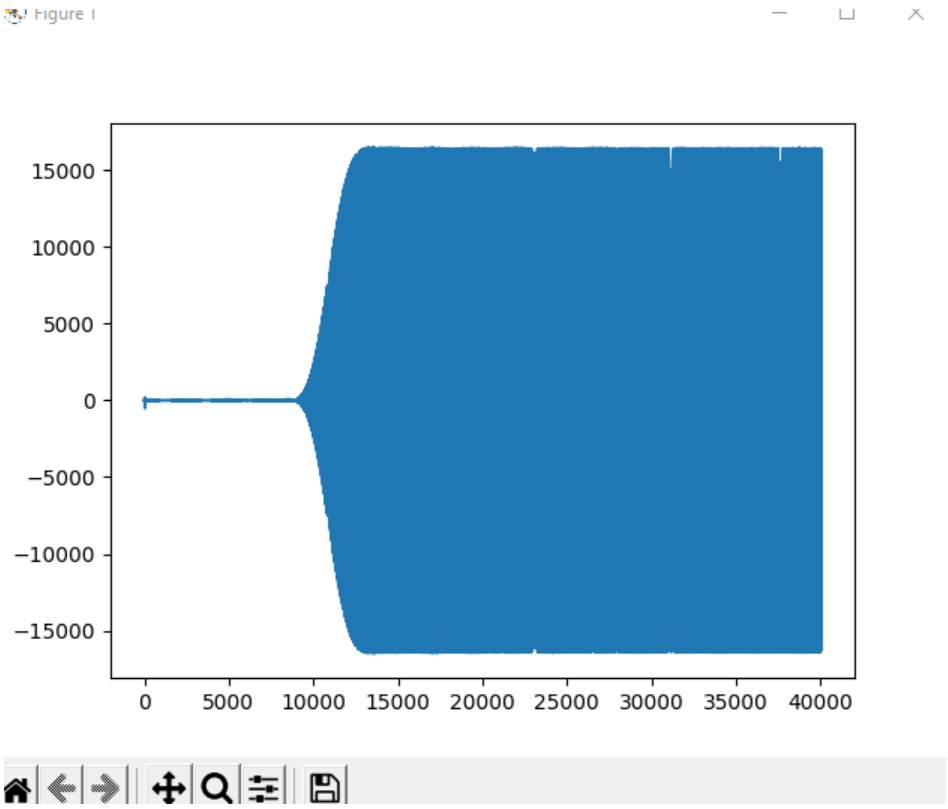

Figura 5 - Filtro passa-alta com sinal Sweep_100_3k4.pcm no VS code

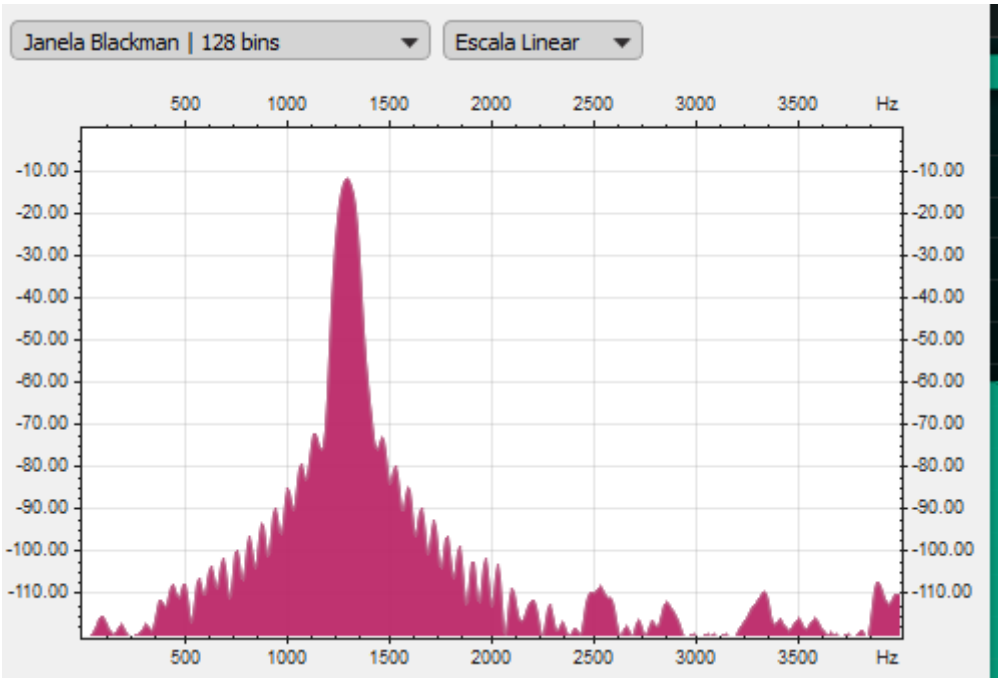

Figura 6 - Filtro passa-alta no sinal sweep no OceanAudio

3.3 – PASSA-BAIXA RESULTADOS

3.3.1 - SINAL TWO_SINS.PCM

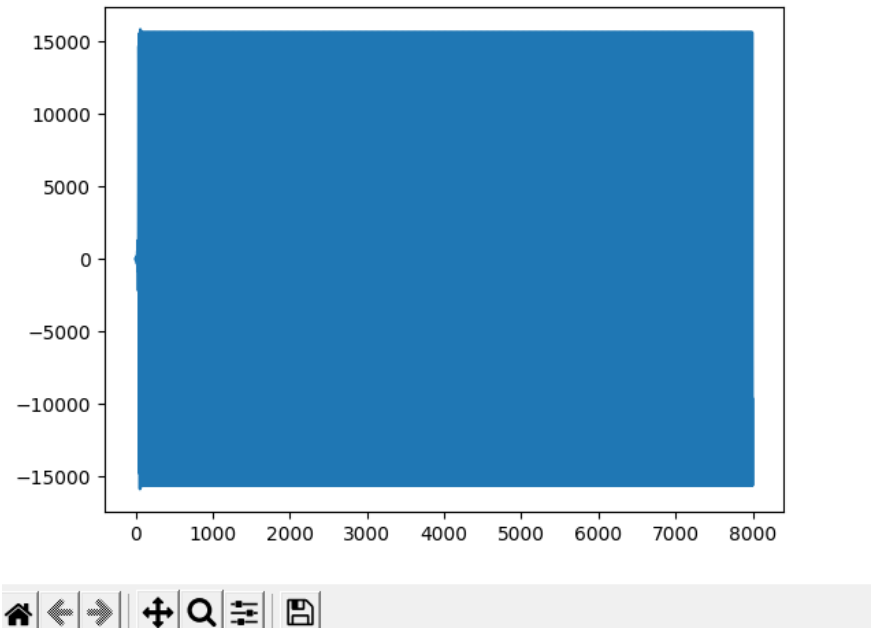

Figura 7 -Filtro passa-baixa sinal two_sins plotado no VS code

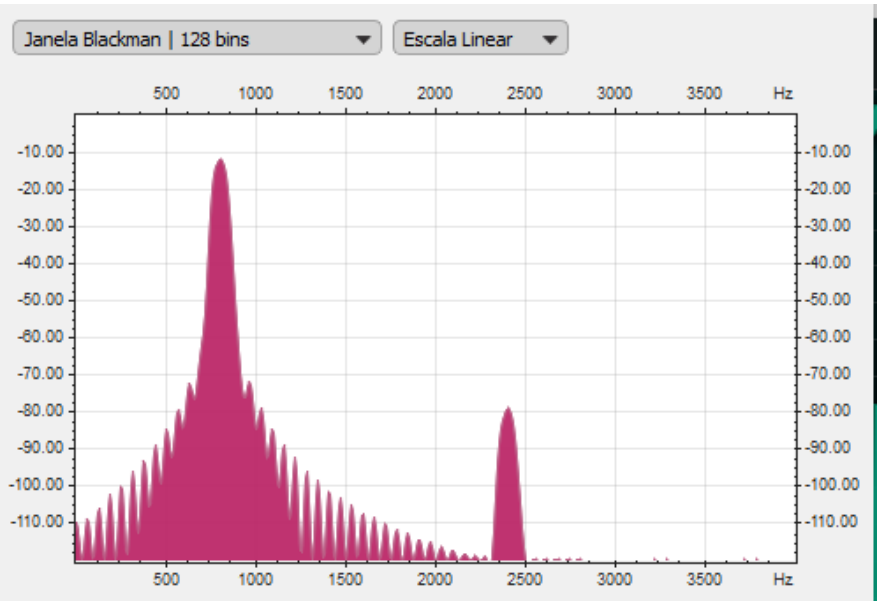

Figura 8 - Filtro passa-baixa sinal two_sins plotado no OceanAudio

3.3.2 - SINAL SWEEP_100_3k4.PCM

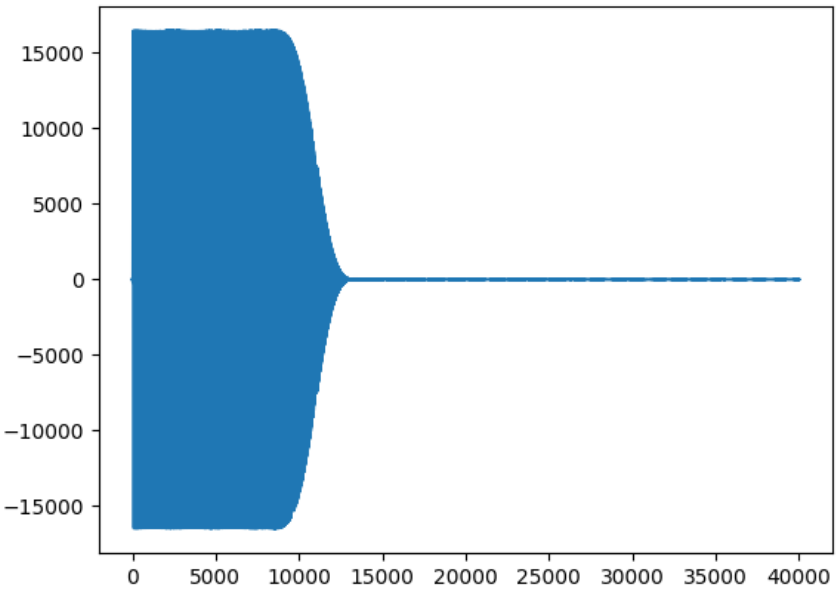

Figura 9 - Filtro passa-baixa sinal sweep_100_3k4.pcm plotado no VS code

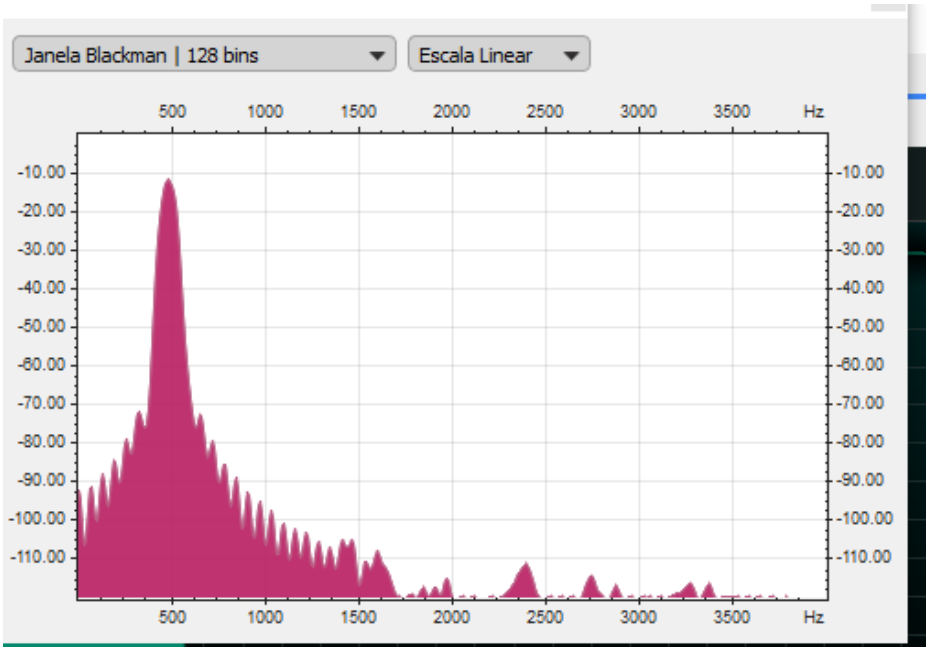

Figura 10- Filtro passa-baixa sinal sweep_100_3k4.pcm plotado no OceanAudio

4. CONCLUSÃO

Com esta implementação foram consolidados os conhecimentos prévios acerca de filtros passa-alta e passa-baixa e obtidos resultados satisfatórios podendo assim visualizar o funcionamento dos filtros corretamente, onde o filtro passa-alta permite a passagem de sinais de alta frequência e atenua sinais de baixa frequência e filtros passa-baixa permitem a passagem de sinais de baixa frequência atenuando sinais de alta frequência.