

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ»  
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)  
Институт №8 «Компьютерные науки и прикладная математика»  
Кафедра 806: «Вычислительная математика и программирование»

# ОТЧЕТ О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ №4

По курсу: «Цифровая обработка изображений»

Тема: «Фильтрация цифровых сигналов»

*Студент:* Чернышёв Д.В.

*Группа:* М8О-107М-22

*Вариант:* 8

*Преподаватель:* Гаврилов К.Ю.

Москва

2023

## СОДЕРЖАНИЕ

1	Задание к лабораторной работе №4.....	3
2	Выполнение лабораторной работы №4 .....	4
3	Выводы лабораторной работы №4 .....	14

## 1 Задание к лабораторной работе №4

### Часть 1

Сформируйте сигнал в виде суммы трех гармонических колебаний с частотами 100 Гц, 1 кГц, 10 кГц. Длительность сигнала составляет 1 с. Постройте

спектр результирующего сигнала и определите области всех гармонических составляющих. Создайте фильтр, выделяющий только сигнал с частотой 100 Гц. Постройте импульсную и частотную характеристики фильтра. Выполните фильтрацию во временной области путем использования функции свертки сигнала с импульсной характеристикой (ИХ) фильтра. Выполните фильтрацию сигнала в частотной области с помощью операции ДПФ. Сравните результаты фильтрации при использовании обоих способов и проанализируйте сигнал на выходе фильтра. Постройте спектр выходного сигнала.

### Часть 2

Дан аудиофайл с записью голоса, на который наложена помеха. Постройте спектр сигнала и определите частотный состав помехи. Постройте режекторный фильтр, подавляющий частоты помехи. С помощью полученного фильтра подавите помехи в заданном сигнале и определите голосовую информацию, содержащуюся заданной аудиозаписи. Постройте спектр отфильтрованного сигнала, сравните его со спектром исходного сигнала и проведите анализ результатов фильтрации. Рекомендация. Для лучшей фильтрации очень сильной помехи, во много раз превосходящей по мощности полезный сигнал, целесообразно в качестве ЧХ фильтра использовать произведение одной и той же ЧХ РФ на себя (2, 3 или более раз).

## 2 Выполнение лабораторной работы №4

Импортируем необходимые библиотеки

```
1 from scipy.signal import kaiserord, lfilter, firwin, freqz
2 import wave as we
3 import numpy as np
4 import matplotlib.pyplot as plt
5
6 from scipy import signal
7 from scipy.io import wavfile
8 from scipy.fft import fft, fftshift
9 from scipy.fft import fftfreq, rfft
10 from scipy.fft import rfftfreq, irfft, ifft
11
12 import os
13 import soundfile as sf
14 from scipy.signal import filtfilt, iirnotch, freqz, butter
15 import librosa
```

### Часть 1

Формирование сигнала

```
1 mu1 = 100
2 mu2= 1000
3 mu3 = 1e4
4
5 N = 1_000
6
7 t = np.linspace(0, 1, N)
8 s = np.sin(2*np.pi*mu1*t)
9     + np.sin(2*np.pi*mu2*t)
10    + np.sin(2*np.pi*mu3*t)
```

## Построение спектра сигнала

```
1 S = np.abs(fft(s))
2 f = fftfreq(len(s), t[1] - t[0])
3
4 plt.plot(f, S)
5 plt.title('Спектр сигнала')
6 plt.xlabel('Частота, Гц')
7 plt.ylabel('Амплитуда')
8 plt.show()
```

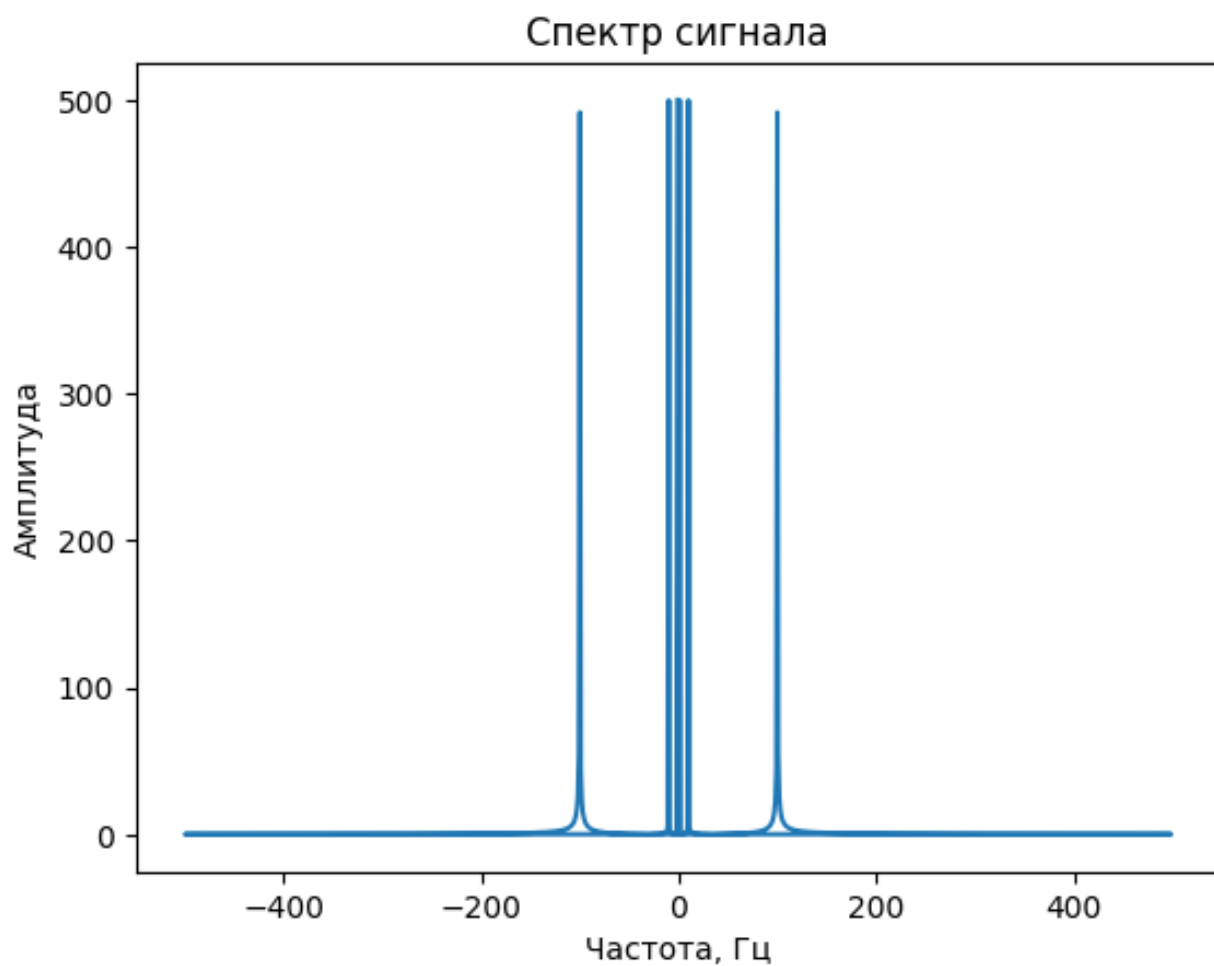
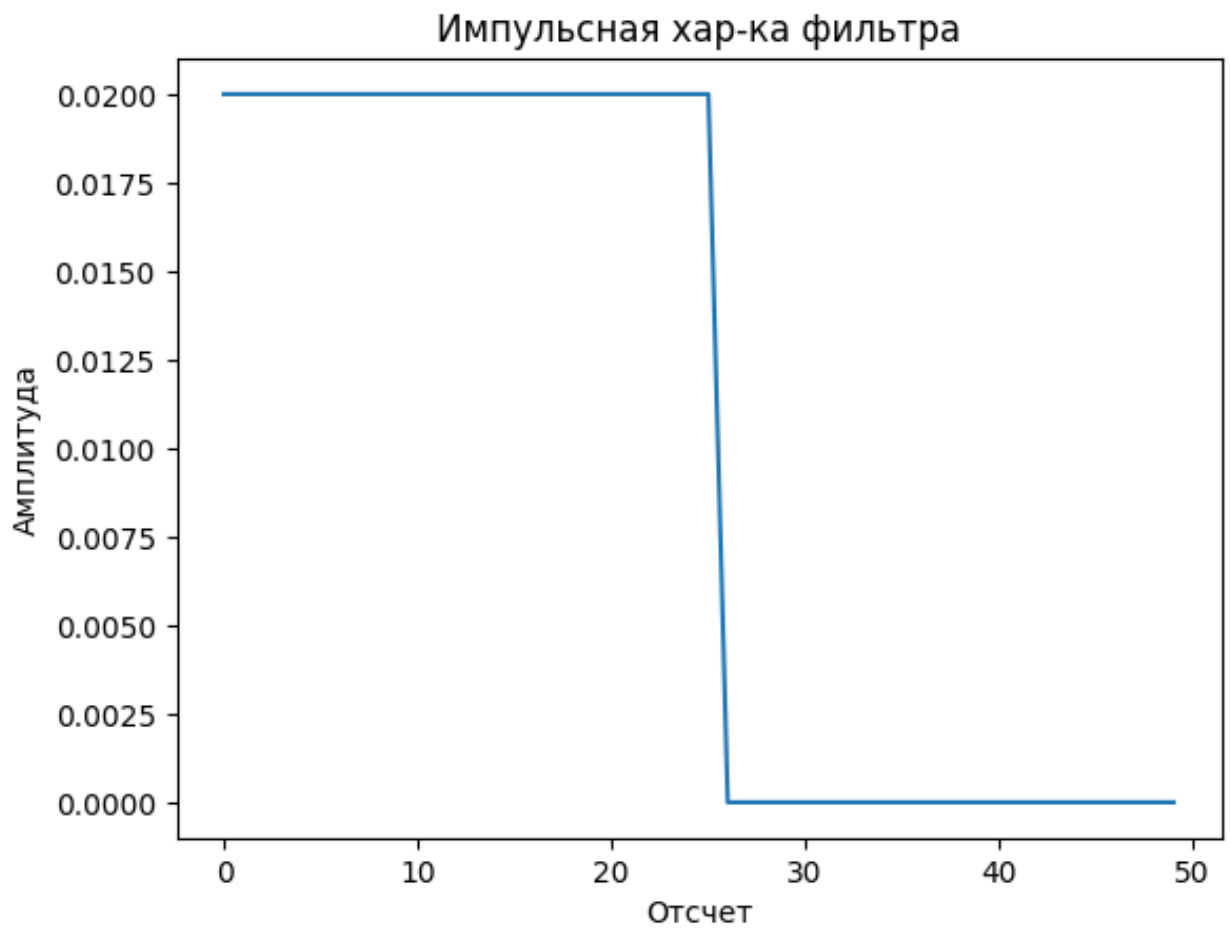


Рис. 1: Спектр сигнала

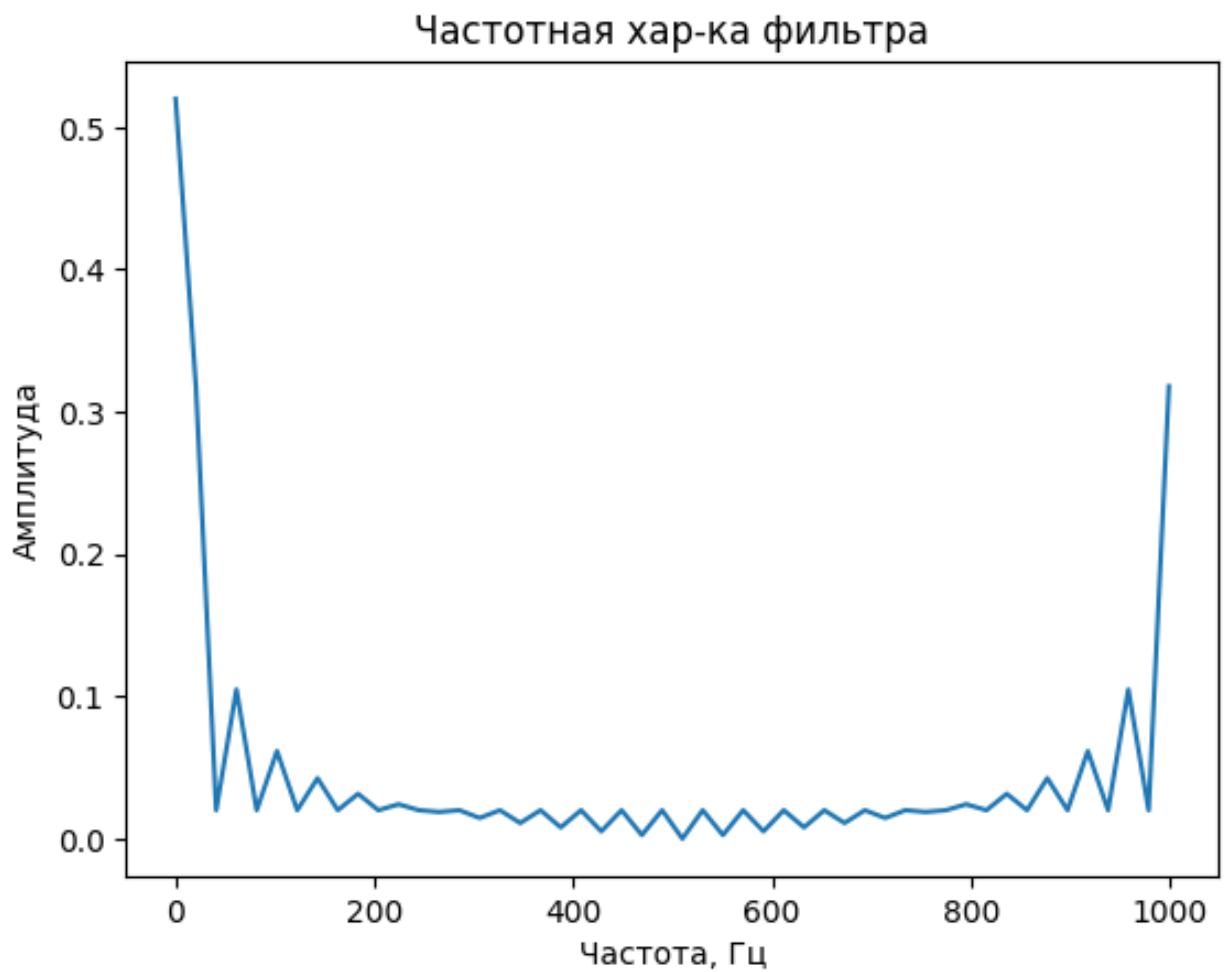
## Построение импульсной и частотной характеристик фильтра

```
1 H = np.abs(fft(h, M))
2 fH = np.linspace(0, 1/(t[1] - t[0]), M)
3
4 plt.plot(h)
5 plt.title('Импульсная хар-ка фильтра')
6 plt.xlabel('Отсчет')
7 plt.ylabel('Амплитуда')
8 plt.show()
```



## Частотная характеристика

```
1 plt.plot(fH, H)
2 plt.title('Частотная хар-ка фильтра')
3 plt.xlabel('Частота, Гц')
4 plt.ylabel('Амплитуда')
5 plt.show()
```



```
1 # Выполнение фильтрации сигнала во временной области
2 y = np.convolve(s, h, mode='same')
3
4 # Выполнение фильтрации сигнала в частотной области
5 sf = fft(s)
6 hf = fft(h, len(s))
```

```

7 yf = sf * hf
8 y = ifft(yf)
9
10 # Построение спектра выходного сигнала
11 Y = np.abs(fft(y))
12 fY = fftfreq(len(y), t[1] - t[0])
13
14 plt.subplot(211)
15 plt.plot(t, s)
16 plt.title('Исходный сигнал')
17 plt.xlabel('Время, с')
18 plt.ylabel('Амплитуда')
19 plt.subplot(212)
20 plt.plot(t, y)
21 plt.title('Выходной сигнал')
22 plt.xlabel('Время, с')
23 plt.ylabel('Амплитуда')
24 plt.show()

```

## Часть 2

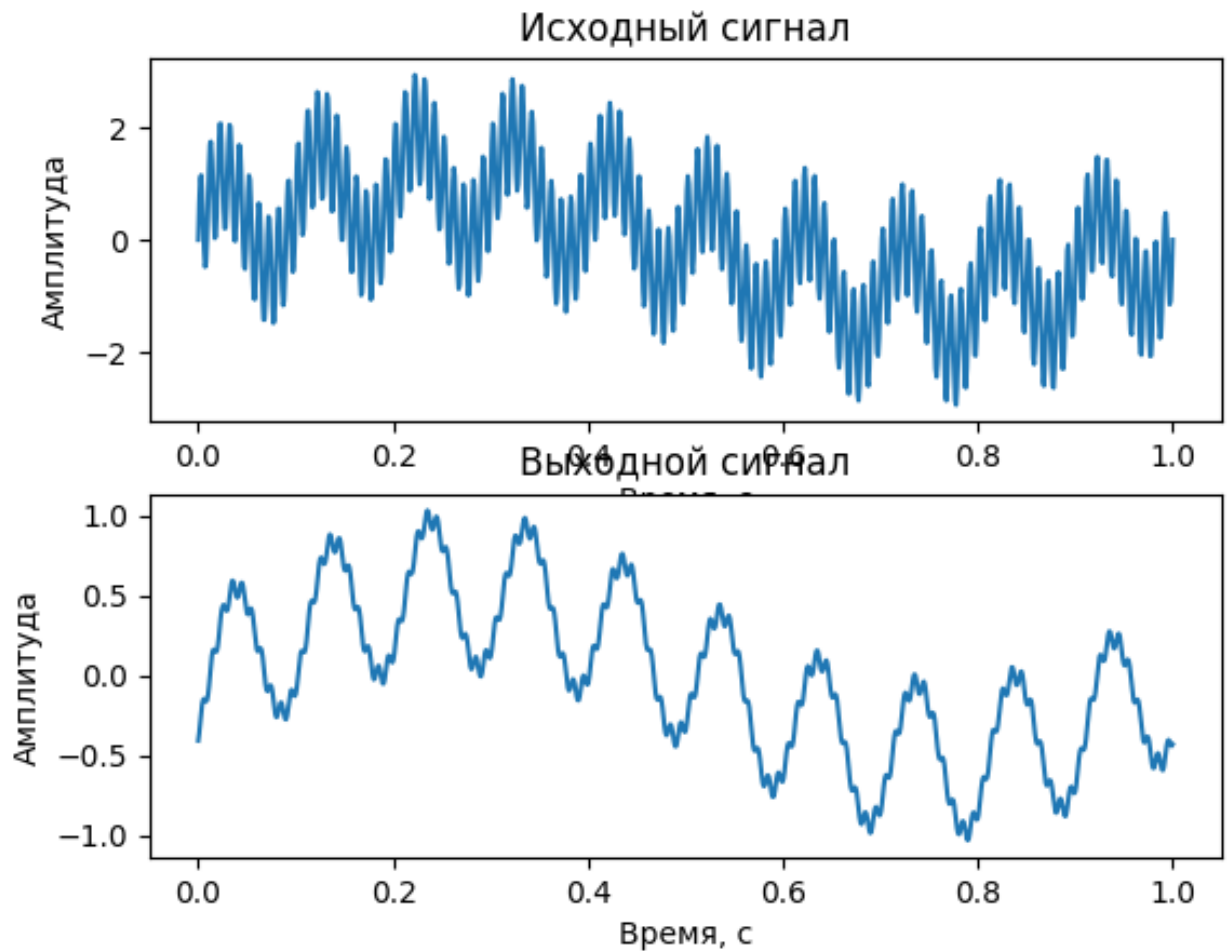
Для начала считаем исходный файл методом ‘wavfile.read’(метод возвращает частоту дискретизации (в сек) и данные из файла LPCM WAV).

```

1 fs, Audiodata = wavfile.read(SOURCE_FILE)
2 print(fs)
3 # Plot the audio signal in time
4 plt.plot(Audiodata)
5 plt.title('Audio signal in time',size=12)

```





Загрузим аудиофайл и построим его спектр:

```

1 y, sr = librosa.load(SOURCE_FILE)
2 print(sr)
3 D = np.abs(librosa.stft(y))
4 plt.figure(figsize=(10, 4))
5 librosa.display.specshow(
6     librosa.amplitude_to_db(D, ref=np.max),
7     y_axis='log',
8     x_axis='time'
9 )
10 plt.title('Спектр сигнала')
11 plt.colorbar(format='%+2.0f dB')
12 plt.tight_layout()

```

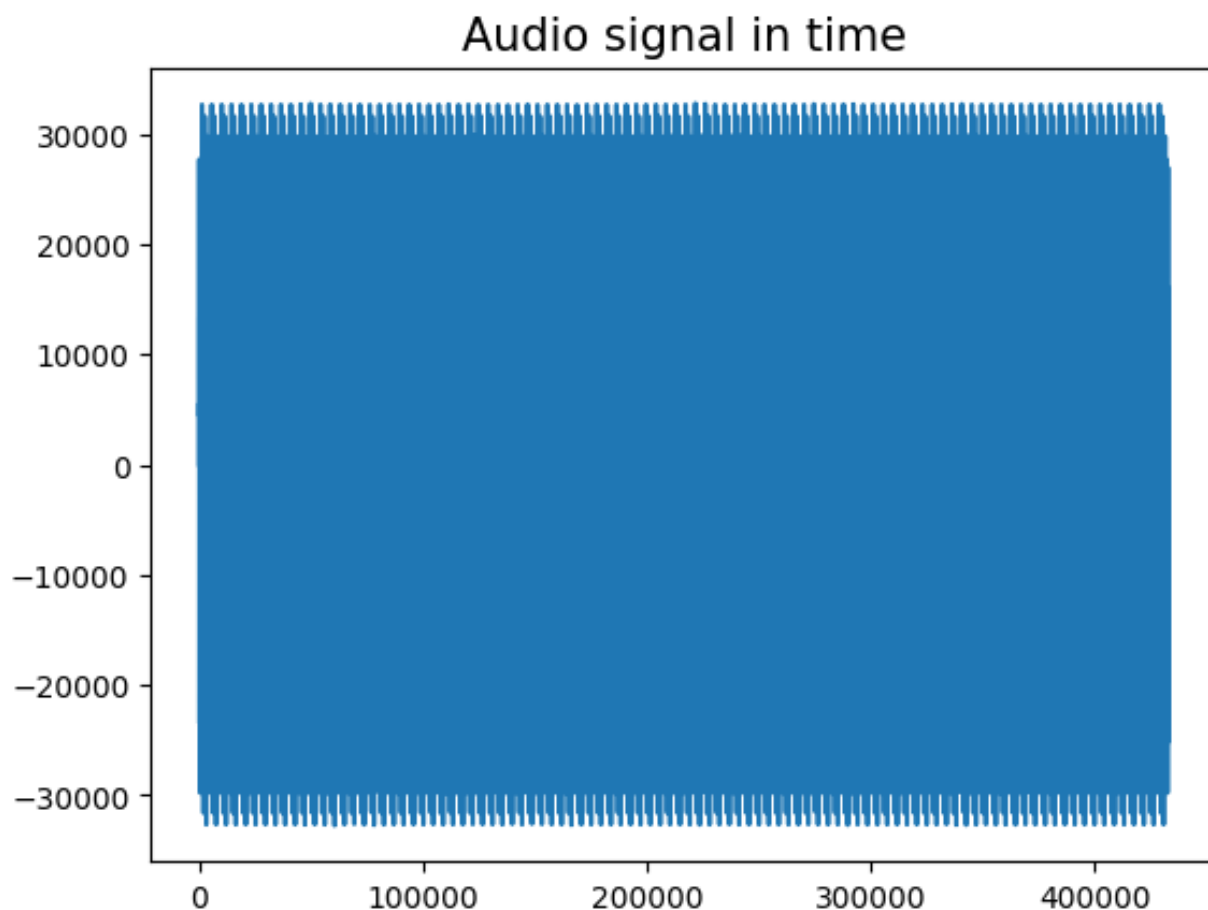


Рис. 2: Исходный аудиосигнал

Далее при использовании язык *Python* может возникнуть проблема с модулем *librosa* , поэтому необходимо модифицировать уже импортированную библиотеку. Для этого используйте команду:

```
1 !pip3 install --upgrade librosa
```

Определяем полосы подавления (в Гц)

```
1 band_stop_1 = [128, 256]
2 band_stop_2 = [128, 256]
3 band_stop_3 = [256, 2*256]
```

Максимальная частота дискретизации/дискретности

```
1 nyquist_freq = sr / 2
```

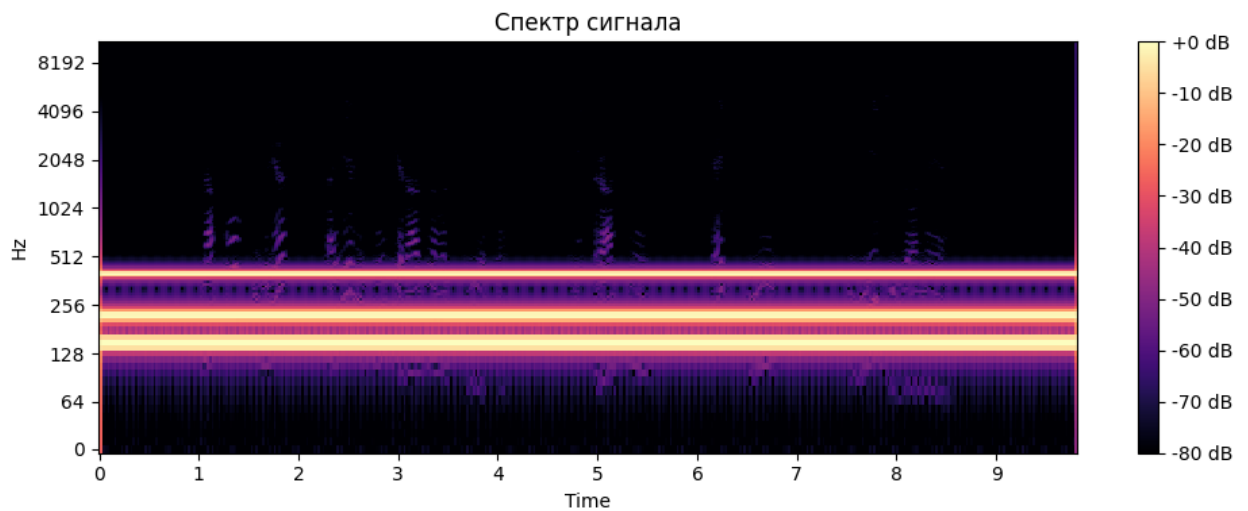


Рис. 3: Спектр исходного сигнала

Нормализация границ полос подавления к интервалу от  $-\pi$  до  $\pi$

```

1 normalized_freqs_bandstop_1= np.array(band_stop_1)
2                               / nyquist_freq
3 normalized_freqs_bandstop_2= np.array(band_stop_2)
4                               / nyquist_freq
5 normalized_freqs_bandstop_3= np.array(band_stop_3)
6                               / nyquist_freq

```

Создаем фильтр

```

1 b, a = signal.butter(
2     4,
3     [normalized_freqs_bandstop_1[0],
4     normalized_freqs_bandstop_1[1]],
5     btype='bandstop'
6 )
7 d, c = signal.butter(
8     4,
9     [normalized_freqs_bandstop_2[0],
10    normalized_freqs_bandstop_2[1]],
11    btype='bandstop'

```

```

12 )
13 f, e = signal.butter(
14     4,
15     [normalized_freqs_bandstop_3[0],
16     normalized_freqs_bandstop_3[1]],
17     btype='bandstop'
18 )

```

Применяем фильтр к сигналу

```

1 y_filtered = signal.filtfilt(b,a,y)
2 y_filtered = signal.filtfilt(d,c,y_filtered)
3 y_filtered = signal.filtfilt(f,e,y_filtered)
4 plt.figure(figsize=(10, 4))
5 D_filtred=np.abs(librosa.stft(y_filtered))
6 librosa.display.specshow(
7     librosa.amplitude_to_db(D_filtred),
8     y_axis='log',
9     x_axis='time'
10 )
11 plt.title('Спектр отфильтрованного сигнала')
12 plt.colorbar(format='%+2.0f dB')
13 plt.tight_layout()

```

Задаем путь к файлу и параметры дискретизации и сохраняем

```

1 filename = os.path.join(
2     OUTPUT_DIR,
3     f"FILTERED_OUT.wav"
4 )
5 sample_rate = sr
6 sf.write(filename, y_filtered, sample_rate, 'PCM_24')

```

Файл получился тихим, необходимо увеличить громкость

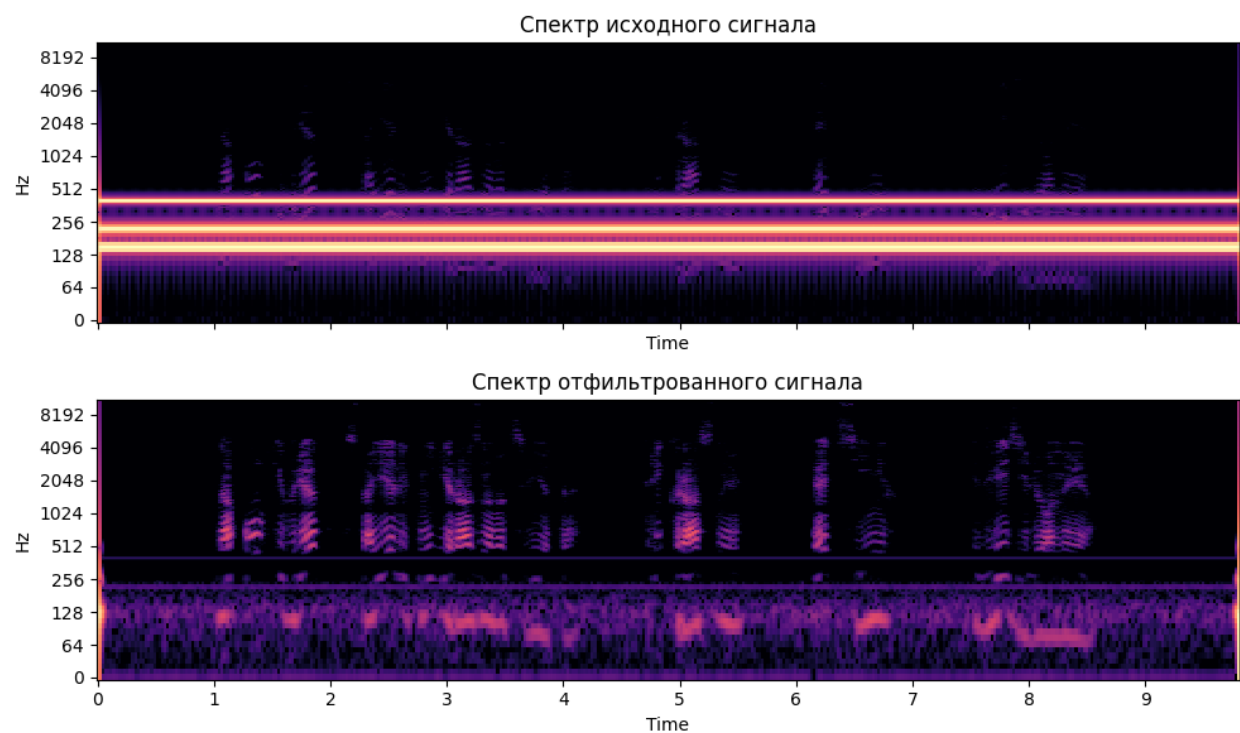


Рис. 4: До и после обработки сигнала

```
1 song = AudioSegment.from_wav(filename)
2 song = song + 50
3 output = os.path.join(
4     OUTPUT_DIR,
5     f"Increased_FILTERED_OUT.wav"
6 )
7 song.export(output, "wav")
```

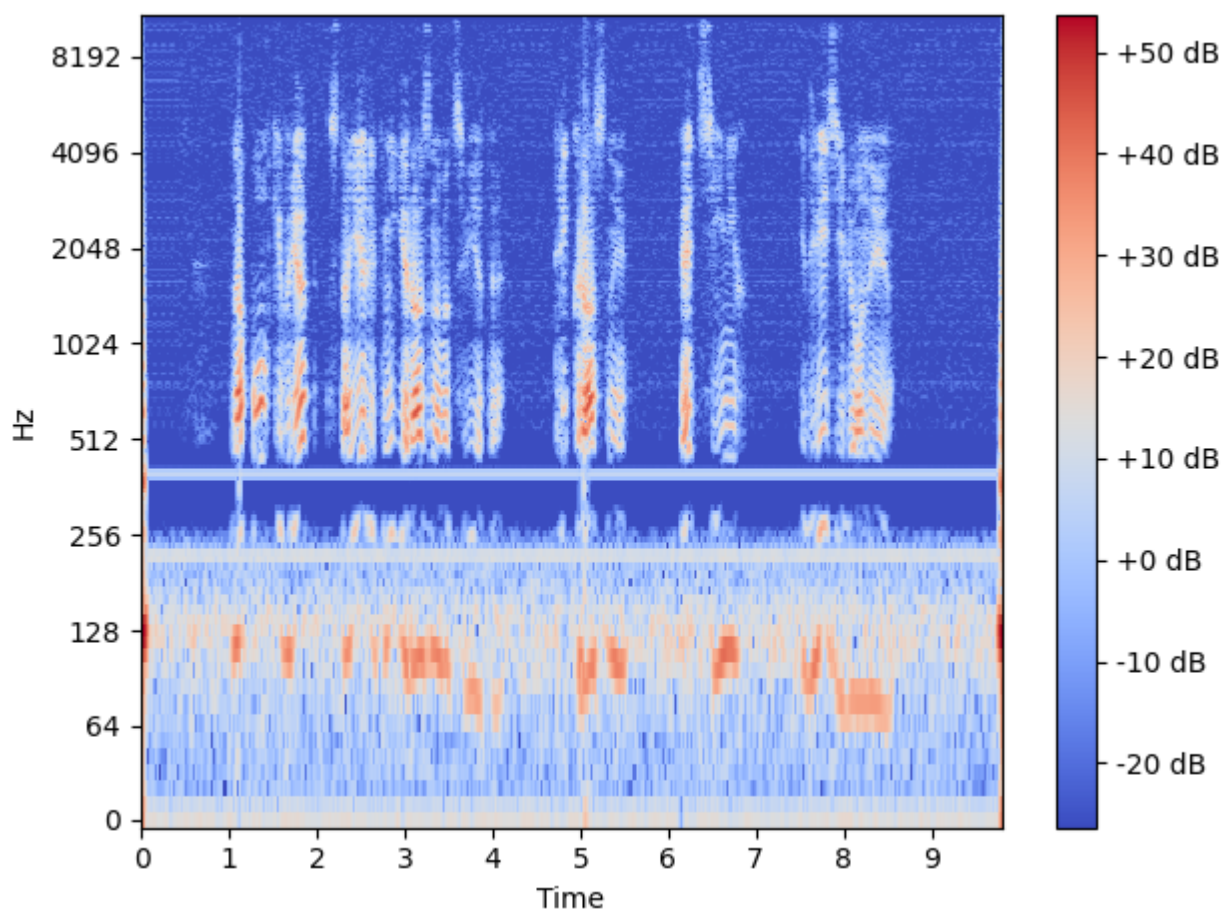


Рис. 5: Спектр усиленного сигнала

Из итогового файла можно узнать что :

На полке в ряд стояли книги разного цвета: 3 красные, 5 синих, 2 зеленые

### 3 Выводы лабораторной работы №4

Выполнив данную работу были изучены методы обработки цифровых сигналов, обработан зашумленный сигнал, из которого получилось извлечь информацию.